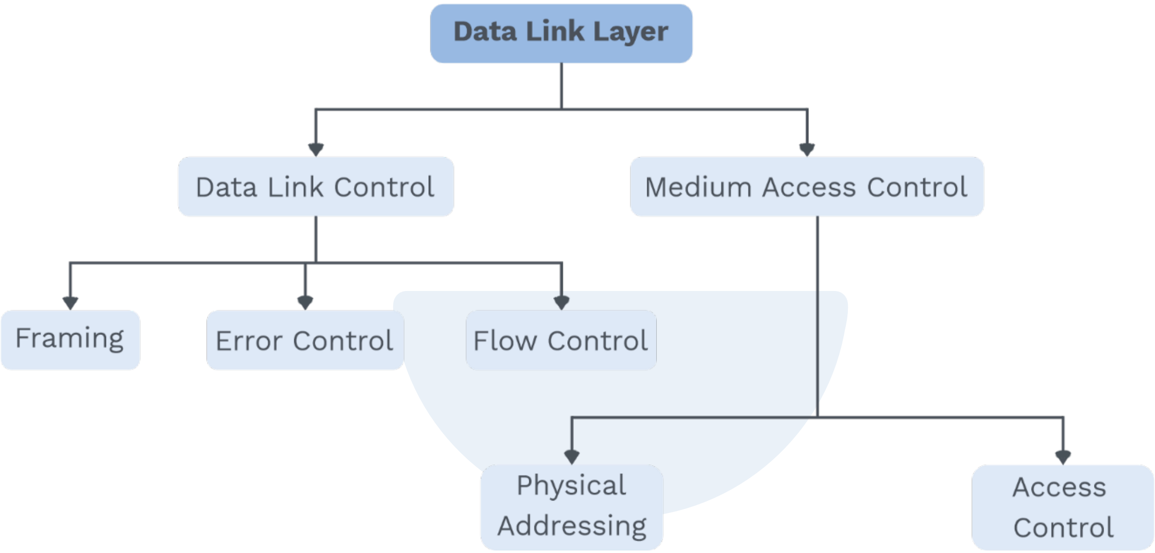
3 Data Link Layer

3.1 DATA LINK LAYER

It has two major functions:

- Data link control
- Medium access control



Data link control:
It includes framing, flow control and error control.

Medium access control:
It tells about the access control.

Data link control:
1) Framing:
We already know the physical layer provides bits synchronization (sender and receiver use the same bits). Now, these bits have to pack into the frame, and this is done by data link layer.

OR

Data link layer also takes packets from the network layer, and enclose them in multiple frames and send it to the physical layer.

Frame is having 3 basic components:

- Frame header
- Payload field
- Frame trailer

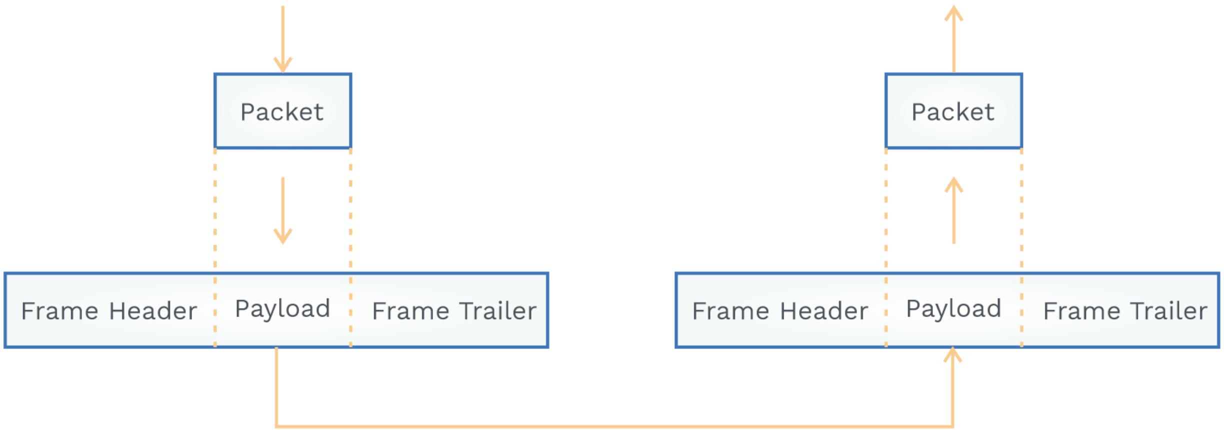


Fig. 3.1 Basic Components of Frame

Concept Building Exercise

- **Is it possible to encapsulate the entire message in a single frame? Is it efficient?**
  Yes, it is possible. But it is not efficient.
- **Why is it not efficient?**
  Because if a single bit error is there, entire frame has to retransmit.

Types of framing:

- Fixed size framing
- Variable size framing

Fixed size framing:

- As the name suggests, it is of Fixed Size.
- You need not worry about the ending of the frame.
- Since Frames are of fixed length, hence no flexibility is possible.

Variable size framing:

- We need to define an end of the frame as well as beginning of the next frame. This can be done in two ways:
  i) Character stuffing
  ii) Bit stuffing

Character stuffing:

- A flag is added at the beginning and at the ending of the frame, which tells the frame has started and ended.
- Flag size is multiple of 8 bit.
- It was used when only the character as a data was exchanged at the data link layer.

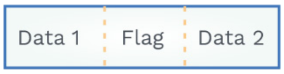
Concept Building Exercise

- **What to do when the data is having other than text, i.e. audio, video, images? Why not go with character stuffing?**
  In this case, we use byte stuffing; we cannot use character stuffing because there may be a chance when FLAG bytes get matched with the data inside the packet.
- **What would you think if you see two continuous FLAG?**
  It means ending of one frame and starting of next frame.
- **How does FLAG look like?**
  It has some special patterns like $(1111110)_2$ or $(0x7E)_{16}$ for HDLC protocol.

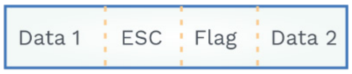
Byte stuffing:

- Add 1 extra byte whenever there is a flag or ESCAPE character in the text.
- Special byte is added to the data section of the frame when the same pattern as that of flag is present inside the frame.
- Now frame has an extra byte called as ESCAPE(ESC).
- This adding of an extra byte is called Byte stuffing.
- But the problem comes when the escape pattern appears in the middle of the data. The solution is add extra ESCAPE.

See figure below!!

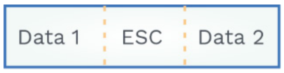


Original Frame

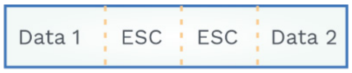


Frame after Byte Stuffing

- When there is ESC pattern in an original frame add another escape byte:



Original Frame



Frame after Byte Stuffing

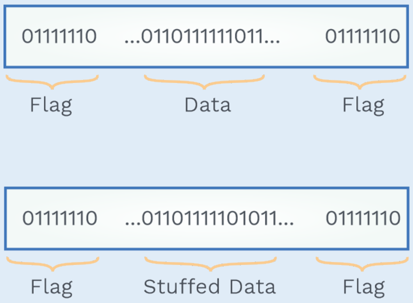
- Byte stuffing has one disadvantage, we have to use 1 byte always. So, there is always a limitation of using 8 bit.
- We can overcome this situation using bit stuffing.

Bit stuffing:

- In this, whenever flag pattern appears in the data section of the frame, to prevent it from looking like a pattern of the flag we add an extra bit to break the pattern.
- Let's say if we have taken 01111110 as Flag, so after bit stuffing, data will become 011111010.

- If we have taken 10000001 as Flag, so after bit stuffing, data will be 100000101.

- **What to do if a FLAG like a pattern appears in data?**
  If a FLAG like a pattern, say 01111110, appears in data, we need to stuff a single bit '0', at the end of 5 consecutive ones, instead of 1 byte(ESC), that we were doing in byte stuffing.



Note:

Real flag is not stuffed by the sender; hence no need to destuffing for real flag at the end of the receiver.

2) Error control:

- It manages both error connection and error detection.
- Basically, it deals with the retransmission of data, and this retransmission is based on Automatic Repeat Request (ARQ).
- There are basically two types of error: Single bit error and Burst error.

Single bit error: Single bit get changes when data reaches to receiver,

11101010 → 11111010

Burst error: When 2 or more bits gets changes,

11101010 → 11111011

Error detecting codes:

In error detection, we can only know the packet is corrupted. Redundant bits are added to detect error, and when the error is detected, retransmission is used to recover from error.

Error correcting codes:

In error detection, we can also know the bit which has been corrupted. Redundant bits are added in such a way, that it will detect and correct the errors.

Concept Building Exercise

- **What do you understand by adding redundant bits?**
  These bits are added by the sender and removed by the receiver. It is done so that errors can be detected and corrected easily.

Error detection:

- Parity checking
- Cyclic redundancy check
- Checksum

Simple parity check codes:

- It can detect an odd number of errors.
- In this k bits, data is changed into n bit code words.
- n = k + 1 (extra 1 bit is called parity bit)

In the case of even parity, if the number of 1's are even then, we add an extra bit by adding 0 else, if the number of 1's is odd, then we add an extra bit of 1 to make the Code bit an even number of 1.

Standard Definition

- A single parity check code is a single bit error detecting code in which:
- n = k + 1 with $d_{min} = 2$.

Concept Building Exercise

What do you understand by d_{min}?

It is called minimum Hamming distance, and it is defined as the number of bits that are changed during the transmission:

$$11111 \longrightarrow 10101$$

In this case $d_{min.} = 2$

Find the hamming distance of the given coding scheme?

d(0000, 1010)

d(1100, 1111)

Sol: 2, 2

What is the minimum Hamming distance for error detection?
$d_{min.}$ = t + 1 (where t bit errors occur during transmission)

If the minimum hamming distance is 2, how many bits of error can be detected?
1 (one bit)

What is the minimum Hamming distance for an error correction?
$d_{min.}$ = 2t + 1

Let's see below diagram and understand how simple parity bits are used in error detection,

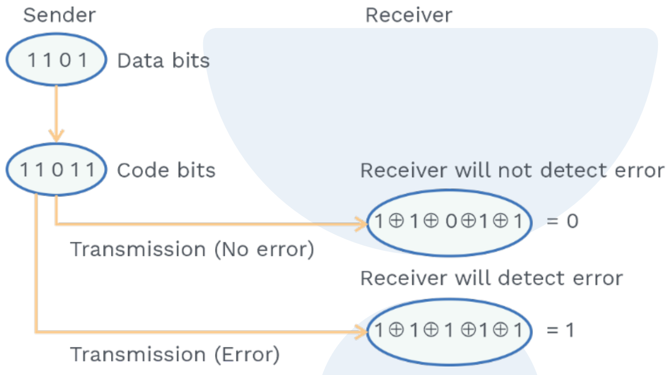


Fig. 3.6 Diagrammatic Representation of Simple Parity Bits Used in Detection

Here we are taking codebits as even parity:

- Databits are sent in the form of codebits by adding single parity.
- **You might be thinking about how to add a parity bit?**
  Add all the bits, and modulo-2 will give the parity bit.
- At the receiver side, it will do modulo-2 of the codebit, and if the result, known as syndrome, is 1, then an error is detected, and if the syndrome is 0, then there is no error.

Note:

A simple parity check code can detect odd number of errors only.

Cyclic redundancy check (CRC):
Common CRC polynomials are:
CRC–32 used in LAN, CRC–8 used in ATM header.

Some rules for generating the generator polynomial,
Rule 1: It should not be divisible by x.
This condition will ensure that all the burst error of length equal

to the length of the polynomial are detected.

Rule 2: It should be divisible by x + 1. This condition will ensure that all the burst error affecting an odd number bits are detected.

Rack Your Brain

CRC generator is $x^3 + x^2 + 1$. Convert it into binary.

CRC generator: 1101 (It is known by both sender as well as receiver)

Code: 1011011

Steps:

1) If CRC G has 4 bits then add 3 bit to the code 1011011000

2) Now mod-2 sum

And do the following at sender side,

```
1101)1011011000
     1101
      1100
      1101
        1110
        1101
          1100
          1101
          001 ⟶Take last (n–1) CRC if CRC is having
               n bits then append it in place of zeroes
```

The correct code which needs to be send is 1011011001.

3) Receiver will check whether the code is correct or not.

```
1101)1011011001
     1101
      1100
      1101
        1110
        1101
         1101
         1101
         0000 ⟶All zero means code is correct
```

- CRC codes can detect the single bit errors, double errors, odd number of errors and burst error.
- Fast when implemented in hardware compared to software.
- The divisor in cyclic codes are normally called generators.

Checksum:

- Checksum bits are usually placed at the end of the message with a complement to the sum function.
- It is used on the internet but not on the data link layer.

What we are using at the data link layer then!! CRC.

Let us understand by taking example,

a) Let's say a set of numbers (10, 20, 30) needs to be sent. Then the sender will send (10,20,30,-60) here -60 is the sum of all the numbers with a negative sign. Now the receiver will add all the numbers, and if the result is zero then there is no error.

b) By taking chunks of 2 bits, Explain how the checksum will work on the sender side if data is 01110001.

We have 01 11 00 01 (2 bit chunks)
Now add using 1 bit compliments 01 + 11 + 00 + 01 = 101
overflow add MSB to LSB 01 + 1 = 10
Take 1 complement of result = 01
Append to the sender data bit and send it to the receiver = 0111000101
At the receiver end we have 01 + 11 + 00 + 01 + 01 = 110
Overflow add MSB to LSB = 10 + 1 = 11
Complement of result = 00

If the result is 0; hence no error, if the result is non zero then error will be detected.

Questions:

1) In CRC, what do you think is the relationship between the size of the divisor or remainder?

Remainder is always one bit smaller than the divisor.

2) CRC generator is $x^3 + x + 1$, data bit are 1101, What will be the CRC which needs to be appended at the databit?

CRC generator $x^3 + x + 1$ = 1011, we will append 3 bit 0 to the data bit. After doing the modulo operation, we got CRC as 010, and Codebit will become 1101001.

```
1011)1101000
     1011
     ----
      1100
      1011
      ----
        1110
        1011
        ----
         1010
         1011
         ----
          001 ⟶ This CRC needs to append
```

3) What is the checksum value which needs to be send for the following two data items:

0x4589 and 0xBA76?

```
 4 5 8 9
 B A 7 6
---------
 F F F F
 0 0 0 0    1's compliment
```

Checksum bit which needs to be sent is 0000

Previous Years' Question

Q. Which of the following statements is TRUE?

a) Both Ethernet frame and IP packet include checksum fields

b) Ethernet frame includes a checksum field, and IP packet includes a CRC field

c) Ethernet frame includes a CRC field, and IP packet includes a checksum field

d) Both Ethernet frame and IP packet include CRC fields

Sol: c) **(GATE-2006)**

Error correcting:

- For error detecting and correcting we will use **hamming codes.**
- Let us first understand relationship between data bits and redundant bits in hamming code.
- Let us take 'd' as data bits and 't' as a redundant bits.
- Total number of bits that has to be transmitted = d + t.
- **Think how many states can the redundant bit discover?**

Its d + t + 1

Note:

Above condition derives a relationship,
$2^t \geq d + t + 1$

Let's say the value of d (message to be transmitted) is 4 then t would be 3.
It would satisfy this equation $2^t \geq d + t + 1$
In this case, t cannot be less than 3 i.e redundant bits cannot be less than 3.

Let us understand Hamming code by taking example:
Note: We calculate here on the basis of even parity bits.

Message: 1011
Here d = 4, now we need to add parity bits for each combination in the powers of 2 as shown below:

7	6	5	4	3	2	1
1	0	1	t_4	1	t_2	t_1

t_1 will take care of 1,3,5,7,... bits
t_2 will take care of 2,3,6,7,... bits

t_4 will take care of 4,5,6,7... bits

7	6	5	4	3	2	1
1	0	1		1		1

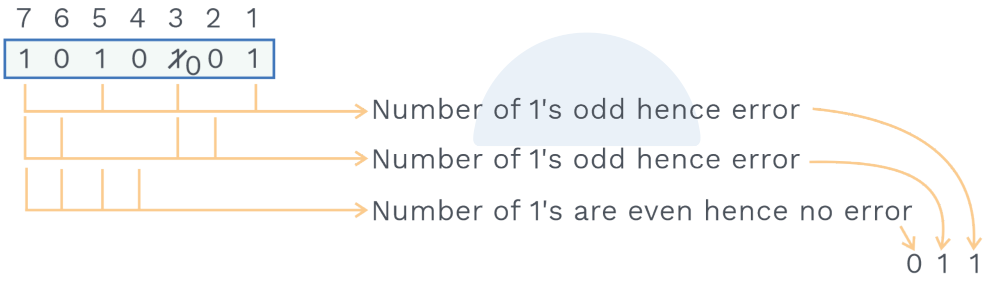
Your ultimate goal should be to make an even number of 1’s for particular parity, in this case, by checking position 3,5,7 we got a number of 1’s as odd, therefore we have to put t_1 = 1 in 1st position in order to make even parity.
Same we will do for t_2 and t_3,

7	6	5	4	3	2	1
1	0	1	0	1	0	1

Now the codebit that has to be transmitted is 1010101.
Let us assume at the receiver side one bit got corrupted, now how hamming code will detect and correct the code.

7	6	5	4	3	2	1
1	0	1	0	~~1~~0	0	1

How receiver will detect?


7 6 5 4 3 2 1
1 0 1 0 ~~1~~0 0 1
Number of 1's odd hence error
Number of 1's odd hence error
Number of 1's are even hence no error
0 1 1

How receiver will correct,
Since, we got to know which position is detected error, here it is 3rd position
Now we can change the 3rd bit and make it correct.

Previous Years’ Question

Q. A computer network uses polynomials over GF(2) for error checking with 8 bits as information bits and uses $x^3 + x + 1$ as the generator polynomial to generate the check bits. In this network, the message 01011011 is transmitted as:

a) 01011011010 **b)** 01011011011
c) 01011011101 **d)** 01011011100

Sol: c) **(GATE-2017)**

Previous Years' Question

Q. A bit-stuffing based framing protocol uses an 8-bit delimiter pattern of 01111110. If the output bit-string after stuffing is 01111100101, then the input bit-string is:

a) 0111110100 **b)** 0111110101

c) 0111111101 **d)** 0111111111

Sol: b) **(GATE-2014)**

Delays in computer network:

Transmission delay:

(t_t) Time which sender takes to transmit a frame on the link.

t_t = L/B (L = length of frame, B = bandwidth of channel).

Propagation delay:

(t_p) Time taken by 1 bit to travel from sender to receiver.

tp = d/v (d= distance between sender to receiver and v is transmission speed).

Queuing delay:

(t_q)Before processing of the frame it has to wait inside the buffer, that waiting time is called queuing delay.

Processing delay:

(t_{pr}) It is the time taken by a node or processor to process the frame. Basically it depends on the speed of the processor.

Flow control:

It deals with how much data the sender can transmit so that the receiver should not overflow.

Standard Definition

Set of procedures used to restrict the amount of data that the sender can send before waiting for an acknowledgement.

Data link layer uses error control, framing and flow control to send data from one node to another node.

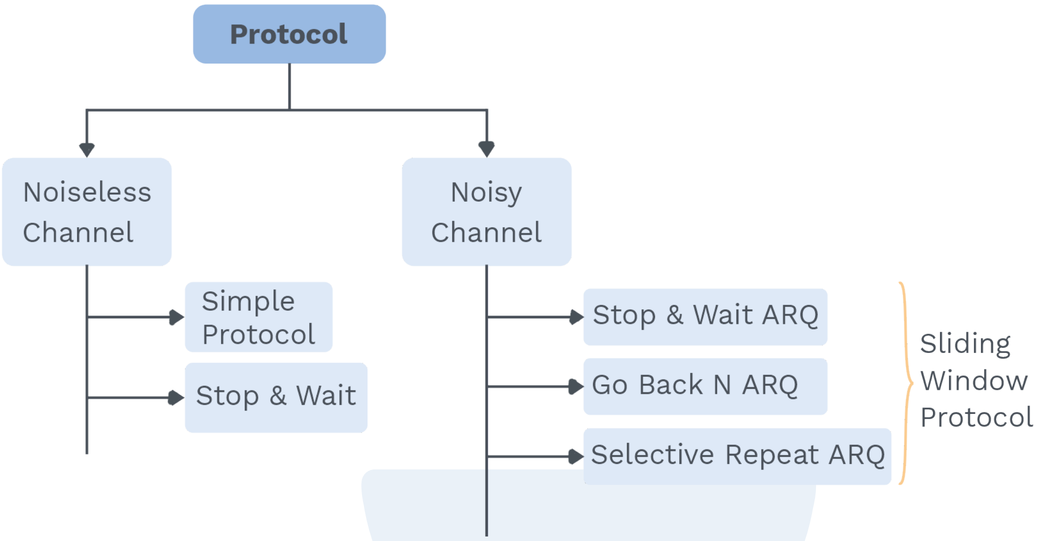


Fig. 3.2 Flow Control Protocols

For noiseless channel:

- We do not need to control errors in this channel.
- Let us assume in this channel, no packet is lost or corrupted.
- We can use simple protocol and stop and wait in this channel.

Simple protocol:

- It has no flow control and no error control.
- We assume that the direction of a packet is from sender to receiver i,e unidirectional.
- In this, the receiver can never discard the packet.
- There is no ack from the receiver.
- Both sender and receiver are constantly running because they do not know when an event is happening.

Note:

What do you understand about the events happening in the above point?

It means the sender does not know when the packet will come from the network layer; it has to constantly check, and the receiver does not know when the packet will come from the physical layer; it also has to constantly check.

Stop and Wait:

- Whenever the Receiver buffer size is full, it starts discarding the packet.
- There should be some mechanism so that sender should slow down its speed.
- **Can you guess what that mechanism is?**
  Sending ACK or acknowledgment from the receiver to sender.

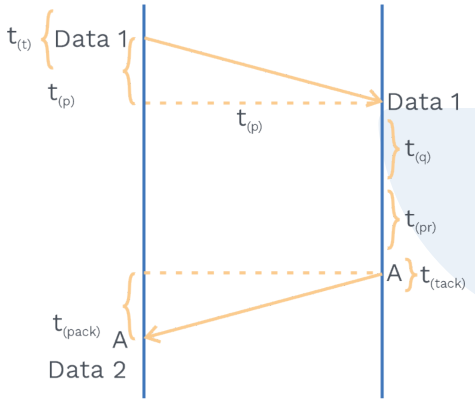
How stop and wait works:

1) Sender sends a data packet and waits for acknowledgement from the receiver.

2) Receiver receives the data packet and sends the acknowledgement to the sender.

3) After having the acknowledgment from the receiver, the sender sends the next packet.

Working:



Grey Matter Alert!

Take bandwidth in power of 10 and data in power of 2 for easy calculation.

The packet needs to be transmitted on a link by sender takes t_t time, and then it needs to propagate from sender to receiver takes t_p time and then at receiver side packet wait inside buffer which is t_q time and after that, it has to process which takes t_{pr} time. Receiver will then transmit the ack on a link, and it will take t_{tack} time.
Now receiver will send ack to sender which takes t_{pack} time.

Total time for sending 1 packet = $t_t + t_p + t_q + t_{pr} + t_{tack} + t_{pack}$

Calculation of link utilization or efficiency or sender utilization:

An assumption we made while calculating the total time of a packet.

1) Queuing delay and processing delay can be ignored.

2) Transmission delay for ack can be ignored.

Total time = $t_t + t_p + t_{pack}$

Total time = $t_t + 2t_p$ ($t_p = t_{pack}$)

Efficiency = Useful time/total time

Useful time = t_t (transmission of packet)

Total time = $t_t + 2t_p$

Efficiency = $t_t / (t_t + 2t_p)$

Efficiency = 1/ (1 + 2a) (a= t_p/t_t)

Calculation of throughput or bandwidth utilization:

Throughput is number of the bits that can be sent in a link per second

Throughput = Efficiency * Bandwidth

= $(t_t / (t_t + 2t_p))$ * Bandwidth

$[t_t=L/B]$

= $L / (t_t + 2t_p)$

Advantage of simple stop and wait:

- Receiver always acknowledges the sender by sending an ack packet.
- As the length of packet increases, efficiency increases.

Limitation of simple stop and wait:

1) Bandwidth is not efficiently utilized.

2) If the data packet gets lost receiver will wait for an infinite amount of time.

3) If ack get lost sender will wait an infinite amount of time.

For noisy channel:

We need to do error control in this channel. **Sliding window technique** is used in this channel

Sliding window protocol:

- Sender and receiver need to deal with only a part of the possible sequence number.
- Available sequence number ≥ Sender window size + Receiver window size.

Stop and wait ARQ:

- It is 1-bit sliding window protocol because Sender window size is 1, and Receiver window size is 1.
- We have seen Stop and wait in the above section, its working, advantage and disadvantage.
- Stop and wait ARQ working is the same as Stop and wait but it solves the limitation of Stop and wait for protocol by adding ARQ.

Rack Your Brain

a) Do you think any other advantage of simple stop and wait.

b) What will be the effect on efficiency if distance increases or decreases.

Grey Matter Alert!

ARQ: Automatic repeat request, its a request method in which receiver ask sender to retransmit the packet, if the packet is having any error.

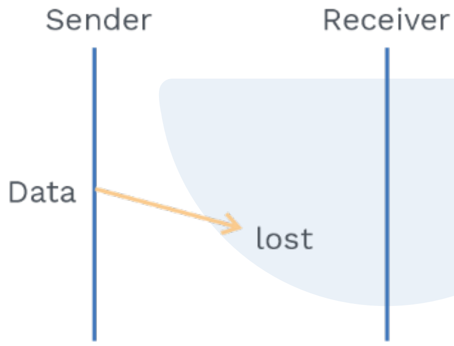
Minimum number of Sequence number required in Stop and wait ARQ?

Available Sequence number ≥ Sender window Size + Receiver window Size
Available Sequence number ≥ 1 + 1
Sequence number ≥ 2 (i.e 0,1)

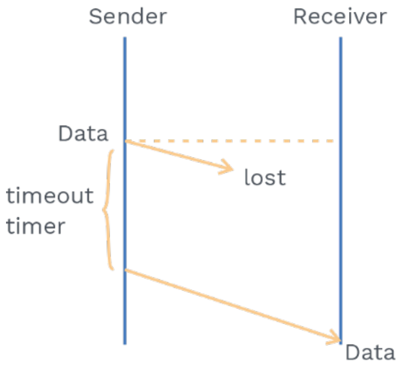
Let us see how it solves the limitation of stop and wait?

a) When data packet is lost?

If the data packet is lost, both the sender and receiver may get into a deadlock.



In order to prevent this situation, after sending the data packet, the sender starts the **time out timer**, to make sure the packet is transmitted in a specified amount of time.



This will prevent from deadlock state.

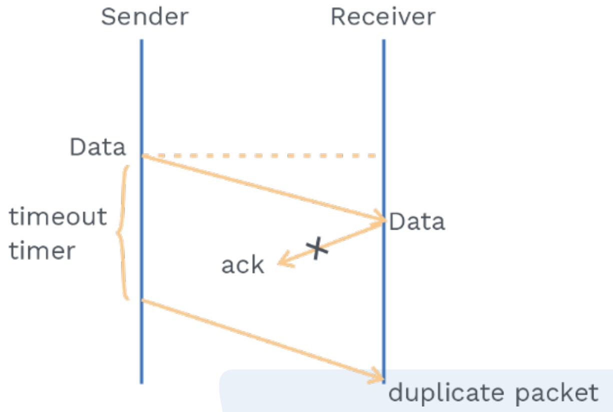
How to prevent deadlock in Stop and wait?

Sender will add time out timer.

Note:

Stop and wait ARQ = Stop and wait + Time out timer

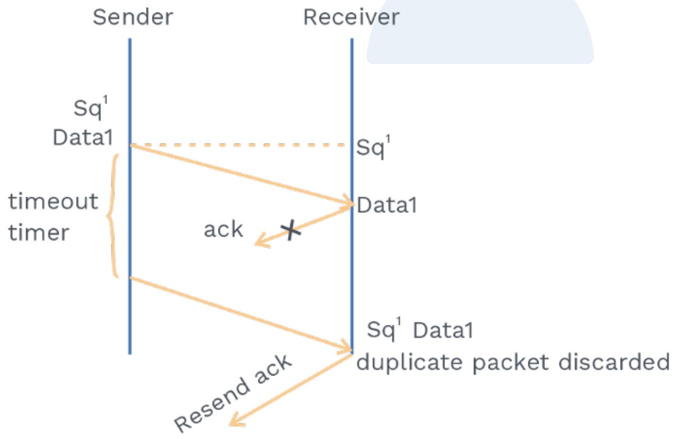
b) When ACK is lost?
Duplicate packet problem arises, see diagram.



If the timer goes timeout and sender does not receive any acknowledgement from the receiver, then the sender will retransmit the packet, but packet will be duplicated.

How to prevent duplicate packet problem in stop and wait?

Add sequence number in data packet, see, figure below.



Note:

Stop and wait, ARQ = Stop and wait + Time out timer + Sequence Number in data packet.

c) When does Ack gets delayed?
Duplicate acknowledgement problem arises.

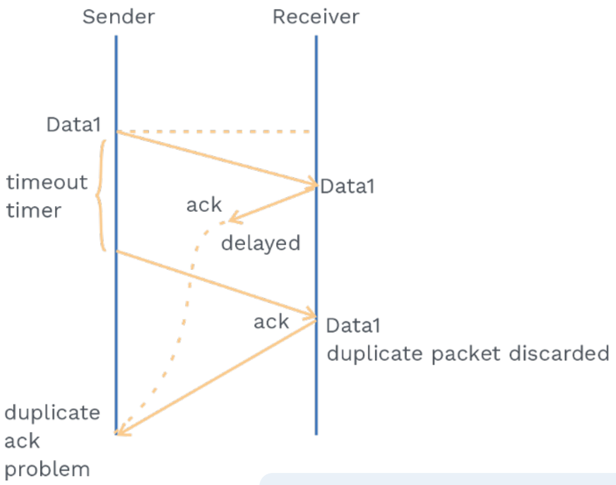


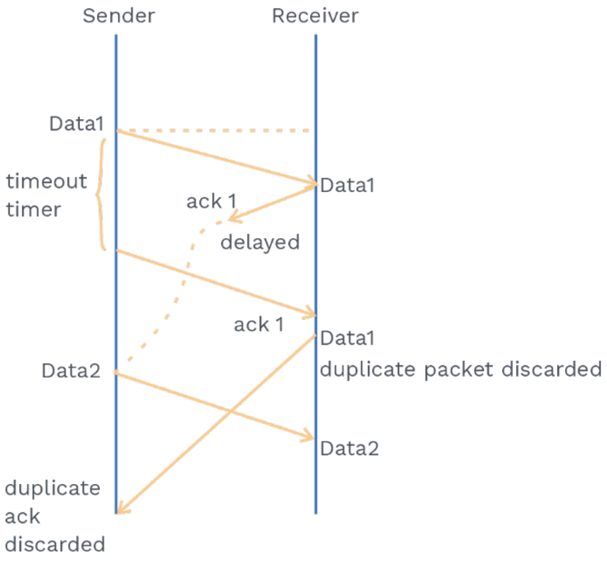
Fig. 3.3 Diagrammatic Representation of Duplicate Acknowledgement Problem

If the timer goes timeout and sender did not receive any acknowledgement from the receiver, then the sender will retransmit the packet, but packet will be duplicated this time and receiver will know that the packet is duplicated due to the sequence number attached to the packet, and it will resend the acknowledgement that will reach at the same time when the first acknowledgement reached, it will create a duplicate acknowledgement problem.

How to prevent duplicate acknowledgement problem?

Add the Sequence number to the acknowledgement.

This time if the sender sees the ack with the same sequence number, it will get to know that acknowledgement is duplicated, see diagram below.



Note:

Stop and wait ARQ = Stop and wait + Time out timer + Sequence Number in data packet + Sequence number in Ack Packet.

Let us understand the difference between "Stop and wait" and "Stop and wait ARQ."

Stop and wait arq	Stop and wait
Since the channel is noisy there is error control mechanism.	Channel is noiseless hence no error control.
Timeout timers added after sending packet.	No concept of timeout timer.
Sequence numbers are added in data packet as well as in acknowledgement packet.	No concept of sequence number or acknowledgement number.

Rack your Brain

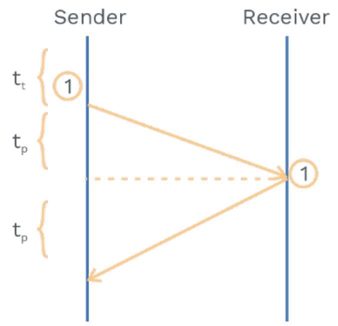
a) Why are sequence numbers added in data packets and acknowledgement packets?

b) Do you think Stop and wait ARQ is efficient?

Limitation of Stop and wait ARQ:

Sender sends data in t_t time, and then it waits for $2t_p$ time. This $2t_p$ actually cause limited efficiency.

Let's consider the given scenario:



- In t_t = 1 packet sends
  In 1 second = $1/t_t$ packet sends
  In a time of $(t_t + 2 * t_p) = (t_t + 2 * t_p)/t_t$ packets can be sent.
- From this we get to know we can transmit (1 + 2a) packet for full efficiency.
  But we are sending only 1 packet.
- Sender can send in $2t_p$ time also.

Grey Matter Alert!

Can you think how we can improve the efficiency of Stop and wait for ARQ?
It can be improved by increasing the window size so that sender should not wait for $2t_p$ time.

PRACTICE QUESTIONS

Q1 **In Stop and wait, 10 packets need to be sent from sender to receiver, which every 4 packets have been lost. What is the total number of the packet that needs to be sent?**

Sol:

1 2 3 4 5 6 7 8 9 1 0

lost (needs retransmission)

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

lost retransmission

1 2 3 4 4 5 6 7 7 8 9 10 10

lost retransmission

Total number of packet that needs to be sent is 13

Q2 **There is a channel between sender and receiver, and the channel is having a problem because of which some packets are getting lost, error probability is 0.5 (i.e while sending 100 packets, 50 packets are lost). How many total number of packets needs to be transmitted if 500 packets need to be sent?**

Sol:

error probability is 0.5 that means
50% of packet get lost

$$\underbrace{500}_{\substack{\text{Sending}\\ \text{500 packet}}} + \underbrace{500 * (0.5)}_{\substack{\text{250 packet}\\ \text{get lost}}} + \underbrace{500 * (0.5)^2}_{\substack{\text{125 packet}\\ \text{get lost}}}.....$$

$$500\left(\frac{1}{1-0.5}\right) = \frac{500}{0.5} = 1000$$

1000 packet needs to be transmitted.

Previous Years' Question

Q. Suppose that the stop-and-wait protocol is used on a link with a bit rate of 64 kilobits per second and 20 milliseconds propagation delay. Assume that the transmission time for the acknowledgement and the processing time at nodes is negligible. Then the minimum frame size in bytes to achieve a link utilization of at least 50 % is___.

a) 160 **b)** 320 **c)** 640 **d)** 220

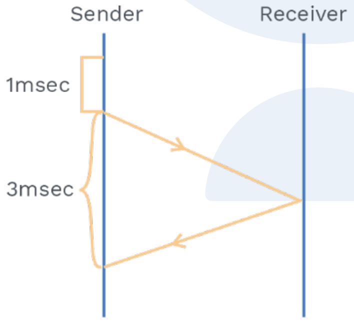
Sol: b) **(GATE-2015)**

Let us understand with example how to increase efficiency in Stop and wait ARQ,

T_t = 1msec and T_p = 1.5msec

efficiency = 1/1+2a

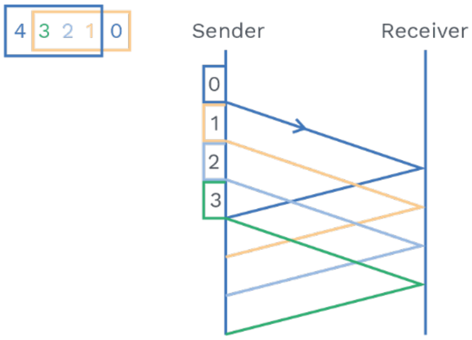
Putting values of T_t and T_p we have efficiency = 0.25



Here, sender is transmitting for 1 msec and waiting for 3 msec.

Let's send 3 more packet in 3 msec:

In the below diagram when the acknowledgement of packet (0) is received, we can send a new packet (4). At this point, the sender came to know packet(0) is received, and it makes the space for the packet (4). This is called the sliding window technique.



Note:

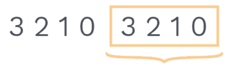
Until the acknowledgement of the first packet comes, sender holds the packet in buffer, that buffer is known as the sender window.

Go Back N ARQ:
What is N here ! Sender window size is N
Points:

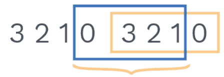
- The size of the sender window must be less than 2^m (where m is the size of sequence number fields in bits), and the size of receiver window is always 1.
- Each time the receiver receives a new frame, it starts a new acknowledgement timer, and if the timer expires, the receiver sends the cumulative acknowledgement for all the frames which are unacknowledged at that moment and sometimes, it uses an independent frame if the receiver wants to acknowledge only one frame.
  It silently discards the frames if frames are corrupted.

Sender window:

- It is the sequence number of the data frames which can be in transmitted. The maximum size of the window is $2^m - 1$.
- The sender window will slide according to one or more valid acknowledgement comes.



Current sender
window size



Window slides after
sending packet (0)

Receiver window size:
In Go back N, Receiver size is always 1.

Go Back N, does not accept the out of order packet.
Given diagram is a flow how packet travels in Go Back N,
Here, N = 4,

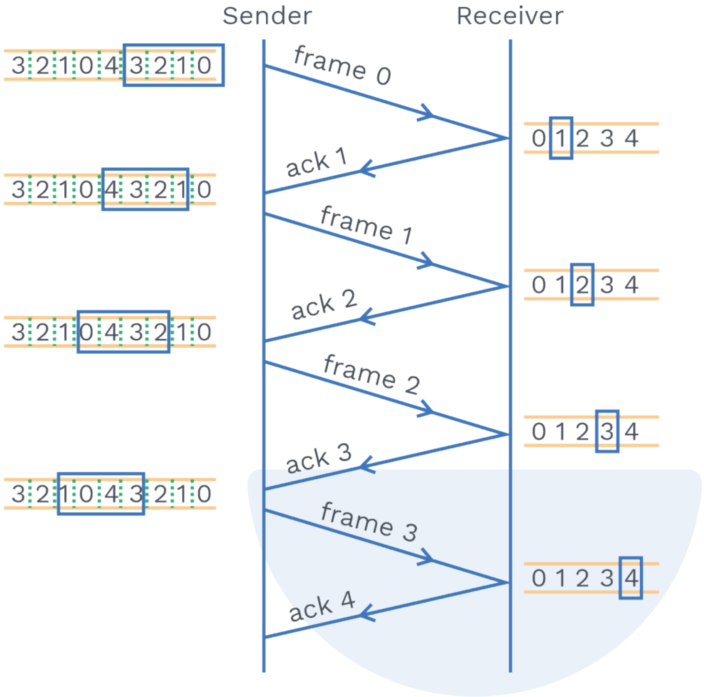
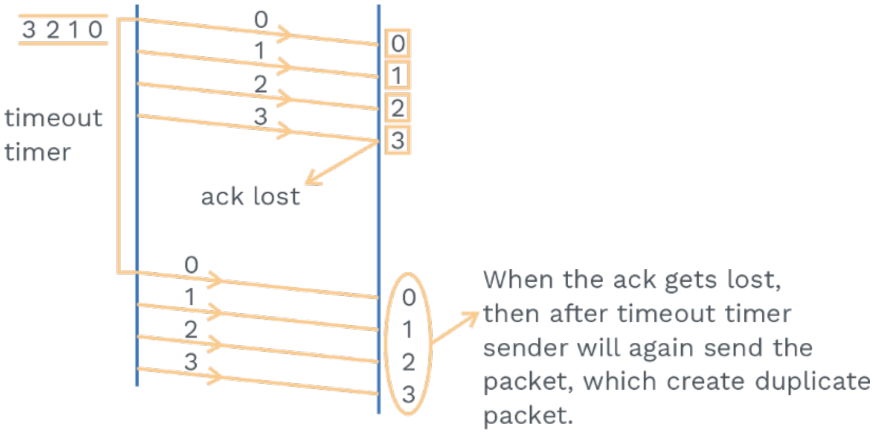


Fig. 3.4 Flow of Packets in Go Back N

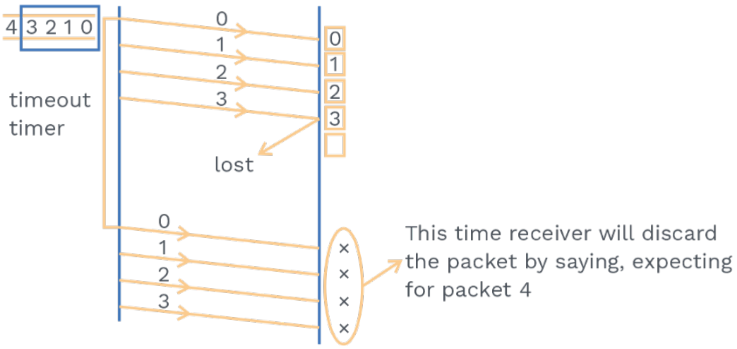
Why is there a need for taking sequence number N+1?
If you see the diagram above Window size is 4, but still, we have taken 5 sequence numbers (i.e 0,1,2,3,4).

In order to understand why we have taken the N+1 sequence number, we must understand what happens when the acknowledgement gets lost in Go Back N and when Sender window = Sequence number?
In the below diagram N = 4 and Sequence number = 4.


3 2 1 0
0
1
2
3
0
1
2
3
timeout timer
ack lost
0
1
2
3
0
1
2
3
When the ack gets lost, then after timeout timer sender will again send the packet, which create duplicate packet.

Here, duplicate packet problem arises.

Now, let us take a situation where N = 4 but Sequence numbers are 5.



This time receiver will discard the packet by saying, expecting for packet 4

Now you get to know why we have taken.

Note:

Maximum sequence number >= Sender window Size + Receiver window size
Maximum sequence number >= N + 1

Q3 **Sender window size = N, Receiver window size = 1, What will be the number of bits required for Sequence number?**

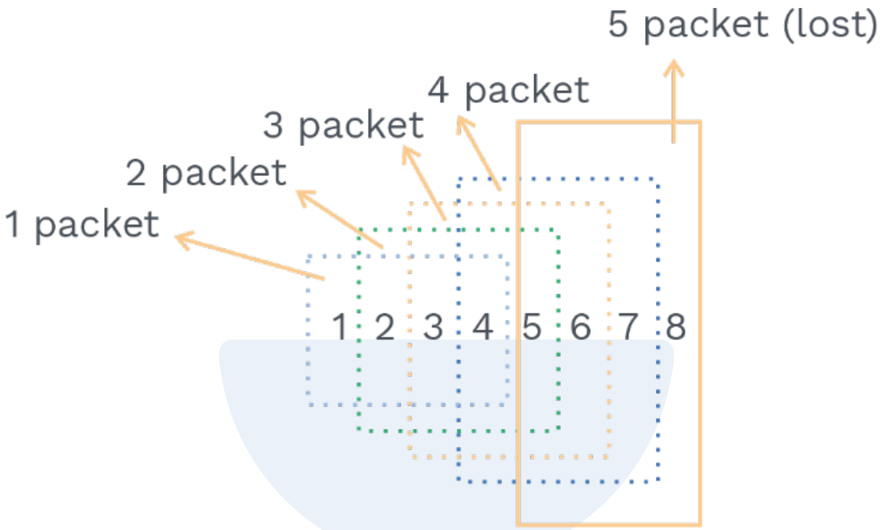
Sol: **$\log_2$(N+1)**

Q4 **Number of bits available in the Sequence number field is 'k', what is the possible Sequence number in GBN protocol?**

Sol: Possible sequence number are from 0 to $2^k - 1$

Q5 **In Go Back 4, assume 8 packets need to be sent from sender to the receiver in which every 5th packet has been lost. What is the total number of packets that needs to be sent?**

Sol: If N = 4, means window size is 4, if any packet gets lost we need to retransmit the entire window,



When the acknowledgement of the 5th packet does not reach the sender side, and there is a timeout timer at the sender side, it will retransmit the entire window.

Note:
The 5th packet contains Sequence numbers from 5,6,7,8. It needs to retransmit.
Now below diagram shows the scenario for remaining packet.

$1\ 2\ 3\ 4\ \boxed{5\ 6\ 7\ 8}\ 5\ \boxed{6\ 7\ 8}\ 6\ 7\ \boxed{8}\ 8$

Total number of packet which needs to retransmit are 16.

Efficiency in Go Back N:

Efficiency = Useful time / Total time

Useful time = Sender Window Size * Transmission time (t_t)

Total time = $t_t + t_p + t_p \Rightarrow t_t + 2t_p$

Efficiency = $N * t_t / (t_t + 2t_p)$

$= N/(1 + 2a)$

Previous Years' Question

Q. A 1 Mbps satellite link connects two ground stations. The altitude of the satellite is 36,504 km, and the speed of the signal is 3 × 10^8 m/s. What should be the packet size for a channel utilization of 25% for a satellite link using the go-back-127 sliding window protocol? Assume that the acknowledgement packets are negligible in size, and that there are no errors during communication.

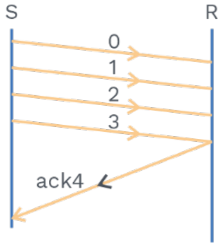
a) 120 bytes **b)** 60 bytes **c)** 240 bytes **d)** 90 bytes

Sol: **a)** **(GATE IT-2008)**

Go back N uses cumulative acknowledgement (mostly) and independent acknowledgement (if required)
Let us see about these acknowledgement.

Cumulative acknowledgment:

- If Kth packet acknowledged, this implies (k-1)th packet is received successfully.
- At the receiver side, it starts the acknowledgement timer, and when it expires, the receiver will send the cumulative acknowledgement for the packet it receives in the meantime.

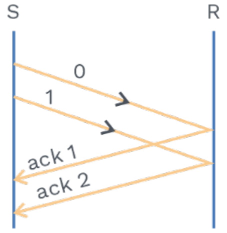


What do you understand by discarding a packet silently in Go back N?
This means it is not going to accept it; due to this, there will be time out at the other end, and entire window is sent in case of GBN.

Independent acknowledgment: Sender will receive acknowledgement for every packet

Rack Your Brain

Acknowledgement timer or timeout timer, which should be greater and why?



Selective repeat ARQ

Points:

- The size of sender window is equal to receiver window size.
- Available Sequence number = Sender window Size + Receiver window size.
  $2^n = 2^{n-1} + 2^{n-1}$ where n bits are used for sequence number
- As soon as the receiver receives the frames, it sends the acknowledgement, and uses the independent acknowledgement.

In Go back, N, out of order packets are not accepted, which means the sender has to send the entire window, which leads to consumption of bandwidth and more traffic. In noisy channels, there are more chances of corrupted packets, then we can use a mechanism called **Selective Repeat ARQ.**

In Selective repeat ARQ, only the damaged the frame needs to be sent; it makes efficient use of bandwidth in the noisy channel.

Rack Your Brain

Do you think Go back N is preferred in a noisy link !!

Note:

Receiver has to do more work in selective repeat, Think why !!

Receiver does not accept the corrupted frame and also does not discard the frame silently, but it will use negative acknowledgement.
Due to the use of negative acknowledgement, the sender need not to wait for timeout timer completion.
Receiver accepts out of order packets.
Efficiency in SR protocol: N/(1+2a).

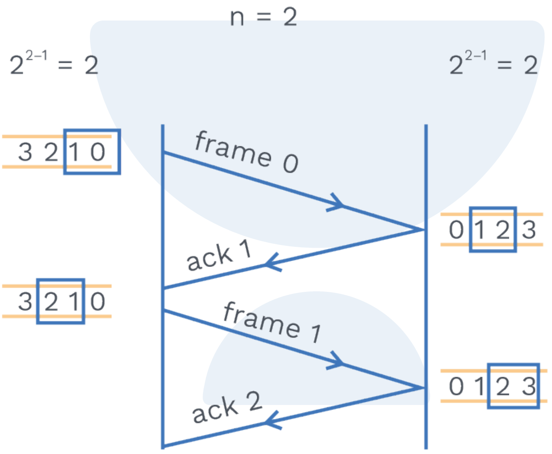
Q6 **What will be the maximum window size that is required for data transmission when SR(Selective repeat) protocol with 5 bit frame sequence number is used______?**

Sol: Maximum window size = 2^{n-1} => which will give 16

It will be, Sender window size + receiver window size

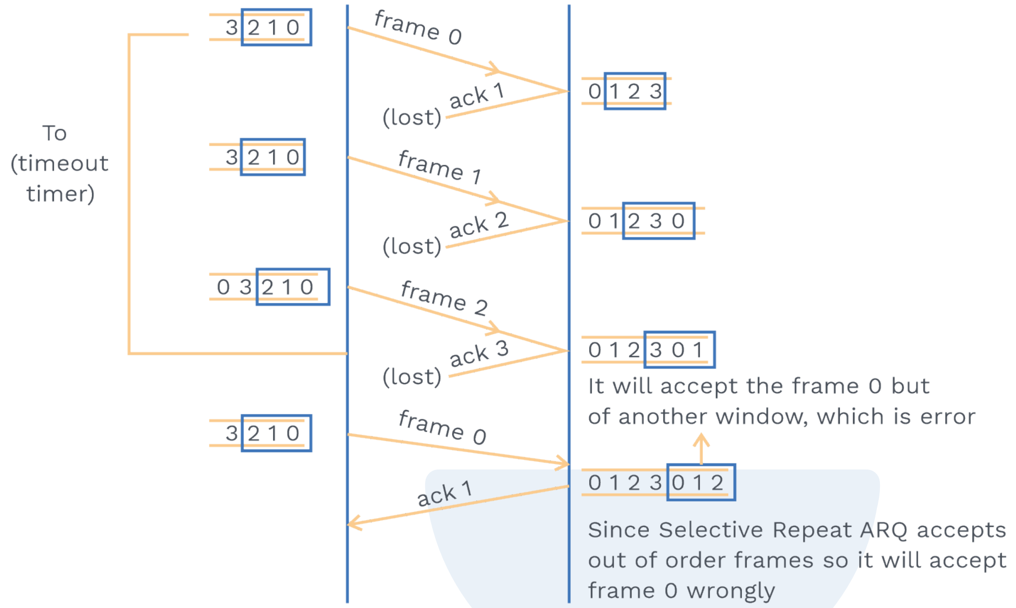
16 + 16 = 32

Lets see how selective repeat works,



What will happen if we take window size greater than 2?
Let's take window size as 3,

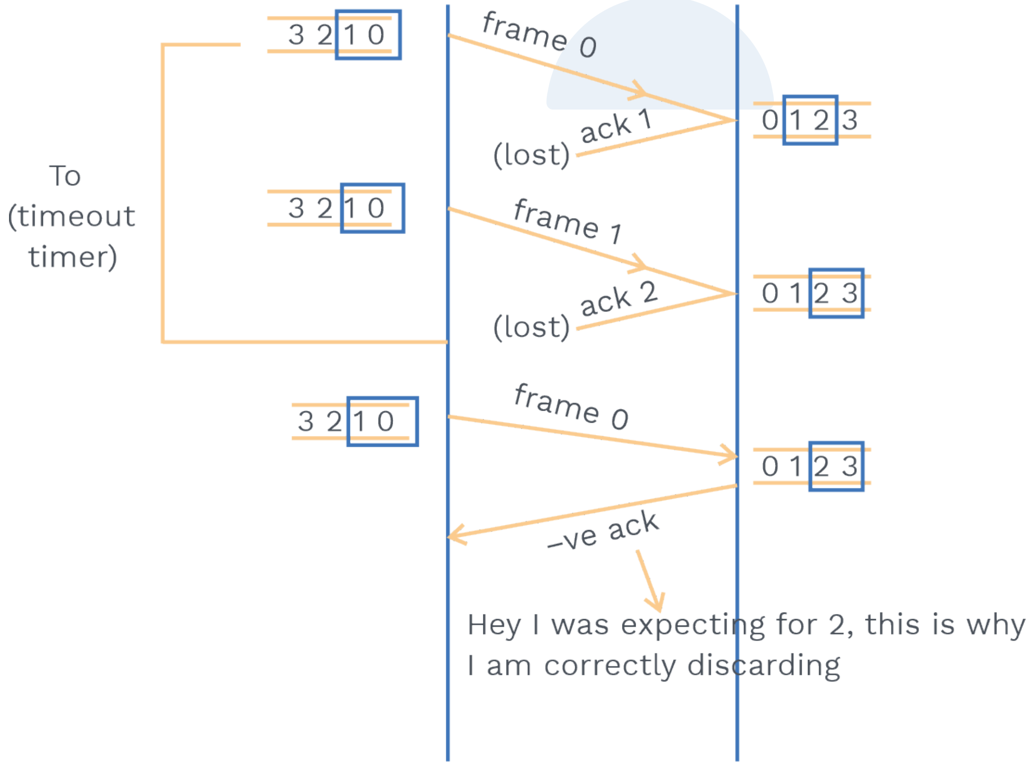
In the below diagram, when all the acknowledgements are lost, the sender will send the frame again after the time out timer of that frame, and the receiver will accept the same frame, but in another window, this is an error.



It will accept the frame 0 but of another window, which is error

Since Selective Repeat ARQ accepts out of order frames so it will accept frame 0 wrongly

Lets see if this problem will arise if we have taken window size as
Now we have taken window size as 2.



Hey I was expecting for 2, this is why I am correctly discarding

If we see the above figure, after timeout, when the sender resends the packet again, receiver will give a negative acknowledgement. This is the reason selective repeat ARQ we will use window size as 2^{n-1}.

Concept Building Exercise

Q.1 What is Piggybacking?

Sol: Piggybacking is used to improve the efficiency of bidirectional transmission. When a frame is carrying data from P to Q, it can also carry control information about frames from Q, and when a frame is carrying data from Q to P, it can also carry control information about frames from P.

Q.2 Compare Go back N and selective repeat?

Sol: In the Go-Back-N ARQ protocol, we can send several frames before receiving acknowledgement. In case a frame is damaged/lost, we need to resend all outstanding frames we have sent before. In SR(Selective repeat) protocol, we easily avoid transmission, which is not necessary by sending only those frames that are either corrupted or missing. This is possible because of negative acknowledgement.

Previous Years' Question

Q. Consider a 128 × 10^3 bits/ second satellite communication link with one-way propagation delay of 150 milliseconds. Selective retransmission (repeat) protocol is used on this link to send data with a frame size of 1 kilobyte. Neglect the transmission time of acknowledgement. The minimum number of bits required for the sequence number field to achieve 100% utilization is __________.

a) 2 **b)** 4 **c)** 6 **d)** 8

Sol: b) **(GATE-2016)**

Comparison of sliding window ARQ protocols:

	Stop and Wait ARQ	Go back N	Selective Repeat
Efficiency	1/(1+2a)	N/(1+2a)	N/(1+2a)
Window size	Sender Window Size = 1 Receiver window size = 1	Sender window size = N Receiver window size = 1	Sender window size = N Receiver window size = N
Sequence numbers required	2	N + 1	2 × N

In data link control, we have seen that if there is a dedicated link between sender and receiver how the protocol works, but the case will change if we do not have a dedicated link.

How to manage in case of cellular networks where channels are not dedicated?

Here comes Medium access control, which is responsible for multiple access resolution.

Lets see categories in Medium access control,

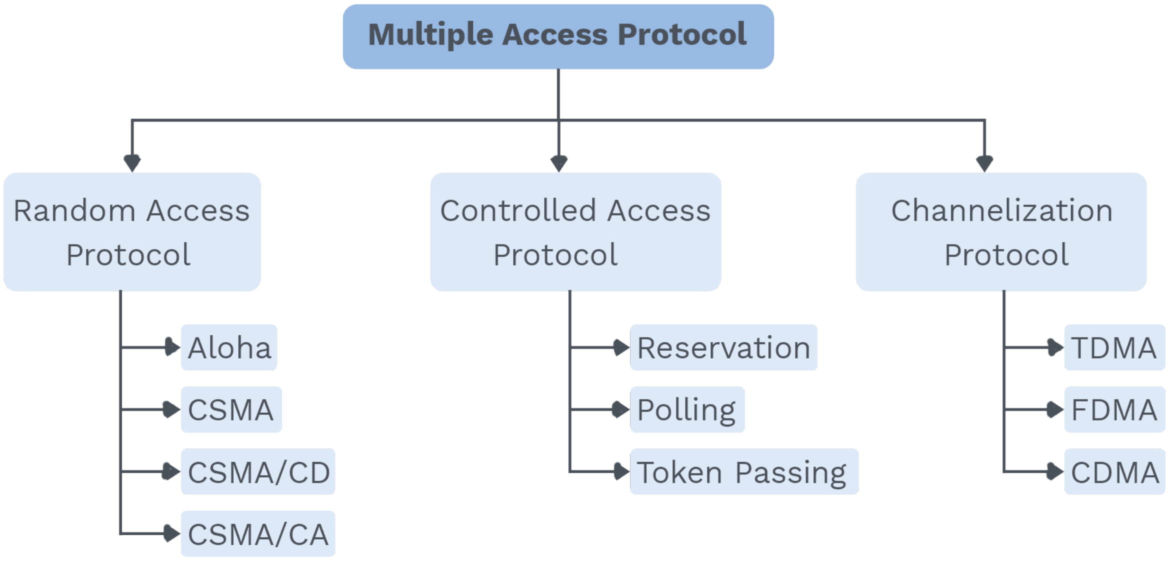


Fig. 3.5 Flow Chart Representing Multiple Access Protocol

Random access:

- It depends on the state of the medium.
- There is no proper time for the station to transmit, that is why the name is Random.
- Each station has a right to the medium without being controlled by another medium.

So, there is collision problem.

Aloha: It was designed for radio, but it can be used on any shared medium.

It can be categorized in 2 ways.

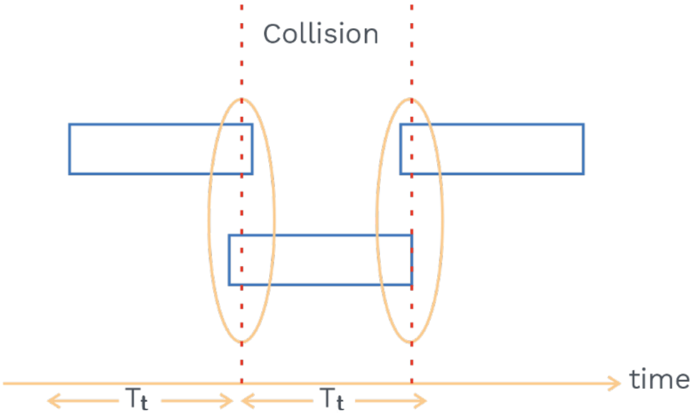
- Pure Aloha
- Slotted Aloha

Pure Aloha:

- It will allow the station to send whenever it has data.
- After sending the data, it waits for acknowledgement from the receiver.
- If it receives the acknowledgement, then the transmission is successful.
- If it doesn't receive the acknowledgement, then the transmission is unsuccessful; after the timeout timer expires, it will resend the data.

Note:

Vulnerable time: This is the time when collision is possible.



Pure aloha vulnerable time = 2 * frame transmission time(T_t)

Throughput in pure Aloha:

Let us assume G = Average number of frame generated during one frame transmitted time.

The throughput for pure ALOHA is $S = G \times e^{-2G}$

The maximum throughput S_{max} = 0.184 when G =(1/2).
G = 1/2, it means when 1/2 frame is transmitted in one T_t time or in other way 1 frame is transmitted in $2T_t$ time.

The maximum efficiency of Pure Aloha is very less due to the large number of collisions.

Q7 **Consider a 100-bits frame is transmitted by a pure Aloha network on a shared channel having, bandwidth of 100 kbps. Calculate the throughput if 250 frames/second is produced by the system(all station together)?**

Sol: Frame transmission time = 1msec

Now station is producing 250 frames in 1 sec, that means (¼) frame in 1 msec
The throughput for pure ALOHA is $S = G \times e^{-2G}$ which will give 0.152.
Now 250 * 0.152 = 38 frames will survive out of 250 frames.

Slotted Aloha:

- In slotted Aloha, data can be transmitted by any station at any given time slot, but the only condition is that a station has to begin its transmission at the start(beginning) of the time slot. A station has to wait until the starting of the next time slot in case it misses the starting of a given slot.
- In this, no station sends the data in the middle.

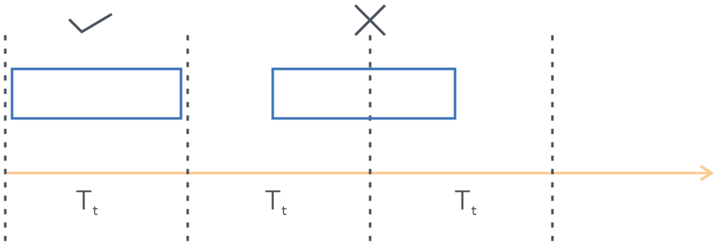


Fig. 3.42 Diagrammatic Representation of Data Transmission in Slotted Aloha

Vulnerable time = T_t

Throughput of slotted Aloha:
The average number of successful transmissions for slotted Aloha is $S = G \times e^{-G}$. The maximum throughput S_{max} is 0.368, when G = 1.

Rack Your Brain

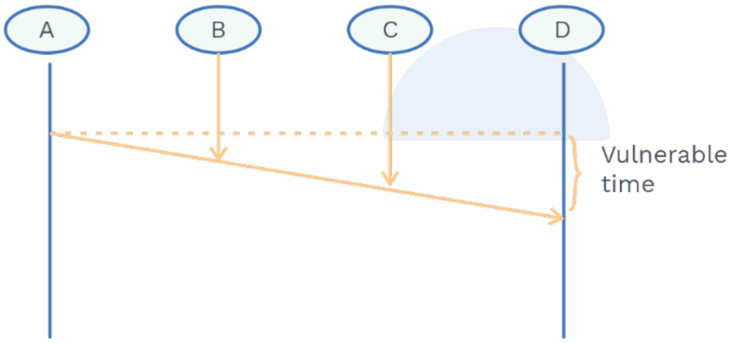
Consider a 100-bits frame is transmitted by a slotted Aloha network on a shared channel having bandwidth of 100 kbps. Calculate the throughput if the system (all stations together) produces 250 frames/second?

Comparison between slotted aloha and pure aloha:

Pure Aloha	Slotted Aloha
Station can send their data at any time	Can transmit only at the beginning of slot
Vulnerable time = 2 * T_t	Vulnerable time = T_t
Throughput = G × e^{-2G}	Throughput = G × e^{-G}

Carrier sense multiple access protocols:

- It was developed to minimize the chance of collision and increase efficiency.
- It was based on the principle “Sense before transmit”.
- The vulnerable time for CSMA is the propagation time Tp. This is the time needed for a signal to propagate from one end of the medium to the other.
- In the given diagram station A sends its data to station D in time T_p, when, station B wants to transmit its senses and similarly, C senses.



What should a station do when the channel is busy or idle?

There are 3 methods for this:
1-persistent method,
non persistent method,
and the p-persistent method

1-persistent method:

If the station finds the line idle, it sends its frame immediately,
If the channel is not idle, the station will continuously sense the channel.

Non persistent method:

If the station finds the channel idle, it sends its frame immediately.
Stations will wait for a random amount of time if they find the channel is busy, and when the channel is idle, it will send the frame again.

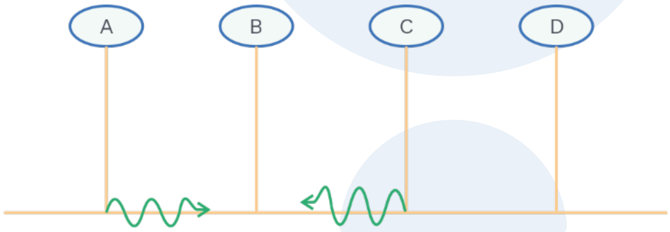
P persistent method:
In this method the station sends its frame with probability p, and it will not transmit with probability q = 1 - p. The station waits for the beginning of the next time slot and checks the line again. Now if the line is idle, it sends the frame with probability p and defers with probability q.

Carrier sense multiple access with collision detection:
Carrier sense multiple access with collision detection (CSMA/CD) elongate the CSMA to handle the collision.

Note:

Each station can sense the carrier only at its point of contact with the carrier.

Let us understand how a station senses the channel.
Station A wants to send data to station C, and station C wants to send data to station A.



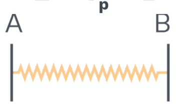
A will think channel is free
let me transmit

C will think channel is free
let me transmit

It can be clearly observed that a collision is about to happen!! How to prevent this?
We need a restriction on the frame size.
Each station must transmit the data packet of size whose transmission delay is at least twice of its propagation delay.

Transmission delay >= 2 x Propagation delay
From this we can derive what will be the minimum size of the packet from ‘B’ to ‘A’.

$$\mathbf{L >= 2 * T_p * B}$$



Collision happens here when A is about to transmit, B can detect collision if it receives collision signal/jamming signal and it’s still transmitting data.

$$T_t \geq T_{P(data)} + T_{P(jamming)}$$

$$\boxed{T_t \geq 2T_p}$$

Rack your Brain

a) What is the minimum packet length when T_p = 1msec and Bandwidth = 1Mbps

b) Guess the differences between Aloha and CSMA/CD

c) Is the throughput of CSMA/CD greater than Aloha?

Efficiency:

Useful time = Transmission delay of data packet = T_t

Total time = Time during collisions + Propagation delay of data packet + T_t

= c x 2 x T_p + T_p + T_t (where c = Number of contention slots).

Efficiency = useful time / total time

T_t /(c x 2 x T_p + T_p + T_t)

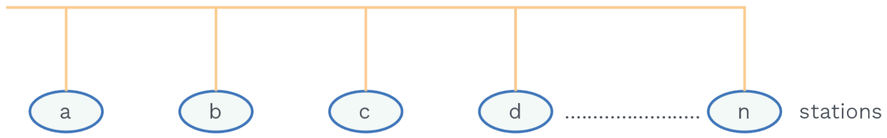
Analysis using probability gives the Average number of collisions before a successful transmission = e

Which leads to c = e

Now Efficiency = T_t /(e x 2 x T_p + T_p + T_t)

Note:

What is the average number of collisions before successful transmission?



P(success) = ${}^nC_1 \times p \times (1-p)^{n-1}$...1

In order to find maximum value we need to differentiate w.r.t. p

dP/dp = 0

On solving we get p = 1/n putting this value in (1) we get

Now P(success)$_{max}$ = $(1-1/n)^{n-1}$

If there are large number of stations n –> ∞

$$= \lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{n-1} = \frac{1}{e}$$

Number of times that a station request before successfully transmitting the data packet,

$$= 1/P_{max} = 1/(1/e) = e$$

Efficiency of standard ethernet = 1/ (1 + 6.44a) where a = T_p / T_t

Points:

- It is used in Wired LAN (802.3)
- Probability of successful transmission = ${}^{n}C_{1}$ x p x $(1-p)^{n-1}$

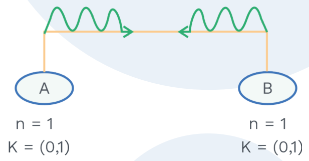
Let us understand; what is Backoff time?

In CSMA / CD protocol, after the occurrence of a collision, the station waits for some random backoff time and then retransmits, and this waiting time for which the station waits before retransmitting the data is called backoff time.

Backoff time = k * timeslot (station chooses number K and timeslot is one round trip time).

Note:

If collision is happening for nth time then station will choose a random number **k** from ($0,2^{n}$ - 1)



Possibilities:

A	B (Description)
0	0 Both station will collide
0	1 Station A will win
1	0 Station B will win
1	1 Both station will collide

Concept Building Exercise

When A = 0 and B = 1, why station A won ! How do you conclude?

Let's see scenario for A = 1 and B = 0

Backoff time for A = 1 * RTT (A has to wait for 1 RTT)

Backoff time for B = 0 *RTT (B does not have to wait)

Note:

In this algorithm, Backoff time increases exponentially
And collision probability decreases exponentially
It shows a capture effect (if the host wins the collisions for one time, it is going to win more numbers of times).

Q8 **Consider a CSMA/CD network having a bandwidth of 10 Mbps, and the minimum frame size is 512 bits for the correct operation of the collision detection process. What will be the minimum frame size when tha bandwidth is increased to 100 Mbps by keeping the propagation delay constant?**

Sol: Frame size = K * data rate

Data rate = 10 Mbps then minimum frame size = 512 bits

Data rate = 100 Mbps then minimum frame size = 5120 bits

Note:

If bandwidth increases, frame size can also increase.

Previous Years' Question

Q. A network with CSMA/CD protocol in the MAC layer is running at 1 Gbps over a 1 km cable with no repeaters. The signal speed in the cable is 2 x 10^8 m/sec. The minimum frame size for this network should be:

a) 10000 bits	**b)** 10000 bytes
c) 5000 bits	**d)** 5000 bytes

Sol: a) **(GATE-2005)**

Previous Years' Question

Q. Consider a network using the pure ALOHA medium access control protocol, where each frame is of length 1,000 bits. The channel transmission rate is 1 Mbps (10^6 bits per second). The aggregate number of transmissions across all the nodes (including new frame transmissions and retransmitted frames due to collisions) is modelled as a Poisson process with a rate of 1,000 frames per second. Throughput is defined as the average number of frames successfully transmitted per second. The throughput of the network (rounded to the nearest integer) is ____________

Sol: 135 **(GATE-2021)**

Controlled access protocol;

Which station will send the packet ! this is done by taking the information from all other station.

In this protocol basically 3 methods are used:

- Polling
- Reservation
- Token Passing

Polling:

There is a Centralized controller which polls 'stations', and gives them an opportunity to send one packet.

All the data which needs to exchange must go through the controller.

Disadvantage in polling:

There is a high overhead of polling messages.

Stations have to depend on controller.

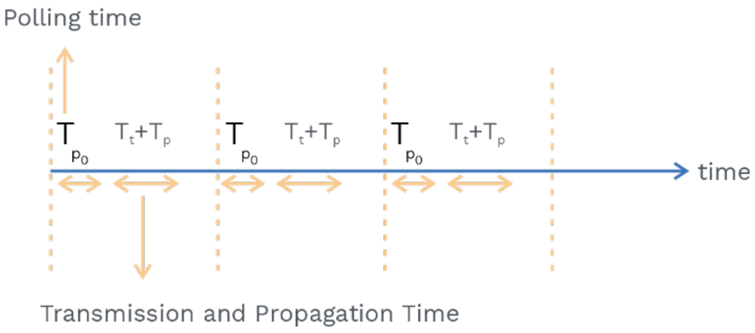


Fig. 3.6 Data Transmission in Polling

Efficiency = Useful time / total time

Useful time = T_t

Total time = $T_{po} + T_p + T_t$

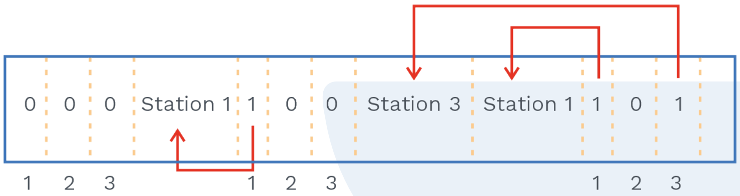
$= T_t / (T_{po} + T_p + T_t)$

Reservation:

Station that wants to send data needs to make a reservation.

Time is divided into intervals.

In each interval, a reservation frame precedes the data frames sent in that interval.



In the above figure there are 3 slots made if 3 stations want to send data. In the first interval, only station 1, and station 3 made a reservation and in the second interval.

Only station 1 made a reservation.

Token passing:

Stations are connected in the form of ring.

Access is granted through the token.

when the station receives the token, it can send a frame (if it has frame) before it passes the token to the next station; if the station does not have a frame simply, it will pass the token to the next station.

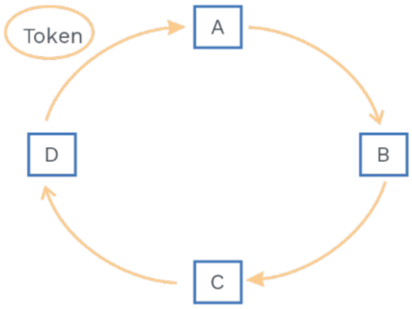


Fig. 3.7 Diagrammatic Representation of Token Ring

Early token reinsertion

Efficiency = 1/(1 + a/N) and

Delayed token reinsertion

Efficiency = 1/(1+{a(1 + 1/N)})

Channelization protocol:

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code between different stations.

Time division multiple access:

Time of the link is divided into fixed-size intervals called time slots or time slices.

Each station can transmit the data in its time slot only.

Let's say there are 3 stations A,B and C.

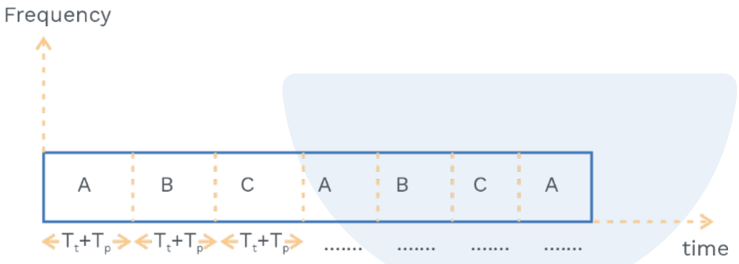


Fig. 3.8 Diagrammatic Representation of TDMA

Slots are given on the basis of Round Robin

Efficiency = Useful time/ total time

$$T_t / (T_t + T_p)$$

Disadvantage:

If the station does not want to send the frame, then also time slot has given, which eventually leads to bad efficiency.

Rack Your Brain

Give the difference between FDMA and FDM !

Frequency division multiple access:

In FDMA, the available bandwidth of the common channel is divided into bands that are separated by guard bands.

Each station has an assigned separate channel.

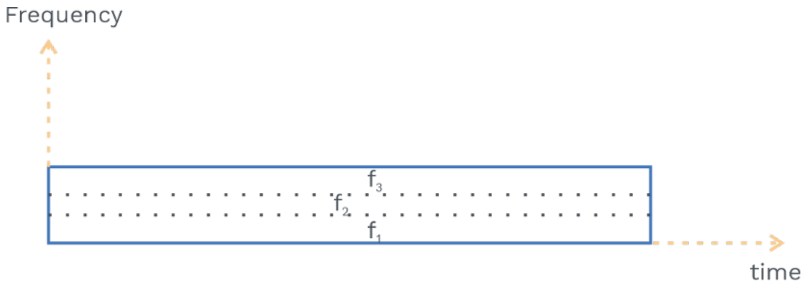


Fig. 3.9 Diagrammatic Representation of FDMA

Wired Lan: Ethernet:

Ethernet is one of the standard LAN technologies used for wired LANs and It is defined under IEEE 802.3.

Till now we have seen Data link layer in two parts.

First one is Data Link Control, and the Second one is Medium Access Control.

But there is only one MAC sublayer in standard ethernet.

Points:

- Topology used is Bus.
- Access control method used is CSMA/CD.
- Encoding Technique used is Manchester.

Let us see Frame format of Ethernet

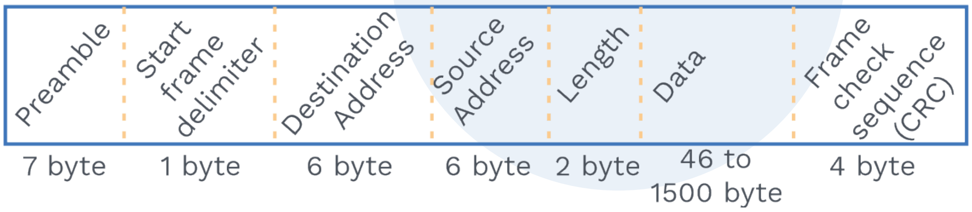


Fig. 3.10 Ethernet Frame Format

Preamble:

a) It contains 7 bytes.
b) It has 0 and 1 in alternate position.
c) It is actually added at the physical layer.
d) It enables frame to synchronize between sender and receiver.

SFD:

It contains 1 byte: 10101011.

SFD tells the station that this is the last chance for synchronization. The last 2 bits are 11 and alerts the receiver that the next field is the destination address.

Destination address: It is a 6 byte field.

It contains the MAC address (physical address) of the destination.

Source address: It is a 6 byte field.

It contains the MAC address (physical address) of the source.

Length: It is a 2 byte field.

Length field describes the number of bytes in the data field.

Note:

The maximum value that can be accommodated in this field = $2^{16} - 1 = 65535$ bytes, but the maximum amount of data that can be sent in an Ethernet frame is 1500 bytes.

Note:

Minimum frame length = 64 bytes
Maximum frame length = 1518 bytes

Data: This field contains actual data also called payload field.
Minimum bytes in data field = 46 bytes.
Maximum bytes in data field = 1500 bytes.

CRC: It has 4 bytes.
This field is used for error detection.

Rack Your Brain

Why 65535 bytes is not allowed in ethernet frame?

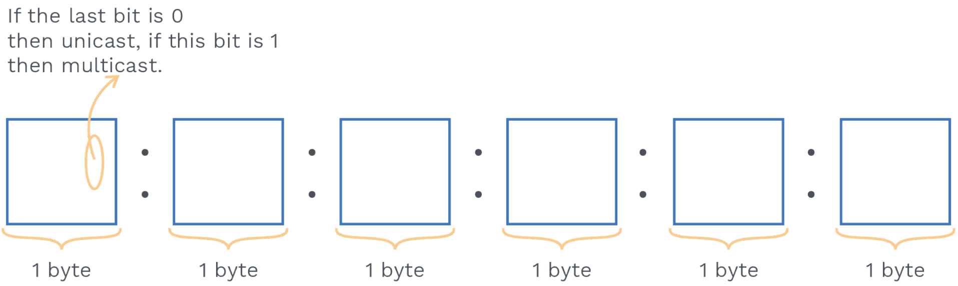
Addressing in ethernet:
Each station has NIC (Network interface card) which has its own physical address.

We already know physical address has 6 bytes.
It is written in hexadecimal code with colon between bytes.
There are 3 types of addresses:

- Unicast
- Multicast
- Broadcast address

Note:

The least significant bit of the first byte defines the type of address. If the bit is 0, the address is unicast; otherwise, it is multicast.



Source address is always unicast.
Destination address can be unicast, multicast or broadcast
The broadcast destination address is a special case of the multicast address in which all bits are 1.

Rack your Brain

Q.1 Define the type of the following destination addresses:

- **a)** 5A:40:20:31:2O:2A
- **b)** 46:60:6B:7E:78:8E
- **c)** FF:FF:FF:FF:FF:FF

Q.2 Why source address cannot be multicast or broadcast?

SOLVED EXAMPLES

Q9 **Compare the data rates for Standard Ethernet, Fast Ethernet, Gigabit Ethernet,**
and Ten-Gigabit Ethernet?

Sol: Standard Ethernet: 10 Mbps, Fast Ethernet: 100 Mbps, Gigabit Ethernet: 1 Gbps, Ten-Gigabit Ethernet: 10 Gbps

Q10 **If an Ethernet destination address is 09:02:02:03:04:05, what is the type of the address (unicast, multicast, or broadcast)? Can this address be a source address?**

Sol: The first byte; i.e 09 => 00001001, the last bit is 1; therefore it is a multicast. No, this address cannot be a source address because a source address cannot be a multicast.

Q11 **The efficiency of Ethernet increases when propagation delay is low, and transmission delay is high. The following statement is true/false.**

Sol: This statement is true because, E = 1/ (1 + 6.44a)

Where, a = T_p/T_t

Q12 **Determine the ratio of the smallest useful data that a frame can carry through ethernet to the largest ethernet frame size?**

Sol: Smallest data size in ethernet frame = 46

Largest frame size = 1518
46/1518

Q13 **Which of the following is/are false about CSMA/CD? (MSQ)**
a) IEEE 802.11 wireless LAN runs CSMA/CD protocol.
b) CSMA/CD is useful for system like ATM
c) Ethernet is based on CSMA/CD protocol
d) CSMA/CD is not suitable for high propagation delay network

Sol: **a,b**

a) IEEE 802.11 wireless LAN runs CSMA/CA protocol.
b) CSMA/CD is not useful for system like ATM because ATM uses interactive methods

Connecting devices:

- Operate at physical layer: Active Hub or repeater
- Operate at physical and data link layer: bridges or 2 layer switches
- Operate at physical, data link layer and etwork layer: Routers or 3 layer switches
Operate at all five layer: Gateways

Repeater:

- It receives the signal at physical layer and before the signal becomes too weak it regenerates the original bit pattern
- A repeater regenerates the signal. It does not amplify signal.

Previous Years' Question

Q) Suppose the round trip propagation delay for a 10 Mbps Ethernet having 48-bit jamming signal is 46.4 μs. The minimum frame size is:

a) 94 **b)** 416
c) 464 **d)** 512

Sol: d) (GATE-2005)

- A repeater connects two **segments** of LAN.
- A repeater forwards every frame, it has no filtering capability (Collision domain remain).

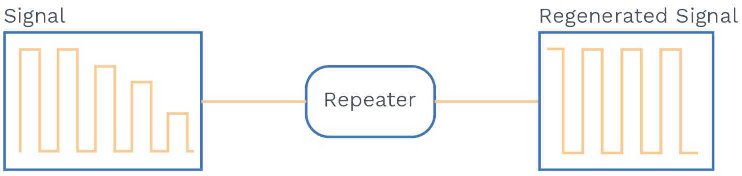


fig 3.11 Regeneration of Signal Using Repeater

Rack your Brain

Difference between repeater and amplifier.

Note:

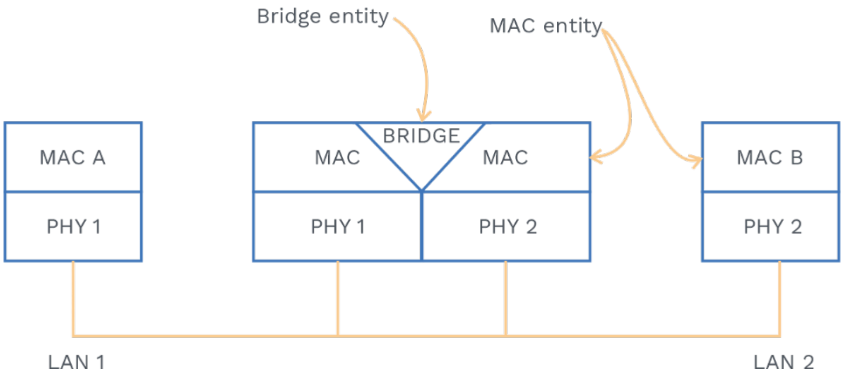
A repeater is not used for connecting two LANs of different protocols.

Active hub:

It is a multiport repeater.
Hubs can also be used to create multiple levels of hierarchy.
Hubs cannot filter data (collision domain remain same).

Bridges:

It works at the physical layer and data link layer.
At the physical layer, it regenerates the signal.
At the data link layer, it can check the MAC address contained in the frame.
It has filtering capability (because it can check the destination MAC address and decide whether the frame has to forward or drop).



Transparent bridge:

- A transparent bridge is a bridge in which the stations are completely unaware of the bridge's existence. It implies that no reconfiguration is required if a bridge is added/deleted from the system.

- It works on transparent and learning process.
- Transparent bridges work fine until there is a redundant bridge which causes a **looping problem.**

Solution of looping problem:

Spanning tree: To prevent the loop path and proper working of forwarding and learning process, there must be only one path between any pair of bridges and that path is maintained using the **Spanning tree algorithm.**

Source routing bridges:

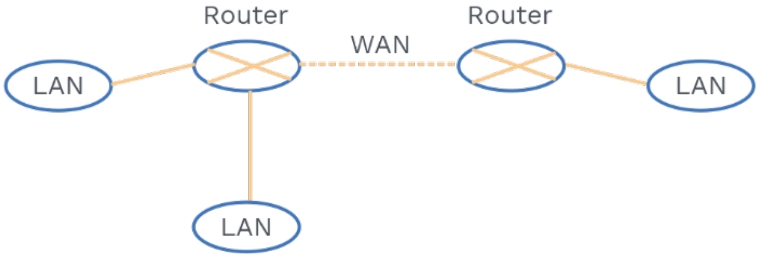
- Here the routing is performed by the source host.
- The frame contains not only the source and destination addresses but also the addresses of all bridges to be visited.
- Source gets these bridge addresses through the exchange of special frames.

Two layer switches:

- It performs at the physical and data link layer.
- A two-layer switch is a bridge having many ports, and it is designed such that it gives a faster(better) performance.
- It takes filtering decisions on the basis of the MAC address of the frame it received. (Each port has a separate collision domain).
- Here, the ports are provided with a buffer.

Router:

- It is used to link two dissimilar LAN.
- It is a 3 layer device which routes the packets based on logical address.
- Routing table is dynamic and updated using a routing protocol.

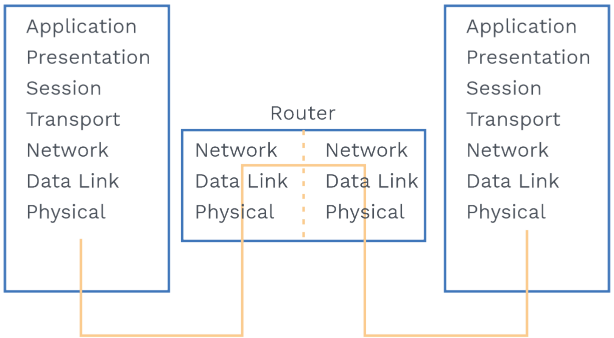


Three layer switch:

- A three-layer switch is a router, which is faster and more sophisticated.
- The switching fabric in a three-layer switch allows faster table lookup and forwarding.
- It can separate broadcast domain.

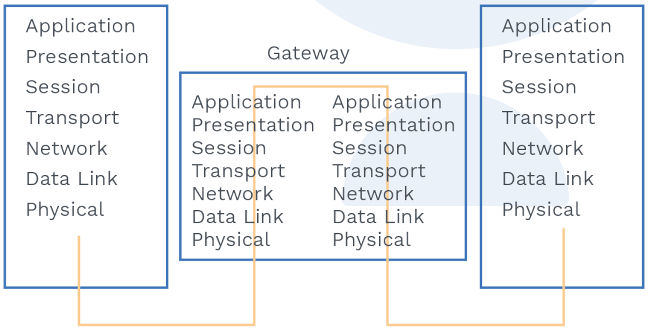
How does a communication through a router happens?

Router can inspect through network layer:



Gateway:

- A gateway is normally a computer that operates in all five layers of the Internet or seven layers of OSI model.
- Gateways can provide security.
- It is used to filter unwanted application-layer messages.



Concept Building Exercise

Q. Which of the following statements are not correct differences between a switch and a hub? *[MSQ]*

a) Switch transmits a signal to all the devices connected to it, hub transmits a signal only to the intended port.

b) Switch works in the physical layer, the hub works at data-link layer

c) Switch works at layer 2 while hub works at layer 1

d) Switch is a smart device, whereas hub is a dumb device

Sol: a) and b)

Hub Transmits a signal to all the devices connected to it; switch transmits a signal only to the intended port

Hub works in the physical layer; Switch works at data-link layer

Previous Years' Question

Q. Which of the following is NOT true with respect to a transparent bridge and a router?

a) Both bridge and router selectively forward data packets

b) A bridge uses IP addresses while a router uses MAC addresses

c) A bridge builds up its routing table by inspecting incoming packets

d) A router can connect between a LAN and a WAN

Sol: b) **(GATE-2004)**

Chapter summary:

- **Data link layer:** Two major function
  1) Data Link Control
  2) Medium Access Control
- Data Link Control performs Framing, Error Control, Flow Control.
- Medium Access Control performs Access Control and Physical Addressing.
- There are two types of framing
  1) Fixed Size Framing
  2) Variable Size Framing
- Error detection can be done by Parity checking, CRC and Checksum.
- Some rules for generating the generator polynomial
  Rule 1: It should not be divisible by x.
  Rule 2: It should be divisible by x+1.
- **Flow control:**
  Set of procedures used to restrict the amount of data that the sender can send before waiting for an acknowledgement.
  Flow control can be done in Noiseless channels and noisy channel.
  In Noiseless channel: Stop and Wait
  In Noisy channel: Stop and Wait ARQ
  Go back N ARQ
  Selective repeat ARQ
- Available Sequence number >= Sender window Size + Receiver window Size

- Stop and Wait ARQ = Stop and Wait + Time Out Timer + Sequence number in Data Packet + Sequence number in ACK Packet.
- Maximum Available Sequence Number in GBN >= N+1.
- Maximum Available Sequence Number in SR >= N+N.
- Maximum throughput of pure Aloha is 18.4%.
- Maximum throughput of slotted Aloha is 36.8%.
- Frame transmission delay in CSMA/CD must be twice than propagation delay.
- Minimum frame length = 64 bytes
- Maximum frame length = 1518 bytes