16/12/22 Unit-4 Data Link Layer & Protocols

* Error in Communication : the message which is transmitted from sender side, if it is altered due to some reason between the path (reason may be electromagnetic field, External interference) then that alteration is known as Error. This may receiver will not receive that data exactly.

Ex: 10110101 → this one alter to '1'

↓

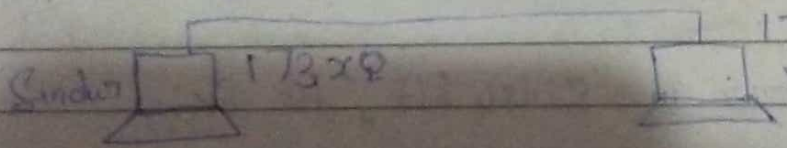
This one alter to '0' then Error.

Types of Error

(i) Single bit Error : There is alteration of only one bit.

(ii) Burst Error : Multiple bits are affected simultaneously.

Now question is that, how receiver will know there is Error or bits are altered? How receiver will conclude this. At machine level it is important to know whether data is received properly or not.



* approach of Error handling.

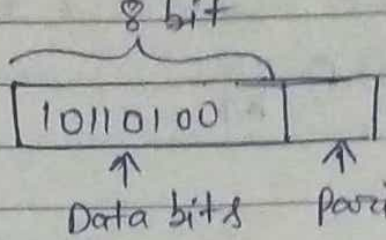
→ Error handling

→ Error Recovery.

* Error detection technique,

(1) Parity Check technique :-

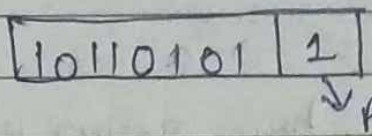
Data is transmitted frame by frame & frame is a logically arranged pattern of data. Bits are contained in the frame. Frame may contain some control information.



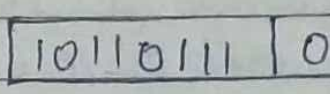
This technique is divided into two Subcategory :-

(i) Even Parity :- It says when a bit pattern is being transmitted the total no. of ones in the bit pattern should be even.

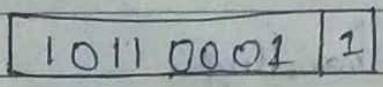
If bits are odd, then put '1' in parity bit



If bits are already even then, put '0' in parity



When there is Error in Single bit, for Ex:



will be detected

no. of 1's are not even so, Error and data will be discarded.

When there will be Error in two bits.

Ex:

10110000	1

No. of 1's are even so Error will not be detected.

Hence, we Conclude that 50% of the Error remains Undetected.

17/12/22

Checksum technique,

Checking the Sum based Concept.

* Binary addition →

5 → 101
6 → 110
1011
↓
Carry

0+0=0
0+1=1
1+0=1
1+1=0 (Carry 1)

5 → 101
3 → + 011
8 1000

1's Complement : 1011 → 0100 (1's complement)

101
010 (1's Complement)
111

adding a binary no. to its 1's Complement gives all bits 1.

(1) The binary bit which is being transmitted is divided into 'K' parts of 'n' bits each. for convenience we take 2 parts.
(2) add both the parts.
(3) find the 1's complement of the sum
(4) Send this 1's complement with binary data.

At the receiver side when receiver will find out the sum of the bits & 1's compliment of the sum which is received with it, all '1' should be obtained.

Ex: Binary data 1010 0001 → divide in 2 parts

```
  1010
+ 0001
------
  1011   Sum
------
  0100   1's compliment
------
```

* Data send at receiver's side

10100001	0100

at receiver side,

```
  1010
  0001
------
  1011   Sum →
  0100
------
  1111   Sum →
```

(1) Receiver will again divide the bits in two parts
(2) Part with sum is calculated.
(3) Then calculated sum is added to the received sum which is actually 1's compliment of the sum at sender's side
(4) If all the bits come to be 1 then data will be received otherwise discarded.

when msb are exchanged?

* Flow & Error Control,

Bandwidth help to find the bit rate of data. Sender se lekar receiver tk jo data transmit karta h to jo media implimented hai ek hi bandwidth ka nahi hoga. Uss bandwidth ka specification alag alag ho sakta hai. Allo, it is possible that data sender is sending & receiver is receiving there architectural configuration may be different. Allo, Speed will be same, it is not at all necessary.
In the implimentation of network, the architectural platform will be different for station to station. Sender jis speed se data bhej raha hai receiver usi speed se propagate karega smoothly, ye jaruri nahi hai. yo Ensure kar bhi lo to receiver ke paas usi speed se receive hoga ye bhi jaruri nahi hai. Processing Speed may vary becoz many devices participate in network.

* Flow Control :- It says control the flow of bits becoz the flow of data should be properly controlled. It is necessary becoz the sender which is sending the data should be properly received and processed by the receiver. Flow control means limit the flow of data, upto which amount that receiver can easily processes in each processing cycle. Ya phir uss amount tak limit kar, jis amount me aage bandwidth allow kar rahi h data ko aage bhejne ke liye. Limit the speed of data in any way.

Error Control,

(i) noiseless channel } Transmission
(ii) noisy channel } channel.

(i) noiseless ÷ no External Interference. So there is no chances of error. It requires only flow control mechanism. They doesn't exist in practical manner. It is only for logical understanding.

(ii) noisy channel ÷ External interference exists. Here the Probability of Error occur bandwidth specifications are different. Includes both flow & Error control.

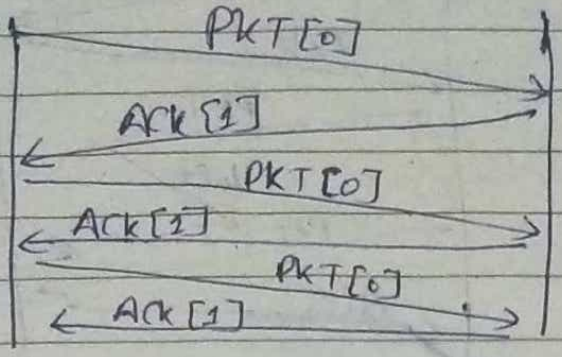
20/12/21

Stop & wait protocol,

It is protocol of flow control. protocol means set of rules. In this Sender will send some data and wait upto the time the receiver is able to process that receive data & it will become ready to receive further data from Sender's side. So that channel do not get overloaded.

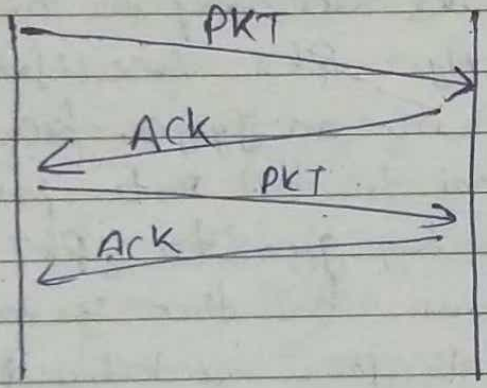
When Stop & wait protocol is implemented Sender is allowed to send a packet. after transmission of 1st packet, Sender will stop & then Sender will wait for the acknowledgement from receiver's side.

Acknowledgement (Ack) is that particular information that informs Sender about the successful receiving & processing of the packet (PKT) and now receiver is ready to receive the next PKT So Sender is allowed to send the next PKT.



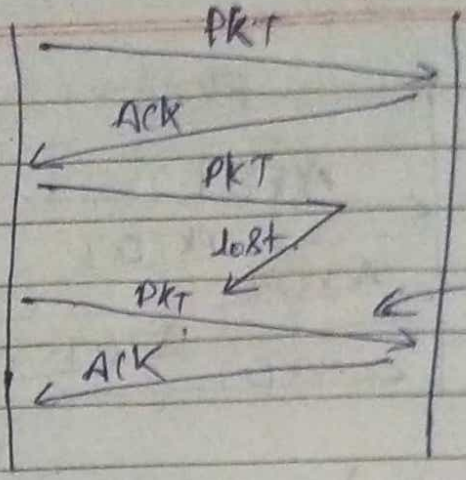
Since noisy Channel is always busy. there may be probability of External interference. So, always there is a prob that there is Error in PKT, PKT loss etc.

Case: 1
ideal Case



Sender ek PKT bhejta hai aur Stop ho jata hai. Stop karke wait karte h, kab tak, jab tak receiver ke side se Ack nahi aa jaaye. To kyo aisa h ki Sender wait hi krta rahega kaafi Samay tak Jab tak receiver Acknowledge nahi karega. To isme ek Timer ka Concept involve hota hai.

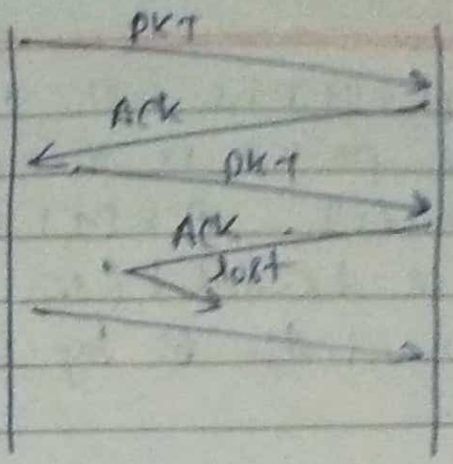
Case: 2
Packet loss



Yahan, when PKT is lost, receiver will not get PKT so receiver will not Acknowledge. Sender has stop after sending PKT. But Sender ko nahi malum that in between path PKT is lost. Sender is waiting for Ack and Receiver is also waiting for PKT. Ack nhi pahuncha to Sender wait hi karta reh jayega. Isi liye timer ka ek role h. We need a timer, if Ack doesn't come till this time. Timer works according to bandwidth & processing speed. Agar PKT successfully pahunch jyga to Ack itne time me aa jyga. bcoz we know bandwidth known hai to bit rate pata lag jata hai. Agar bit rate pata lag gya to, PKT me bits kya specification agar known hai then ye pata lag jyga, ki PKT ko receiver tak jaane me kitna time lagega and same with the Ack. So if we add both the time of PKT and Ack including some marginal time we can get the timing for timer.

In Case of PKT lost, Sender will send the PKT & wait till timer is over. If timer is over & still Ack is not received, Sender will assume that PKT is lost somewhere. So, Sender will retransmit the PKT. Again Ack will be received and transmission will continue.

Case: 3



In this case, pkt is sent successfully at receiver side but Ack sent by receiver is lost. So when timer is over sender will assume that pkt is not reached at receiver. So it will retransmit the same pkt bcoz Ack is not received but pkt is received at receiver side. So, this is a case of pkt duplication at the receiver's side. So we need a strategy to manage pkt duplication.

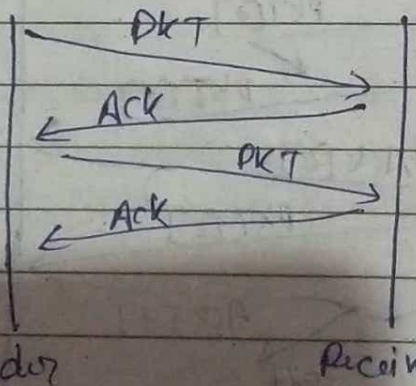
When pkt[0] is transmitted, it is acknowledge by Ack[1]. It means receiver is expecting

pkT[1]. pkT[1] is lost. So pkT[1] again sent. Ack[0] means packet is received. No, receiver is expecting pkT[0]. Ack[1] lost. Now pkT[0] again sent packet is discarded (NC; pkT[0]). Now receiver acknowledge by Ack[1] and expecting pkT[1].

Ack[0] → informs receiver is expecting pkT[0]
Ack[1] → receiver is expecting pkT[1].

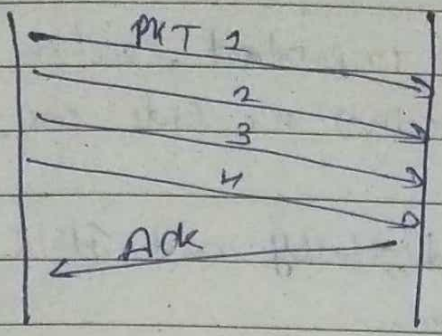
Q.) What is Stop & wait?
Q.) What is flow control & role of Stop & wait in that?
Q.) What are cases of Stop & wait?
Q.) What is the use of acknowledging pkT[1] by Ack[0] & pkT[0] by Ack[1].
Q.) How cases of pkT loss & Ack loss handled.

21/12/21



In this case, when 1st pkT is sent we get 1st Ack and for 2nd pkT other Ack is sent. So there are four transmission out of which two are message part and rest two are acknowledgement part. Jeha hi pkT bhej rahe the waha 2 hi bheje aur 2

Acknowledgment ke liye use kiye. So bandwidth utilization reduces to 50% in ideal case and reduces more in case of PKT loss & Ack lost.

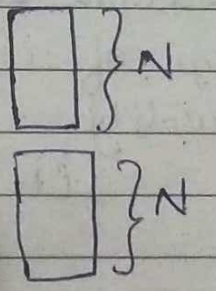


4 PKT send at one time & one Ack, then Efficiency increases to 80%

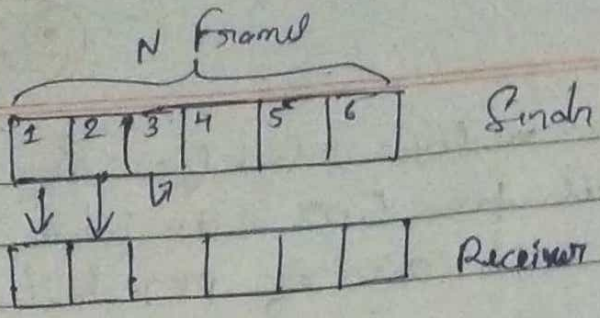
So if we ~~note~~ acknowledge multiple PKT's with Single Acknowledgment then bandwidth Efficiency Can be improved.

* GO-BACK-N ARQ (or Sliding window protocol)

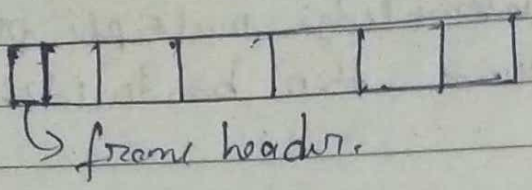
'N' no. of PKTS are transmitted in one time. 'N' no. of packets transmitted and we get a Cummulative acknowledgment ~~that~~ which acknowledge that 'N' no. of frames has reached.



Send window ⇒ Logical arrangement of N frames is called Send window. It is an abstract Concept. a kind of data Structure, inside which frames are logically arranged. We will send one window & when we will get Ack for that proceed to next window.

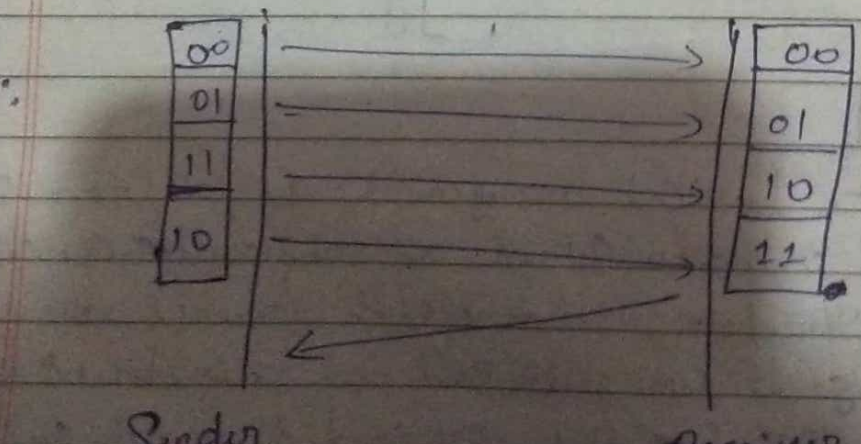


Numbering of frame is important, taki properly numbered then only receiver ke side me finally compose kar payega.
Transmission happens digitally and it is based on binary number system.

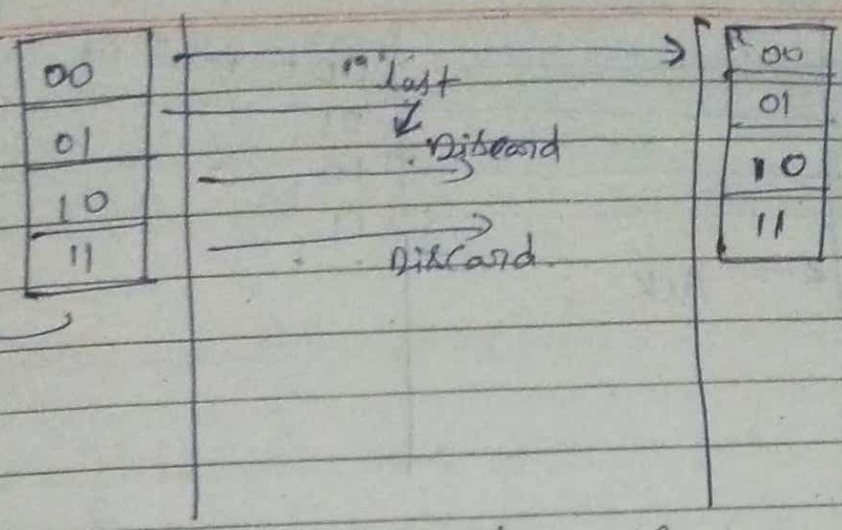


If we use 2 bit how many frames we can make.
So, a window consist of 4 frames in this case. So, window size depends on the no. of bits used.

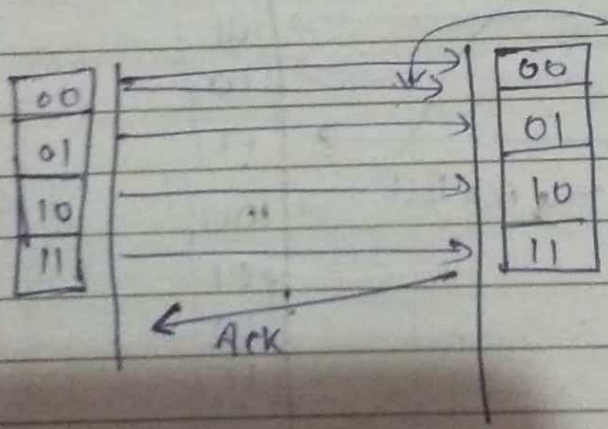
If we use n bits in frame header, how many frame can be addressed uniquely.
So, size of window should be 2^n.
But in reality, it is ~~2^n~~ $2^n - 1$



Case: 2
Frame lost



In this case, one frame is successfully received & 2nd frame is lost, now receiver is expecting frame with frame header 01. Sender doesn't know that ~~frame is~~ frame is lost and will send frame [10] and [11] but receiver will discard them and receiver will not acknowledge. Now, Sender will wait for Ack till the timer is over. When timer is over and Ack is not receive then Sender will assume frame is lost. So, Sender will 'Go-Back-N' i.e; 'go to Nth place.

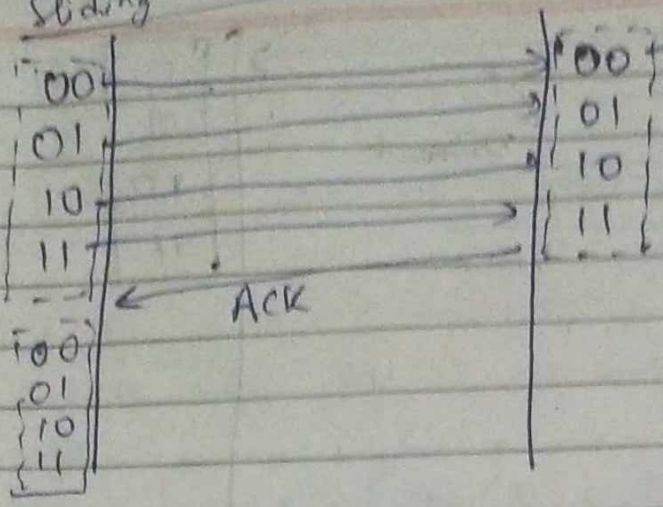


It will again send frame from Nth place and if receiver has already received that frame then it will discard that frame and transmit the next frame.

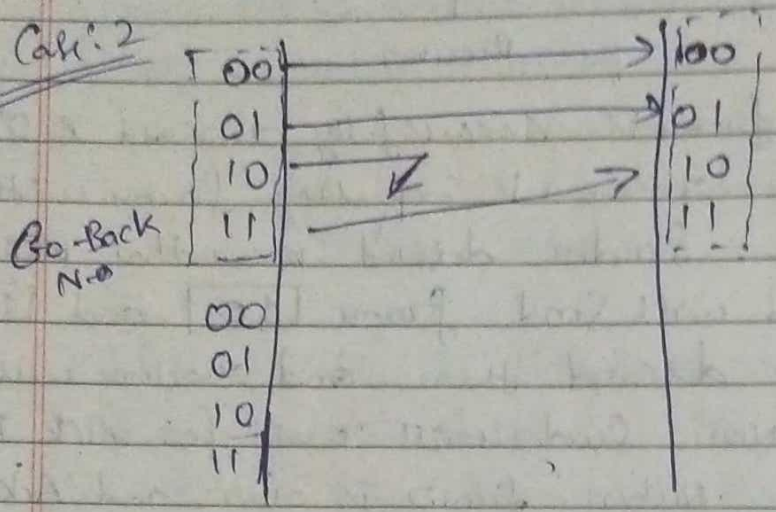
Revision

Sliding window Protocol

Case: 1

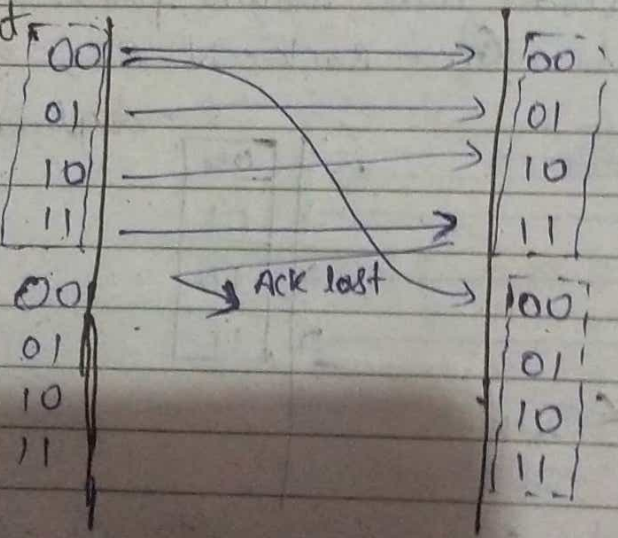


Case: 2



Case: 3

Acknowledgment loss:

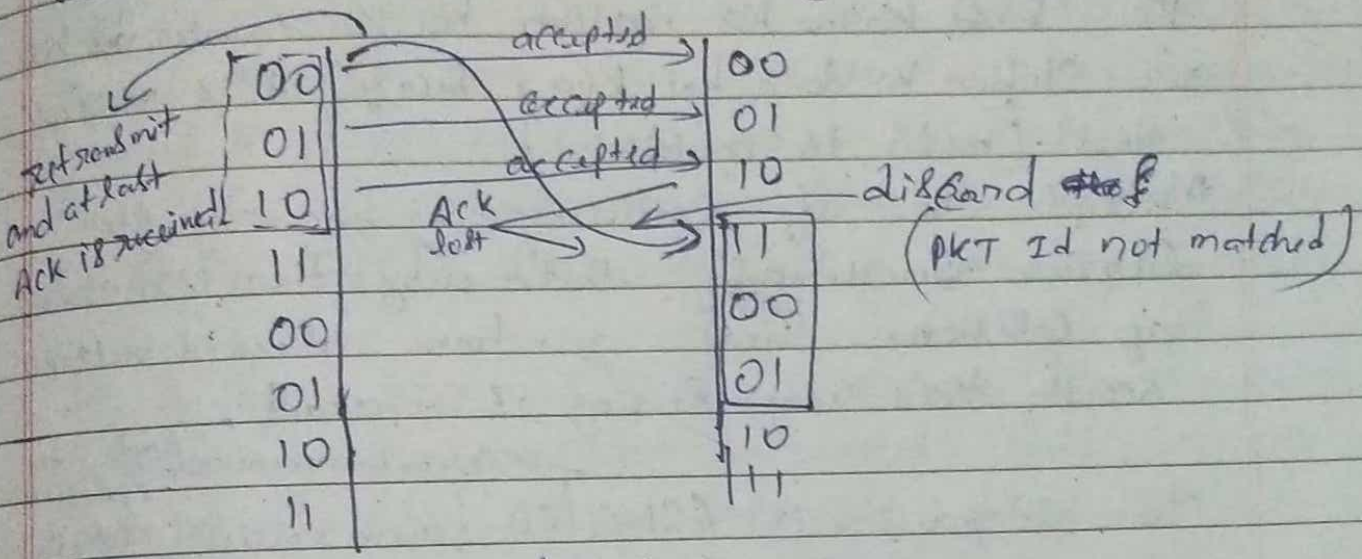


when Ack is last sender assume karega ki packet loss ho gaya. the sender will again go back to nth place and find 00 (PKT Id) of first window and will be received by the 2nd window. Since, PKT Id 00 (of 2nd window matched) then it will be received.

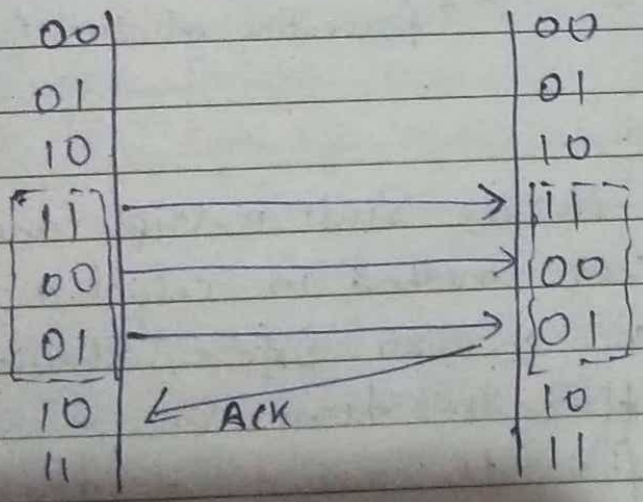
So there will be a wrong receiving. Sender is retransmitting the 1st window again.
This is the drawback of Go-Back-N protocol.

* To overcome this drawback, :

If n bits are used for n pkt's, 2^n ~~bits~~ Packets can be uniquely addressed or window size is 2^n. Logically this is correct but, practically it is $\boxed{2^n - 1}$



Slide to next window.



In Case of Ack lost, if Ack is lost & we kept the size of window n then there will be duplication of packet and receiver ke side me wrong packet receive ho jata hai Instead jab $2^n - 1$ tab correctly receive hota hai.