Experiment No: - 7

Experiment Name: - Implementation of Sliding Window Protocol.

Name: - Shrikrushna C Gundre

Roll No: - DCSE 12

PRN No: - 2122010191

✞ Stop and Wait ARQ:

The Stop and Wait ARQ offers error and flow control, but may cause big performance issues as sender always waits for acknowledgement even if it has next packet ready to send. Consider a situation where you have a high bandwidth connection and propagation delay is also high (you are connected to some server in some other country through a high-speed connection), you can't use this full speed due to limitations of stop and wait.

- **Transmission Delay (Tt):** Time to transmit the packet from host to the outgoing link. If B is the Bandwidth of the link and D is the Data Size to transmit.

$$Tt = D/B$$

- **Propagation Delay (Tp):** It is the time taken by the first bit transferred by the host onto the outgoing link to reach the destination. It depends on the distance d and the wave propagation speed s (depends on the characteristics of the medium).

$$Tp = d/s$$

- **Efficiency:** It is defined as the ratio of total useful time to the total cycle time of a packet. For stop and wait protocol,

Total cycle time = Tt(data) + Tp(data) + Tt(acknowledgement) +Tp(acknowledgement)

= Tt(data) + Tp(data) + Tp(acknowledgement)

= Tt + 2*Tp

Since acknowledgements are very less in size, their transmission delay can be neglected.

Efficiency = Useful Time / Total Cycle Time

= Tt/(Tt + 2*Tp) (For Stop and Wait)

= 1/(1+2a) [Using a = Tp/Tt]

Effective Bandwidth(EB) or Throughput – Number of bits sent per second.

EB = Data Size(D) / Total Cycle time(Tt + 2*Tp)
Multiplying and dividing by Bandwidth (B),
= (1/(1+2a)) * B [Using a = Tp/Tt]
= Efficiency * Bandwidth

Capacity of link: If a channel is Full Duplex, then bits can be transferred in both the directions and without any collisions. Number of bits a channel/Link can hold at maximum is its capacity.

Capacity = Bandwidth(B) * Propagation(Tp)
For Full Duplex channels,
Capacity = 2*Bandwidth(B) * Propagation(Tp)

In Stop and Wait protocol, only 1 packet is transmitted onto the link and then sender waits for acknowledgement from the receiver. The problem in this setup is that efficiency is very less as we are not filling the channel with more packets after 1st packet has been put onto the link. Within the total cycle time of Tt + 2*Tp units, we will now calculate the maximum number of packets that sender can transmit on the link before getting an acknowledgement.

In Tt units ----> 1 packet is Transmitted.

In 1 units ---- > 1/Tt packet can be Transmitted.

In Tt + 2*Tp units -----> (Tt + 2*Tp)/Tt

packets can be Transmitted

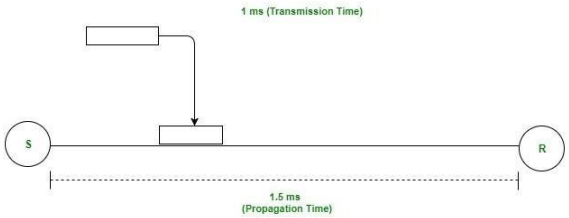
------> 1 + 2a [Using a = Tp/Tt]

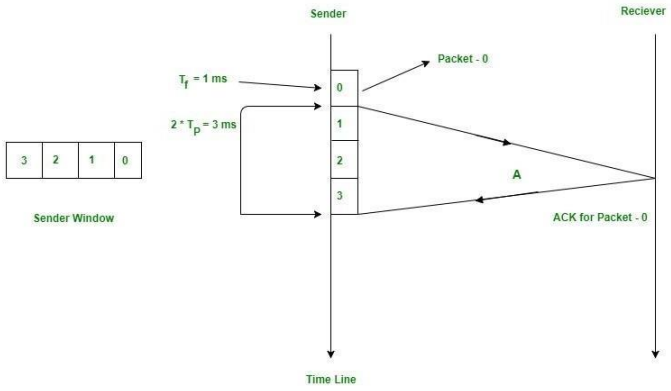
Maximum packets That can be Transmitted in total cycle time = 1+2*a Let

me explain now with the help of an example.

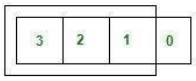
Consider Tt = 1ms, Tp = 1.5ms.

In the picture given below, after sender has transmitted packet 0, it will immediately transmit packets 1, 2, 3. Acknowledgement for 0 will arrive after 2*1.5 = 3ms. In Stop and Wait, in time 1 + 2*1.5 = 4ms, we were transferring one packet only. Here we keep a window of packets that we have transmitted but not yet acknowledged.

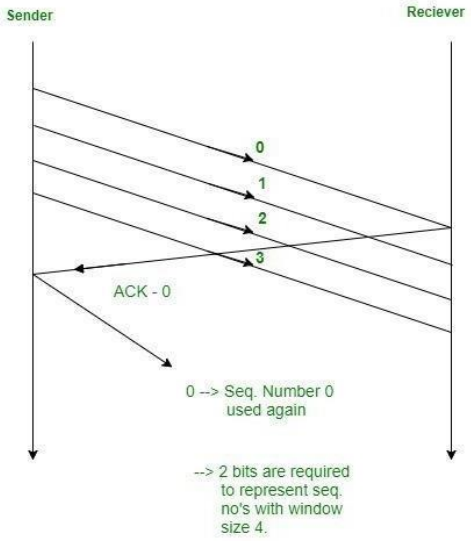


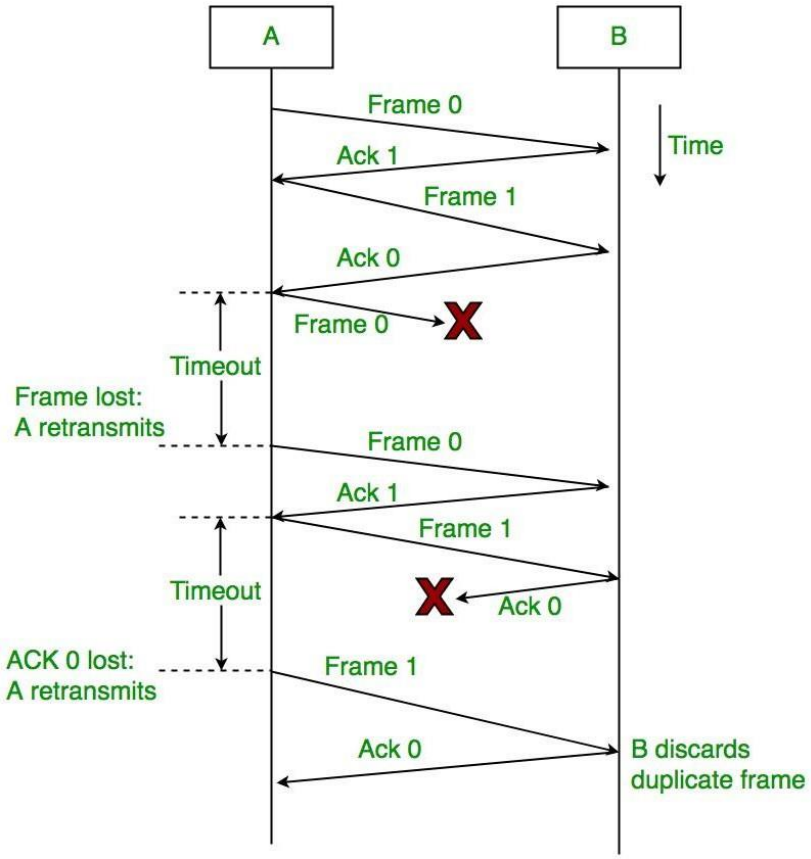


After we have received the Ack for packet 0, window slides and the next packet can be assigned sequence number 0. We reuse the sequence numbers which we have acknowledged so that header size can be kept minimum as shown in the diagram given below.



Window Slided On Receiving Ack of Packet - 0





- **Code**

```
#include<stdio.h> int

main()

{

   int w,i,f,frames[50];     system("clear");

printf("Enter window size: ");     scanf("%d",&w);

printf("\nEnter number of frames to transmit: ");

scanf("%d",&f);


   printf("\nEnter %d frames: ",f);

   for(i=1;i<=f;i++)

scanf("%d",&frames[i]);

   printf("\nWith sliding window protocol the frames will be sent in the following manner
```

```
(assuming no corruption of frames)\n\n");
   printf("After sending %d frames at each stage sender waits for acknowledgement sent by
the receiver\n\n",w);     for(i=1;i<=f;i++)
   {
      if(i%w==0)
      {
         printf("%d\n",frames[i]);           printf("Acknowledgement of above frames
sent is received by sender\n\n");
      }       else
printf("%d ",frames[i]);
   }
   if(f%w!=0)       printf("\nAcknowledgement of above frames sent is received
by sender\n");     return 0;
}
```

- **Output**

```
Enter window size: 3

Enter number of frames to transmit: 5

Enter 5 frames: 12
55
88
99
46

With sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)

After sending 3 frames at each stage sender waits for acknowledgement sent by the receiver

12 55 88
Acknowledgement of above frames sent is received by sender

99 46
Acknowledgement of above frames sent is received by sender
```