What is Flow-Control in networking?

In a network, the sender sends the data and the receiver receives the data. But suppose a situation where the sender is sending the data at a speed higher than the receiver is able to receive and process it, then the data will get lost. **Flow-control** methods will help in ensuring this. The flow control method will keep a check that the senders send the data only at a speed that the receiver is able to receive and process. So, let's get started with the blog and learn more about flow control.

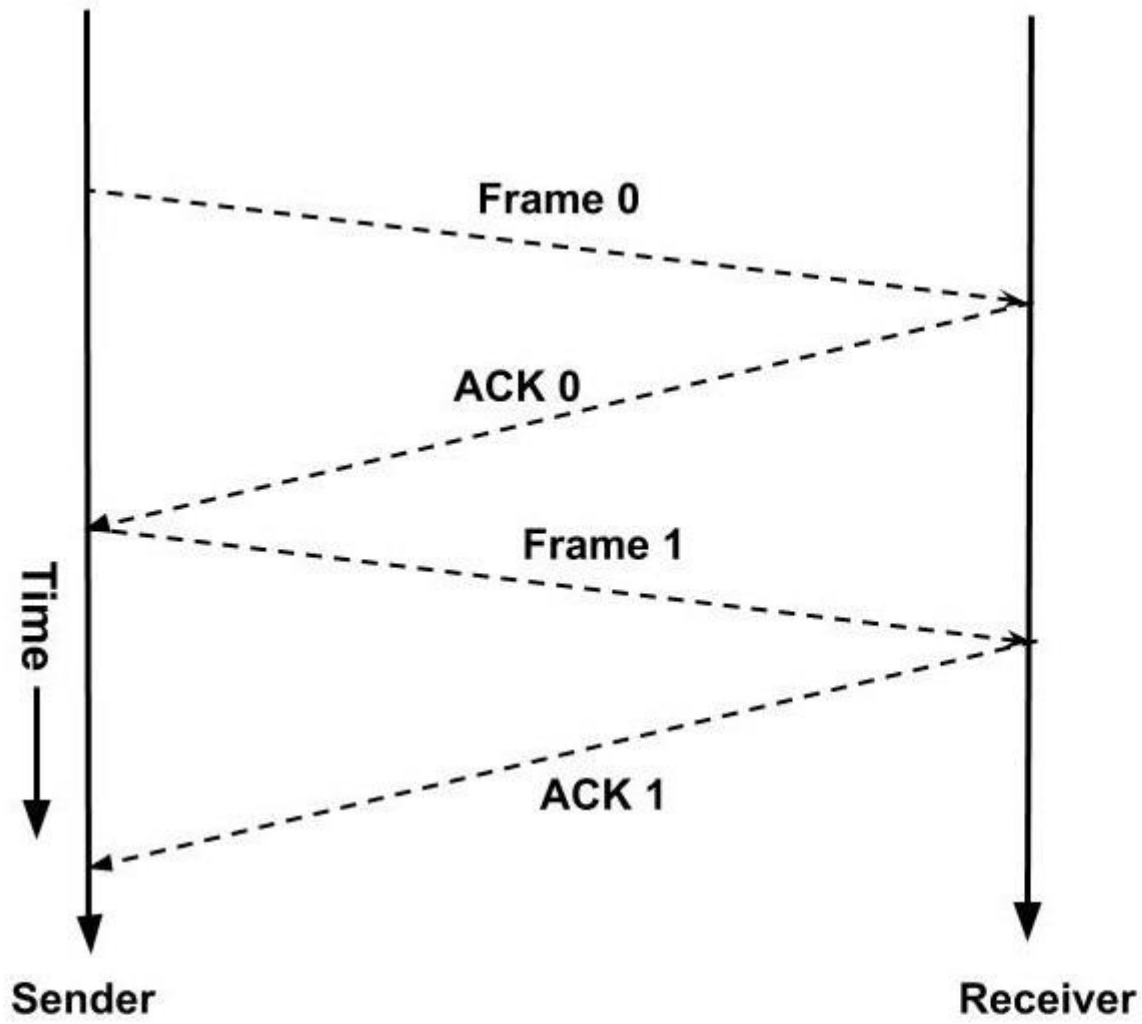
Flow Control

Flow control tells the sender how much data should be sent to the receiver so that it is not lost. This mechanism makes the sender wait for an acknowledgment before sending the next data. There are two ways to control the flow of data:

1. Stop and Wait Protocol
2. Sliding Window Protocol

Stop and Wait Protocol

It is the simplest flow control method. In this, the sender will send one frame at a time to the receiver. Until then, the sender will **stop and wait** for the acknowledgment from the receiver. When the sender gets the acknowledgment then it will send the next data packet to the receiver and wait for the acknowledgment again and this process will continue. This can be understood by the diagram below.



Suppose if any frame sent is not received by the receiver and is lost. So the receiver will not send any acknowledgment as it has not received any frame. Also, the sender will not send the next frame as it will wait for the acknowledgment for the previous frame which it had sent. So a deadlock situation can be created here. To avoid any such situation there is a time-out timer. The sender will wait for this fixed amount of time for the acknowledgment and if the acknowledgment is not received then it will send the frame again.

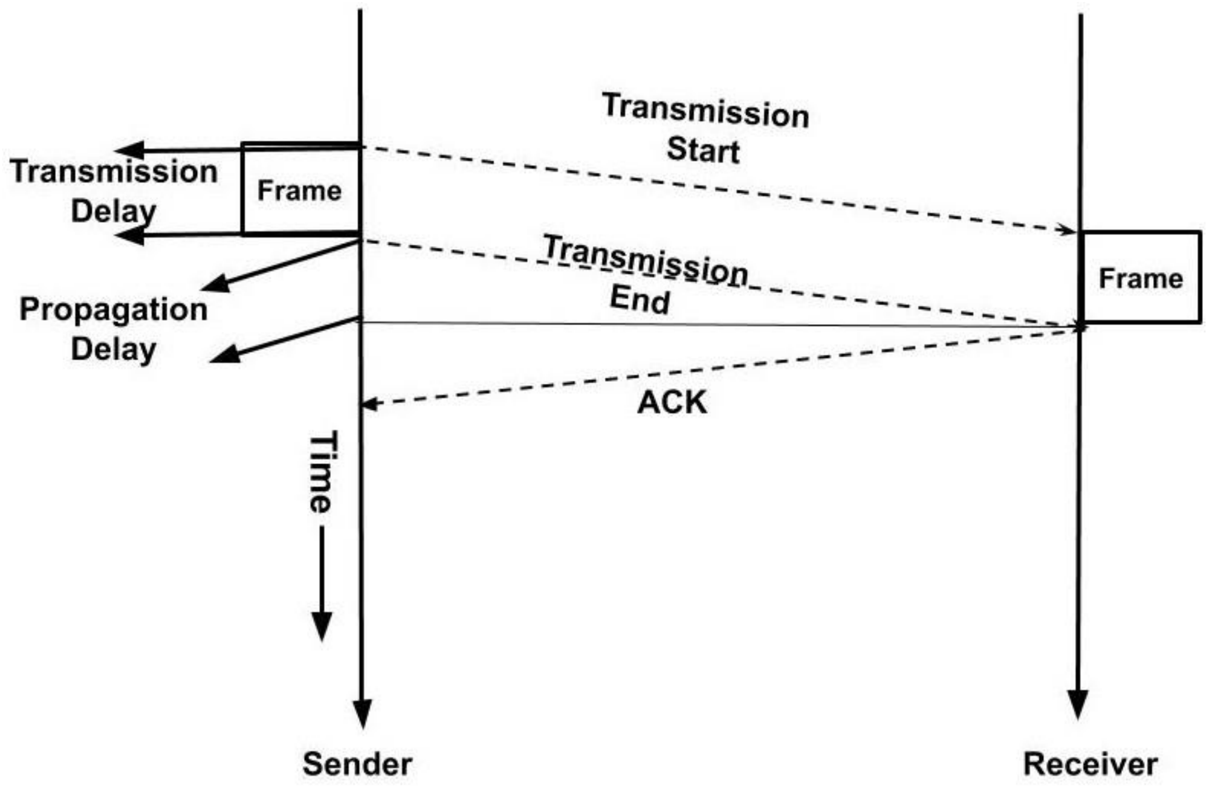
There are two types of delays while sending these frames:

- **Transmission Delay:** Time taken by the sender to send all the bits of the frame onto the wire is called transmission delay. This is calculated by dividing the data size(D) which has to be sent by the bandwidth(B) of the link.

Td = D / B

- **Propagation Delay:** Time taken by the last bit of the frame to reach from one side to the other side is called propagation delay. It is calculated by dividing the distance between the sender and receiver by the wave propagation speed.

Tp = d / s ; where d = distance between sender and receiver, s = wave propagation speed



The propagation delay for sending the data frame and the acknowledgment frame is the same as distance and speed will remain the same for both frames. Hence, the total time required to send a frame is

Total time= Td(Transmission Delay) + Tp(Propagation Delay for data frame) + Tp(Propagation Delay for acknowledgment frame)

The sender is doing work only for **Td** time and for the rest **2Tp** time the sender is waiting for the acknowledgment.

Efficiency = Useful Time/ Total Time

$\eta = Td / (Td + 2Tp)$

Advantages of Stop and Wait Protocol

1. It is very simple to implement.

Disadvantages of Stop and Wait Protocol

1. We can send only one packet at a time.
2. If the distance between the sender and the receiver is large then the propagation delay would be more than the transmission delay. Hence, efficiency would become very low.
3. After every transmission, the sender has to wait for the acknowledgment and this time will increase the total transmission time.

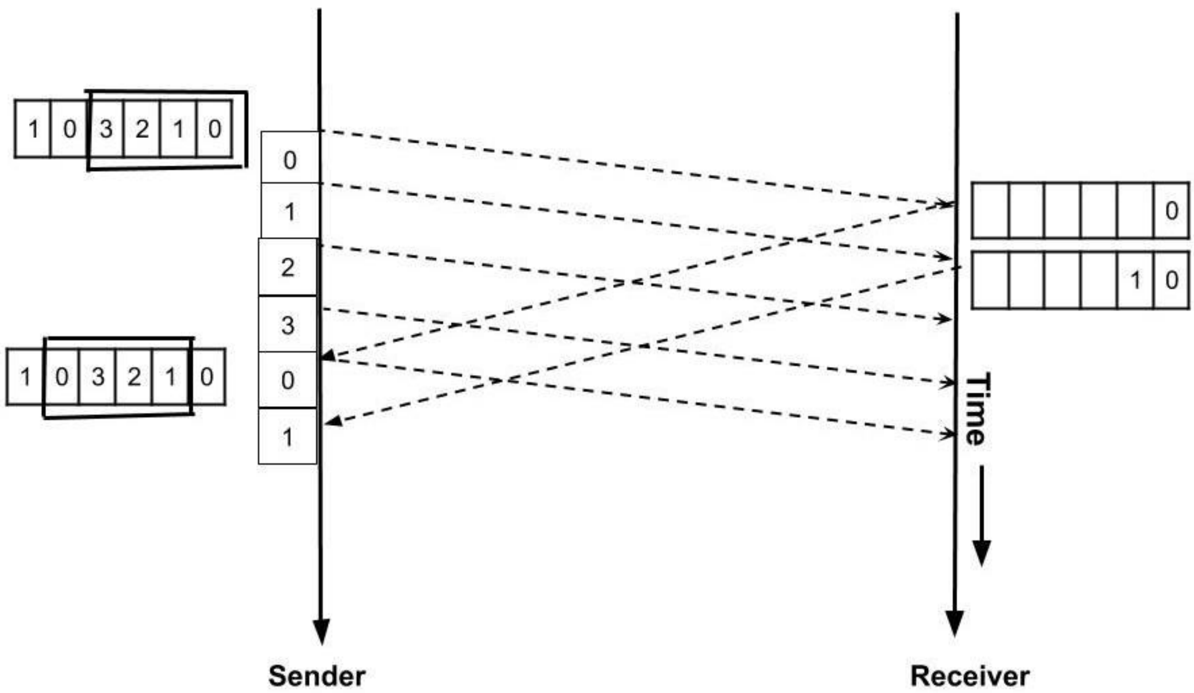
Sliding Window Protocol

As we saw that the disadvantage of the stop and wait protocol is that the sender waits for the acknowledgment and during that time the sender is idle. In sliding window protocol we will utilize this time. We will change this waiting time into transmission time.

A **window** is a buffer where we store the frames. Each frame in a window is numbered. If the window size is **n** then the frames are numbered from the number 0 to n-1. A sender can send n frames at a time. When the receiver sends the acknowledgment of the frame then we need not store that frame in our window as it has already been received by the receiver. So, the window in the sender side **slides** to the next frame and this window will now contain a new frame along with all the previous unacknowledged frames of the window. **At any instance of time window will only contain the unacknowledged frames.** This can be understood with the ***example*** below:

1. Suppose the size of the window is 4. So, the frames would be numbered as 0,1,2,3,0,1,2,3,0,... so on.

2. Initially, the frames in the window are 0,1,2, 3. Now, the sender starts transmitting the frames. The first frame is sent, then second and so on.
3. When the receiver receives the first frame i.e. frame 0. Then it sends an acknowledgment.
4. When the acknowledgment is received by the sender then it knows that the first frame has been received by the receiver and it need not keep its record. So, the **window slides** to the next frame.
5. The new window contains the frame 1, 2, 3, 0. In this way, the window slides hence the name sliding window protocol.



Sliding Window Protocol

Using sliding window protocol, the efficiency can be made maximum i.e. 1. In sliding window protocol we are using the propagation delay time also for the transmission. For doing this we the sender should be sending the data frame all the time i.e for Td+2Tp time. So, **what should be the number of packets such that the efficiency is maximum?**

We will apply a simple unitary method to find this. In Td units of time, we can send one packet. So in one unit of time, we can send 1/Td packets. We have total time as Td+2Tp. Therefore, in Td+2Tp time we can send (Td+2Tp)/Td packets. Let a=Tp/Td. So, if we send 1+2a packets then the efficiency is 1.

Td units of time → 1 packet transmitted

1 unit of time → (1/ Td) packet transmitted

Td + 2Tp units of time → (Td + 2Tp) / Td packets transmitted

This is how the flow of data is controlled using the above two mechanisms.