## Transmission and Propagation Delay

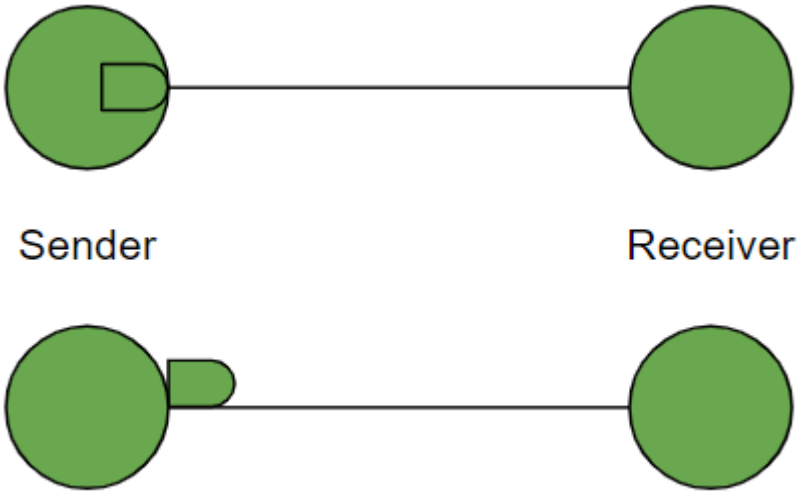
**Delays in Packet switching :**

1. Transmission Delay
2. Propagation Delay
3. Queuing Delay
4. Processing Delay

**Transmission Delay :**

Time taken to put a packet onto link. In other words, it is simply time required to put data bits on the wire/communication medium. It depends on length of packet and bandwidth of network.

```
Transmission Delay = Data size / bandwidth = (L/B) second
```

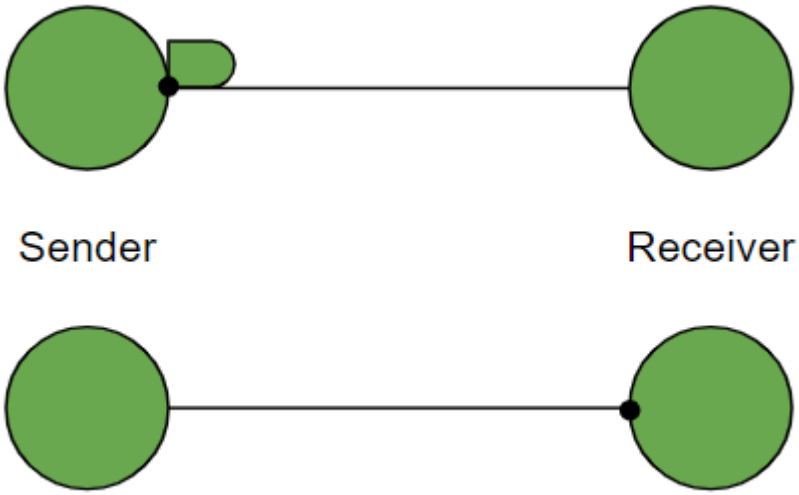





For example,
Let the link bandwidth of a network be 100 bits/second and the packet length be 1000 bits. Therefore the tarnsmission delay for the following network will be 1000/100 = 10 seconds.

**Propagation delay :** Time is taken by the first bit to travel from sender to receiver end of the link. In other words, it is simply the time required for bits to reach the destination from the start point. Factors on which Propagation delay depends are Distance and propagation speed.

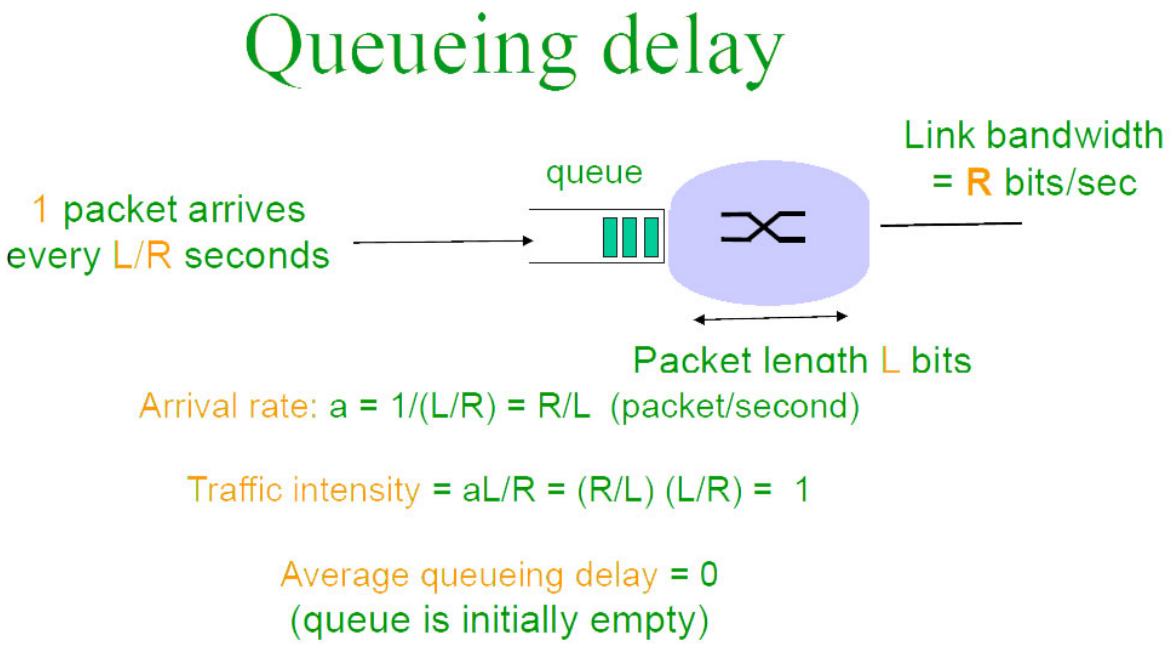
```
Propagation delay = distance/transmission speed = d/s
```




Lets take another example with the distance as 30 km, velocity being 3 x 10^8 meter/second.
Therefore, the propagation delay = distance/velocity = (30 * 10^3)/(3 * 10^8) = 0.1 milli-second.

**Queuing Delay :** Queuing delay is the time a job waits in a queue until it can be executed. It depends on congestion. It is the time difference between when the packet arrived Destination and when the packet data w essed or executed. It may be caused by

mainly three reasons i.e. originating switches, intermediate switches or call receiver servicing switches.




```
Average Queuing delay = (N-1)L/(2*R)

where N = no. of packets

      L=size of packet

      R=bandwidth
```

**Processing Delay :** Processing delay is the time it takes routers to process the packet header. Processing of packets helps in detecting bit-level errors that occur during transmission of a packet to the destination. Processing delays in high-speed routers are typically on the order of microseconds or less.
In simple words, it is just the time taken to process packets.

```
Total time or End-to-End time
= Transmission delay + Propagation delay+ Queuing delay
                                  + Processing delay
```

*For M hops and N packets -*

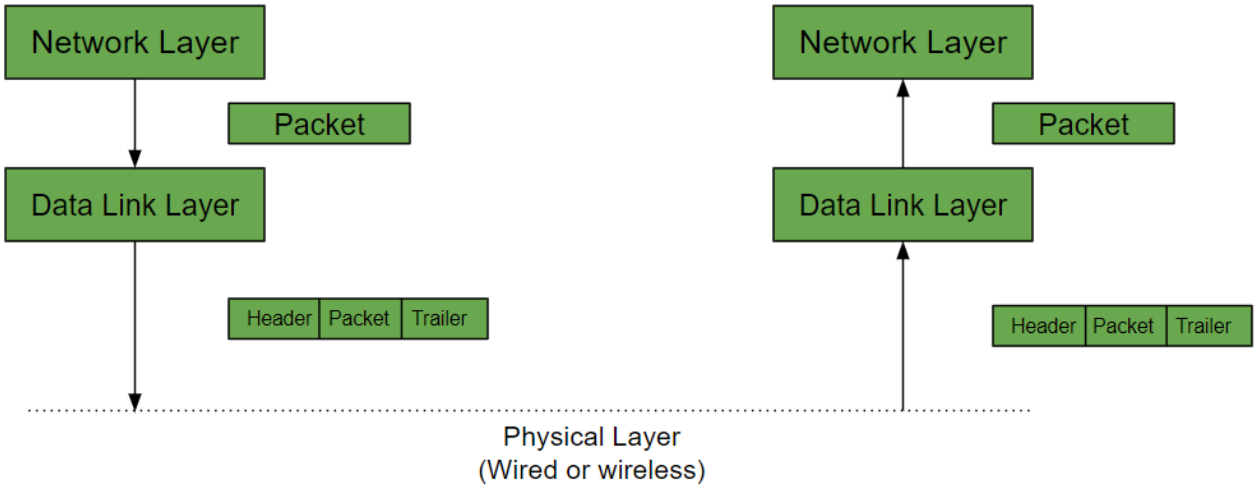
```
Total delay
```

```
= M*(Transmission delay + propagation delay)+

                (M-1)*(Processing delay + Queuing delay) +

                (N-1)*(Transmission delay)
```



## Data Link Layer

The data link layer is responsible for the node to node delivery of the message. The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer. When a packet arrives in a network, it is the responsibility of DLL to transmit it to the Host using its MAC address. The working is as follows:



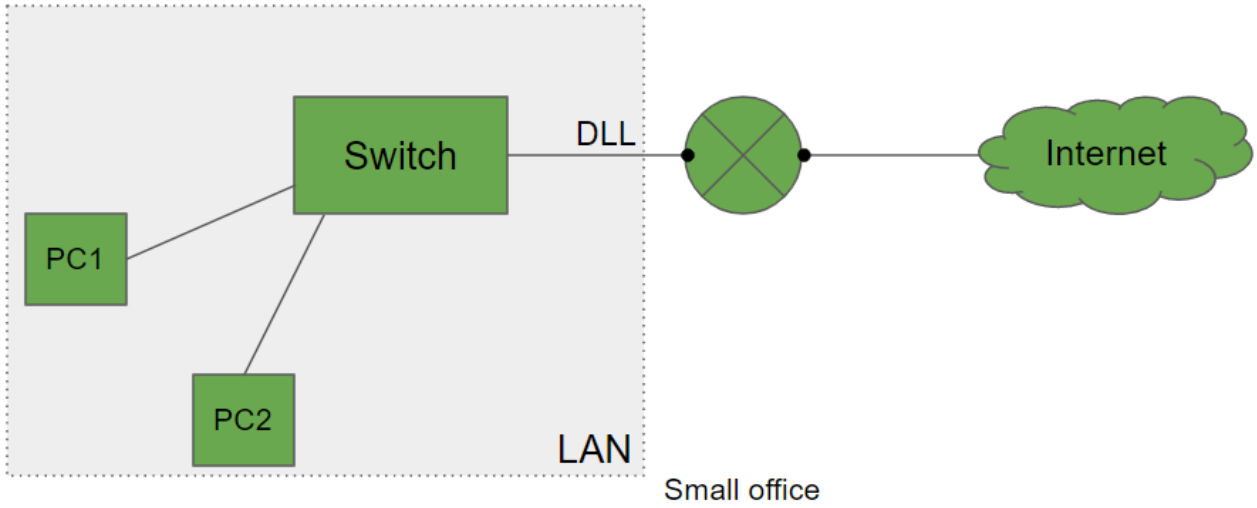

Data Link Layer is divided into two sub layers :

1. Logical Link Control (LLC)
2. Media Access Control (MAC)

The packet received from the Network layer is further divided into frames depending on the frame size of NIC(Network Interface Card). DLL also encapsulates Sender and Receiver's MAC address in the header.

The Receiver's MAC address is obtained by placing an ARP(Address Resolution Protocol) request onto the wire asking "Who has that IP address?" and the destination host will reply with its MAC address.

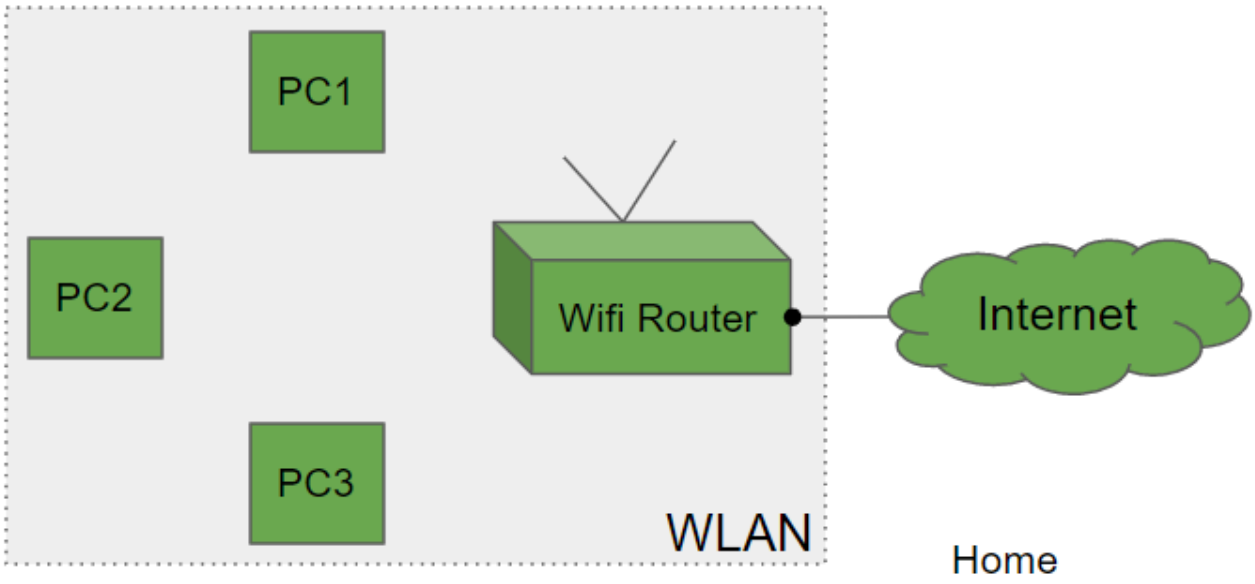
Now let's see how the DLL works in a sr ce:






The switch is used to connect multiple computers or laptops which in turn is connected to a router. This is then connected to the internet. All the 1-to-1 connection is done using DLL. The setup is called LAN as they are all connected in Local Area Network.

Now let's see how the DLL works in a small office:




WLAN as they are all connected in Wireless Local Area Network. This network might have a collision.

The functions of the Data Link layer are :

1. **Framing:** Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. This can be accomplished by attaching special bit patterns to the beginning and end of the frame.

2. **Physical addressing:** After creating frames, Data link layer adds physical addresses (MAC address) of sender and/or receiver in the header of each frame.
3. **Error Detection:** Data link layer provides the mechanism of error control in which it detects and retransmits damaged or lost frames.
4. **Error and Flow Control:** The data rate must be constant on both sides else the data may get corrupted thus, flow control coordinates that amount of data that can be sent before receiving acknowledgement.
5. **Access control:** When a single communication channel is shared by multiple devices, MAC sub-layer of data link layer helps to determine which device has control over the channel at a given time.

** Packet in Data Link layer is referred as **Frame**.*
*** Data Link layer is handled by the NIC (Network Interface Card) and device drivers of host machines.*
**** Switch & Bridge are Data Link Layer devices.*



## - Stop and Wait ARQ

### Characteristics

- Used in Connection-oriented communication.
- It offers error and flow control
- It is used in Data Link and Transport Layers
- Stop and Wait ARQ mainly implements Sliding Window Protocol concept with Window Size 1

### Useful Terms:

- **Propagation Delay:** Amount of time taken by a packet to make a physical journey from one router to another router.

  Propagation Delay = (Distance between routers) / (Velocity of propagation)

- RoundTripTime (**RTT**) = 2* Propagation Delay
- TimeOut (**TO**) = 2* RTT
- Time To Live (**TTL**) = 2* TimeOut. (Maximum TTL is 180 seconds)

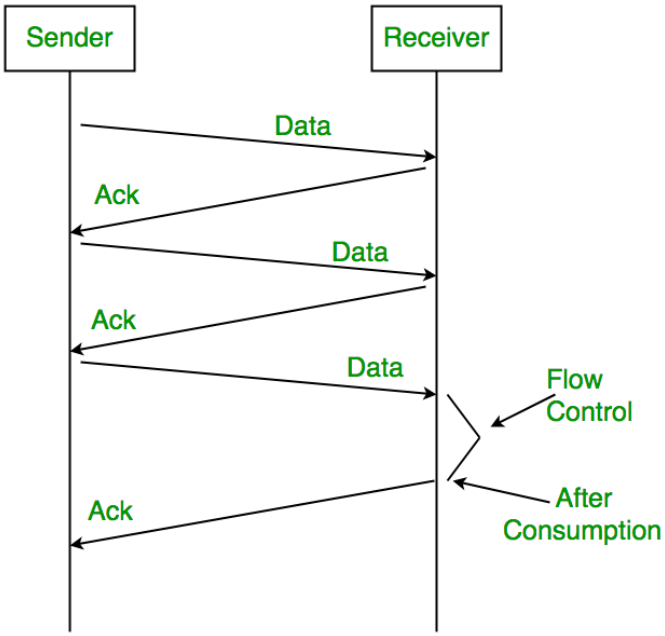
### Simple Stop and Wait

Sender:

Rule 1) Send one data packet at a time.
Rule 2) Send next packet only after receiving acknowledgement for previous.

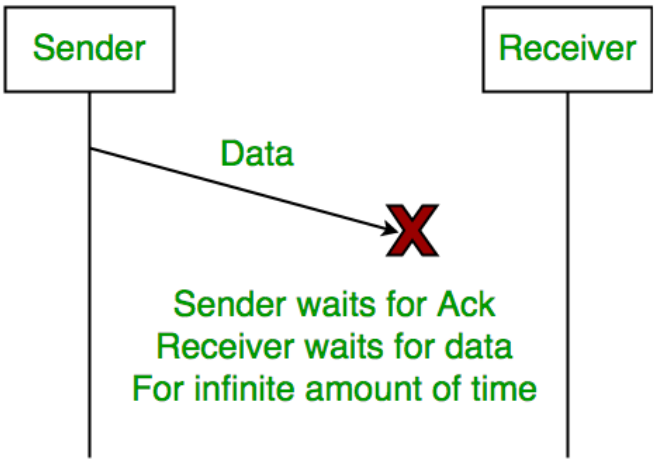
Receiver:

Rule 1) Send acknowledgement after receiving and consuming of the data packet.
Rule 2) After consuming packet acknowledgement need to be sent (Flow Control)

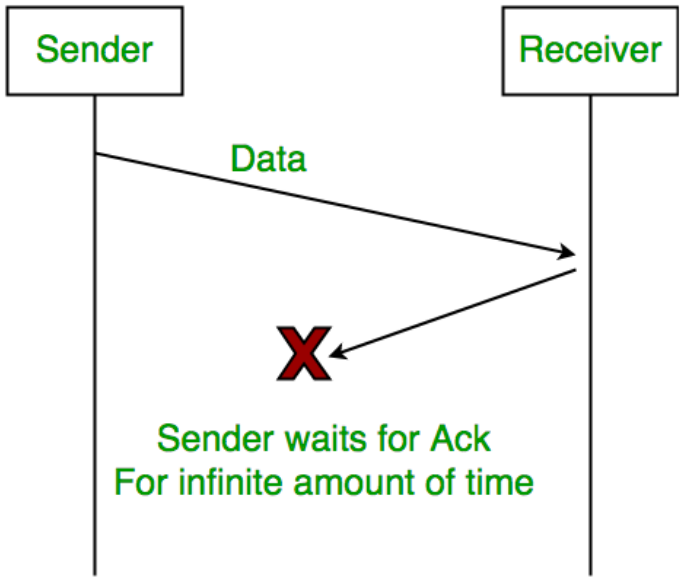



# Problems :

**1. Lost Data**



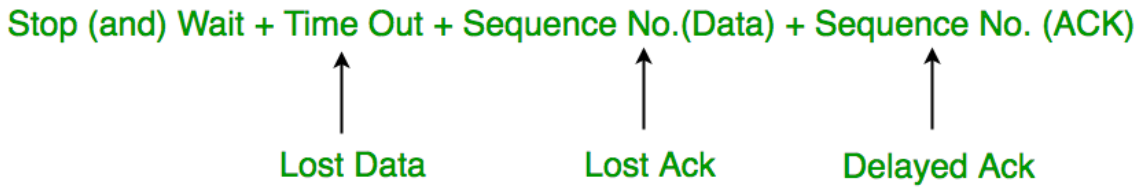

**2. Lost Acknowledgement:**



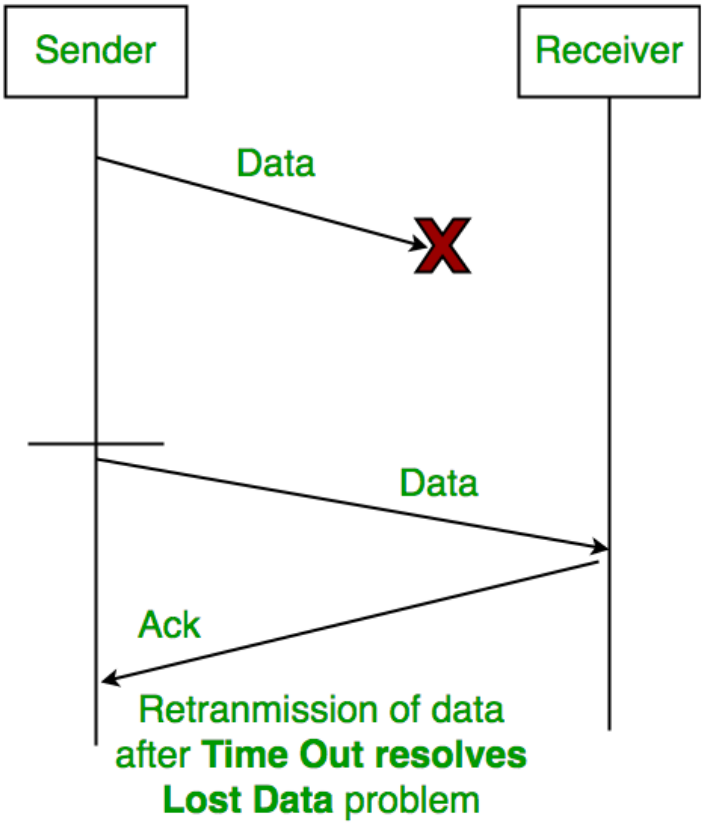

**3. Delayed Acknowledgement/Data:** After timeout on sender side, a long delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

## Stop and Wait ARQ (Automatic Repeat Request)

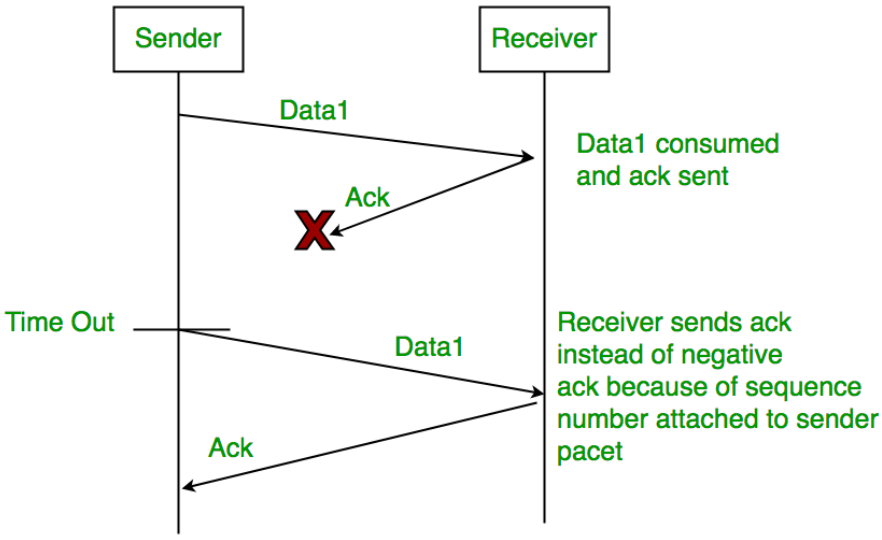
Above 3 problems are resolved by Stop and Wait ARQ (Automatic Repeat Request) that does both error control and flow control.




**1. Time Out:**



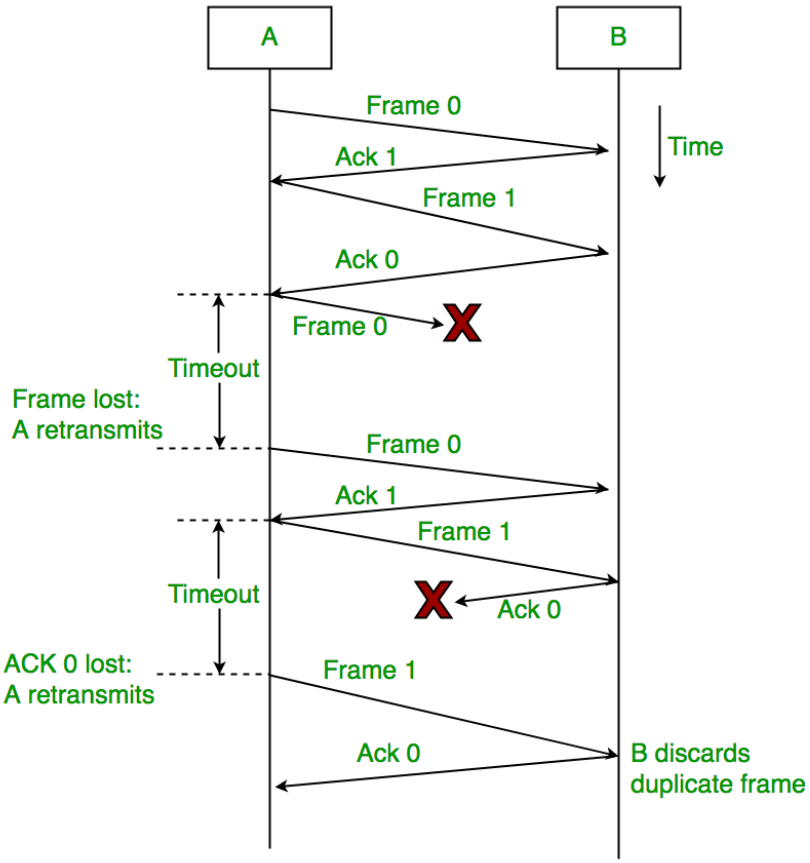

**2. Sequence Number (Data)**




**3. Delayed Acknowledgement:** This is resolved by introducing sequence number for acknowledgement also.

## Working of Stop and Wait ARQ:

1) Sender A sends a data frame or packet with sequence number 0.
2) Receiver B, after receiving data frame, sends an acknowledgement with sequence number 1 (the sequence number of next expected data frame or packet)
There is only a one-bit sequence number that implies that both sender and receiver have a buffer for one frame or packet only.




Characteristics of Stop and Wait ARQ:

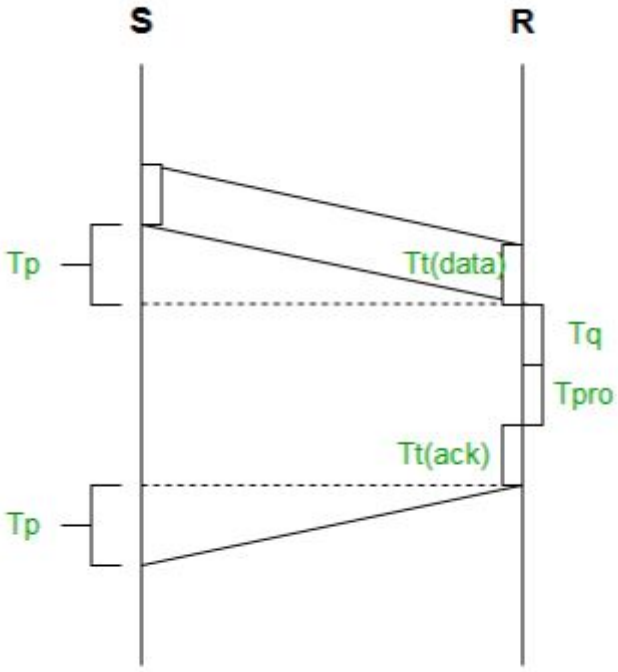
- It uses the link between sender and receiver as half-duplex link
- Throughput = 1 Data packet/frame per RTT
- If Bandwidth*Delay product is very high, then stop and wait protocol is not so useful. The sender has to keep waiting for acknowledgements before sending the processed next packet.
- It is an example for "**Closed Loop OR connection-oriented** " protocols
- It is a special category of SWP where its window size is 1
- Irrespective of number of packets sender is having stop and wait for protocol requires only 2 sequence numbers 0 and 1

The Stop and Wait ARQ solves main three problems 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). To solve this problem, we can send more than one packet at a time with a larger sequence number.

So Stop and Wait ARQ may work fine where propagation delay is very less, for example, LAN connections, but performs badly for distant connections like satellite connection.

**Efficiency:** Stop and Wait is a flow control protocol. In which the sender sends one packet and waits for the receiver to ackn e and then it will send the next packet.

In case if the acknowledgement is not received, the sender will retransmit the packet. This is the simplest one and easy to implement. but the main disadvantage is the efficiency is very low.




Total time taken to send one packet,

```
= Tt(data) + Tp(data) + Tq + Tpro + Tt(ack) + Tp(ack)


Since,

Tp(ack) = Tp(data)


And,

Tt(ack) << Tt(data).


So we can neglect Tt(ack)

Tq = 0 and Tpro = 0
```

```
Hence,

Total time = Tt(data) + 2 * Tp
```

Where,

```
Tt(data) : Transmission delay for Data packet

Tp(data) : propagation delay for Data packet

Tq: Queuing delay

Tpro: Processing delay

Tt(ack): Transmission delay for acknowledgment

Tp(ack) : Propagation delay for acknowledgment
```

We know that the **Efficiency (η)**,

```
= Useful time / Total cycle time.

= Tt / (Tt + 2*Tp)

= 1 / (1+2*(Tp/Tt))

= 1 / (1+2*a)



where,

a = Tp / Tt
```

**Throughput:** Number of bits send per second, which is also known as Effective Bandwidth or Bandwidth utilization.

```
Throughput,

= L/(Tt + 2*Tp)

= ((L/BW)*BW)/(Tt + 2*Tp)

= Tt/(Tt + 2*Tp) * BW
```

```
= 1/(1 + 2a) * BW


Hence, Throughput

= η * BW


where,

BW : BandWidth

L : Size of Data packet
```

**Factors affecting Efficiency:**

```
n = 1/(1 + 2*(Tp/Tt)

= 1/(1 + 2*(d/v)*(BW/L))


where,

d = distance between source and receiver

v = velocity
```

Let's see an example.

**Example:** Given,

```
Tt = 1ms

Tp = 2ms

Bandwidth = 6 Mbps


Efficiency(η)

= 1/(1 + a)

= 1/(1 + (2/1))
```

```
= 1/3

= 33.33 %


Throughput

= η * BW

= (1/3) * 6

= 2 Mbps
```



**Note:** As we can observe from the above given formula of Efficiency that:

1. On increasing the distance between source and receiver the Efficiency will decrease. Hence, Stop and Wait is only suitable for small area network like LAN. It is not suitable for MAN or WAN, as the efficiency will be very low.
2. If we increase the size of the Data packet, the efficiency is going to increase. Hence, it is suitable not for small packets. Big data packets can be send by Stop and Wait efficiently.

## - Selective Repeat Protocol

**Why Selective Repeat Protocol?** The go-back-n protocol works well if errors are less, but if the line is poor it wastes a lot of bandwidth on retransmitted frames. An alternative strategy, the selective repeat protocol, is to allow the receiver to accept and buffer the frames following a damaged or lost one.

Selective Repeat attempts to retransmit only those packets that are actually lost (due to errors) :

- Receiver must be able to accept packets out of order.
- Since receiver must release packets to higher layer in order, the receiver must be able to buffer some packets.

**Retransmission requests :**

- **Implicit -** The receiver acknowledges every good packet, packets that are not ACKed before a time-out are assumed lost or in error.Notice that this approach must be used to be sure that every packet is eventually received.
- **Explicit -** An explicit NAK (selective reject) can request retransmission of just one packet. This approach can expedite the retransmission but is not strictly needed.

- One or both approaches are used in practice.

**Selective Repeat Protocol (SRP) :** This protocol(SRP) is mostly identical to GBN protocol, except that buffers are used and the receiver, and the sender, each maintains a window of size. SRP works better when the link is very unreliable. Because in this case, retransmission tends to happen more frequently, selectively retransmitting frames is more efficient than retransmitting all of them. SRP also requires a full-duplex link. Backward acknowledgements are also in progress.



- Sender's Windows ( Ws) = Receiver's Windows ( Wr).
- Window size should be less than or equal to half the sequence number in SR protocol. This is to avoid packets being recognized incorrectly. If the size of the window is greater than half the sequence number space, then if an ACK is lost, the sender may send new packets that the receiver believes are retransmissions.
- Sender can transmit new packets as long as their number is with W of all unpacked packets.
- Sender retransmit un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.
- Receiver ACKs all correct packets.
- Receiver stores correct packets until they can be delivered in order to the higher layer.

- In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.

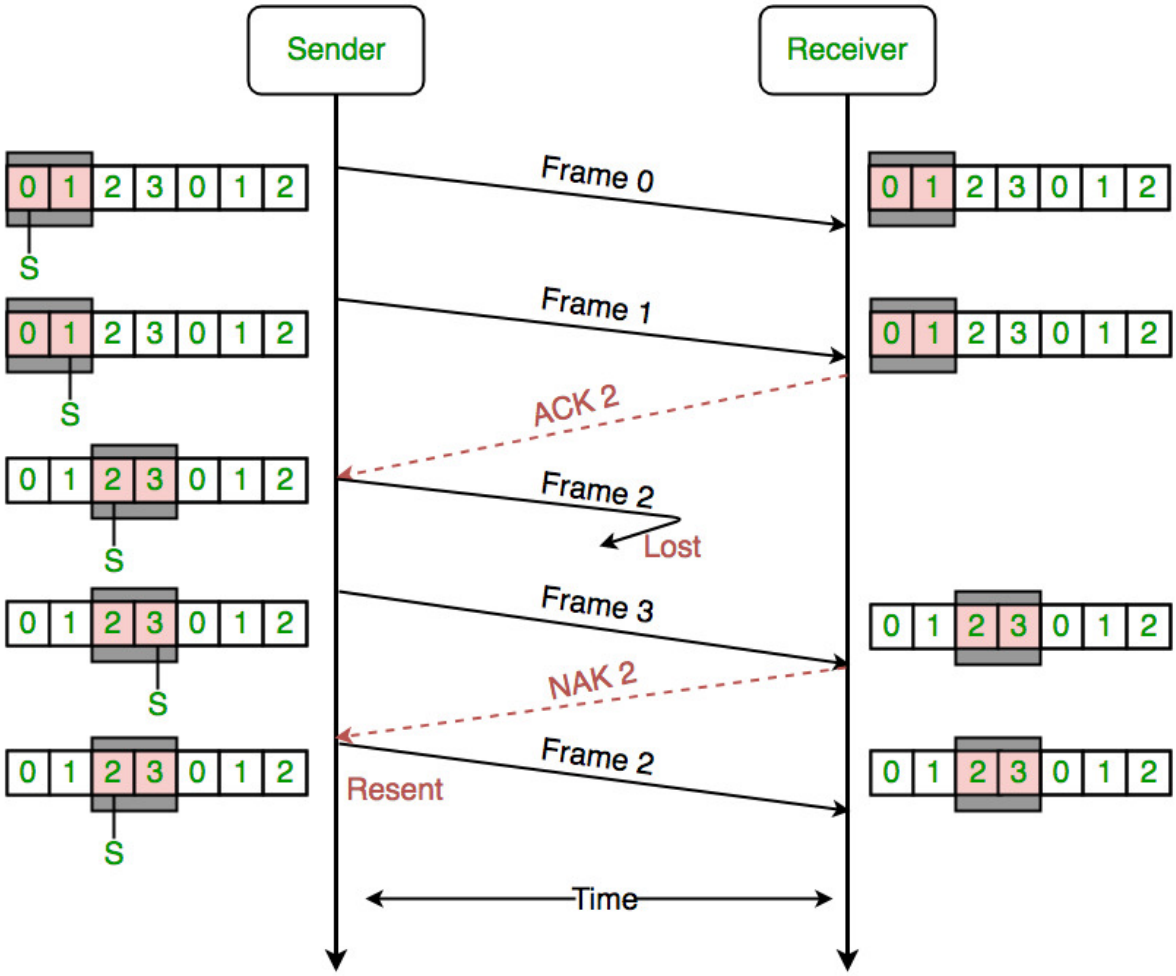



**Figure -** the sender only retransmits frames, for which a NAK is received

Efficiency of Selective Repeat Protocol (SRP) is same as GO-Back-N's efficiency :

```
Efficiency = N/(1+2a)
Where a = Propagation delay / Transmission delay
Buffers = N + N
Sequence number  = N(sender side)  + N  ( Receiver Side)
```

**Difference between Stop and Wait, GoBackN and Selective Repeat:**

1. **Stop and Wait -** The sender sends the packet and waits for the ACK (acknowledgement) of the packet. Once the ACK reaches the sender, it transmits the next packet in row. If the ACK is not received, it re-transmits the previous packet again.

2. **Go Back N -** The sender sends N packets which is equal to the window size. Once the entire window is sent, the sender then waits for a cumulative ACK to send more packets. On the receiver end, it receives only in-order packets and discards out-of-order packets. As in case of loss, the entire window would be re-

transmitted.

3. **Selective Repeat -** The sender sends packet of window size N and the receiver acknowledges all packet whether they were received in order or not. In this case, the receiver maintains a buffer to contain out-of-order packets and sorts them.The sender selectively re-transmits the lost packet and moves the window forward.



**Differences:**

| Properties | Stop and Wait | Go Back N | Selective Repeat |
|---|---|---|---|
| Sender window size | 1 | N | N |
| Receiver Window size | 1 | 1 | N |
| Minimum Sequence number | 2 | N+1 | 2N |
| Efficiency | 1/(1+2*a) | N/(1+2*a) | N/(1+2*a) |
| Type of Acknowledgement | Individual | Cumulative | Individual |
| Supported order at Receiving end | - | In-order delivery only | Out-of-order delivery as well |
| Number of retransmissions in case of packet drop | 1 | N | 1 |

Where,

- a = Ratio of Propagation delay and Transmission delay,
- At N=1, Go Back N is effectively reduced to Stop and Wait,/li>
- As Go Back N acknowledges the packed cumulatively, it rejects out-of-order packets,

- As Selective Repeat supports receiving out-of-order packets (it sorts the window after receiving the packets), it uses Independent Acknowledgement to acknowledge the packets.

## - Sliding Window Background

In sliding window protocol, the sender sends more than one frames to the receiver side and re-transmit the frame which are/is damaged or suspected. Efficiency of sliding window protocol is more than Stop-and-Wait Protocol. Sender window size of sliding window protocol is N. Receiver window size of sliding window protocol may 1 or N. In sliding window protocol, sorting may be or may not be necessary. The efficiency of the sliding window protocol is N/(1+2*a).

The Stop and Wait ARQ offers error and flow control, but may cause big performance issues as sender always waits for ackno ment even if it has next packet ready to send. Consider a situation where you ha h bandwidth connection and

propagation delay is also high (you are connected to some server in some other country though a high speed connection), you can't use this full speed due to limitations of stop and wait.

Sliding Window protocol handles this efficiency issue by sending more than one packet at a time with a larger sequence numbers. The idea is same as pipelining in architectures.



# Few Terminologies :

**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(L) / Total Cycle time(Tt + 2*Tp)
Multiplying and dividing by Bar    (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)
```



# Concept Of Pipelining

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]
```

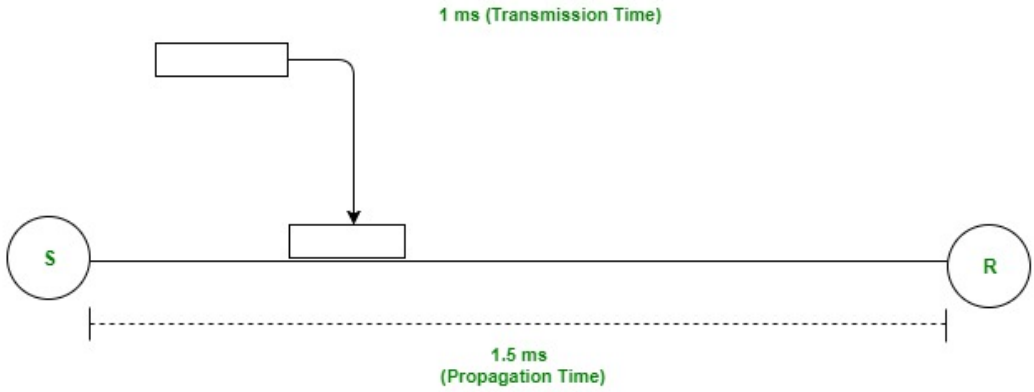
Maximum packets That can be Transmitted in total cycle time = 1+2*a

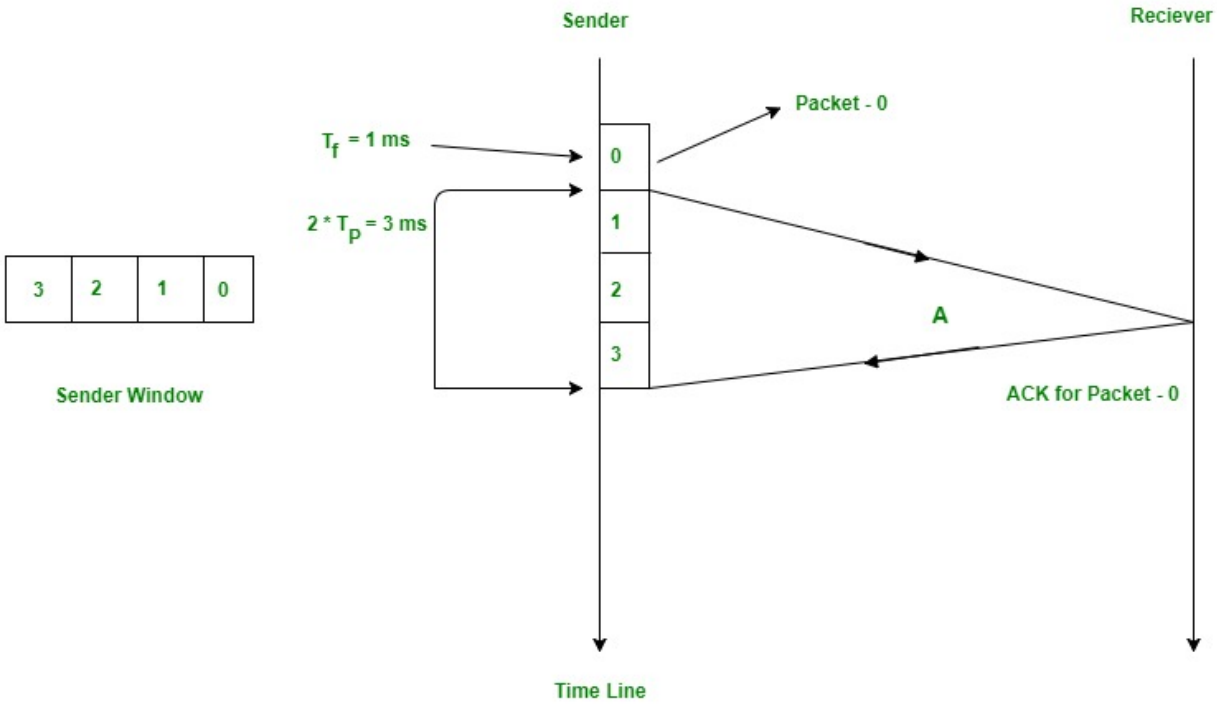
Let me explain now with the help of an example.

Consider Tt = 1ms, Tp = 1.5ms.

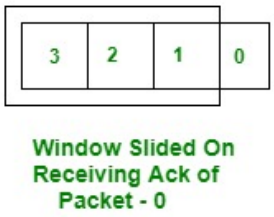
In the picture given below, after sender has transmitted packet 0, it will immediately transmit packets 1, 2, 3. Acknowledgement for 0 will arrive after 2*1.5 = 3ms. In Stop and Wait, in time 1 + 2*1.5 = 4ms, we were transferring one packet only. Here we keep

a **window of packets which we have transmitted but not yet acknowledged**.

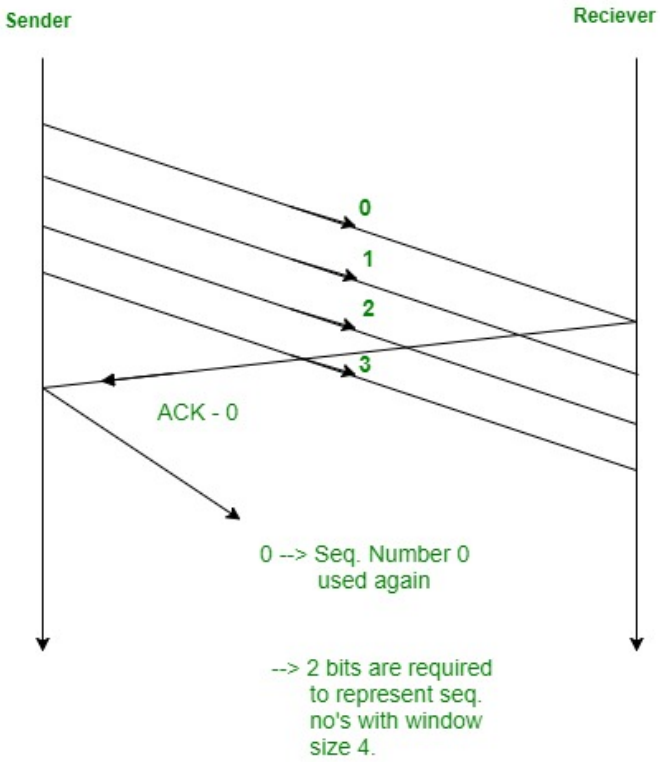



After we have received the Ack for packet 0, window slides and the next packet can be assigned sequence number 0. We reuse the sequence numbers which we have acknowledged so that header size can be kept minimum as shown in the diagram given below.




Window Slided On Receiving Ack of Packet - 0






## Go Back N

Sliding Window Protocol is actually a theoretical concept in which we have only talked about what should be the sender window size (1+2a) in order to increase the efficiency of stop and wait arq. Now we will talk about the practical implementations in which we take care of what should be the size of receiver window. Practically it is implemented in two protocols namely :

1. Go Back N (GBN)
2. Selective Repeat (SR)

In this article, we will explain you about the first protocol which is GBN in terms of three main characteristic features and in the last part we will be discussing SR as well as comparison of both these protocols

**Sender Window Size (WS)** It is N itself. If we say the protocol is GB10, then Ws = 10. N should be always greater than 1 in order to implement pipelining. For N = 1, it reduces to Stop and Wait protocol.

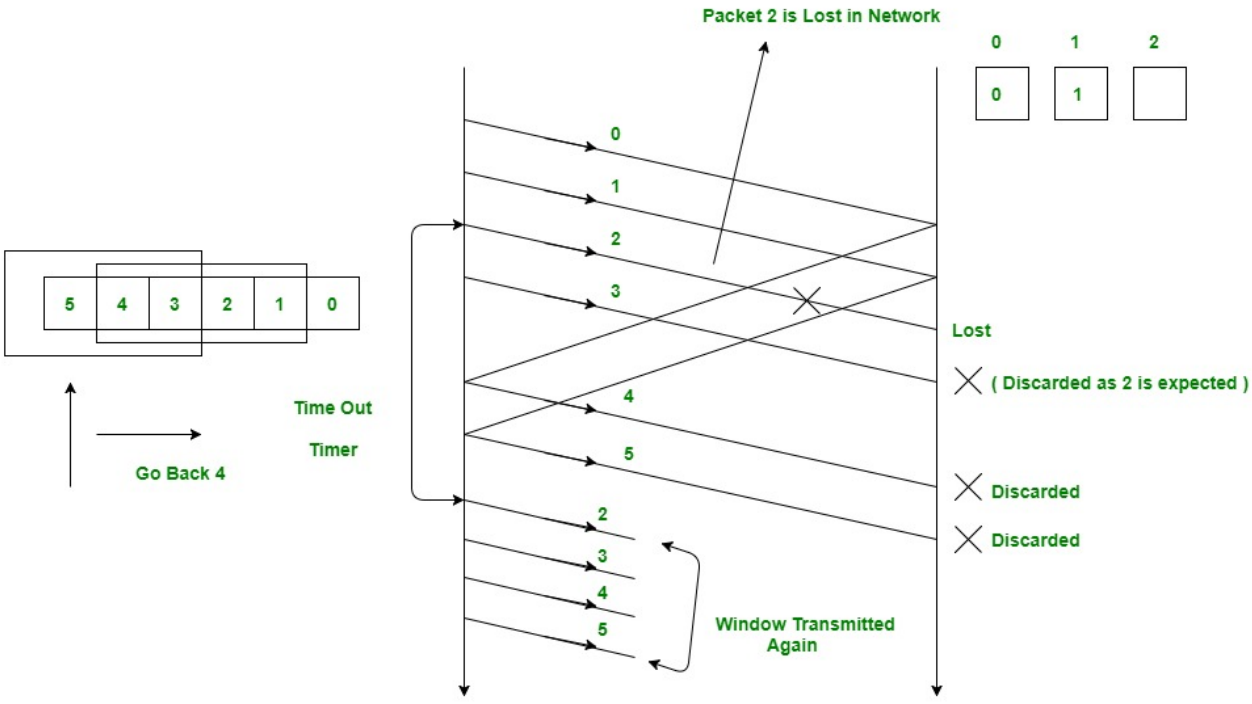
Efficiency Of GBN = N/(1+2a) Where a = Tp/Tt

If B is the bandwidth of the channel, the ve Bandwidth or Throughput =

Efficiency * Bandwidth = (N/(1+2a)) * B.

**Receiver Window Size (WR)** WR is always 1 in GBN.

Now what exactly happens in GBN, we will explain with the help of an example. Consider the diagram given below. We have a sender window size of 4. Assume that we have lots of sequence numbers just for the sake of explanation. Now the sender has sent the packets 0, 1, 2 and 3. After acknowledging the packets 0 and 1, the receiver is now expecting packet 2 and sender window has also slide to further transmit the packets 4 and 5. Now suppose the packet 2 is lost in the network, the Receiver will discard all the packets which sender has transmitted after packet 2 as it is expecting sequence number of 2. On the sender side for every packet send there is a time out timer which will expire for packet number 2. Now from the last transmitted packet 5 senders will go back to the packet number 2 in the current window and transmit all the packets till packet number 5. That's why it is called Go Back N. Go back means the sender has to go back N places from the last transmitted packet in the unacknowledged window and not from the point where the packet is lost.

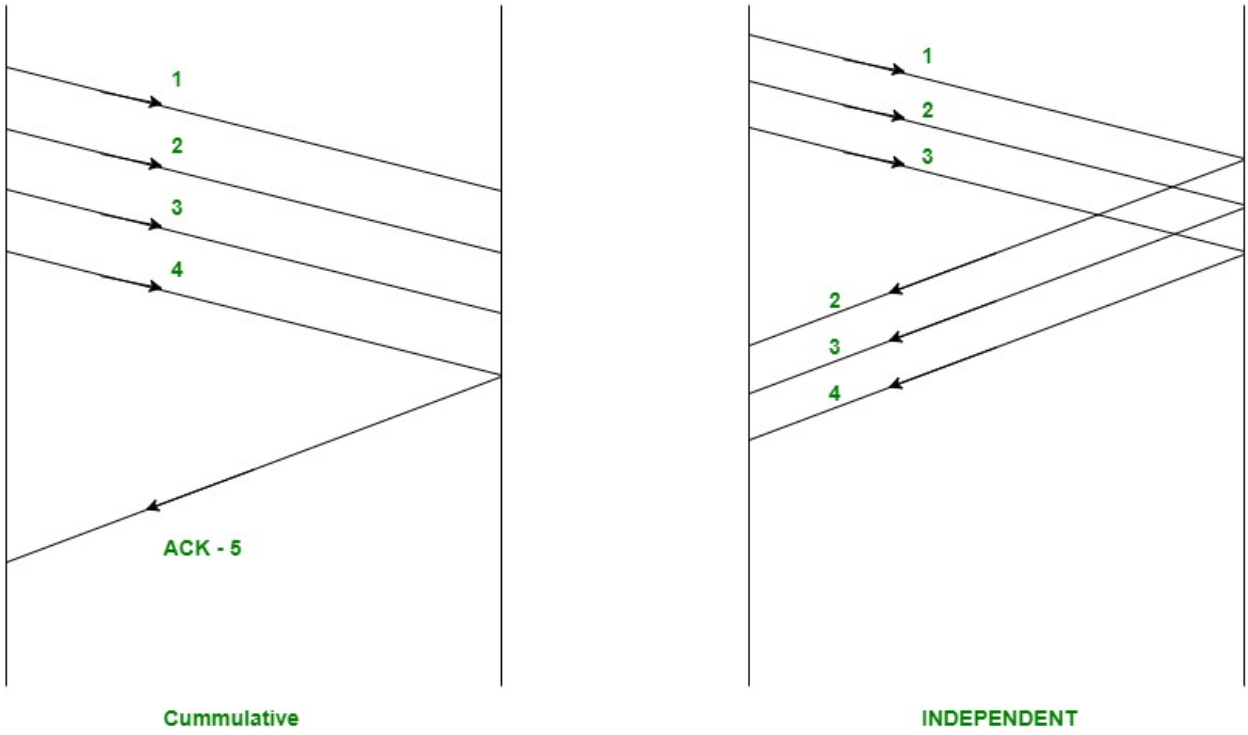



## Acknowledgements

There are 2 kinds of acknowledgements namely :

- **Cumulative Ack** - One acknowledgement is used for many packets. Main advantage is traffic is less. Disadvantage is less reliability as if one ack is loss that would mean that all the packets sent are lost.

- **Independent Ack** - If every packet is going to get acknowledgement independently. Reliability is high here but disadvantage is that traffic is also high since for every packet we are receiving independent ack.




GBN uses Cumulative Acknowledgement. At the receiver side, it starts an acknowledgement timer whenever receiver receives any packet which is fixed and when it expires, it is going to send a cumulative Ack for the number of packets received in that interval of timer. If the receiver has received N packets, then the Acknowledgement number will be N+1. An important point is Acknowledgement timer will not start after the expiry of first-timer but after the receiver has received a packet.
Time out timer at the sender side should be greater than Acknowledgement timer.

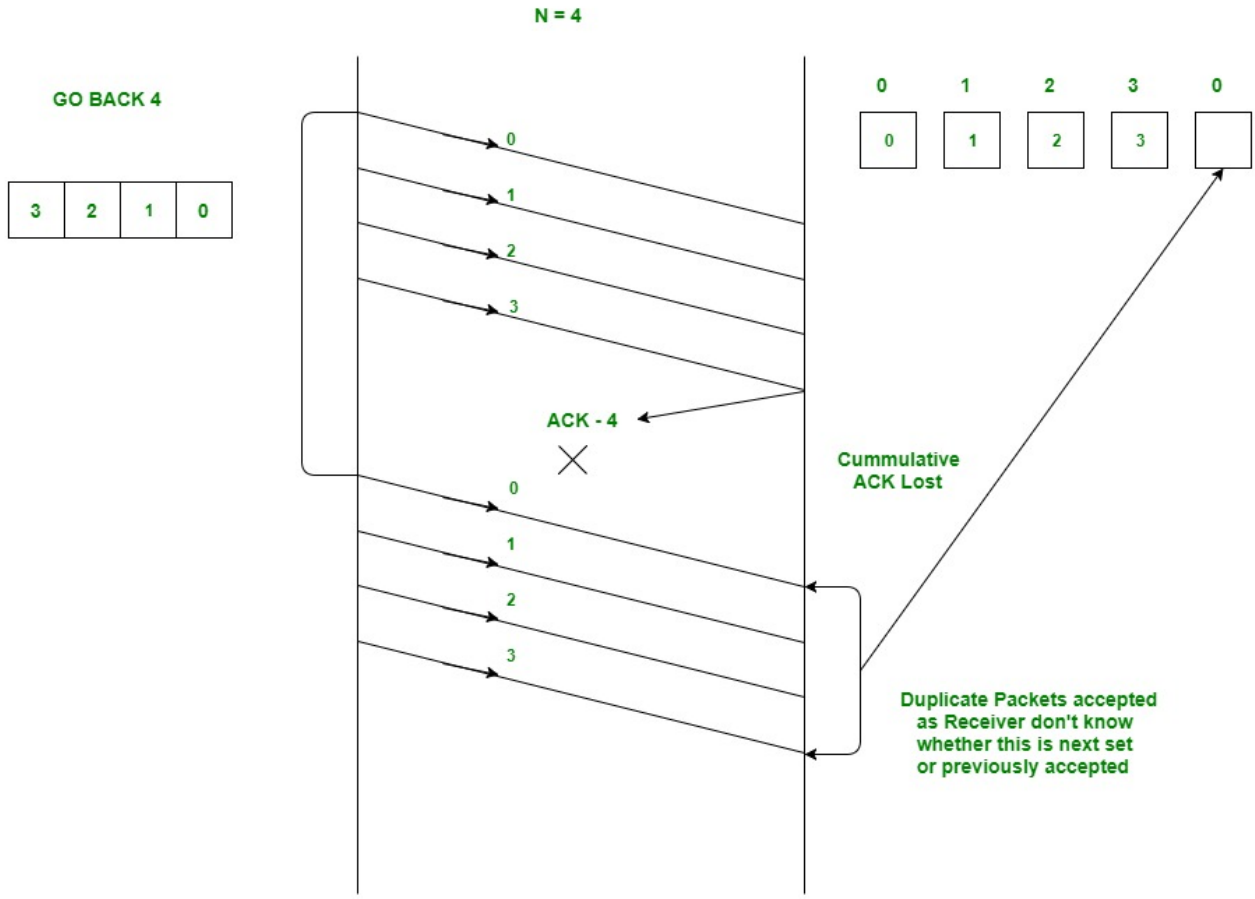
**Relationship Between Window Sizes and Sequence Numbers** We already know that sequence numbers required should always be equal to the size of the window in any sliding window protocol.

Minimum sequence numbers required in GBN is N+1.
Bits Required will be ceil(log2(N+1)).
The extra 1 is required in order to avoid the problem of duplicate packets as described below.

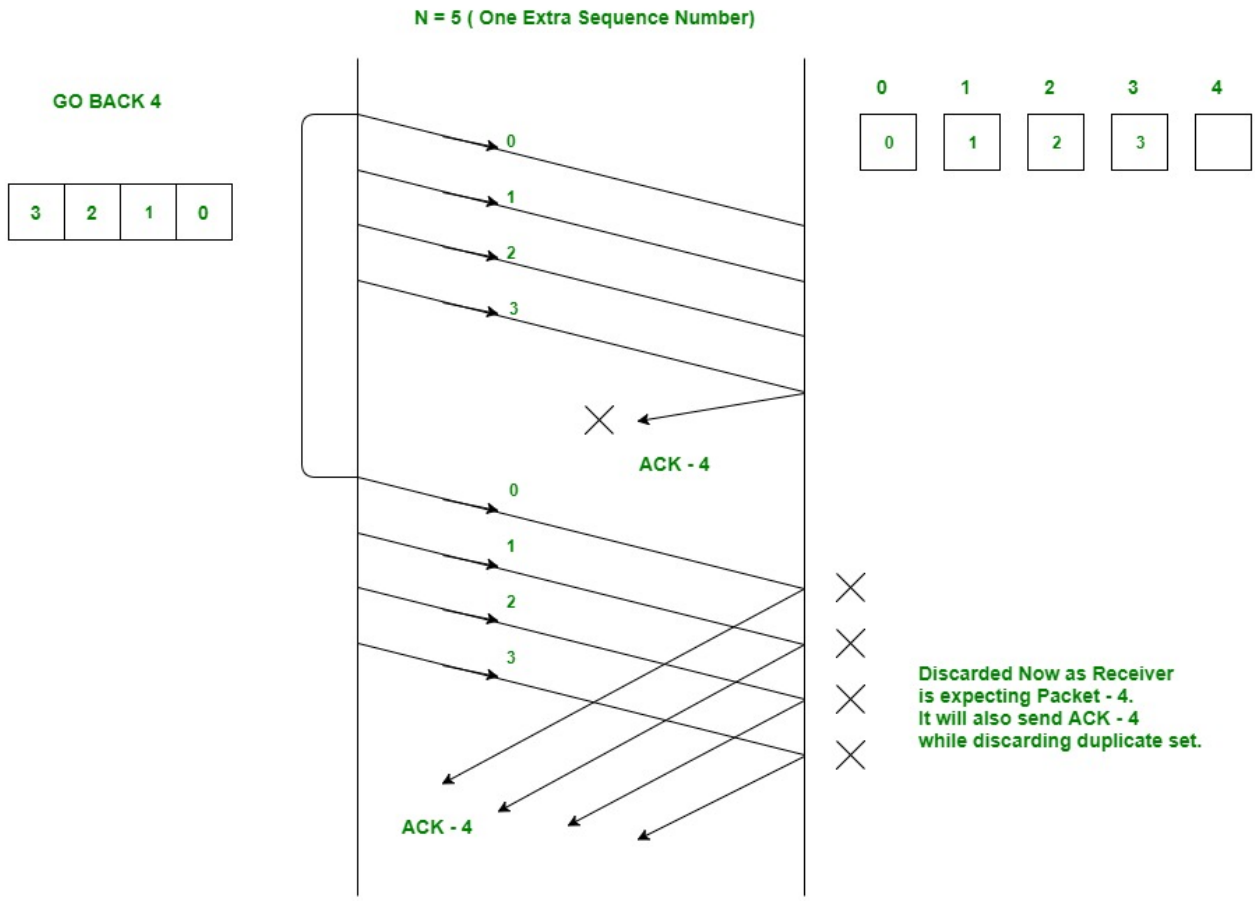
Consider an example of GB4. Sender window size is 4, therefore, we require a minimum of 4 sequence numbers to label each packet in the window. Now suppose receiver has received all the packets(0, 1, 2 and 3 sent by sender) and hence is now waiting for packet number 0 again(We can not use 4 as we have only 4 sequence numbers

available since N = 4). Now suppose the cumulative ack for the above 4 packets is lost in the network. On the sender side, there will be a timeout for packet 0 and hence all the 4 packets will be transmitted again. The problem now is receiver is waiting for a new set of packets which should have started from 0 but now it will receive the duplicate copies of the previously accepted packets. In order to avoid this, we need one extra sequence number. Now the receive could easily reject all the duplicate packets which were starting from 0 because now it will be waiting for packet number 4( We have added an extra sequence number now).

Trying with Sequence numbers 4.




Now Trying with one extra Sequence Number.

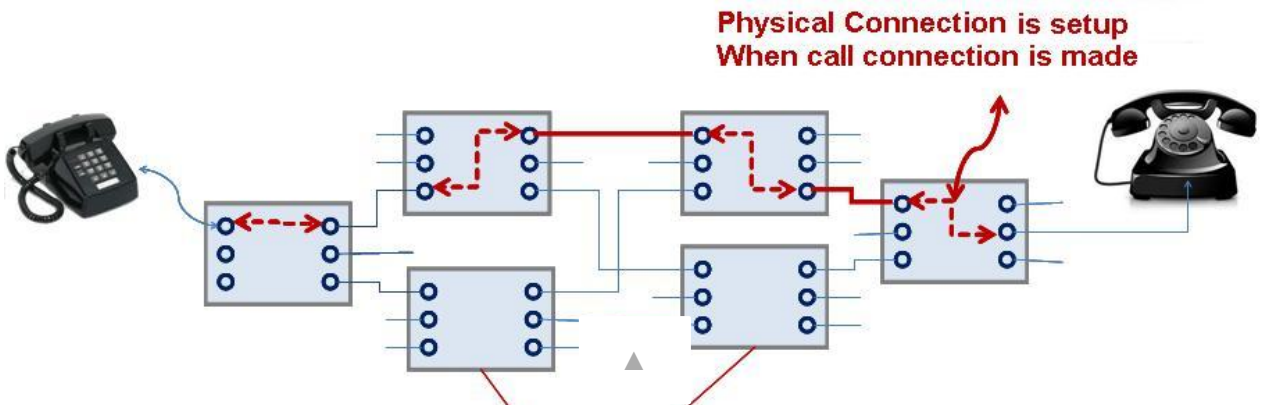



In the next article, we will explain Selective repeat and comparison between the 2 protocols.

## - Circuit Switching v/s Packet Switching

Circuit-Switching and Packet-Switching are 2 standard ways of transmitting packets between two end-devices over a network. Circuit-Switching is a historically used scheme which has been currently replaced by Packet-Switching.

**Circuit Switching** In circuit-switching, network resources are dedicated to establish a connection between the end-devices. Thus, a dedicated fixed path is established where data is transmitted without delays (as there is no concept of network congestion). A telephone system works under this scheme.




Key points of circuit switching:

- suitable for continuous transmission (dedicated line)
- guaranteed data-rate
- Inefficient (no transmission even if line is free)
- Under-utilization of resources in most cases



**Packet-Switching** Packet switching is a method of transferring the data to a network in the form of packets. In order to transfer the file fast and efficient manner over the network and minimize the transmission latency, the data is broken into small pieces of variable length, called **Packet**. At the destination, all these small-parts (packets) has to be reassembled, belonging to the same file. A packet composes of payload and various control information. No pre-setup or reservation of resources is needed.
Packet Switching uses **Store and Forward** technique while switching the packets; while forwarding the packet each hop first store that packet than forward. This technique is very beneficial because packets may get discarded at any hop due to some reason. More than one path is possible between a pair of source and destination. Each packet contains Source and destination address using which they independently travel through the network. In other words, packets belonging to the same file may or may not travel through the same path. If there is congestion at some path, packets are allowed to choose different path possible over an existing network
In packet-switching, data is broken down into several packets which are transmitted and routed over the network (according to protocols). No dedicated line is established. i.e. some packets may follow some other path, causing delay and out-of-order reception of data.

Some of the key points of packet switching are:

- Efficient utilisation of network resources
- out-of-order reception of packets
- Transmission delay (variable data-rate)

Packet-Switched networks were designed to overcome the *weaknesses* of Circuit-Switched networks since circuit-switched networks were not very effective for small messages.

**Advantage of Packet Switching over Circuit Switching :**

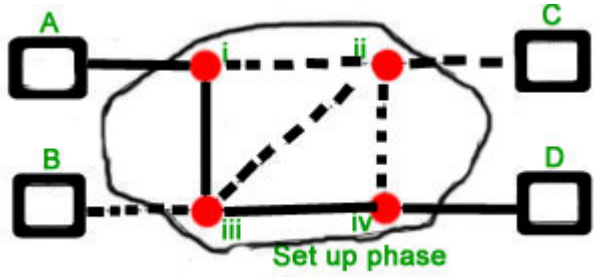
- More efficient in terms of bandwidth, since the concept of reserving circuit is not there.
- Minimal transmission latency.
- More reliable as destination can detect the missing packet.
- More fault tolerant because packets may follow different path in case any link is down, Unlike Circuit Switching.
- Cost effective and comparatively cheaper to implement.

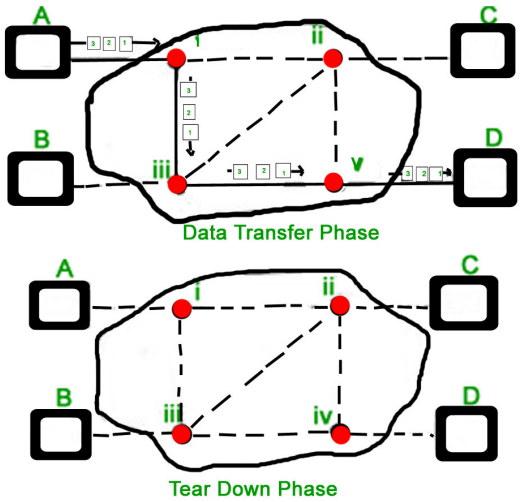
**Disadvantage of Packet Switching over Circuit Switching :**

- Packet Switching don't give packets in order, whereas Circuit Switching provides ordered delivery of packets because all the packets follow the same path.
- Since the packets are unordered, we need to provide sequence numbers to each packet.
- Complexity is more at each node because of the facility to follow multiple path.
- Transmission delay is more because of rerouting.
- Packet Switching is beneficial only for small messages, but for bursty data (large messages) Circuit Switching is better.

**Modes of Packet Switching :**

1. **Connection-oriented Packet Switching (Virtual Circuit) :-** Before starting the transmission, it establishes a logical path or virtual connection using signalling protocol, between sender and receiver and all packets belongs to this flow will follow this predefined route. Virtual Circuit ID is provided by switches/routers to uniquely identify this virtual connection. Data is divided into small units and all these small units are appended with help of sequence number. Overall, three phases takes place here- Setup, data transfer and tear down phase.






Phases in virtual circuit packet switching

All address information is only transferred during setup phase. Once the route to destination is discovered, entry is added to switching table of each intermediate node. During data transfer, packet header (local header) may contain information such as length, timestamp, sequence number etc.
Connection-oriented switching is very useful in switched WAN. Some popular protocols which use Virtual Circuit Switching approach are X.25, Frame-Relay, ATM and MPLS(Multi-Protocol Label Switching).

2. **Connectionless Packet Switching (Datagram) :-** Unlike Connection-oriented packet switching, In Connectionless Packet Switching each packet contains all necessary addressing information such as source address, destination address and port numbers etc. In Datagram Packet Switching, each packet is treated independently. Packets belonging to one flow may take different routes because routing decisions are made dynamically, so the packets arrived at destination might be out of order. It has no connection setup and teardown phase, like Virtual Circuits.
Packet delivery is not guaranteed in connectionless packet switching, so the reliable delivery must be provided by end systems using additional protocols.

```
A---R1---R2---B

A is the sender (start)
R1, R2 are two routers that store and forward data
B is receiver(destination)
```

To send a packet from A to B there are delays since this is a Store and Forward network.



Report An Issue

If you are facing any issue on this page. Please let us know.

Company-wise
Topic-wise
How to begin?
Write Interview Experience
Internships
Videos