Unit-3
Topic-1

Medium Access protocols

Stop and Wait Protocol

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)

<u>Simple Stop and Wait</u>

Sender:

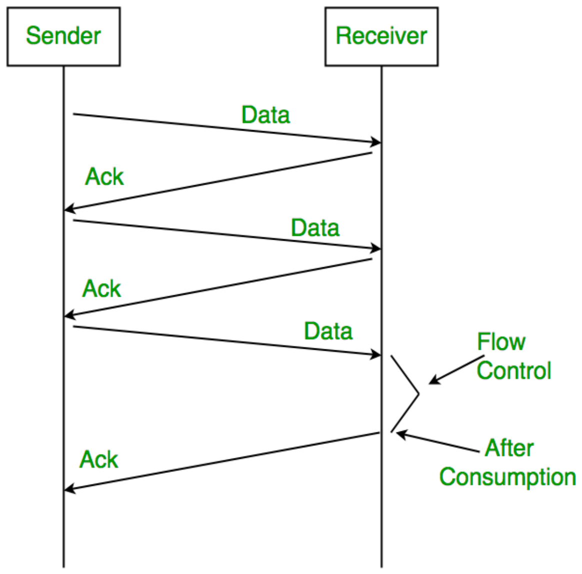
Rule 1) Send one data packet at a time.

Rule 2) Send next packet only after receiving acknowledgement for previous.

Receiver:

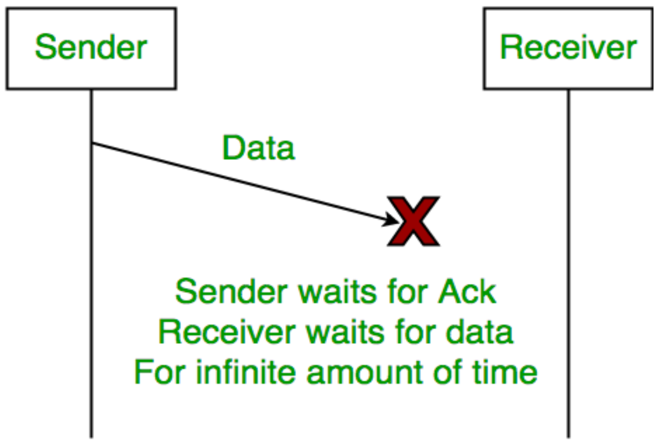
Rule 1) Send acknowledgement after receiving and consuming of data packet.

Rule 2) After consuming packet acknowledgement need to be sent (Flow Control)

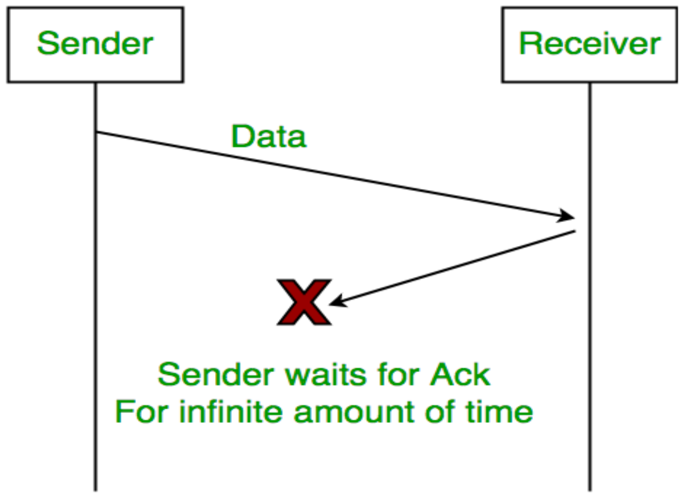


Problems :

1. **Lost Data**



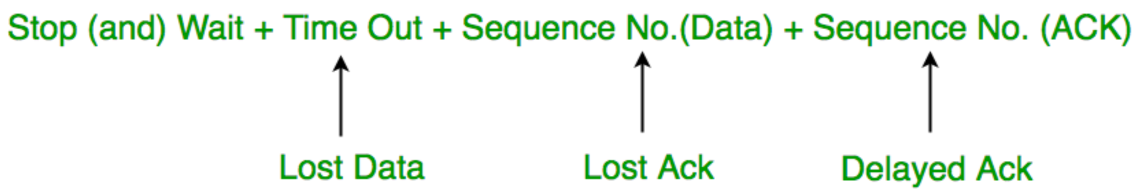
2. **Lost Acknowledgement:**



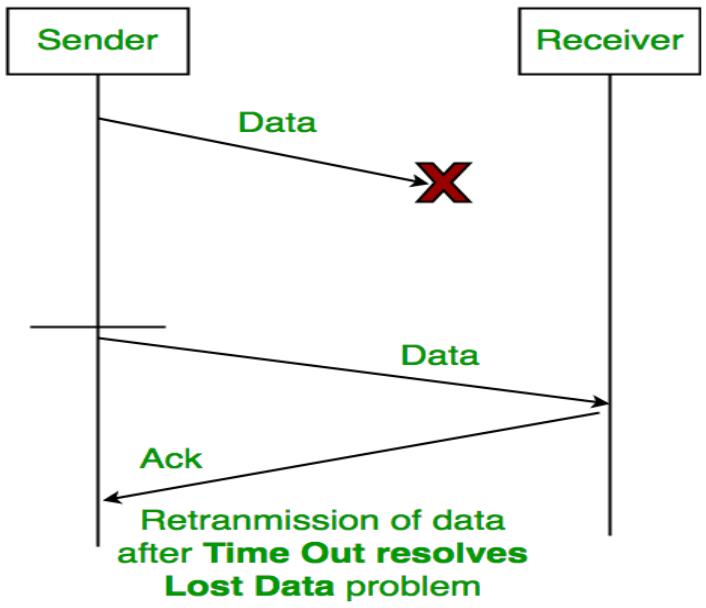
3. **Delayed Acknowledgement/Data:** After timeout on sender side, a long delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

Stop and Wait ARQ (Automatic Repeat Request)

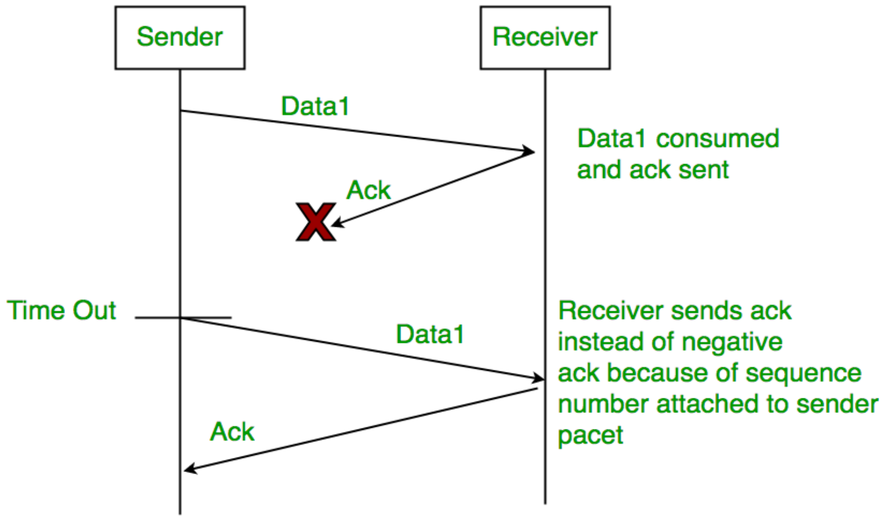
Above 3 problems are resolved by Stop and Wait ARQ (Automatic Repeat Request) that does both error control and flow control.


Stop (and) Wait + Time Out + Sequence No.(Data) + Sequence No. (ACK)
Lost Data
Lost Ack
Delayed Ack

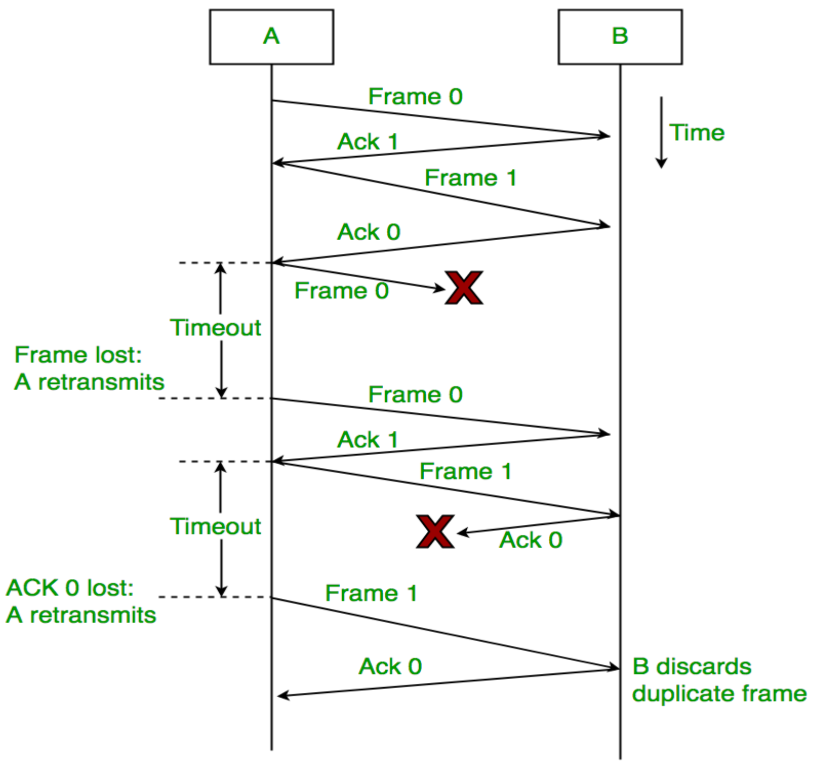
1. **Time Out:**



2. **Sequence Number (Data)**



3. **Delayed Acknowledgement:**
This is resolved by introducing sequence number for acknowledgement also.
<u>Working of Stop and Wait ARQ:</u>
1) Sender A sends a data frame or packet with sequence number 0.
2) Receiver B, after receiving data frame, sends and acknowledgement with sequence number 1 (sequence number of next expected data frame or packet)
There is only one bit sequence number that implies that both sender and receiver have buffer for one frame or packet only.



Characteristics of Stop and Wait ARQ:

- It uses 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 an special category of SWP where its window size is 1
- Irrespective of number of packets sender is having stop and wait 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 though a high speed connection). To solve this problem, we can send more than one packet at a time with a larger sequence numbers. We will be discussing these protocols in next articles.

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.

https://www.youtube.com/watch?v=n09DfvemnTQ

https://www.youtube.com/watch?v=oHaQOOW49fQ

Unit-3
Topic-2

Go-Back-N ARQ

Go-Back-N ARQ (Go-Back-N automatic repeat request) is a flow control protocol where the sender continues to send several frames specified by a window size even without receiving feedback from the receiver node. It can be said that it's a special case of the general sliding window protocol where the transmitter or sender window size is **N** and the receiver's window size is **1**, which means that it can transmit N frames to the receiving node before waiting for a feedback.

Now the receiver's duty is to keep the track of the sequence number of the next frame it expects to receive and sends the feedback after every data packet it receives. Once the sender has sent all the frames in its window, it makes sure that it has then received all the feedbacks of the transmitted data packets, if in any case, it hasn't received a feedback of any data packet and the time out timer expires, it then resends all the data packets again, starting from the lost data packet to the final data packet.

Let's take an example

Consider a sender has to send data packets indexing from **p1 to p5**, it sends all the data packets in order (from p1 to p5), but the receiver has only received p1 and the data packet p2 is lost somewhere in the network, then the receiver declines all the data packets after p2 (i.e. p3, p4, p5) because the receiver is waiting for packet p2 and will not accept any other data packet than that. So, now as the time out time index of p2 expires, the sender goes back 3 packets and starts sending all the data packets from p2 to p5 again.

We can see the process in the illustration below,

Sender window size (WS):

In Go-Back-N ARQ, **N** is the sender window size, which we can see in the above example was **5**. Now, here **N should be greater than 1** in order to implement pipelining. **If N=1**, then our system reduces to **Stop & Wait protocol**.

Now the efficiency of Go-Back-N ARQ = **N/(1+2a)**, where **a = tp/tt**.

Where **tp** is **propagation delay** and **tt** is the **transmission delay**

Also, **tt = D/B**; and here **D = data size** and **B = bandwidth**

And **tp = d/s**, here **d = distance** and **s = wave propagation speed**.

Now to find the effective bandwidth (or throughput),

Effective bandwidth = efficiency * bandwidth, which means,

Effective bandwidth = **(N/(1+2a)) *B**

Receiver window size (WR):

WR is always equal 1 in the case of Go-Back-N ARQ.

Feedbacks/Acknowledgements

There are basically 2 types of feedbacks/acknowledgments:

1. **<u>Cumulative Ack</u>:** Here we use only one feedback for many data packets, because of this the main advantage we get is that the traffic is less. But it can also result in a huge drawback which is, if one acknowledgment is lost then it means that all the data packets transmitted are lost.

2. **Independent Ack:** Here every data packet gets acknowledged independently. Here the reliability is high, but the main drawback is high traffic.

Go-Back-N ARQ uses a cumulative acknowledgment technique, which means receiver starts an acknowledgment timer whenever it receives a data packet which is fixed & when it expires, it will transmit a cumulative acknowledgment for the number of data packets received in that interval of time out timer. Now, if the receiver has received N data packets, then the feedback/acknowledgment number is going to be N+1. The important point here is that the acknowledgment timer will not start after the expiry of the first-timer, but it will do so when the receiver has received a data packet. The thing which should be kept in mind is that the time out timer at the sender node must be greater than the acknowledgment timer.
https://www.youtube.com/watch?v=QD3oCelHJ20
https://www.youtube.com/watch?v=F2Vu-JqNY_4

Unit-3
Topic-3

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 maintain 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 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 windows size 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 unACKed 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.

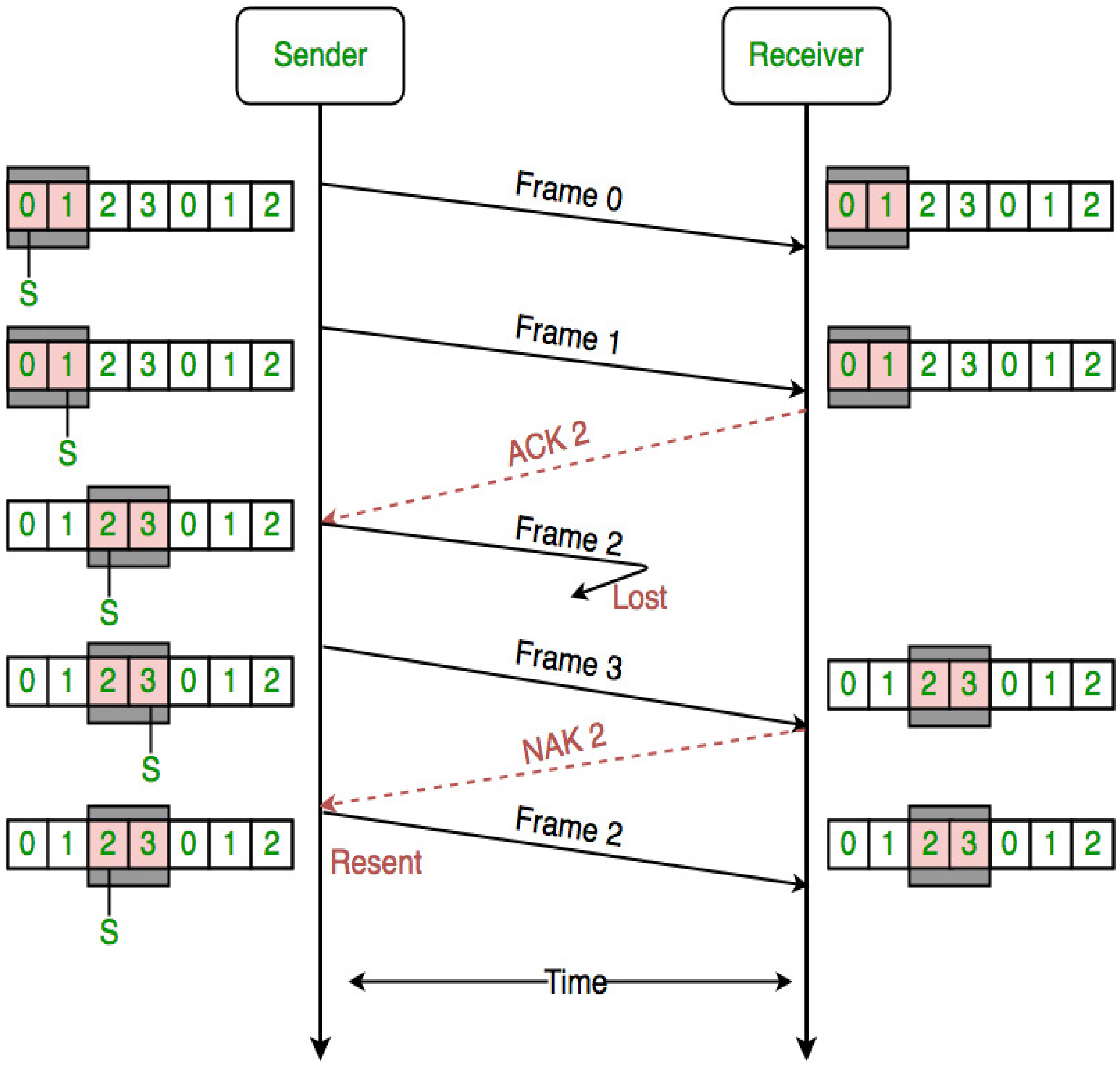


Figure – the sender only retransmits frames, for which a NAK is received
Efficiency of Selective Repeat Protocol (SRP) is same as GO-Back-N's efficiency :
Efficiency = N/(1+2a)
Where a = Propagation delay / Transmission delay
Buffers = N + N
Sequence number = N(sender side) + N (Receiver Side)

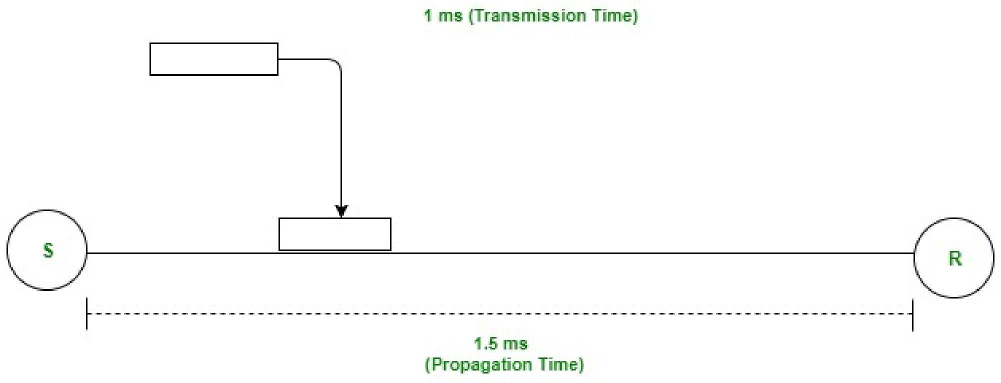
The Stop and Wait ARQ offers error and flow control, but may cause big performance issues as sender always waits for acknowledgement even if it has next packet ready to send. Consider a situation where you have a high bandwidth connection and propagation delay is also high (you

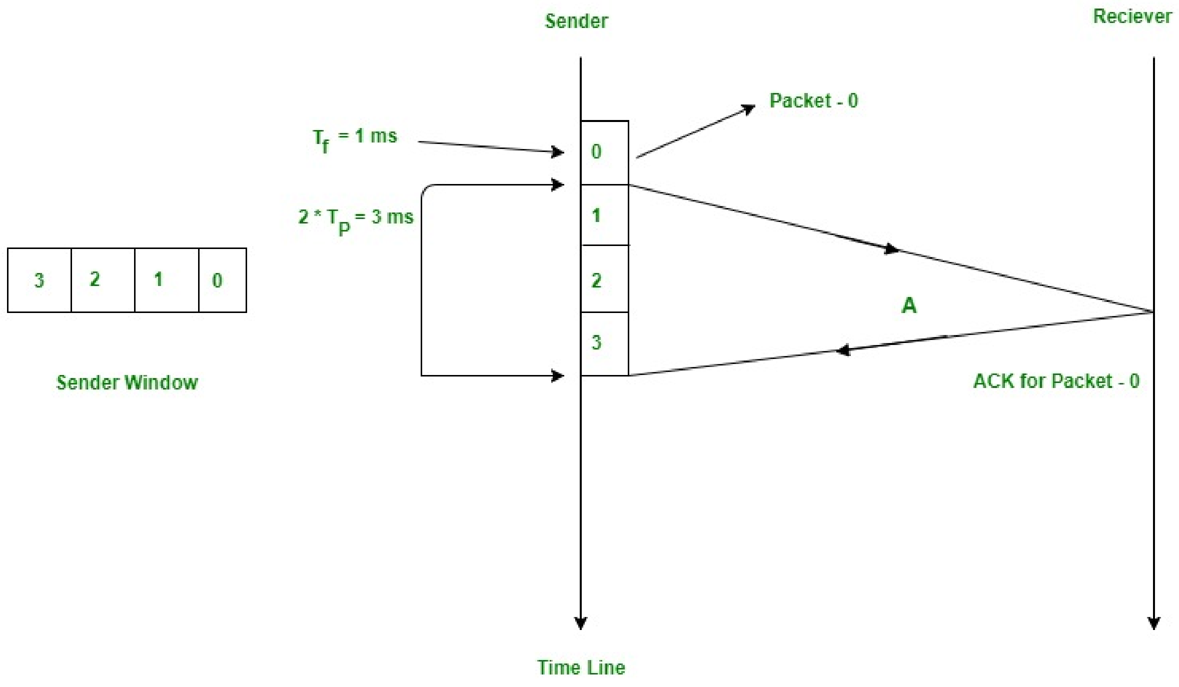
are connected to some server in some other country though a high speed connection), you can't use this full speed due to limitations of stop and wait.
Sliding Window protocol handles this efficiency issue by sending more than one packet at a time with a larger sequence numbers. The idea is same as pipelining 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 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)

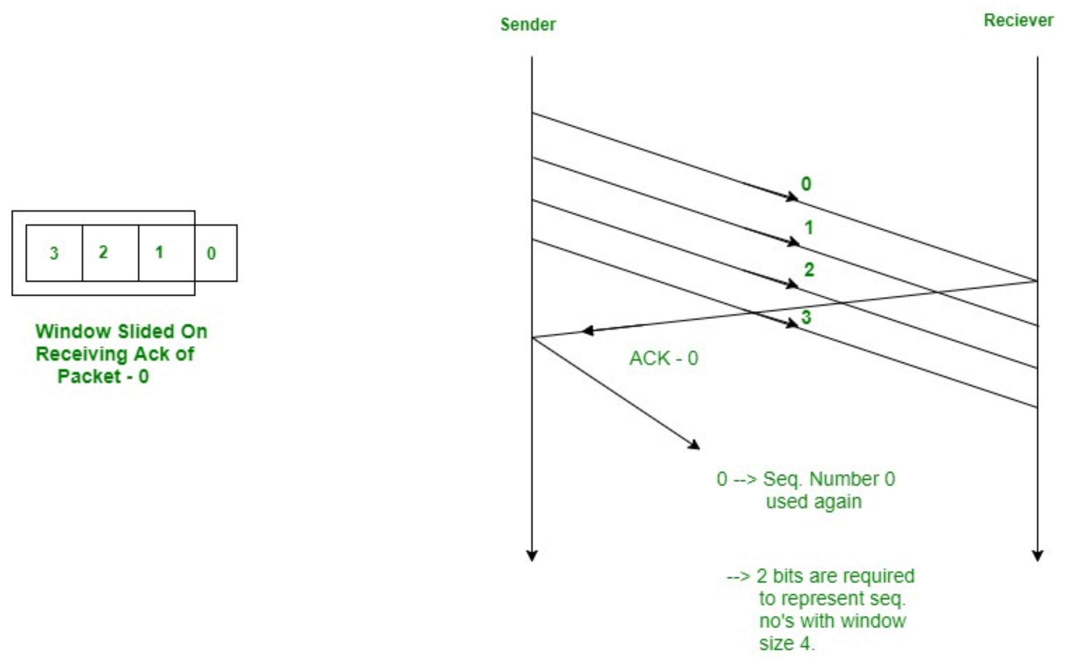
For Full Duplex channels,
Capacity = 2*Bandwidth(B) * Propagation(Tp)
Concept Of Pipelining
In Stop and Wait protocol, only 1 packet is transmitted onto the link and then sender waits for acknowledgement from the receiver. The problem in this setup is that efficiency is very less as we are not filling 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.


Sender
Reciever
3
2
1
0
Window Slided On Receiving Ack of Packet - 0
0
1
2
3
ACK - 0
0 --> Seq. Number 0 used again
--> 2 bits are required to represent seq. no's with window size 4.

Minimum Number Of Bits For Sender window (Very Important For GATE)
As we have seen above,
Maximum window size = 1 + 2*a   where a = Tp/Tt

Minimum sequence numbers required = 1 + 2*a.
All the packets in the current window will be given a sequence number. Number of bits required to represent the sender window = ceil(log2(1+2*a)).
But sometimes number of bits in the protocol headers is pre-defined. Size of sequence number field in header will also determine the maximum number of packets that we can send in total cycle time. If N is the size of sequence number field in the header in bits, then we can have 2^N sequence numbers.
Window Size ws = min(1+2*a, 2^N)
If you want to calculate minimum bits required to represent sequence numbers/sender window, it will be **ceil(log2(ws))**.
https://www.youtube.com/watch?v=WfIhQ3o2xow
https://www.youtube.com/watch?v=W6Xcp79oIbc

Unit-3
Topic-4

Why Piggybacking?
Communications are mostly full – duplex in nature, i.e. data transmission occurs in both directions. A method to achieve full – duplex communication is to consider both the communication as a pair of simplex communication. Each link comprises a forward channel for sending data and a reverse channel for sending acknowledgments.
However, in the above arrangement, traffic load doubles for each data unit that is transmitted. Half of all data transmission comprise of transmission of acknowledgments.

So, a solution that provides better utilization of bandwidth is piggybacking. Here, sending of acknowledgment is delayed until the next data frame is available for transmission. The acknowledgment is then hooked onto the outgoing data frame. The data frame consists of an *ack* field. The size of the *ack* field is only a few bits, while an acknowledgment frame comprises of several bytes. Thus, a substantial gain is obtained in reducing bandwidth requirement.

Working Principle

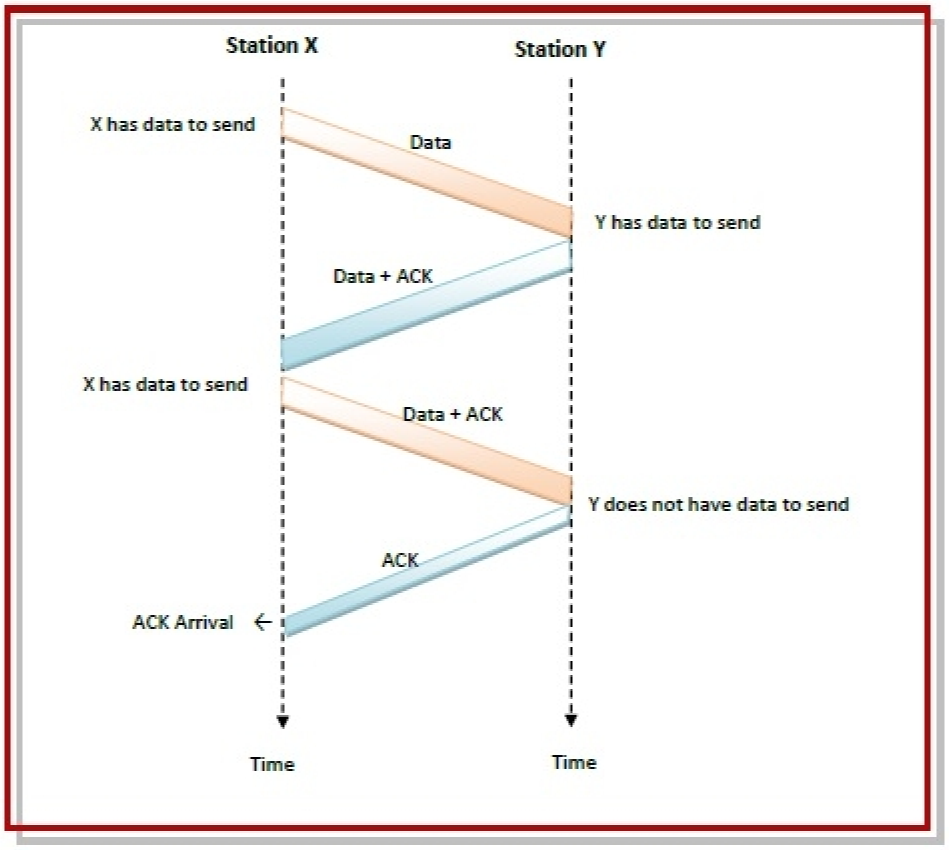
Suppose that there are two communication stations X and Y. The data frames transmitted have an acknowledgment field, *ack* field that is of a few bits length. Additionally, there are frames for sending acknowledgments, ACK frames. The purpose is to minimize the ACK frames.

The three principles governing piggybacking when the station X wants to communicate with station Y are:

1. If station X has both data and acknowledgment to send, it sends a data frame with the *ack* field containing the sequence number of the frame to be acknowledged.
2. If station X has only an acknowledgment to send, it waits for a finite period of time to see whether a data frame is available to be sent. If a data frame becomes available, then it piggybacks the acknowledgment with it. Otherwise, it sends an ACK frame.
3. If station X has only a data frame to send, it adds the last acknowledgment with it. The station Y discards all duplicate acknowledgments. Alternatively, station X may send the data frame with the *ack* field containing a bit combination denoting no acknowledgment.

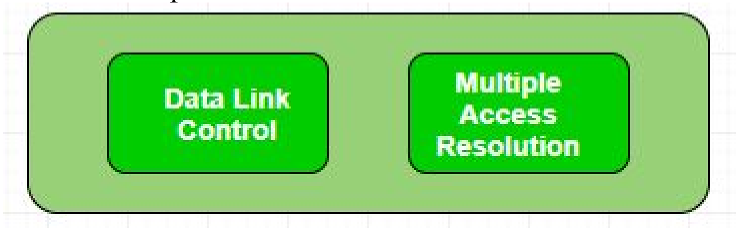
Example

The following diagram illustrates the three scenario −



Multiple Access Protocols in Computer Network
The Data Link Layer is responsible for transmission of data between two nodes. Its main functions are-

- Data Link Control
- Multiple Access Control

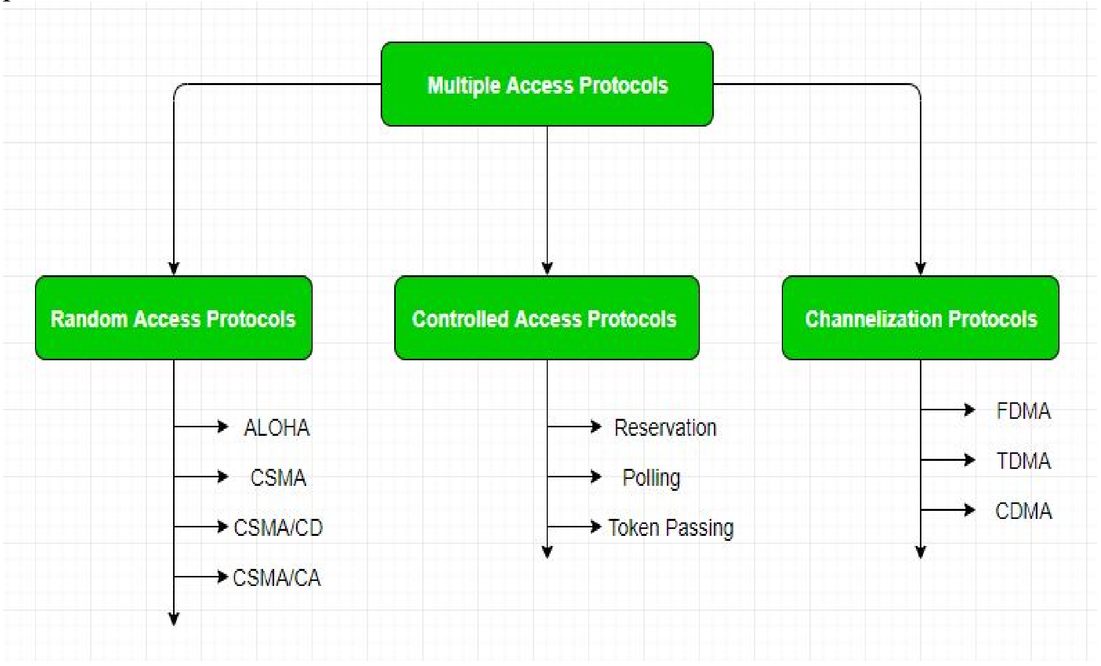


Data Link control –
The data link control is responsible for reliable transmission of message over transmission channel by using techniques like framing, error control and flow control.

Unit-3
Topic-5

Multiple Access Control

If there is a dedicated link between the sender and the receiver then data link control layer is sufficient, however if there is no dedicated link present then multiple stations can access the channel simultaneously. Hence multiple access protocols are required to decrease collision and avoid crosstalk. For example, in a classroom full of students, when a teacher asks a question and all the students (or stations) start answering simultaneously (send data at same time) then a lot of chaos is created(data overlap or data lost) then it is the job of the teacher (multiple access protocols) to manage the students and make them answer one at a time.
Thus, protocols are required for sharing data on non dedicated channels. Multiple access protocols can be subdivided further as –



1. Random Access Protocol: In this, all stations have same superiority that is no station has more priority than another station. Any station can send data depending on medium's state(idle or busy). It has two features:

1. There is no fixed time for sending data
2. There is no fixed sequence of stations sending data

The Random access protocols are further subdivided as:

(a) ALOHA – It was designed for wireless LAN but is also applicable for shared medium. In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled.

- **Pure Aloha:**
  When a station sends data it waits for an acknowledgement. If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time (Tb) and re-sends the data. Since different stations wait for different amount of time, the probability of further collision decreases.
- Vulnerable Time = 2* Frame transmission time
- Throughput = G exp{-2*G}

Maximum throughput = 0.184 for G=0.5

- **Slotted Aloha:**
  It is similar to pure aloha, except that we divide time into slots and sending of data is allowed only at the beginning of these slots. If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.
- Vulnerable Time = Frame transmission time
- Throughput = G exp{-*G}

Maximum throughput = 0.368 for G=1

(b) CSMA – Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data. If it is idle then it sends data, otherwise it waits till the channel becomes idle. However there is still chance of collision in CSMA due to propagation delay. For example, if station A wants to send data, it will first sense the medium.If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.

CSMA access modes-

- **1-persistent:** The node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally(with 1 probability) as soon as the channel gets idle.
- **Non-Persistent:** The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.
- **P-persistent:** The node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it send with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems.

- **O-persistent:** Superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

(c) CSMA/CD – Carrier sense multiple access with collision detection. Stations can terminate transmission of data if collision is detected.

(d) CSMA/CA – Carrier sense multiple access with collision avoidance. The process of collisions detection involves sender receiving acknowledgement signals. If there is just one signal(its own) then the data is successfully sent but if there are two signals(its own and the one with which it has collided) then it means a collision has occurred. To distinguish between these two cases, collision must have a lot of impact on received signal. However it is not so in wired networks, so CSMA/CA is used in this case.

CSMA/CA avoids collision by:

1. **Interframe space –** Station waits for medium to become idle and if found idle it does not immediately send data (to avoid collision due to propagation delay) rather it waits for a period of time called Interframe space or IFS. After this time it again checks the medium for being idle. The IFS duration depends on the priority of station.
2. **Contention Window –** It is the amount of time divided into slots. If the sender is ready to send data, it chooses a random number of slots as wait time which doubles every time medium is not found idle. If the medium is found busy it does not restart the entire process, rather it restarts the timer when the channel is found idle again.
3. **Acknowledgement –** The sender re-transmits the data if acknowledgement is not received before time-out.

2. Controlled Access:

In this, the data is sent by that station which is approved by all other stations.

3. Channelization:

In this, the available bandwidth of the link is shared in time, frequency and code to multiple stations to access channel simultaneously.

- **Frequency Division Multiple Access (FDMA) –** The available bandwidth is divided into equal bands so that each station can be allocated its own band. Guard bands are also added so that no to bands overlap to avoid crosstalk and noise.
- **Time Division Multiple Access (TDMA) –** In this, the bandwidth is shared between multiple stations. To avoid collision time is divided into slots and stations are allotted these slots to transmit data. However there is a overhead of synchronization as each station needs to know its time slot. This is resolved by adding synchronization bits to each slot. Another issue with TDMA is propagation delay which is resolved by addition of guard bands.
- **Code Division Multiple Access (CDMA) –** One channel carries all transmissions simultaneously. There is neither division of bandwidth nor division of time. For example, if there are many people in a room all speaking at the same time, then also perfect reception of data is possible if only two person speak the same language. Similarly data from different stations can be transmitted simultaneously in different code languages.

https://www.youtube.com/watch?v=YAjfUc7Tt24