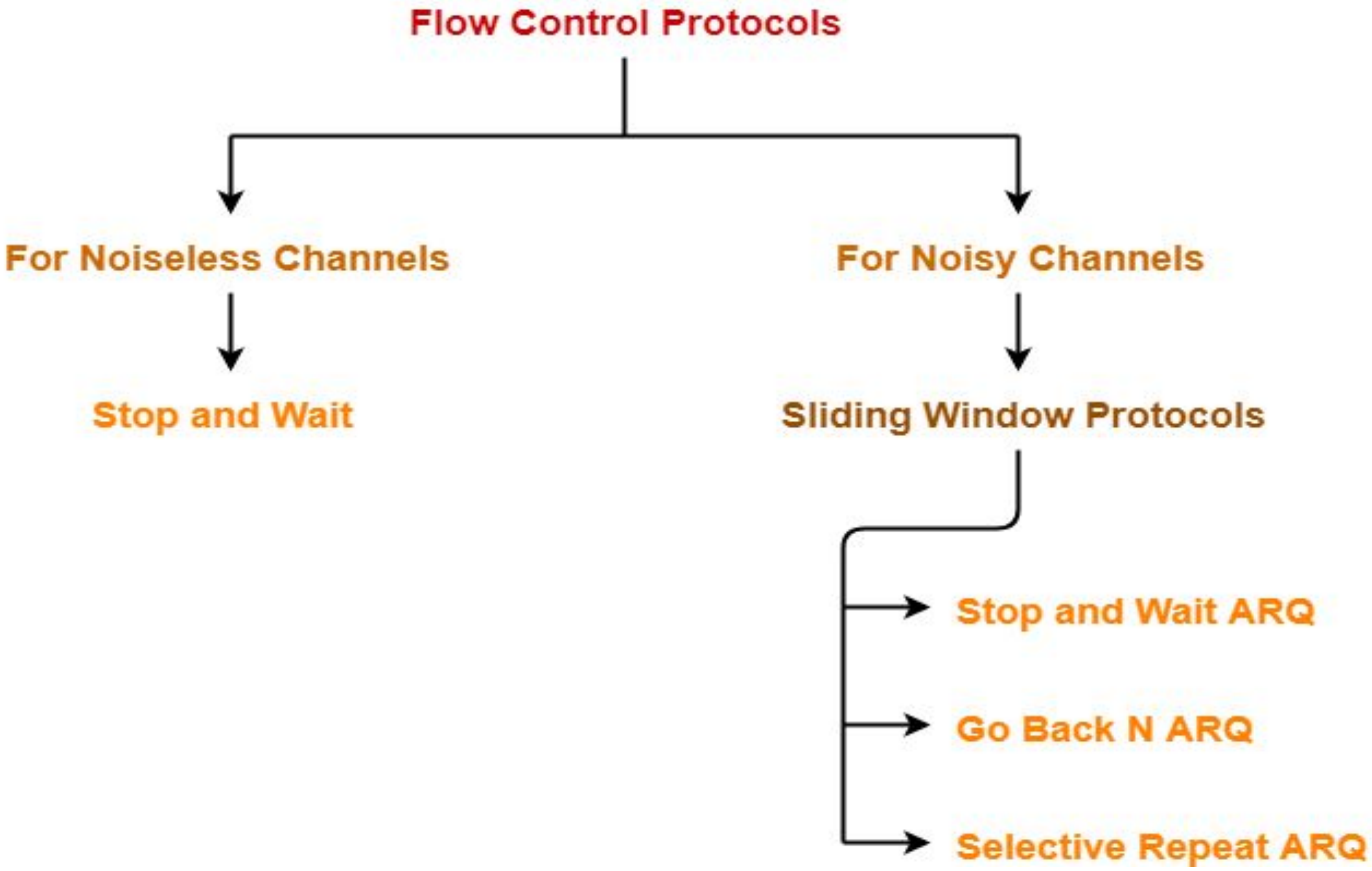
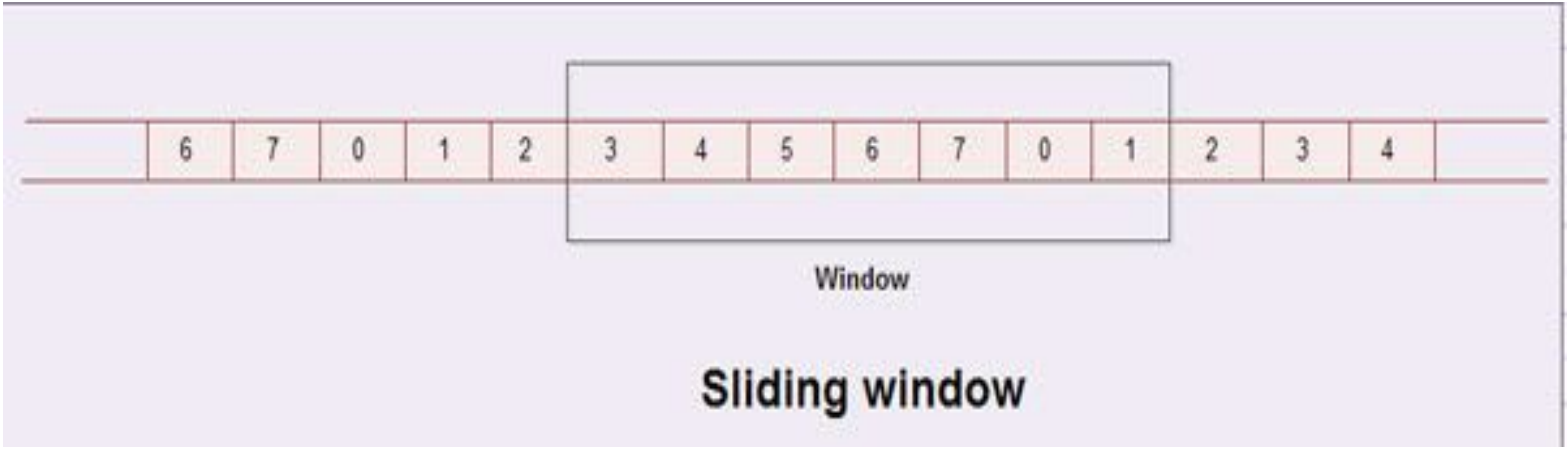
CN: WEEK 6

Sliding Window Protocol


Flow Control Protocols
For Noiseless Channels
For Noisy Channels
Stop and Wait
Sliding Window Protocols
Stop and Wait ARQ
Go Back N ARQ
Selective Repeat ARQ

- Sliding window protocol is a flow control protocol.
- With the help of the sliding window technique, multiple frames can be sent at a time by the sender before receiving any acknowledgment from the receiver.
- Sliding Window protocols make the use of TCP(transmission control protocol).
- The receiver can send the acknowledgment of multiple frames transmitted by the sender using a single ACK frame.
- In the Sliding Window protocols, the term sliding window mainly refers to the imaginary box that can hold the frames of both the sender side as well as receiver side.



- The windows have a specific size in which the frames are numbered modulo- n, which means they are numbered from 0 to n-1. For e.g. if n = 8, the frames are numbered 0, 1,2,3,4,5,6, 7, 0, 1,2,3,4,5,6, 7, 0, 1, ...

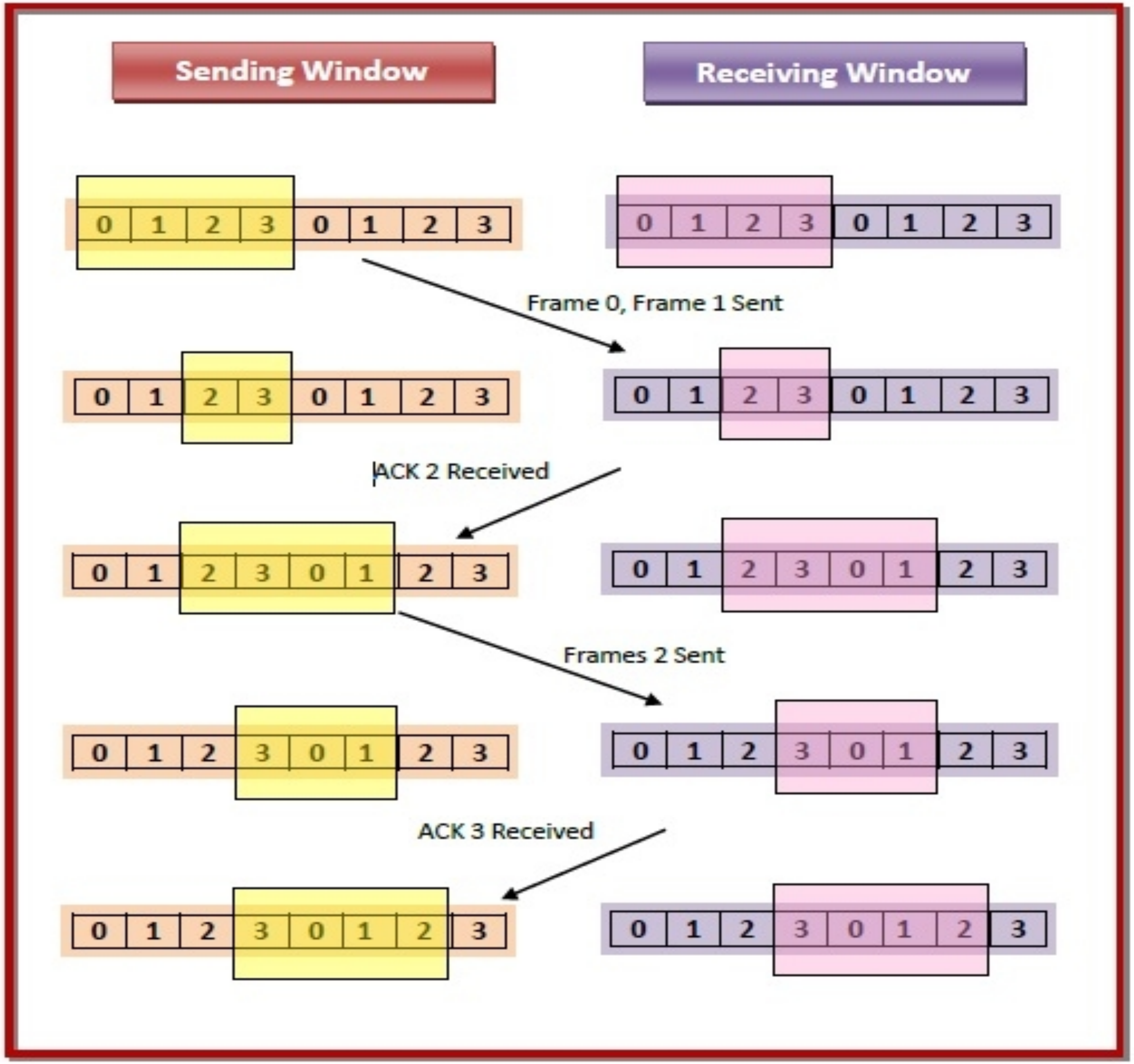
- When the receiver sends an ACK, it includes the number of next frame it expects to receive. For example in order to acknowledge the group of frames ending in frame 4, the receiver sends an ACK containing the number 5. When sender sees an ACK with number 5, it comes to know that all the frames up to number 4 have been received.

- **In these protocols, the sender has a buffer called the sending window and the receiver has buffer called the receiving window.**

- **The size of the sending window determines the sequence number of the outbound frames.**

- **The size of the receiving window is the maximum number of frames that the receiver can accept at a time**

Suppose that we have sender window and receiver window each of size 4. So the sequence numbering of both the windows will be 0,1,2,3,0,1,2 and so on. The following diagram shows the positions of the windows after sending the frames and receiving acknowledgments.


Sending Window
Receiving Window
0 1 2 3 0 1 2 3
0 1 2 3 0 1 2 3
Frame 0, Frame 1 Sent
0 1 2 3 0 1 2 3
0 1 2 3 0 1 2 3
ACK 2 Received
0 1 2 3 0 1 2 3
0 1 2 3 0 1 2 3
Frames 2 Sent
0 1 2 3 0 1 2 3
0 1 2 3 0 1 2 3
ACK 3 Received
0 1 2 3 0 1 2 3
0 1 2 3 0 1 2 3

Therefore, the sliding window of sender shrinks from left when frames of data are sending. The sliding window of the sender expands to right when acknowledgments are received.

The sliding window of the receiver shrinks from left when frames of data are received. The sliding window of the receiver expands to the right when acknowledgement is sent.

In a sliding window protocol, optimal sender window size = 1 + 2a

$$\text{Efficiency } (\eta) = \frac{T_t}{T_t + 2 \times T_p}$$

To get 100% efficiency, we must have-

$$\eta = 1$$

$$T_t / (T_t + 2T_p) = 1$$

$$T_t = T_t + 2T_p$$

Thus,

To get 100% efficiency, transmission time must be $T_t + 2T_p$ instead of T_t.

This means sender must send the frames in waiting time too.

Now, let us find the maximum number of frames that can be sent in time $T_t + 2T_p$.

In time T_t, sender sends one frame.

Thus, In time $T_t + 2T_p$, sender can send $(T_t + 2T_p) / T_t$ frames i.e. 1+2a frames.

Thus, to achieve 100% efficiency, window size of the sender must be 1+2a.

Required Sequence Numbers-

Each sending frame has to be given a unique sequence number.

Maximum number of frames that can be sent in a window = 1+2a.

So, minimum number of sequence numbers required = 1+2a.

To have 1+2a sequence numbers,

Minimum number of bits required in sequence number field = $\lceil \log_2(1+2a) \rceil$

Choosing a Window Size-

The size of the sender's window is bounded by-

1. Receiver's Ability-

Receiver's ability to process the data bounds the sender window size.

If receiver can not process the data fast, sender has to slow down and not transmit the frames too fast.

2. Sequence Number Field-

Number of bits available in the sequence number field also bounds the sender window size.

If sequence number field contains n bits, then 2^n sequence numbers are possible.

Thus, maximum number of frames that can be sent in one window = 2^n.

For n bits in sequence number field, Sender Window Size = min $(1+2a, 2^n)$

Efficiency-

Efficiency of any flow control protocol may be expressed as-

$$\text{Efficiency } (\eta) = \frac{\text{Number of frames sent in one window}}{\text{Total number of frames that can be sent in one window}}$$

OR

$$\text{Efficiency } (\eta) = \frac{\text{Sender Window Size in the Protocol}}{\text{Optimal Sender Window Size}}$$

OR

$$\text{Efficiency } (\eta) = \frac{\text{Sender Window Size in the Protocol}}{1 + 2a}$$

Q1. What is the efficiency In Stop and Wait ARQ protocol.

In **Stop and Wait ARQ**, sender window size = 1.

Thus,

Efficiency of Stop and Wait ARQ = 1 / 1+2a

Q2. If transmission delay & propagation delay in a sliding window protocol are 1 msec and 49.5 msec respectively, then-

a. What should be the sender window size to get the maximum efficiency?

b. What is the minimum number of bits required in the sequence number field?

c. If only 6 bits are reserved for sequence numbers, then what will be the efficiency?

Given-

Transmission delay = 1 msec

Propagation delay = 49.5 msec

To get the maximum efficiency,

sender window size = 1 + 2a

$= 1 + 2 \times (T_p / T_t)$

$= 1 + 2 \times (49.5 \text{ msec} / 1 \text{ msec})$

$= 1 + 2 \times 49.5 = 100$

Thus,

For maximum efficiency,

sender window size = 100

Minimum number of bits required in the sequence number field

$= \lceil \log_2(1+2a) \rceil$

$= \lceil \log_2(100) \rceil$

$= \lceil 6.8 \rceil = 7$

Thus,

Minimum number of bits required in the sequence number field = 7

If only 6 bits are reserved in the sequence number field, then-

Maximum sequence numbers possible = $2^6 = 64$

Efficiency

= Sender window size in the protocol / Optimal sender window size

= 64 / 100 = 64%

Q3. A 3000 km long trunk operates at 1.536 Mbps and is used to transmit 64 byte frames and uses sliding window protocol. If the propagation speed is 6 µsec / km, how many bits should the sequence number field be?

Given-

Distance = 3000 km

Bandwidth = 1.536 Mbps

Packet size = 64 bytes

Propagation speed = 6 µsec / km

Calculating Transmission Delay-

Transmission delay (T_t)

= Packet size / Bandwidth

= 64 bytes / 1.536 Mbps

= (64 x 8 bits) / (1.536 x 10^6 bits per sec)

= 333.33 µsec

Calculating Propagation Delay-

For 1 km, propagation delay = 6 µsec

For 3000 km, propagation delay = 3000 x 6 µsec = 18000 µsec

Calculating Value Of 'a'-

$a = T_p / T_t$

a = 18000 µsec / 333.33 µsec

a = 54

Calculating Bits Required in Sequence Number Field-

Bits required in sequence number field

$= \lceil \log_2(1+2a) \rceil = \lceil \log_2(1 + 2 \times 54) \rceil$

$= \lceil \log_2(109) \rceil = \lceil 6.76 \rceil$

= 7 bits

Thus,

Minimum number of bits required in sequence number field = 7

With 7 bits, number of sequence numbers possible = 128

Q4. Compute approximate optimal window size when packet size is 53 bytes, RTT is 60 msec and bottleneck bandwidth is 155 Mbps.

Given-

Packet size = 53 bytes

RTT = 60 msec

Bandwidth = 155 Mbps

Calculating Transmission Delay-

Transmission delay (T_t)

= Packet size / Bandwidth

= 53 bytes / 155 Mbps

= (53 x 8 bits) / (155 x 10^6 bits per sec)

= 2.735 μsec

Calculating Propagation Delay-

Propagation delay (T_p)

= Round Trip Time / 2

= 60 msec / 2 = 30 msec

Calculating Value of 'a'-

a = T_p / T_t

a = 30 msec / 2.735 μsec

a = 10968.921

Calculating Optimal Window Size-

Optimal window size

= 1 + 2a

= 1 + 2 x 10968.921

= 21938.84

Thus, approximate optimal window size

= 21938 frames.

Q5. Host A is sending data to host B over a full duplex link. A and B are using the sliding window protocol for flow control. The send and receive window sizes are 5 packets each. Data packets (sent only from A to B) are all 1000 bytes long and the transmission time for such a packet is 50 µs. Acknowledgement packets (sent only from B to A) are very small and require negligible transmission time. The propagation delay over the link is 200 µs. What is the maximum achievable throughput in this communication?

Given-

Sender window size = Receiver window size = 5

Packet size = 1000 bytes

Transmission delay (T_t) = 50 µs

Propagation delay (T_p) = 200 µs

Calculating Bandwidth-

We know,

Transmission delay = Packet size / Bandwidth

So, Bandwidth

= Packet Size / Transmission delay (T_t)

= 1000 bytes / 50 µs

= (1000 x 8 bits) / (50 x 10^{-6} sec)

= 160 Mbps

Calculating Value of 'a'-

$a = T_p / T_t$

a = 200 µsec / 50 µsec

a = 4

Calculating Optimal Window Size-

Optimal window size

= 1 + 2a

= 1 + 2 x 4

= 9

Calculating Efficiency-

Efficiency (η)

= Sender window size / Optimal window size

= 5 / 9 = 55.55%

Calculating Maximum Achievable Throughput-

Maximum achievable throughput

= Efficiency (η) x Bandwidth

= 0.5555 x 160 Mbps

= 88.88 Mbps

Q6. Station A uses 32 byte packets to transmit messages to station B using sliding window protocol. The round trip delay between A and B is 80 msec and the bottleneck bandwidth on the path between A and B is 128 Kbps. What is the optimal window size that A should use?

Given-

Packet size = 32 bytes

Round Trip Time = 80 msec

Bandwidth = 128 Kbps

Calculating Transmission Delay-

Transmission delay (T_t)

= Packet size / Bandwidth

= 32 bytes / 128 Kbps

= (32 x 8 bits) / (128 x 10^3 bits per sec)

= 2 msec

Calculating Propagation Delay-

Propagation delay (T_p)

= Round Trip Time / 2

= 80 msec / 2

= 40 msec

Calculating Value of 'a'-

$a = T_p / T_t$

a = 40 msec / 2 msec

a = 20

Calculating Optimal Window Size-

Optimal window size

= 1 + 2a

= 1 + 2 x 20 = 41

Implementation of Sliding Window Protocol

Go back N Protocol

In Go-Back-N, if a sent frame is damaged then all frames are retransmitted till the last packet.

Selective Repeat Protocol

In Selective Repeat, only the damaged frames are retransmitted.

Go Back N
Protocol

- Go back N protocol is an implementation of a sliding window protocol where sender window size is N and receiver window size is always 1. Suppose we say that Go-Back-3, which means that the three frames can be sent at a time before expecting the acknowledgment from the receiver.

- It uses pipelining concept in which multiple frames can be sent before receiving the acknowledgment of the first frame. If we have total five frames and it is Go-Back-3, which means that three frames can be sent, i.e., frame no 1, frame no 2, frame no 3 can be sent before expecting the acknowledgment of frame no 1.

- The number of frames that can be sent at a time totally depends on the size of the sender's window. So, we can say that 'N' is the number of frames that can be sent at a time before receiving the acknowledgment from the receiver.

- If the acknowledgment of a frame is not received in the predefined time period, then all the frames available in the current window will be retransmitted. Suppose we have sent the frame no 5, but we didn't receive the acknowledgment of frame no 5, and the current window is holding three frames, then all these three frames will be retransmitted.

Note 1:

In Go back N, sender window size is N and receiver window size is always 1.

- *Sender window size = N. Example in Go back 10, sender window size will be 10.*
- *Receiver window size is always 1 for any value of N.*

Note 2:

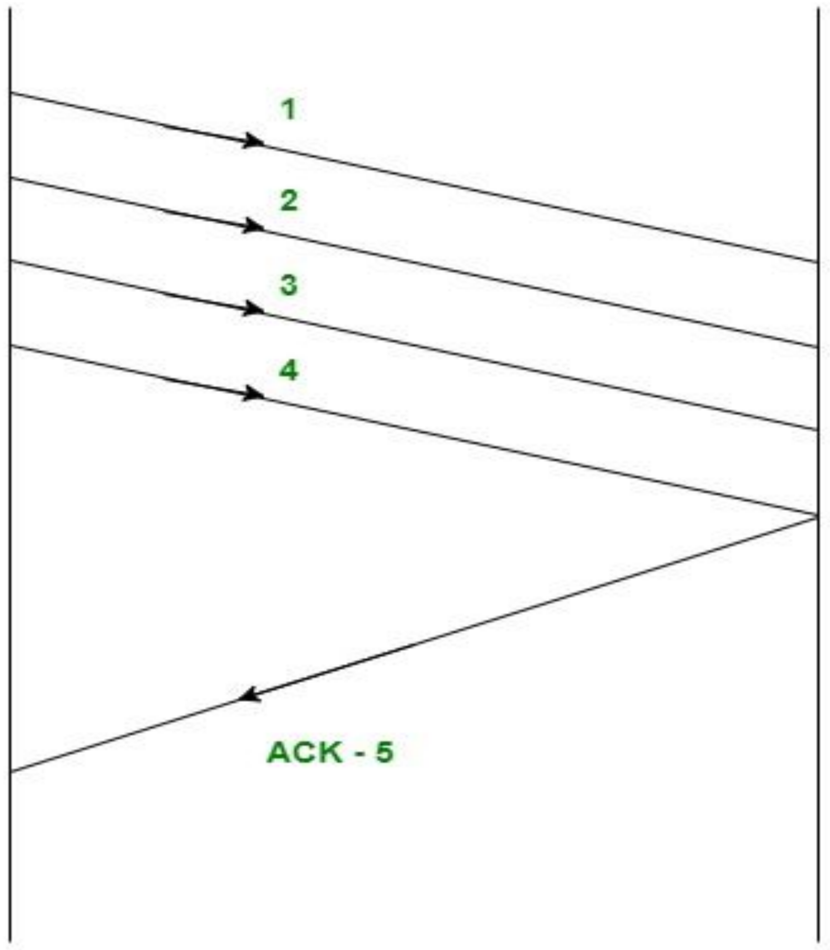
Go back N uses cumulative acknowledgements.

- ***Receiver maintains an acknowledgement timer.***
- ***Each time the receiver receives a new frame, it starts a new acknowledgement timer.***
- ***After the timer expires, receiver sends the cumulative acknowledgement for all the frames that are unacknowledged at that moment.***
- ***A new acknowledgement timer does not start after the expiry of old acknowledgement timer.***
- ***It starts after a new frame is received.***

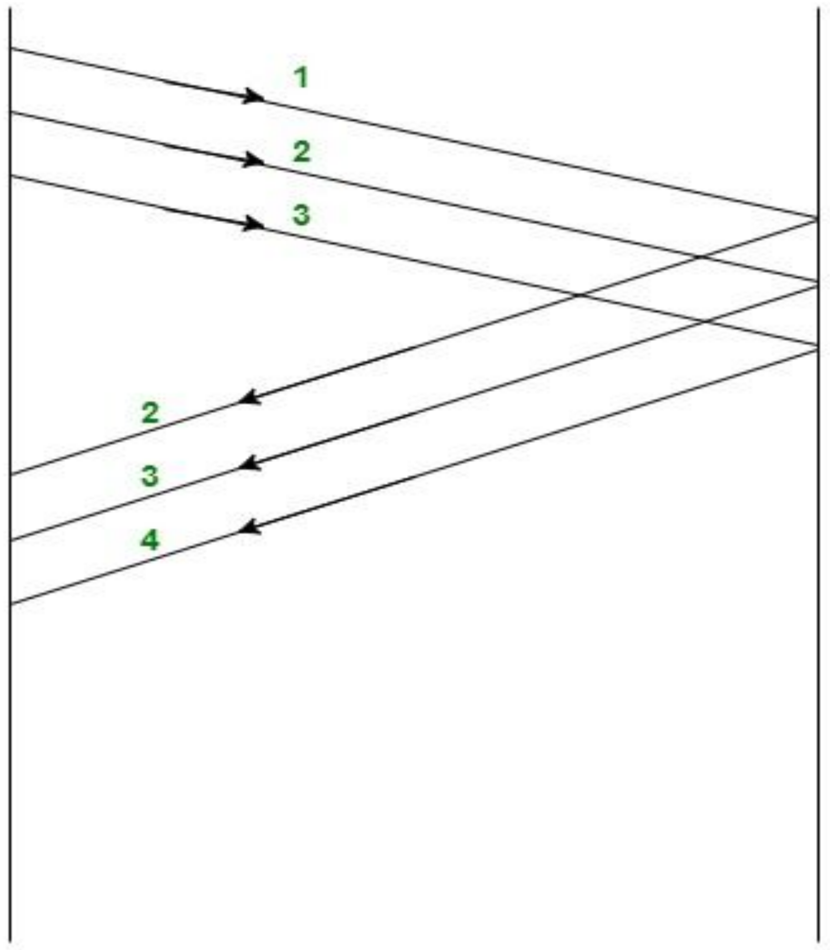
Note 3:

Go back N may use independent acknowledgements too.

- ***Go back N may use independent acknowledgements too if required.***
- ***The kind of acknowledgement used depends on the expiry of acknowledgement timer.***
- ***Consider after the expiry of acknowledgement timer, there is only one frame left to be acknowledged.***
- ***Then, Go back N sends the independent acknowledgement for that frame.***



Cummulative



INDEPENDENT

<u>Cumulative Ack:</u> **One acknowledgement is used for many packets. The main advantage is traffic is less. A disadvantage is less reliability as if one ack is the loss that would mean that all the packets sent are lost.**

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

Note 4:

Go back N does not accept the corrupted frames and silently discards them.

- *If receiver receives a frame that is corrupted, then it silently discards that frame.*
- *The correct frame is retransmitted by the sender after the time out timer expires.*
- *Silently discarding a frame means- “Simply rejecting the frame and not taking any action”*

Note 5:

Go back N does not accept out of order frames and silently discards them.

- ***If receiver receives a frame whose sequence number is not what the receiver expects, then it silently discards that frame.***
- ***All the following frames are also discarded.***
- ***This is because receiver window size is 1 and therefore receiver can not accept out of order frames.***

Note 6:

Go back N leads to retransmission of entire window if for any frame, no ACK is received by the sender.

- *Receiver silently discards the frame if it founds the frame to be either corrupted or out of order.*
- *It does not send any acknowledgement for such frame.*
- *It silently discards all the following frames too.*

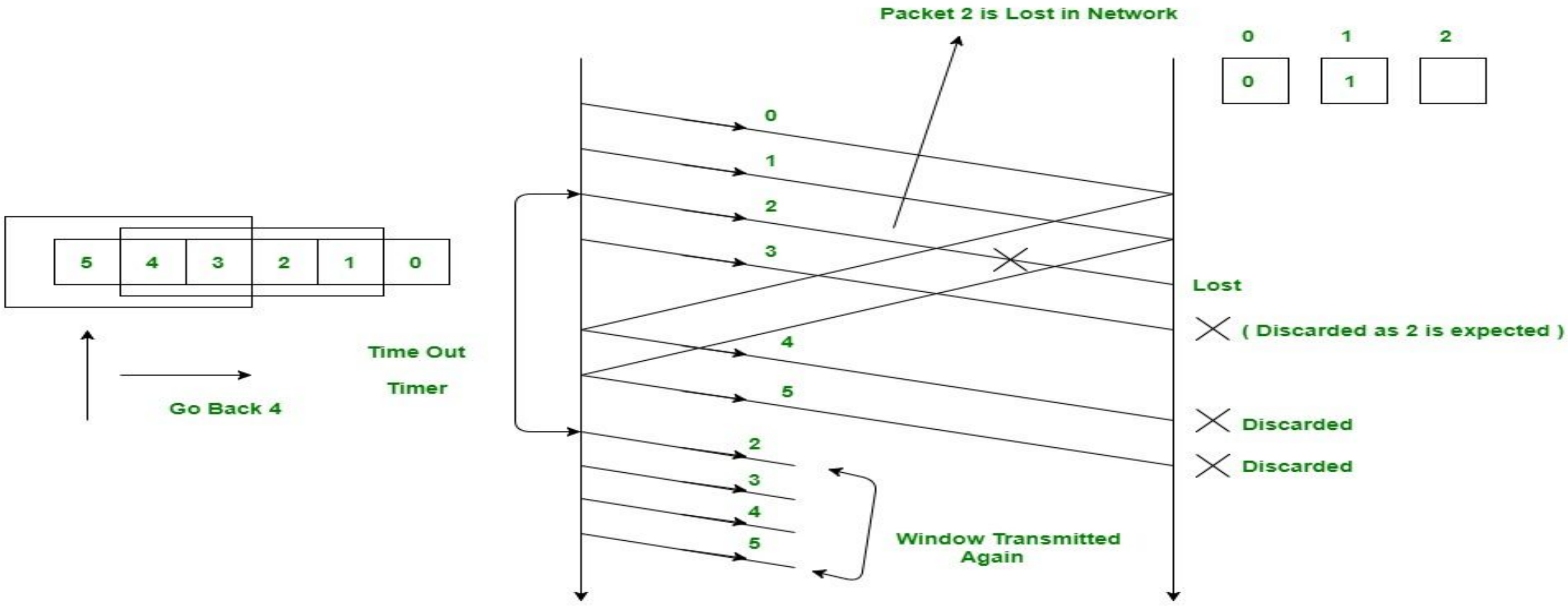
If for any particular frame, sender does not receive any acknowledgement, then it understands that along with that frame, all the following frames must also have been discarded by the receiver. So, sender has to retransmit all the following frames too along with that particular frame. Thus, it leads to the retransmission of entire window.

That is why, the protocol has been named as "Go back N".

Consider the diagram. We have sender window size of 4. Now the sender has sent the packets 0, 1, 2 and 3. After acknowledging the packets 0 and 1, receiver is now expecting packet 2 and sender window has also slided to further transmit the packets 4 and 5.

Now suppose the packet 2 is lost in the network, 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 sender 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 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.



Note 7:

Go back N leads to retransmission of lost frames after expiry of time out timer.

- *Consider a frame being sent to the receiver is lost on the way.*
- *Then, it is retransmitted only after time out timer expires for that frame at sender's side.*

Efficiency of Go back N-

Efficiency = Sender Window Size in Protocol / (1 + 2a)

In Go back N protocol, sender window size = N.

$$\boxed{\text{Efficiency of Go back N} = N / (1 + 2a)}$$

Practice Problems of Go Back N Protocol

Q1. A 20 Kbps satellite link has a propagation delay of 400 ms. The transmitter employs the "go back n ARQ" scheme with n set to 10. Assuming that each frame is 100 bytes long, what is the maximum data rate possible?

Given-

Bandwidth = 20 Kbps

Propagation delay (T_p) = 400 ms

Frame size = 100 bytes

Go back N is used where N = 10

Calculating Transmission Delay-

Transmission delay (T_t)

= Frame size / Bandwidth

= 100 bytes / 20 Kbps

= (100 x 8 bits) / (20 x 10^3 bits per sec)

= 0.04 sec = 40 msec

Calculating Value Of 'a'-

$a = T_p / T_t$

a = 400 msec / 40 msec

a = 10

Calculating Efficiency-

Efficiency (η)

= N / (1+2a)

= 10 / (1 + 2 x 10)

= 10 / 21

= 0.476

= 47.6 %

Calculating Maximum Data Rate Possible-

Maximum data rate possible or Throughput

= Efficiency x Bandwidth

= 0.476 x 20 Kbps

= 9.52 Kbps $\cong$ 10 Kbps

Q2. A 1 Mbps satellite link connects two ground stations. The altitude of the satellite is 36504 km and speed of the signal is 3 x 108 m/sec. What should be the packet size for a channel utilization of 25% for a satellite link using go back 127 sliding window protocol?

Given-

Bandwidth = 1 Mbps

Distance = 2 x 36504 km = 73008 km

Propagation speed = 3×10^8 m/sec

Efficiency = 25% = 1/4

Go back N is used where N = 127

Let the packet size be L bits.

Calculating Transmission Delay-

Transmission delay (T_t)

= Packet size / Bandwidth

= L bits / 1 Mbps

= L μsec

Calculating Propagation Delay-

Propagation delay (T_p)

= Distance / Speed

= $(73008 \times 10^3$ m$) / (3 \times 10^8$ m/sec$)$

= 24336×10^{-5} sec

= 243360 μsec

Calculating Value of 'a'-

$a = T_p / T_t$

a = 243360 μsec / L μsec

a = 243360 / L

Calculating Packet Size-

Efficiency (η) = N / (1+2a)

Substituting the values, we get-

$1/4 = 127 / (1 + 2 \times 243360 / L)$

$1/4 = 127 \times L / (L + 486720)$

$L + 486720 = 508 \times L$

$507 \times L = 486720$

$L = 960$

Q3. Consider a network connecting two systems located 8000 km apart. The bandwidth of the network is 500×10^6 bits per second. The propagation speed of the media is 4×10^6 meters per second. It is needed to design a Go back N sliding window protocol for this network. The average packet size is 10^7 bits. The network is to be used to its full capacity. Assume that processing delays at nodes are negligible. Then, find the minimum size in bits of the sequence number field.

Given-

Distance = 8000 km

Bandwidth = 500×10^6 bps

Propagation speed = 4×10^6 m/sec

Packet size = 10^7 bits

Now,

For using the network to its full capacity, Efficiency (η) = 1

Efficiency (η) = 1 when sender window size = 1+2a

Calculating Transmission Delay-

Transmission delay (T_t)

= Packet size / Bandwidth

= 10^7 bits / (500×10^6 bits per sec)

= 1 / 50 sec

Calculating Propagation Delay-

Propagation delay (T_p)

= Distance / Speed

= 8000 km / (4×10^6 m/sec)

= 2 sec

Calculating Value of 'a'-

$a = T_p / T_t$

a = 2 sec / 0.02 sec

a = 100

Calculating Sender Window Size-

Sender window size

= 1 + 2a

= 1 + 2 x 100

= 201

Calculating Minimum Size Of Sequence Number Field-

Minimum number of bits required in the sequence number field

$= \lceil \log_2(1+2a) \rceil = \lceil \log_2(201) \rceil$

$= \lceil 7.65 \rceil = 8$

Thus, Minimum size of sequence number field = 8 bits.

Selective Repeat Protocol

- The Go-back-N ARQ protocol works well if it has fewer errors and the link is reliable. But if there is a lot of error in the frame, lots of bandwidth is lost in sending the frames again.

- To overcome this problem, an alternative technique known as Selective Repeat Protocol.

- Selective Repeat (SR) protocol allows the receiver to accept & buffer the data packets following a damaged or lost one and then retransmits only those data packets which were lost or damaged in the network channel during transmission.

- As an example, consider you are sending data packets p1 to p5, now p1 & p2 are received successfully but p3 got lost or damaged in the network channel, now unlike Go-Back-N ARQ, as per the selective repeat protocol, the sender will send the p4 & p5, which the receiver will receive and after that, the selective repeat protocol will retransmit p3.

- Hence, only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.

Note 1:

In SR protocol, sender window size is always same as receiver window size.

- ***Sender window size = Receiver window size***
- ***The size is of sender greater than 1 otherwise the protocol will become Stop and Wait ARQ.***
- ***If n bits are available for sequence numbers, then-***

Sender window size = Receiver window size = $2^n/2 = 2^{n-1}$

Note 2:

SR protocol uses independent acknowledgements only.

- ***Receiver acknowledges each frame independently.***
- ***As receiver receives a new frame from the sender, it sends its acknowledgement.***

Note 3:

SR protocol does not accept the corrupted frames but does not silently discard them.

- *If receiver receives a frame that is corrupted, then it does not silently discard that frame.*
- *Receiver handles the situation efficiently by sending a negative acknowledgement (NACK).*
- *Negative acknowledgement allows early retransmission of the corrupted frame.*
- *It also avoids waiting for the time out timer to expire at the sender side to retransmit the frame.*

Note 4:

SR protocol accepts the out of order frames.

- ***Consider receiver receives a frame whose sequence number is not what the receiver expects.***
- ***Then, it does not discard that frame rather accepts it and keeps it in its window.***

Note 5:

SR protocol requires sorting at the receiver's side.

- ***Receiver window is implemented as a linked list.***
- ***When receiver receives a new frame, it places the new frame at the end of the linked list.***
- ***When the received frames are out of order, receiver performs the sorting.***
- ***Sorting sorts the frames in the correct order.***

Note 6:

SR protocol requires searching at the sender's side.

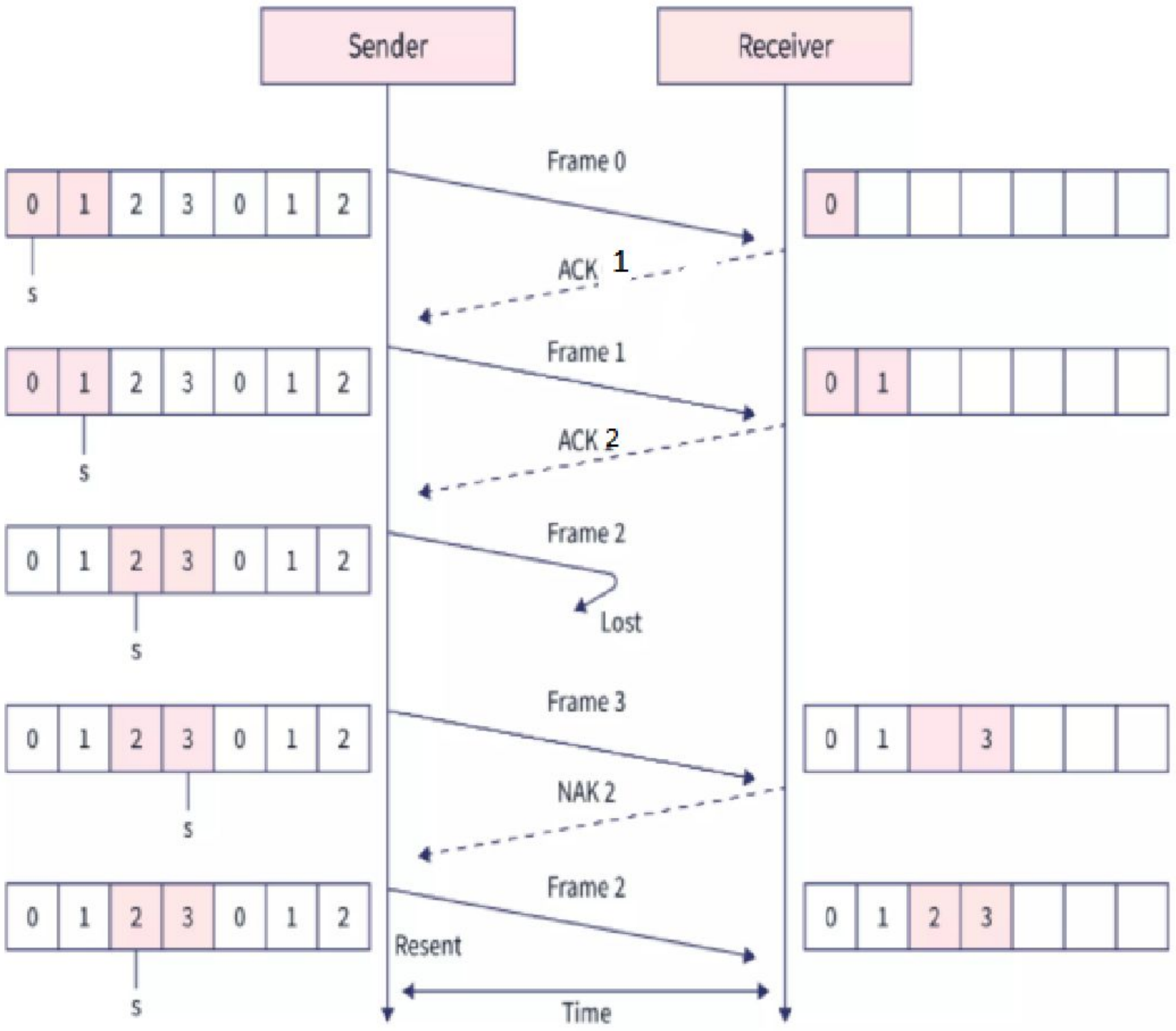
- ***Receiver does not reject the out of order frames.***
- ***Receiver accepts the out of order frames and sort them later.***
- ***Thus, only the missing frame has to be sent by the sender.***
- ***For sending the missing frame, sender performs searching and finds the missing frame.***
- ***Then, sender selectively repeats that frame.***
- ***Thus, only the selected frame is repeated and not the entire window.***
- ***That is why, the protocol has been named as "Selective Repeat Protocol".***

Note 7:

SR protocol leads to retransmission of lost frames after expiry of time out timer.

- *Consider a frame being sent to the receiver is lost on the way.*
- *Then, it is retransmitted only after time out timer expires for that frame at sender's side.*

The sender only retransmits frames, for which a NAK is received



Efficiency of SR Protocol-

Efficiency of any flow control protocol is given by-

Efficiency = Sender Window Size in Protocol / (1 + 2a)

In selective repeat protocol, if sender window size = N, then-

$$\boxed{\text{Efficiency of SR Protocol} = N / (1 + 2a)}$$

Justify the statement The maximum window size possible of sender and receiver = 2^{n-1}

We know-

- With n bits, total number of sequence numbers possible = 2^n.
- In SR Protocol, sender window size = receiver window size = W (say)

For any sliding window protocol to work without any problems,

Min Available Sequence Numbers = Sender window size + Receiver window size

So, we have-

$$2^n = W + W = 2W$$

$$W = 2^{n-1}$$

Therefore, maximum window size possible of sender and receiver = 2^{n-1}

Q. Consider a 128×10^3 bits/sec satellited communication link with one way propagation delay of 150 msec. Selective repeat protocol is used on this link to send data with a frame size of 1 KB. Neglect the transmission time of acknowledgement. Calculate the minimum number of bits required for the sequence number field to achieve 100% utilization.

Given-

Bandwidth = 128×10^3 bits/sec

Propagation delay (T_p) = 150 msec

Frame size = 1 KB

Now,

- To achieve 100% utilization, efficiency must be 100%.
- Efficiency is 100% when sender window size is optimal i.e. 1+2a

Calculating Transmission Delay-

Transmission delay (T_t)
= Frame size / Bandwidth
= 1 KB / (128×10^3 bits per sec)
= 64 msec

Calculating Value of 'a'-

$a = T_p / T_t$
a = 150 msec / 64 msec
a = 2.34

Calculating Optimal Sender Window Size-

Optimal sender window size
= 1 + 2a
= 1 + 2 x 2.34
= $\lceil 5.68 \rceil$ = 6

Calculating Number Of Sequence Numbers Required-

In SR Protocol, sender window size and receiver window size are same.
So, sender window size = receiver window size = 6

Now,

For any sliding window protocol, minimum number of sequence numbers required
= Sender window size + Receiver window size
= 6 + 6 = 12

Calculating Bits Required in Sequence Number Field-

To have 12 sequence numbers,

Minimum number of bits required in sequence number field
= $\lceil \log_2(12) \rceil$ = 4

Thus,

- Minimum number of bits required in sequence number field = 4
- With 4 bits, number of sequence numbers possible = 16
- We use only 12 sequence numbers and rest 4 remains unused.

Go-Back-N Protocol	Selective Repeat Protocol
In Go-Back-N Protocol, if the sent frame are find suspected then all the frames are re-transmitted from the lost packet to the last packet transmitted.	In selective Repeat protocol, only those frames are re-transmitted which are found suspected.
Sender window size of Go-Back-N Protocol is N.	Sender window size of selective Repeat protocol is also N.
Receiver window size of Go-Back-N Protocol is 1.	Receiver window size of selective Repeat protocol is N.
Go-Back-N Protocol is less complex.	Selective Repeat protocol is more complex.
In Go-Back-N Protocol, neither sender nor at receiver need sorting.	In selective Repeat protocol, receiver side needs sorting to sort the frames.
In Go-Back-N Protocol, type of Acknowledgement is cumulative.	In selective Repeat protocol, type of Acknowledgement is individual.
In Go-Back-N Protocol, Out-of-Order packets are NOT Accepted (discarded) and the entire window is re-transmitted.	In selective Repeat protocol, Out-of-Order packets are Accepted.
In Go-Back-N Protocol, if Receives receives a corrupt packet, then also, the entire window is re-transmitted.	In selective Repeat protocol, if Receives receives a corrupt packet, it immediately sends a negative acknowledgement and hence only the selective packet is retransmitted.

Metrics	Stop and Wait ARQ	Go back N	Selective Repeat	Remarks
Efficiency	1 / (1+2a)	N / (1+2a)	N / (1+2a)	Go back N and Selective Repeat gives better efficiency than Stop and Wait ARQ.
Window Size	Sender Window Size = 1 Receiver Window Size = 1	Sender Window Size = N Receiver Window Size = 1	Sender Window Size = N Receiver Window Size = N	Buffer requirement in Selective Repeat is very large. If the system does not have lots of memory, then it is better to choose Go back N.
Min sequence numbers	2	N+1	2 X N	Selective Repeat requires large number of bits in sequence number field.
Retransmissions required if a packet is lost	Only the lost packet is retransmitted	The entire window is retransmitted	Only the lost packet is retransmitted	Selective Repeat is far better than Go back N in terms of retransmissions required.
Bandwidth requirement	Low	High as even if a single packet is lost, entire window has to be retransmitted.	Moderate	Selective Repeat is better than Go back N in terms of bandwidth requirement.
CPU usage	Low	Moderate	High due to searching and sorting required at sender and receiver side	Go back N is better than Selective Repeat in terms of CPU usage.
Difficulty in Implementation	Low	Moderate	Complex as it requires extra logic for sorting and searching	Go back N is better than Selective Repeat in terms of implementation difficulty.
Acknowledgements	Uses independent acknowledgement for each packet	Uses cumulative acknowledgements (may use independent acknowledgements also)	Uses independent acknowledgement for each packet	Sending cumulative acknowledgements reduces the traffic in the network but if it is lost, then the ACKs for all the corresponding packets are lost.
Transmission	Half duplex	Full duplex	Full duplex	Go back N and Selective Repeat are better in terms of channel usage.

INFERENCE

- Go back N is more often used than other protocols.
- SR protocol is less used because of its complexity.
- Stop and Wait ARQ is less used because of its low efficiency.
- Depending on the context and resources availability, Go back N or Selective Repeat is employed.
- Selective Repeat and Stop and Wait ARQ are similar in terms of retransmissions.
- Go back N and Selective Repeat are similar in terms of efficiency if sender window sizes are same.
- SR protocol may be considered as a combination of advantages of Stop and Wait ARQ and Go back N.
- SR protocol is superior to other protocols but because of its complexity, it is less used.

END