Flow Control

1) Stop-And-Wait ARQ

A sender, station A, has 3500 bytes to send and can transmit a maximum of 1000 bytes in each frame as payload i.e. MTU = 1000 bytes.

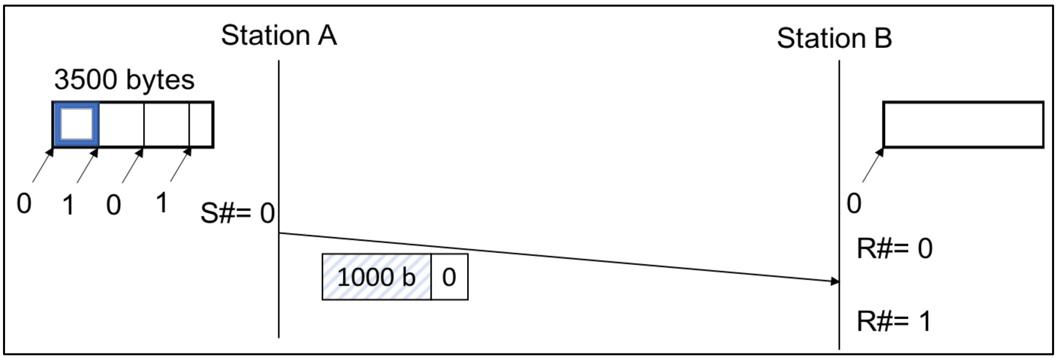
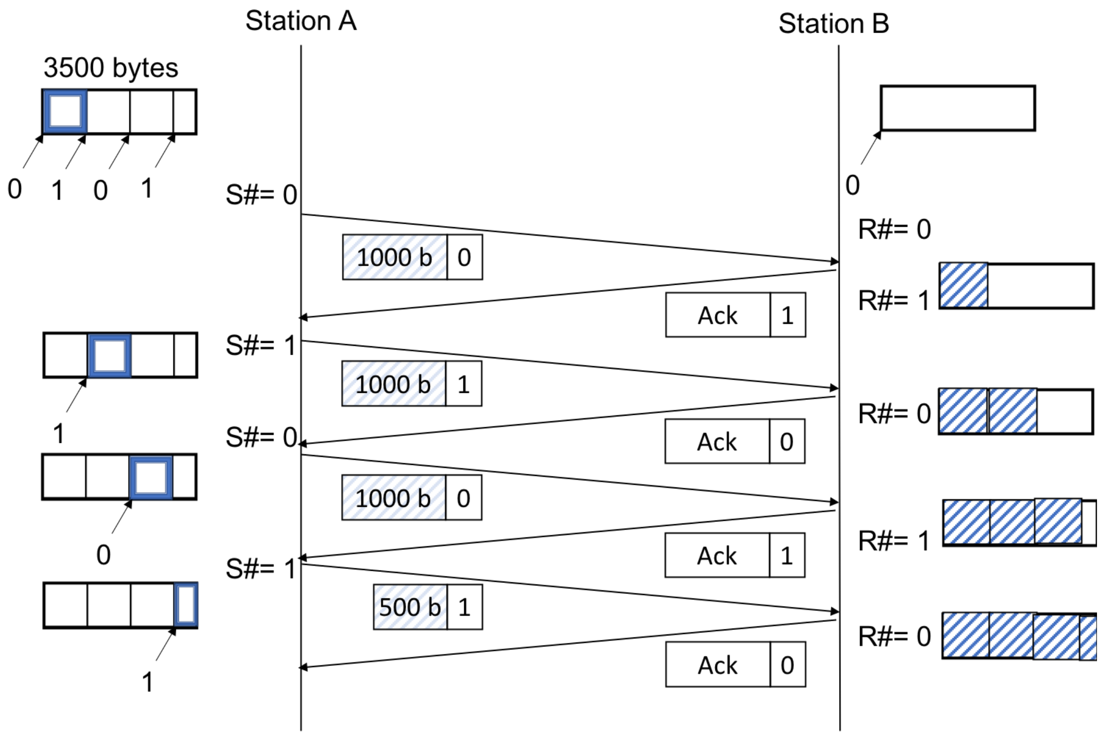


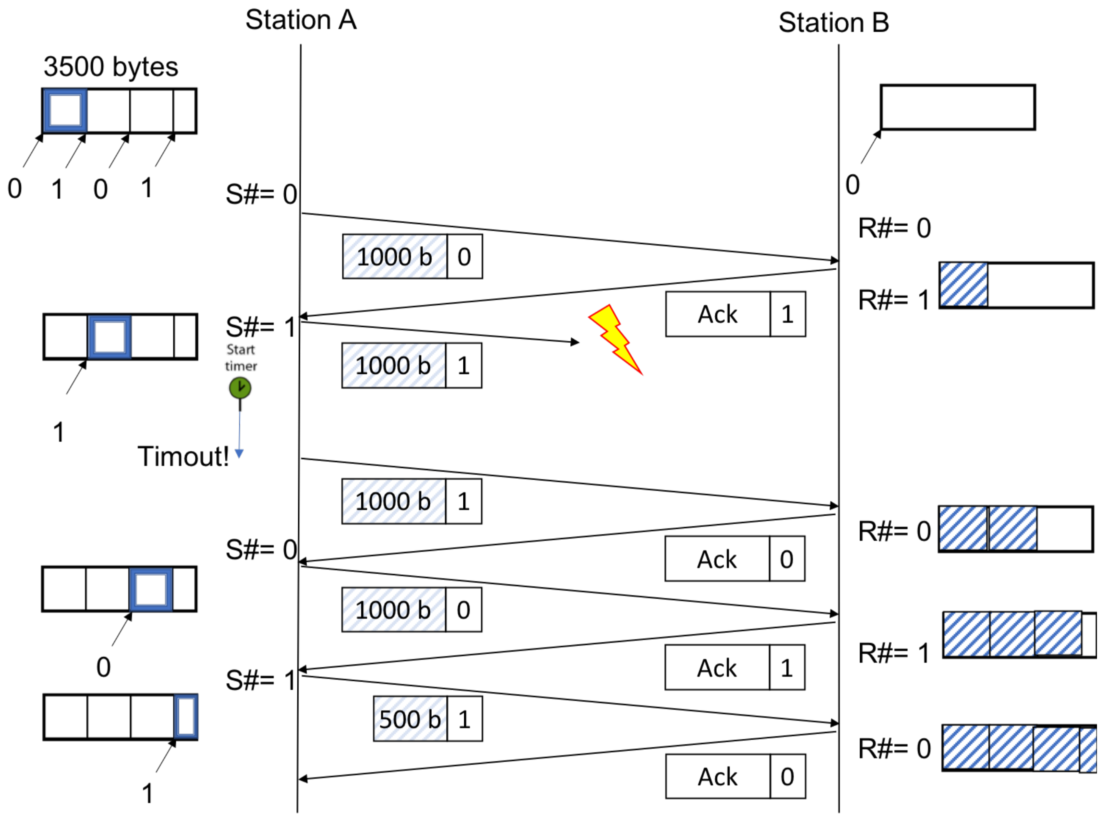
Figure 1: Start of a transmission of 3500 bytes using Stop&Wait. The header may require additional information.

I. Complete the picture in figure 1 for a transmission using Stop&Wait ARQ assuming that all transmissions will succeed.



Station A transmits 3 times 1000 bytes and once 500 bytes with sequence numbers alternating between 0 and 1. Without any interference, station B replies to every frame with an acknowledgement that includes the next expected sequence number.
The header should include a type field, for example one byte, to inform the receiver if the frame includes data or an acknowledgement.

II. What happens if the transmission of the 2nd frame experiences some interference? Visualise the progress of the transmission with a diagram.



Station A needs to start a timer with each frame that it transmits. If this timer runs out, station A will retransmit the current packet. The receiver, in this case station B, does not perform any action unless it has to send data of its own or receives a data frame from station A.

III. What information would you include in the header, what size would this header have and how much overhead would be introduced for the transfer of 3500 bytes.

The header above should include a field indicating the type of the frame e.g. a data frame or an acknowledgement. Assuming that the protocol should implement piggy-backing i.e. use one bit for the sequence number from the sender and one bit for an acknowledgement, the first six bits of the first byte of the frame could be used to indicate the type of the frame.
For Stop&Wait, this means that the header for each data frame will consist of one byte and each data frame will require one acknowledgement consisting of one byte. So, in order to transfer 3500 byte, this protocol would introduce 8 bytes of overhead.

2) Go-Back-N ARQ

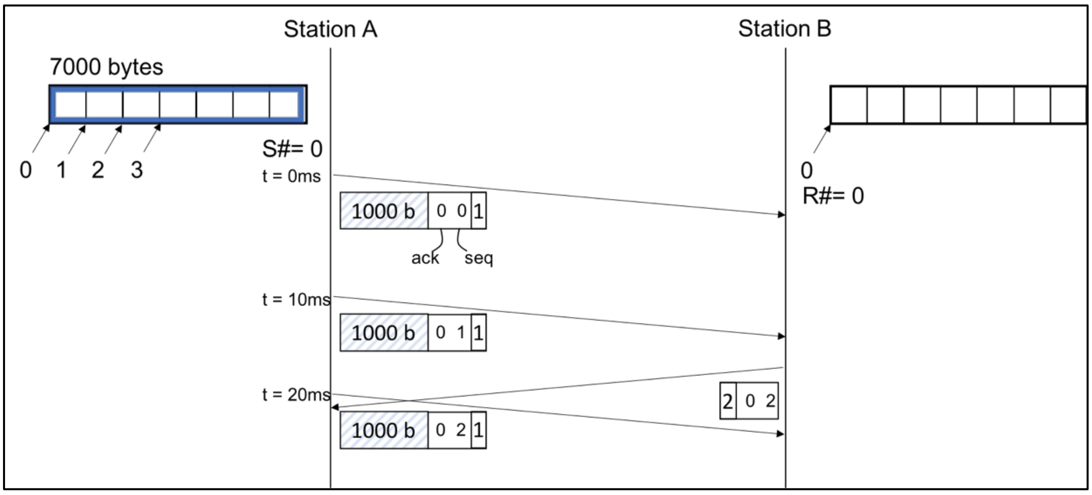
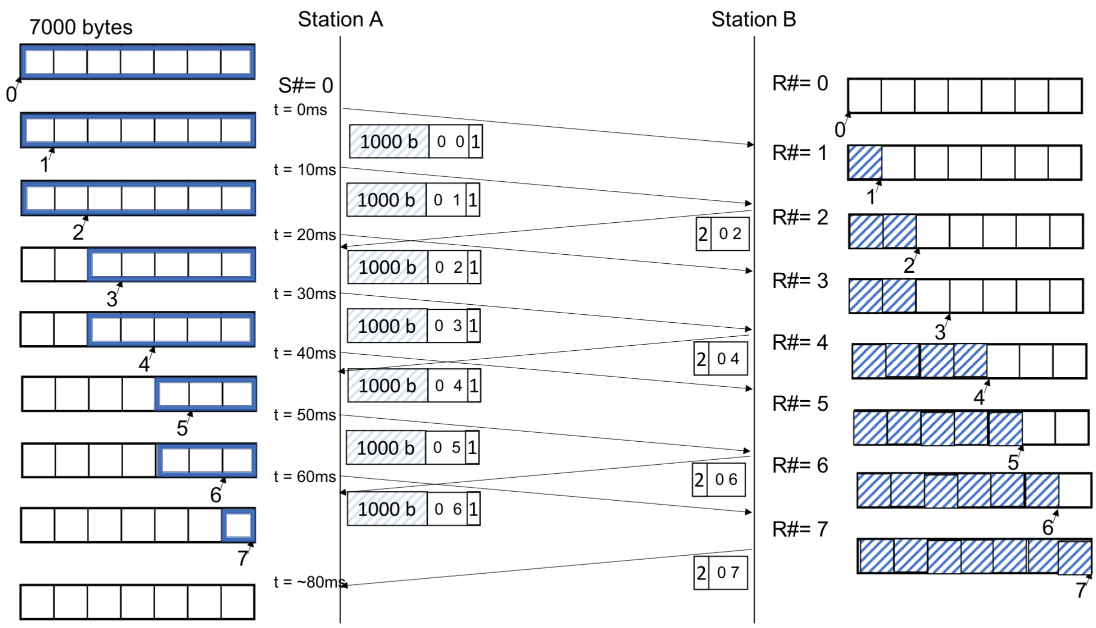


Figure 2: Start of a transmission of 7000 bytes using GoBackN ARQ. The wider frame inside the buffer of available data indicates the window of frames that can be transmitted.

Assume your header includes 1 byte to identify the protocol, 1 byte to identify the type of the frame, 3 bits for a sequence number of the frame and 3 bits to indicate the sequence number of the next expected frame i.e. the acknowledgement.

prot.	type	seq	ack
15	1 = data 2 = ack		
1 byte	1 byte	3 bit	3 bit

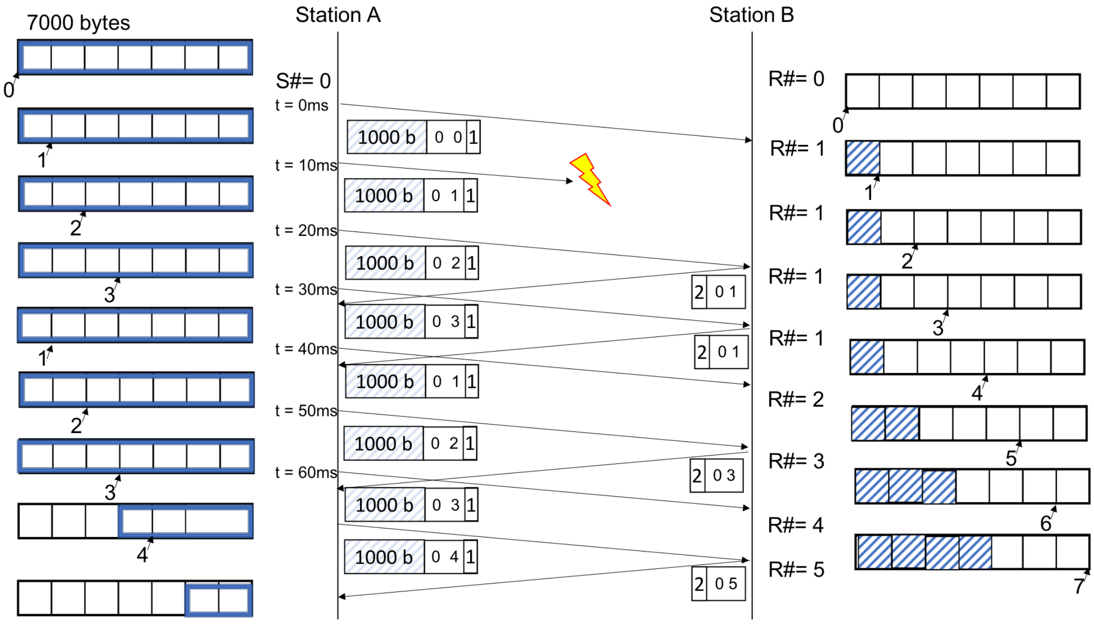
I. Complete the picture in figure 2 using GoBackN ARQ with the sender transmitting a frame every 10ms and the receiver acknowledging every 2nd frame, assuming that all transmissions will succeed.



Station A transmits 7 data frames, each with its own header and 1000 bytes payload. In the diagram the byte indicating the protocol type has been omitted. The bold frame in the diagram indicates the window of frames that station A is allowed to transmit. In this case, the window

is pushed forward whenever station A receives an acknowledgement. Station B responds to every 2nd incoming data frame with an acknowledgement. In its buffer, station B only indicates the frame that it currently expects.
Because the header has 3 bits to represent sequence numbers, the window at the sender side is 7 frames long.

II. What happens if the transmission of the 2nd frame experiences some interference? Visualise the progress of the transmission with a diagram.

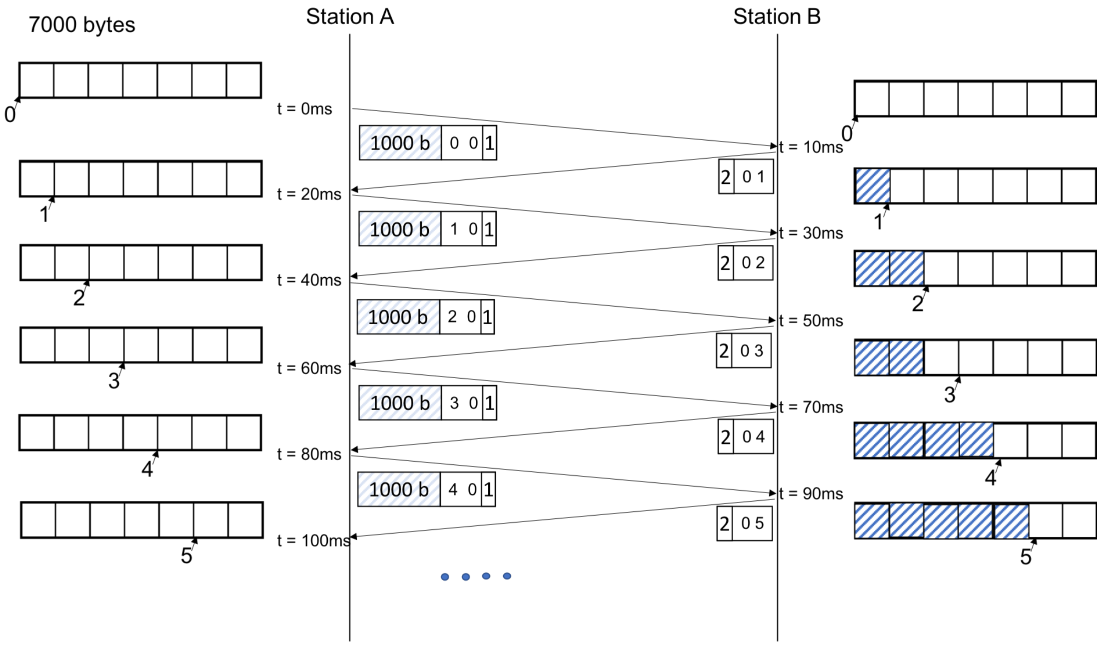


Station B would receive frame #2 without having received frame #1 and would send back an acknowledgement that it has received everything up to frame #1. Once station A receives the acknowledgement that station B expects frame #1, station A will reset its pointer to frame #1 and transmit frame #1,2, and 3 again. Station B will always only accept the next expected frame. Note: Acknowledgements do not increase the sequence number.

III. What information would you include in the header, what size would this header have and how much overhead would be introduced for the transfer of 7000 bytes.
If the header suggested above would contain a byte to indicate the protocol version, a byte to indicate the type of the frame and a byte to hold the sequence number and the number for the acknowledgement, the header would consist of 3 bytes and the acknowledgements would consist of 3 bytes. In the case, where not interference takes place and every 2nd frame is acknowledged, 7+4 frames will be exchanged, so the overhead is 11 times 3 bytes ie. 33 bytes in order to transfer 7000 bytes of payload.

If a source address and a destination address of 4 bytes, a byte to indicate options, a flag byte at the start and end of the frame and 4 bytes for a checksum would be included, the overhead would rise to 3 bytes +8 bytes addresses +1 byte options + 2 bytes flag bytes +4 bytes checksum = 18 bytes overhead per frame or 198 bytes for the exchange of 11 frames.
In the case, where frames are filled with the maximum payload i.e. maximum transmission unit (MTU) defined by the protocol, the overhead of the header should generally be neglectable. However, the overhead introduced through headers may become an issue when the payload in frames drops to a few bytes and the size of the header dominates the amount of data being exchanged.

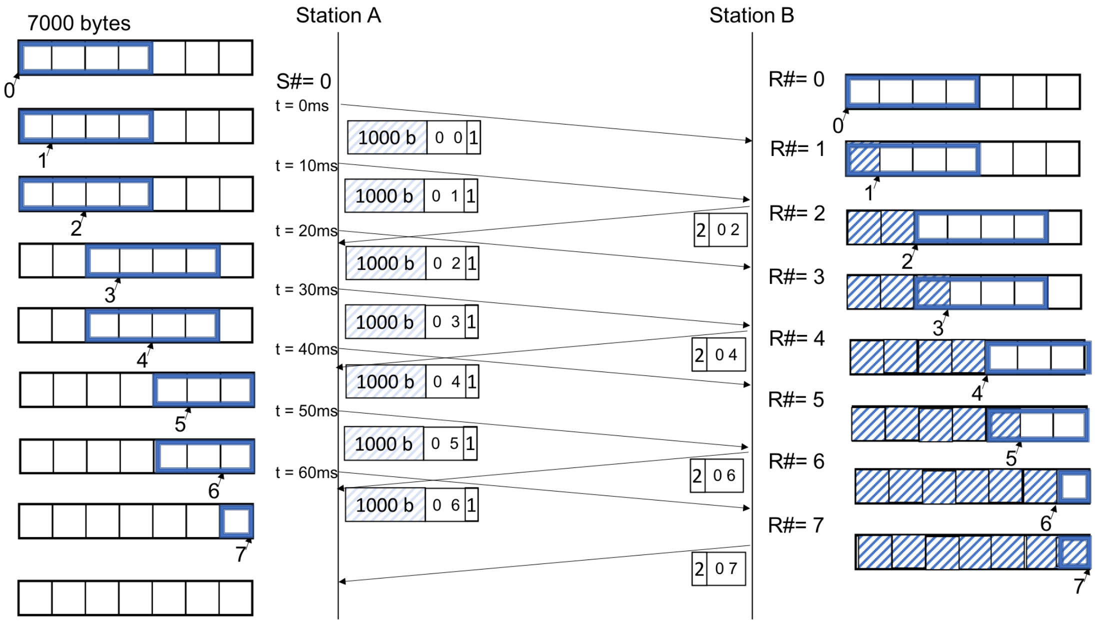
IV. If you would have used Stop&Wait ARQ, how long would have the transmission of 7000 bytes taken, given that both transmissions of data and acknowledgements would take 10ms?



A Stop&Wait approach would complete in around 140ms, whereas a GoBackN approach may complete in around 80ms.
Note: These times are only indicative and not realistic i.e. specific to a protocol. They are intended to give you an idea of the scale of timing involved in communication. A second may be considered an eternity for the processing of a few instructions on a processor and very long for the transmission of a number of bytes; however, in comparison to the processing of a few instruction on a processor, the transmission of a number of bytes is always considered very long.

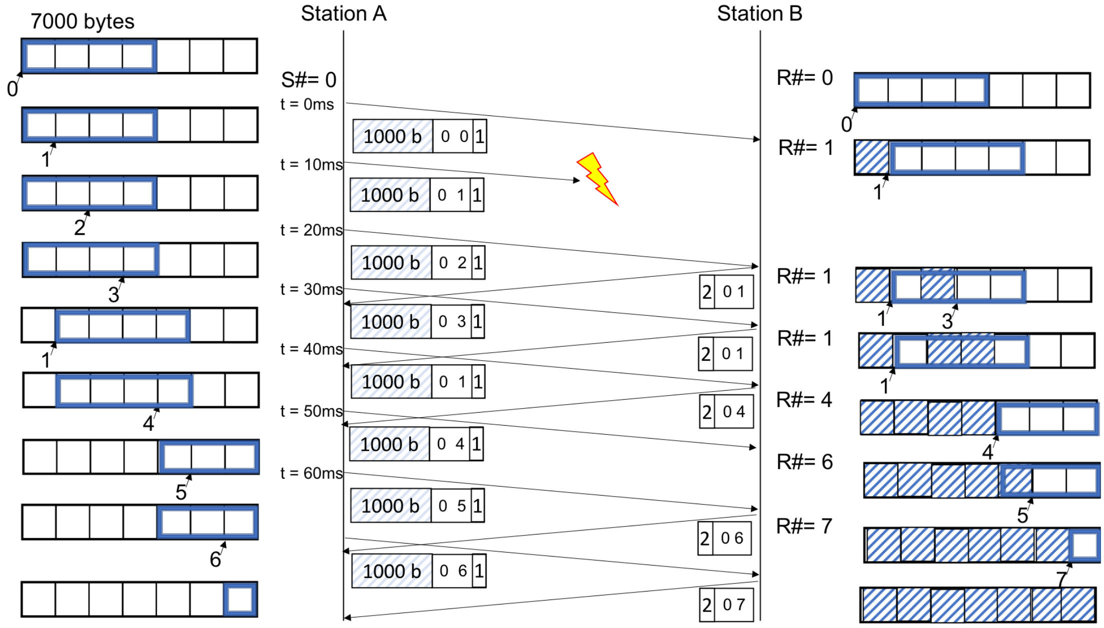
3) Selective Repeat ARQ

I. Complete the picture in figure 2 for a transmission using Selective Repeat ARQ instead of GoBackN ARQ assuming that all transmissions will succeed.



A case for Selective Repeat without any interference will look very similar to a case for GoBackN. The difference would be in the size of the window at the sender side and the introduction of a window at the receiver side. For Selective Repeat, the windows at both sides will be limited to 4 frames for a 3-bit sequence number i.e. the sender would be able to send 4 frames before it would expect a reply to the 1st frame from the receiver. The receiver would accept any of the 4 frames indicated in its window. Without any interference and the acknowledgement of every 2nd frame, the scenario would look very similar to the scenario in 2.I.

II. What happens if the transmission of the 2nd frame experiences some interference? Visualise the progress of the transmission with a diagram.



In the case of a lost frame in Selective Repeat, the receiver would reply with a negative acknowledgement whenever it receives a frame that does not correspond to the next frame it is expecting, it will reply with a negative acknowledgement, indicating the missing frame. The receiver will still accept any frame with a sequence number inside its window. Once it has received a retransmission of an outstanding frame, the receiver will push its window forward to the next expected frame and reply with an acknowledgement that indicates the next expected frame to the sender.

III. What information would you include in the header, what size would this header have and how much overhead would be introduced for the transfer of 7000 bytes?

Similar to 2.III, the header may include a byte to indicate the protocol version and a byte to indicate the type of the frame. It may be useful to increase the number of bits for the sequence number and the number for the acknowledgement to 4 bits each in order to allow for larger windows. The header would still consist of 3 bytes and the acknowledgements would still consist of 3 bytes.

IV. What is the maximum window size for a Go-Back-N ARQ protocol and a Selective Repeat ARQ protocol in this scenario?

The window size for GoBackN is limited to 2^m-1 where m indicates the number of bits used to encode the sequence number e.g. for a 3-bit sequence number, the window size would be 7 frames.
The window size for Selective Repeat is limited to 2^{m-1} where m indicates the number of bits used to encode the sequence number e.g. for a 3-bit sequence number, the window size would be 4 frames.

4) Bandwidth-Delay Product

a) Describe the term "Bandwidth-Delay Product" and calculate it for a connection that has a bandwidth of 2 Mbit/s, uses frames of the size of 1000 bit and a round-trip time of 50ms. What is the usage of the total bandwidth? What happens if the bandwidth is changed to 1 Gbit/s and the round-trip time to 10ms?

The bandwidth delay product would be $50 * 10^{-3}$ s * $2 * 10^{6}$ b/s = 100,000 bits i.e. a sender could transmit 100kbits before expecting a response from a receiver.
If this would be scaled upwards to 1 Gbit/s and the round-trip time reduced $10 * 10^{-3}$ s * $1 * 10^{9}$ b/s = 10,000,000 bits. Assuming a full Ethernet frame of 1536 bytes/ 12288 bits and a Stop&Wait approach, the sender would wait for a response while it could send another 9,987,712 bits

---

Sample Exam Question

(1c) Assume you have a connection between two stations that are limited in processing power and storage capacity. Suggest a flow control mechanism that would be suitable for this connection, explain the details of this mechanism and justify your choice by contrasting the mechanism against an alternative mechanism.