**Stop and Wait ARQ**

The Stop and Wait ARQ is derived from the Stop and Wait Protocol which relies on acknowledgements (ACKs) from the receiver to maintain a consistent flow of data during transmission. It adds error control mechanism to the Stop and Wait Protocol.

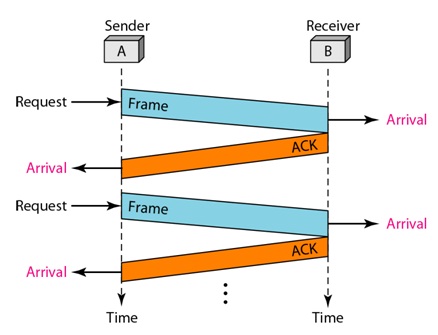


Figure 1: Stop and Wait Protocol

An error is said to occur during transmission of data frames when either the frames get lost or they get corrupted. When a receiver receives a corrupted frame, it silently discards it and sends no acknowledgements to the sender. Since the sender receives no ACKs in both the cases, it assumes that the frame was not received by the receiver and after a time out period, the sender re-sends the frames. The sender knows which frame to send because it keeps a copy of the last sent frame.

Additionally, the sender also uses sequence numbers to number each frame that it sends. This helps the receiver in identifying duplicate frames. The sequence number alternates between 0 and 1. The ACKs from the receiver contain the sequence number of the next frame that it expects from the sender. This is called the acknowledgement number. If the receiver has received frame with sequence number 0, it sends 1 as the ACK and vice versa.

Another case in which an error is said to occur is when the ACK from the receiver is lost. The sender will not know if the frame was lost or the ACK was lost. Hence the sender re-sends after the time out.

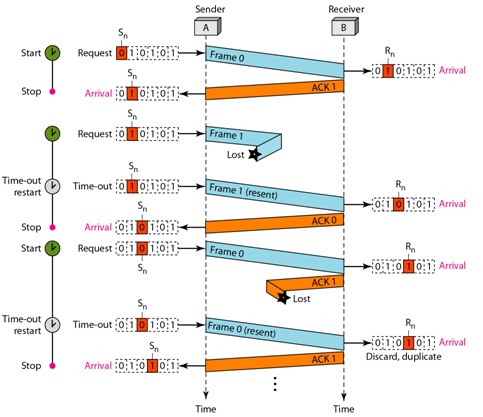


Figure 2: Stop and Wait ARQ

Figure 2 shows an example of Stop-and-Wait ARQ. Frame 0 is sent and acknowledged. Frame 1 is lost and resent after the time-out. The resent frame 1 is acknowledged and the timer stops. Frame 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.

**Efficiency of Stop and Wait ARQ**

Since the sender can send only one frame before receiving an ACK, there will be only frame at a time in the channel connecting the sender and the receiver. As a result, the capacity of the channel is not fully utilized. Also, the sender can send no other frame before it receives an ACK for the previous frame. Hence the Stop and Wait ARQ is not efficient.

**Go-Back N ARQ**

The Go-Back N ARQ overcomes the disadvantage of the Stop and Wait ARQ by allowing the sender to send more than one frame before receiving ACKs. This keeps the channel busy and increases efficiency.

The mechanism is similar to the Stop and Wait ARQ except this protocol allows more than one frame to be transmitted at a time. The sender keeps a copy of all the last sent frames until it receives ACKs for those. It also starts a timer

This protocol uses the concept of a sliding window which defines the number of frames that a sender can send at a time. Also, the frames need to assigned sequence numbers. If the header of a frame allows m bits to be used for the sequence number, the sequence numbers can range from 0 to 2m – 1. For example, if m = 2, the sequence numbers can range from 0 to 3. The sequence numbers repeat after the end of the range is reached. So, the sequence numbers are:-

0, 1, 2, 3, 0, 1, 2, 3, 0, ……..

The maximum size of the sliding window can be 2m – 1. Here, the maximum size can be 3. The following depicts an instance of the sliding window:-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ………. | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | ……… |

Frames already acknowledged Frames sent but Frames that cannot be sent

not acknowledged

1. Send window before sliding

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ………. | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | ……… |

1. Send window after sliding

Figure 3: Send window