Flow Control Policies of DLL

Overview

On the network, the sender sends information, so the receiver receives the information. But assuming that the sender can receive and process the data faster than the receiver, then the information will be diverted. Flow control methods will help ensure this. The flow control method will ensure that the sender only sends notifications that the receiver can receive and process. Flow control tells the sender what percentage of data needs to be sent to the receiver not to be lost. This mechanism allows the sender to wait for confirmation before sending subsequent data.
An ideal network should be prepared to ensure that knowledge is transferred to the target host without errors in the DLL(Data Link Layer). The system must ensure that the information received is the same as the information transmitted for many applications. Many factors will eventually change the information and can corrupt the transmitted bits. Data is often destroyed during transmission. Some applications often require error detection and correction. Some applications can tolerate a small number of errors. For example, random audio or video transmission errors can also be accepted, but in text messages, equivalent errors are unacceptable.

Stop and Wait Protocol

In this protocol, the sender will send one frame to the receiver at a time. The sender will stop and wait for confirmation from the receiver. This time (the time between sending the message and receiving the confirmation) is the sender's timeout, so the sender is completely inactive during this period. When the sender acknowledges receipt (ACK), it will send the following packets to the receiver and wait for the confirmation again; this process will continue because the sender has information to send. The following figure will help understand this:

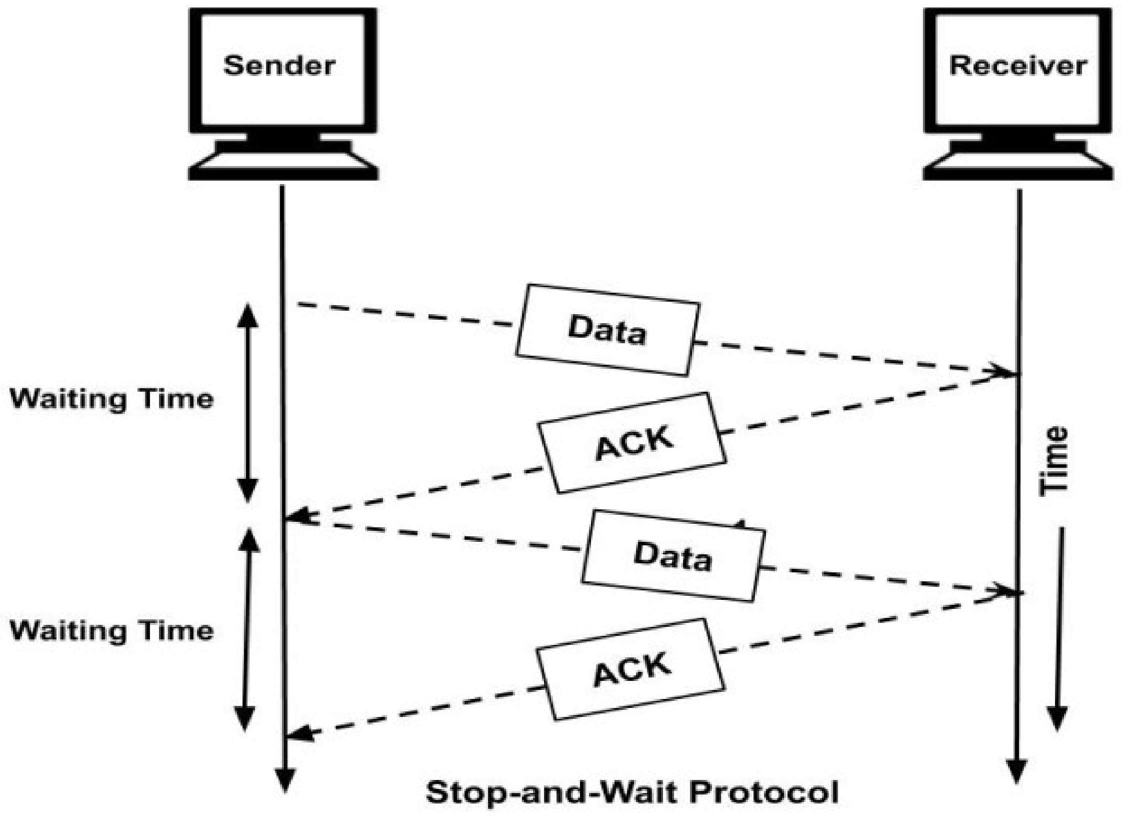

Figure 1: Stop and Wait Protocol

Suppose that any frame sent is not received by the receiver and is lost. Therefore, the receiver will not send an acknowledgment because it has not received any frames. In addition, the sender will not send subsequent frames because it will wait for an acknowledgment of an earlier frame that it has shipped. Therefore, there is often a stalemate here. To avoid any such situation, there is a timeout timer. The sender will wait for this fixed time to receive the confirmation, and if it does not receive the ack, it will resend the frame. It allows us to assume that the channel is noisy and can introduce errors in the data transmitted through it. Channel noise can damage the frames, or they can drift completely. Let's understand this through simple steps:

- The sender first transmits a frame, then stops the transmission and waits for confirmation from the receiver.
- If a positive acknowledgment is received (meaning the receiver is ready to receive subsequent frames), go to step 1 above.
- If a negative acknowledgment (NACK) is received (the receiver cannot receive subsequent frames), it waits for a positive acknowledgment (ACK) from the receiver.

Go Back N ARQ

Go back was introduced because the "stop and wait protocol" is not efficient because the sender can only send one data packet at a time and must wait for the next frame to be transmitted until the previous frame is recognized. In this way, the stop-and-wait protocol wastes channel bandwidth and even increases the trigger delay. This is usually because the stop and wait protocol does not use the concept of pipes. A pipeline can be a general concept in which post-task processing starts before the end of the previous task. In the network, the idea of pipes specifies that the source can send several frames before acknowledging the main transmission frame. The sliding window is an imaginary box in the transmitter and receiver. This window also keeps the frame in transmission as the receiving endpoint and provides an upper limit on the number of frames transmitted before an acknowledgment is obtained. The range that the sender pays attention to is called the sending window, and the range that the receiver gives priority to is called the receiving window.

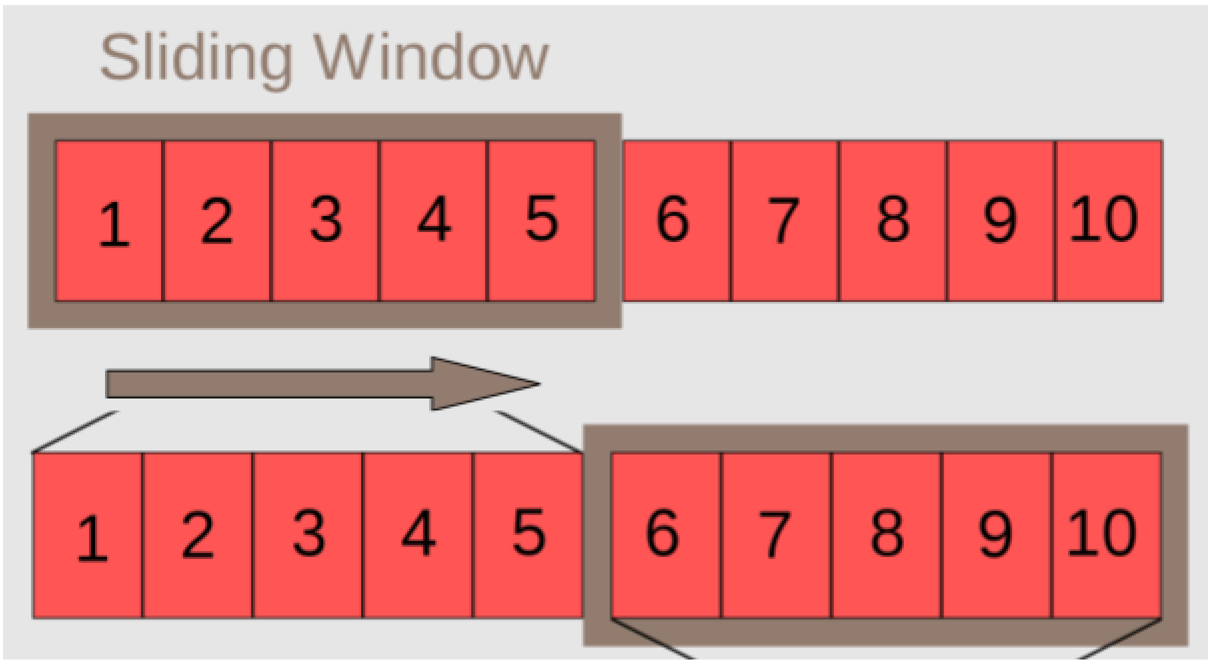

Figure 2: Sliding Window

GoBackN ARQ, the sender's dimension is N, so the receiver's window size is usually 1.

- The use of cumulative acknowledgments in this protocol means that the receiver maintains an identification timer; whenever the receiver receives a replacement frame from the sender, it starts the replacement identification timer. When the timer expires, the receiver sends a cumulative acknowledgment of all structures that the receiver did not recognize at that time.

- It should be noted that the new confirmation timer will only start after receiving the replacement frame, and the old confirmation timer will not start after the timeout.
- If the receiver receives a corrupted frame, it silently discards it, so the sender will retransmit the correct frame after the timeout timer expires. Therefore, the receiver silently discards corrupted frames. By silently discarding, we mean: "Only reject the frame, and take no action on the frame."
- If the confirmation timer expires, it is assumed that there is only one frame left for confirmation. In this case, the receiver sends a separate acknowledgment of the frame.
- If the receiver receives messy frames, it just discards all frames.
- If the sender does not receive any confirmation, the entire frame window will be retransmitted.
- Use of GoBackN ARQ protocol results in retransmission of lost frames

Selective Repeat ARQ

The function of the selective repetition protocol is that it allows the receiver to accept and buffer the packets following the damaged or missing packages and then only retransmit the boxes that are lost or damaged in the network channel during transmission. This protocol is similar to the GoBackN ARQ protocol or an improved version of GBN ARQ. The difference is that a buffer is used here, and both the receiver and the sender maintain a size window. This selective repetition sometimes works best when network links are generally unreliable. Because here, in this case, retransmission tends to occur more frequently. Selective frame retransmission is much more efficient than full-frame retransmission. Selective retransmission also requires a full-duplex channel to transmit packets uniformly. And feedback/confirmation.

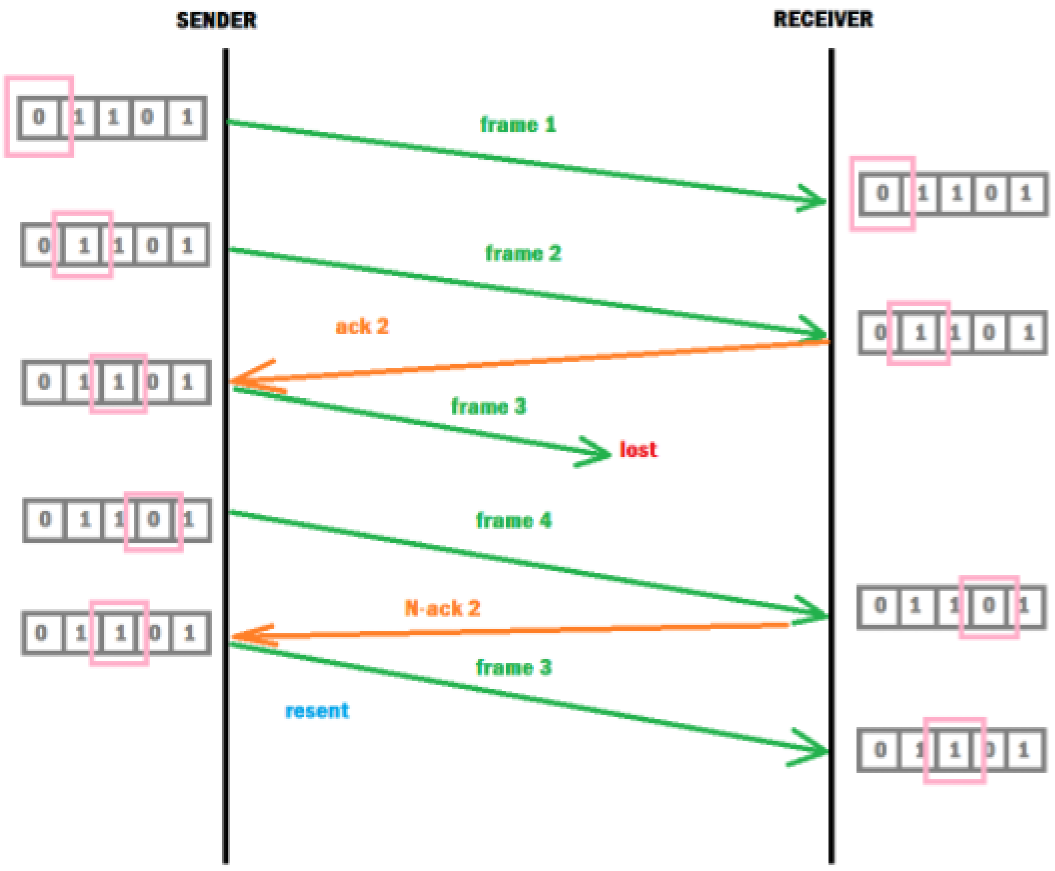

Figure 3: Selective Repeat

Now, in the above figure, the sender first sends frame 1 (i.e., 0), the receiver immediately recognizes the frame, and then sends frame 2, which is also recognized. We will see that frame3 is lost during transmission and cannot succeed at the receiver, but the next frame, i.e., frame4, is transmitted and accepted by the receiver. Then the receiver sends a negative confirmation, thanks to the sender for only transmit and send frame3 again Instead of retransmitting all frames starting from frame3 (this happens in GoBackN ARQ). The efficiency of the selective repeat protocol is the same as that of the GoBackN ARQ protocol.

efficiency = N / (1 + 2a)

where a = propagation delay / transmission delay

buffer = N + N

serial number = N (sender) + N (receiver)