1 Flow and Error Control

To ensure reliable communication, there needs to exist flow control (managing the amount of data the sender sends), and error control (that data arrives at the destination error free).

Flow and error control needs to be done at several layers. For node-to-node links, flow and error control is carried out in the data-link layer. For end-point to end-point, flow and error control is carried out in the transport layer.

2 Flow Control

Flow control tells the sender how much data to send. It makes the sender wait for some sort of an acknowledgment (ACK) before continuing to send more data.

There are two primary methods of flow control: Stop-and-wait, and Sliding Window.

2.1 Stop And Wait

Stop And Wait is a simple scheme, where the sender has to wait for an acknowledgment of every frame that it sends. It sends a frame, waits for acknowledgment, then it sends another frame, and again, waits for acknowledgment.

The trouble with this scheme is that it's very slow. For every frame that is sent, there needs to be an acknowledgment, which takes a similar amount of propagation time to get back to the sender.

The advantage is simplicity.

2.2 Sliding Window

The whole idea behind Sliding Window is not to wait for an acknowledgment for individual frames, but to send a few frames (and then get an acknowledgment that acknowledges several frames at the same time).

It works by having the sender and receiver have a "window" of frames. The sender can send as many frames as would fit into a window. The receiver, upon receiving enough frames, will respond with an acknowledgment of all frames up to a certain point in the window. The window it then said to "slide", and the whole thing starts again (the sender sends more frames, the receiver gets more frames, sends an acknowledgment of those frames, etc.)

Each frame has to be numbered in relation to the sliding window. For a window of size N, frames get a number from 0 to $N - 1$. Subsequent frames get a number *mod* N.

3 Error Control

Error control involves retransmission of the lost, damaged, or corrupted frame. The scheme is called ARQ, for Automatic Repeat Request.

The general scheme works this way: the sender sends the data. If data arrives without any problems, the receiver sends out an ACK message (acknowledgment). If the data has a problem (corrupt), the receiver sends out an NAK (negative acknowledgment). Upon getting a NAK message, the sender retransmits. There is also a timer; which allows for data retransmission if the original message or ACK or NAK got lost.

There are several forms: Stop-and-Wait ARQ, and Sliding Window ARQ.

3.1 Stop-and-Wait ARQ

Each frame has an alternating bit (0 or 1). The sender sends out the message, and starts the timeout counter.

If the receiver gets the message without any problems, it sounds out an ACK message, otherwise it sends out a NAK message.

If the sender gets an NAK message, it retransmits the sent message.

If the timer expires (and no NAK nor ACK arrived) then the sender retransmits the frame. If it happened that the message was received fine, but the ACK got lost, then the even/odd bit in the data frame will alert the receiver to the duplicate frame.

3.2 Sliding Window ARQ

There are two categories of Sliding Window ARQ: Go-back-n, and Selective reject.

Both of these schemes keep a copy of all the sent frames until they have been acknowledged. The receiver has the option of responding with a ACK (acknowledgment) or NAK (negative acknowledgment). Each frame needs to be numbered. The ACK indicates the next expected frame number, while a NAK identifies the broken frame that needs to be resent. There is also a timer, in case the sender doesn't hear an ACK or NAK, the sender will resend the data from the last time it got an ACK or a NAK.

3.2.1 Go-back-n

In the Go-back-n scheme, upon an error, the sender retransmits all the frames that came after the error. For example, sender may send frames 1,2,3,4 and get an NAK with a value of 2. The NAK acknowledges everything that came before it, and asks for frame 2 (and subsequent frames) to be resent.

3.2.2 Selective reject

Selective reject allows the sender to reject any particular frame, without having the sender resend all the following frames. For example, the sender may send frames 1,2,3,4 and the receiver may send a NAK with value of 2. At which point, the sender retransmits frame 2. If frames 3,4 were received without any problems, then the receiver acknowledges them with a ACK 4 (note that ACK now points to the last successful frame, as opposed to the next expected frame).

In order to support Selective reject, the sender needs to maintain all data that hasn't been aknowledged yet. The receiver also needs to have sorting (rearranging) logic to be able to re-order frames that may have arrived out of sequence (or some frames that were damaged and retransmitted out of sequence). The sender needs to be able to find any particular frame for retransmission.