ARQ and Flow Control

Automatic Repeat reQuest (ARQ) is a technique to deal with errored or lost frames by retransmitting them. Flow control is the term for various techniques to stop or slow down transmissions to match the rate at which a receiver can accept data. After this lecture you should be able to: explain how ACK frames ensure error-free transmissions; select an appropriate type of ARQ (from stop-and-wait, go-back-N and selective repeat) based on channel error rate and delay; draw clock and data waveforms for a synchronous interface and identify the setup and hold times; and select appropriate flow-control method(s) to avoid under- and over-flow on synchronous and asynchronous interfaces and on frame-oriented networks.

Retransmission Protocols

To ensure that no data is lost, most frame-oriented data communication protocols require that the receiver acknowledge correct reception of each frame.

When a message is received without errors the receiver sends a special packet back to the sender that acknowledges correct reception. This is called an ACK frame.

If there can be multiple outstanding (unacknowledged) frames, each frame and the corresponding ACK will contain a unique frame identifier (typically a sequence number).

If the transmitter does not receive the appropriate ACK frame within a certain amount of time (the timeout) it retransmits the frame.

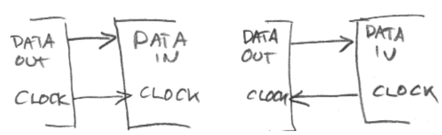
There are many variations on this "Stop and Wait" ARQ that try to make it more efficient:

- the transmitter does not wait for an ACK after each frame. This requires that the transmitter retain all unacknowledged frames so they can be retransmitted if necessary.
- in the simplest case, if a frame is lost all of the frames starting with the first lost frame are retransmitted ("go back N ARQ").
- the receiver can selectively ACK specific frames rather than requiring retransmission of all frames since the last acknowledged frame (called "selective repeat ARQ").
- the receiver can acknowledge multiple frames at the same time. This cuts down on the number of ACK frames required.
- another way to reduce the ACK overhead is to "piggyback" the ACK information onto data frames being sent in the reverse direction.
- the receiver can transmit a negative acknowledgment (NACK) if a frame is received in error (assuming the source and/or packet sequence number are known) rather than waiting for the transmitter to time out

Exercise 1: A data communication system operates at 1 Mb/s and uses 10000-bit data frames and 100-bit ACK frames. What are the frame durations? What is the throughput if there is no channel delay and no errors? If the round-trip channel delay is a 0.5s (typical for satellite links)? If go-back-N ARQ is used, assuming N is larger than the number of frames transmitted in 0.5 seconds?

Synchronous vs Asynchronous Interfaces

Synchronous interfaces include both data and clock signals. A symbol (typically one bit but possibly multiple bits in parallel) is transferred over the interface on each clock edge. The clock can be supplied by either the source or the sink as shown below:



A synchronous interface does not require that the source and sink agree on the baud rate since the clock controls the transfer of data. In fact, in many cases the clock period can vary from bit to bit.

Exercise 2: What are the definitions of clock setup and hold times?

Flow Control

It is possible for the data to overflow the buffers available in the sink.

It is also possible, for a synchronous interface that uses a clock supplied by the sink, to have a data "underflow" when no data is available at the source.

To avoid underflows and overflows we can use "flow control" signals. For example, a "clear-to-send" signal output by the sink or a "data ready" signal output by the source.

Another flow control method is to use acknowledgment frames. The sink can output an ACK for the previous frame only when there is room for another frame.

Another flow control method is to use insert escape sequences that have no effect or null frames in the data stream if there is no data to send.

Exercise 3: Which of the above flow control methods can be used with asynchronous interfaces? With source-clocked synchronous interfaces? With sink-clocked synchronous interfaces?