ARQ and Channel Codes

Retransmission Protocols

To ensure that no data is lost, most 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 the transmitter does not receive the ACK frame within a certain amount of time (the timeout) it retransmits the frame.

There are a number of ways that this process can be made more efficient:

- the transmitter does not have to wait for an ACK after each frame. This requires that each frame have a sequence number and the transmitter must retain unacknowledged frames so they can be retransmitted if necessary ("go back N ARQ").
- the receiver can acknowledge multiple frames at the same time.
- the receiver can selectively ACK specific frames rather than requiring retransmission of all frames since the last acknowleged frame (selective ACK).
- the ACKs can be "piggybacked" onto data frames being sent in the reverse direction.
- the receiver can transmit a negative acknowledgement (NACK) if a frame is received in error (assuming the source is known) rather than the transmitter having to wait for a timeout

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?

Error Detection

In order to detect errors in received frames the transmitter computes additional bits called "parity bits" and adds them to the end of the frame. The receiver computes the parity bits in the same way using the received data and compares them to the received parity bits. If the transmitted and received parity bits match then either there were no errors or the errors were such that the errors resulted in the same parity. The probability of the latter event is called the undetected error probability. Good error detecting codes try to make this probability as low as possible.

Checksums

A simple way to check for errors in a frame of data is to compute the sum of the byte (or word) values in a frame of data. The sum is computed modulo the word size of the check sum. The additive complement of the sum is then appended to the packet so that an error-free packet has a zero checksum. This is the type of error-detection used by TCP/IP frames used on the Internet.

Checksums are typically used by higher-level protocols since they are easy to compute in software.

Parity Bits

The simplest type of parity is a single parity bit. Typically the parity bit is computed as the modulo-2 sum of all of the bits in the message.

Exercise 2: What is a modulo-2 sum? What is the modulo-2 sum of 1, 0 and 1? What is the modulo-2 sum if the number of 1's is an even number?

The most common example of a single parity bit is a parity bit added to each ASCII character. Most serial interfaces can be configured to compute and append a parity bit to each ASCII character. This bit can be either the sum of all of the bits ("even parity") or it's complement ("odd parity").

Block Codes

More complex channel codes use multiple parity bits. A block code is used in which each block of n bits contains k data bits and $n-k$ parity bits. This is called an (n,k) code.

Exercise 3: How many different code words (different blocks) does an (n,k) code have? How many different patterns of $n-k$ parity bits are there?

The *Hamming Distance* is the number of bits that differ between two code words. The performance of a code is generally determined by the minimum (Hamming) distance between code words.

Exercise 4: What is the Hamming distance between the codewords 11100 and 11011?

The *rate* of a code is the ratio of information bits to total bits, or k/n. This is a measure of the efficiency of the code. Typically there is a tradeoff between the error-correcting ability (minimum distance) and the rate of a code.

FEC Coding and Minimum Distance Decoding

It is possible to correct errors with certain block codes. These types of codes are called Forward Error Correcting (FEC) codes.

Conceptually, a receiver can correct errors by choosing the valid codeword that has the smallest Hamming distance from the received codeword. This is because codewords with fewer errors are more likely to happen than those with more errors.

Exercise 5: A block code has two valid codewords, 101 and 010. The receiver receives the codeword 110. What is the Hamming distance between the received codeword and each of the valid codewords? What codeword should the received decide was sent?

Howver, with large codewords it is not possible to do a simple exhaustive search through all possible codewords to find the one with the minimum distance. There is a large field of study devoted to the design of codes which can be efficiently decoded. We will consider some of these codes in the next lecture.