2: Error Detection and Correction

Thursday 26th of October, 2017

Intro

How can a receiver know that a packet has been corrupted in transit?

You can send redundant bits – for example, send two copies of all your data, and compare them to check if they're the same on the receiving side. It's unlikely that errors would happen in the same place in both packets.

However, now you're doubling all your data, so effectively halving your throughput.

Parity

Make sure every packet has an even number of ones (use an extra bit at the end, set it to 1 if you have an odd number of ones in the packet – now you have an even number). The receiver can count the number of 1s when receiving it, to see if it matches up.

Using one bit like this is known as single-bit parity. The problem is, what if two 1s are flipped to 0s? Then there are still an even number of bits – single-bit parity can only detect single-bit errors.

Two-Dimensional Bit Parity

Arrange the bits into a table, and do a parity bit for every row and column, and for the resulting parity row/column pair.

One bit per packet chunk, and a parity chunk at the end.

This isn't used in practice, because if you do have errors, they're likely to affect a burst of more than 4 or 5 bits, which is the limit for this method of error detection.

What We Want

We have to calculate this at every hop and for every packet, so we want our error correction method to be powerful but not take long to calculate.

Cyclic Redundancy Check (CRC)

Simple to implement in hardware (shift registers and XOR gates), so it's cheap and fast. It's also very good at detecting errors.

Treat bit-strings (packets) as polynomials with coefficients of 1 and 0 only – every bit is a term in the polynomial. 10101001 -> x^7 + x^5 + x^3 + x^0.

What is the extra bit I need to add to a message to make it divisible by a certain number?

Sender and receiver agree ahead of time on a generator polynomial G(x). Take the original message polynomial D(x), and convert it to a new polynomial P(x), which should be divisible exactly by G(x).

The sender transmits P(x), and the receiver receives P(x) + E(x), where E(x) represents corrupted bits. If E(x) is 0, we have no errors. (Or, E(x) was such that P(x) + E(x) was still divisible by G(x))

[...]

Forward Error Correction (FEC)

Variants of CRC that allow the receiver to correct the errors as well as detect them.

In cellular networks, adaptive FEC can be used, where the number of redundancy bits used is increased if your signal is poor (and errors are more likely).