5 Error-Detecting Codes

Part (1) Scheme with extra k distinct packets/symbols

Below, I'll first present a brute force explanation of a scheme of sending an extra k packets works (assume a maximum of k errors). This means that Alice would be sending out values corresponding to $x = 1, 2, ..., n, n+1, ..., n+k$ where the first n packets represent her encoded message.

First, if any erasure error occurs, then Bob would easily detect it and throw away the message, so I'll only discuss the case where a maximum of k general errors occur.

Let's think of Bob as a person who loves doing Lagrange interpolation, so he picks out all the combinations of the $n+k$ packets he received from Alice, and uses Lagrange interpolation to calculate all the corresponding polynomials. Since there are a maximum of k errors, so there are at least n non-corrupted packets/symbols, and let's call one set of n non-corrupted symbols as S.

By assumption, Bob must have picked the set of n symbols, S, at some point during his Lagranging process, which would allow him to deduce a unique polynomial $P(x)$, and by the proerty of polynomials, this is the original polynomial that Alice used to extend her message. Now, if there are any corrupted packets, let one such error be at $x = e$. Since Alice used $P(x)$ as her encoding polynomimal, so the original symbol should be $P(e)$, and let the corrupted symbol be r_e such that $P(e) \neq r_e$.

Now, consider the set of n symbols, S', which consists of the first $n-1$ elements of S and the symbol corresponding to $x = e$, which is r_e. Again, by assumption, Bob must have picked the set of n symbols, S', at some point during his Lagranging process, from which he would deduce a unique polynomial $P'(x)$. Here, we have that $P'(e) = r_e \neq P(e)$. Thus, $P(x)$ must be different from $P'(x)$ by the property of polynomials, which implies that if any error exists, Bob would deduce at least two different polynomials (under $GF(p)$) at some time during his Lagrange interpolation process. In this situation, he can infer that the transmitted code contains at least one error, and throw away the message.

On the other hand, if no errors exist in the transmitted message, then every set of n symbols Bob picked would all let him deduce the same polynomial, which is the original $P(x)$ Alice used.

Thus, this scheme of adding k extra, distinct packets allows Bob to detect if the transmitted code contains at least one error.

(Proof of minimality on the next page)

Part (2) Minimality of k extra packets

We proceed to show with a counterexample that adding any lesser number of symbols is not good enough. Consider $k = 2, n = 2$. By my scheme, Alice need to extend her message by at least 2 packets. Now, suppose Alice only sends 1 extra packet.

Consider this situation: suppose Alice wants to send a simple message "bc", which corresponds to the 2 characters "1", "2" over a modem where $a = 0, b = 1, c = 2, d = 3, e = 4$.

Thus, Alice wishes to transmit $m_1 = 1, m_2 = 2$, and we can decide that the unique polynomials of degree $n - 1 = 1$ is $P(x) = x$ by simple observation (or via Lagrange interpolation). By assumption, Alice only sends one extra packet, which is $P(3) = 3$, so the encoded message sent by Alice is $c_1 = 1, c_2 = 2, c_3 = 3$.

Now, since the maximum number of errors is $k = 2$, suppose we have all 2 errors such that the last two symbols of the message got corrupted, and we have $r_1 = 1, r_2 = 3, r_3 = 5$. Then Bob, not knowing how many errors were present, would naturally deduce $P'(x) = 2x - 1$ without noticing any errors, since this is a correct degree 1 polynomial corresponding to the corrupted message Bob received. Thus, he would decode Alice's message as $m'_1 = 1, m'_2 = 3$, which would translate into "bd", which is different from the original message.

Thus, this is a counterexample where adding any lesser number of symbols than proposed my scheme would not necessarily help Bob detect an error.