Imperial College London

Notes on interleaving of codewords

IEEE P802.3dj 200 Gb/s, 400 Gb/s, 800 Gb/s, and 1.6 Tb/s Ethernet Task Force

Omer S. Sella

Imperial College London

Background

- Reed-Solomon code introduced in 802.3bj uses 10 bit symbols and a generator polynomial.
- There have been a few presentations on interleaving, but the context for this presentation only requires
  - ran_3dj_01a_230206
  - ran_3df_02a_2211
- i.e., interleaving that respects the code structure, which is 10 bits for RS FEC as introduced by 802.3bj

Imperial College London

Assumptions

- Multiple codewords that belong to the same code, i.e., generated by a common polynomial $\mathrm{g}(X)$
  - $c_0(X) \quad = \sum_{k=0}^{n} c_0^k X^k = g(X) \times a_0(X)$

  ...
  - $c_{S-1}(X) = \sum_{k=0}^{n} c_{S-1}^k X^k = g(X) \times a_{S-1}(X)$
- Interleaving of one symbol from each codeword at a time, so:
- $c_0^0, c_1^0, \dots c_{S-1}^0, c_0^1, c_1^1, \dots c_{S-1}^1, \dots$
  - Lower script refers to message / codeword index, and upper script refers to index within a message / data

Imperial College London

Spacing:

- For a polynomial with coefficients $f^0, \dots f^n$ (upper script index, not power), i.e.:
- $f(X) = f^0 + f^1 X + f^2 X^2 + \dots f^n X^n = \sum_{k=0}^{n} f^k X^k$
- Inserting $S - 1$ zero coefficients between existing coefficients, i.e.:
- $f^0 + 0 \cdot X + \dots + 0 \cdot X^{S-1} + f^1 X^S + 0 \cdot X^{S+1} \dots$
- Is obtained through a change of variable $X \rightarrow X^S$, i.e.:
- $f^0 + 0 \cdot X + \dots + 0 \cdot X^{S-1} + f^1 X^S + 0 \cdot X^{S+1} \dots = f(X^S)$
  - Replacing X with X^S

Lifting:

- For a polynomial with coefficients $f^0, \ldots f^n$ (upper script index, not power), i.e.:
- $f(X) = f^0 + f^1 X + f^2 X^2 + \ \ldots f^n X^n = \ \sum_{k=0}^{n} f^k X^k$
- Lifting by T i.e.:
- $f^0 \cdot X^T + \ f^1 X^{T+1} + \cdots$
- Is obtained by multiplying $f(X)$ by X^T, i.e.:
- $X^T \cdot f(X)$

Imperial College London

Symbol-wise interleaving in polynomial notation:

- Start with $0 \leq i < S$ codewords:
- $c_i(X) = c_i^0 + c_i^1 X + c_i^2 X^2 + \ \dots c_i^n X^n c_0(X^{n-1}) = \ \sum_{k=0}^{n} c_0^k X^{(n-1)^k}$
- Introduce spacing between symbols, by insertion of $(s - 1)$ zeros:
- $c_i(X) \to c_i(X^s) = c_i^0 + 0 \cdot X + \ \dots + 0 \cdot X^{s-1} + \ c_i^1 \cdot X^s$
  - *(same change of variable for all)*
- Now lift codeword i for $0 \leq i < S$ by X^i to obtain:
- $X^i \cdot c_i(X^S) = X^i \cdot c_i^0 + X^i \cdot 0 \cdot X + \ \dots + X^i \cdot 0 \cdot X^{s-1} + X^i \cdot c_i^1 \cdot X^s$
- Add:
- $\sum_{i=0}^{S-1} X^i c_i(X^S)$ ← this is the interleaving result of the codewords.

Imperial College London

Observation:

- If $c_i(X) = g(X) \cdot a_i(X)$ is a codeword in the code generated by $g(X)$, then:
- $c_i(X^S)$ is a codeword in the code generated by $g(X^S)$
  - $c_i(X\ ) = g(X) \cdot a_i(X)$
  - $c_i(X^S) = g(X^S) \cdot a_i(X^S)$
- And so is the lift by X^i, i.e.:
  - $X^i \cdot c_i(X^S) = X^i \cdot g(X^S) \cdot a_i(X^S) = g(X^S) \cdot (X^i \cdot a_i(X^S))$
- And therefore, so is the sum$\sum_{i=0}^{S-1} X^i c_i(X^S)$ ← this is the interleaving result of the codewords.

Imperial College London

Conclusions:

- Interleaving of S codewords as presented (symbol-wise) can be achieved by encoding using the polynomial $g(X^S)$ instead of $g(X)$.
- If the underlying code has message length k and codeword length n, then:
  - the resulting code is of message length $S \cdot k$ and codeword length $S \cdot n$

- We only used the fact that the base code was generated by $g(X)$, so the same could be done for any other code which is generated by a polynomial (BCH for example).

Imperial College London

- If the underlying code can fix error bursts of length b then then:
  - The resulting code can fix error bursts of length $S \cdot b$

- Was already proven differently in:
  - ran_3dj_01a_230206
  - ran_3df_02a_2211

(claim, when it comes to data) the following architectures are equivalent: