Chapter 7

Burst-Error Correction

1. Introduction

- In burst error channels, errors occur in clusters.

- An error pattern,

  \[
  \bar{e} = (e_0, e_1, e_2, \ldots, e_{n-1}),
  \]

  is said to be a burst of length \( l \) if its nonzero components are confined to \( l \) consecutive positions, say \( e_j, e_{j+1}, \ldots, e_{j+l-1} \), the first and the last of which are nonzero, i.e., \( e_j = e_{j+l-1} = 1 \).

- For examples, the error pattern,

  \[
  \bar{e} = (0000101100100000),
  \]

  is a burst of length 7.
A linear code which is capable of correcting all error bursts of length $l$ or less but not all error bursts of length $l+1$ is called an $l$-burst-error-correcting code. The code is said to have burst-error-correcting capability $l$.

For an $l$-burst-error-correcting code, all the error burst of length $l$ or less can be used as coset leaders of a standard array.

**Reiger Bound**: The burst-error-correcting capability $l$ of an $(n,k)$ code is at most $\left\lfloor (n-k)/2 \right\rfloor$, i.e.,

$$l \leq \left\lfloor (n-k)/2 \right\rfloor$$

Codes meet the Reiger bound are called **optimal** codes.
For a cyclic burst-error-correcting code, it can correct bursts with one part at one end and one part at the other end as shown in Figure 7-1. These bursts are called **end-around bursts**.

**Figure 7-1.** An end-around burst
2. Known Codes and Coding Techniques for Correcting Bursts

- Fire codes
- Binary RS codes
- Interleaving technique
- Product codes
- Concatenation
- Cascading
3. Fire Codes

- They are cyclic codes and were discovered by P. Fire in 1959.

- Let $\overline{p}(X)$ be a binary irreducible polynomial of degree $m$. Let $\rho$ be the smallest integer such that $\overline{p}(X)$ divides $X+1$. The integer $\rho$ is called the period of $\overline{p}(X)$.

- Let $l \leq m$ such that $2l-1$ is not divisible by $\rho$.

- Let $n=\text{LCM}(2l-1, \rho)$.

- Define the following polynomial:

$$\overline{g}(X) = (X^{2l-1} + 1) \cdot \overline{p}(X)$$

- Then $\overline{g}(X)$ is a factor of $X^n + 1$, and has degree $2l-1+m$. 
The cyclic code generated by
\[ \overline{g}(X) = (X^{2l-1} + 1) \cdot \overline{p}(X) \]
is a **Fire** code which is capable of correcting any single burst of errors of length \( l \) or less (including the end-around bursts). The code has the following parameters:

\[
\begin{align*}
n &= \text{LCM}(2l-1, \Box) \\
n-k &= 2l-1+m.
\end{align*}
\]

**Example 7-1:** the polynomial \( \overline{p}(X) = 1 + X^2 + X^5 \) irreducible and has period \( \Box = 31 \) \( \Box \) Let \( l=m=5 \). Clearly \( \Box = 31 \) does not divide \( 2l-1=9 \). Then

\[
\overline{g}(X) = (X^9 + 1) \cdot (1 + X^2 + X^5) \\
= 1 + X^2 + X^5 + X^9 + X^{11} + X^{14}
\]
generates a Fire code with

\[
\begin{align*}
n &= \text{LCM}(31,9)=279 \\
n-k &= 2l-1+m=2\times5+5-1=14.
\end{align*}
\]

This code is capable of correcting any error burst of length 5 or less.
4. Decoding of Burst-Error-Correcting Codes

- Decoding consists two basic steps:
  1. Error-pattern determination, and
  2. Burst location determination.

- These two steps can be easily achieved by error-trapping decoding.
  The basic concept is to trap the error burst in a (syndrome) shift register by cyclically shifting the received vector $\tilde{r}$.

- Let $\tilde{r}(X)$ and $\tilde{e}(X)$ be the received and error polynomial respectively.

- Let
  $$\bar{s}(X) = s_0 + s_1 X + \ldots + s_{n-k-1} X^{n-k-1}$$
  be the syndrome of $\tilde{r}(X)$ which is the remainder obtained from dividing $\tilde{r}(X)$ by the generator polynomial $\bar{g}(X)$. 

Recall that $\bar{s}(X)$ is actually equal to the remainder of the error polynomial $\bar{e}(X)$ dividing by $\bar{g}(X)$,

$$\bar{e}(X) = \bar{a}(X) \cdot \bar{g}(X) + \bar{s}(X).$$

Suppose the errors in $\bar{e}(X)$ are confined to the $l$ high-order parity bit positions:

$$X_{n-k-l}, X_{n-k-l+1}, ..., X_{n-k-1}$$

Then,

$$\bar{e}(X) = e_{n-k-l}X_{n-k-l} + e_{n-k-l+1}X_{n-k-l+1} + ... + e_{n-k-1}X_{n-k-1}$$

Dividing $\bar{e}(X)$ by the generator polynomial $\bar{g}(X)$, we find that

$$\bar{e}(X) = 0 \cdot \bar{g}(X) + \bar{s}(X) = \bar{s}(X).$$
• The $l$ high-order syndrome bits,

$$s_{n-k-l}, s_{n-k-l+1}, \ldots, s_{n-k-1}$$

are identical to the errors in $\tilde{e}(X)$.

• The other $n-k-l$ low-order syndrome bits are zeros, i.e.,

$$s_0 = s_1 = \cdots = s_{n-k-l-1} = 0.$$

• Thus, when the received polynomial $\tilde{r}(X)$ is completely shifted into the syndrome register, the error pattern is **trapped** in the $l$ high-order stages of the syndrome register; and the other $n-k-l$ low order stages contain zeros.
Suppose the errors in $\bar{e}(X)$ are not confined to the $l$ high-order parity bit positions, but confined to $l$ consecutive positions (including the end-around case). For example,

Then, after a certain number of shifts of $\bar{r}(X)$, say $i$ cyclic shifts, the errors in $\bar{e}(X)$ will be shifted into the $l$ high-order parity bit positions of $\bar{r}^{(i)}(X)$.

At this instant, the errors are trapped in the $l$ high-order stages of the syndrome register, and the other $n-k-l$ low-order stages of the syndrome register contain zeros.
Knowing the number of shifts, i, (shorted in a counter), we can determine the location of burst in $\tilde{e}(X)$.

Then, error correction is done by adding the error pattern to $\tilde{r}(X)$ at the right location.

A general error-trapping decoder is shown in Figure 7-2.

**Error Trapping Decoder**

![Diagram of Error Trapping Decoder]

Figure 7-2
5. Binary RS Codes

- Consider a $t$-symbol-error correcting RS code $C$ of length $2^m-1$ with from $GF(2^m)$.
- The binary code derived from $C$ by representing each code symbol by a $m$-bit byte has length $n = m(2^m-1)$ and number of parity bits $n-k = 2^{mt}$.
- This binary RS code is capable of correcting any single burst of length $m(t-1)+1$ or less because such a burst can only affect $t$ or fewer symbols in the original RS code $C$. 
Example 7-4: Consider the NASA standard (255, 223) RS code over GF(2^8). It is capable of correcting $t=16$ symbol errors. The binary code derived from this RS code has length

$$n = 8 \times 255 = 2040,$$

and dimension

$$k = 8 \times 233 = 1784.$$ 

Hence it is a (2040, 1784) binary RS code. This code is capable of correcting any single burst of length

$$l = 8 \times (16 - 1) + 1 = 121$$

Or less.
6. Interleaving Technique

- Let $C$ be an $(n, k)$ linear code.
- Suppose we take $\lambda$ code words from $C$ and arrange them into $\lambda$ rows of an $\lambda \times n$ array as shown in figure 7-4. This structure is called block interleaver.

![Interleaved Array Diagram](image)

Figure 7-4  An interleaved array
Then we transmit this **code array** column by column in serial manner. By doing this, we obtain a vector of $n$ digits.

- Note that two consecutive bits in the same codeword are now separated by $\lambda-1$ positions.
- Actually, the above process simply **interleaves** $\lambda$ codewords in $C$. The parameter $\lambda$ is called **interleaving degree** (or **depth**).
- There are $(2k)^\lambda = 2^{k\lambda}$ such interleaved sequences and they form a $(\lambda n, \lambda k)$ linear code, called an interleaved code, denoted $C(\lambda)$.
- If the base code $C$ is a cyclic code with generator polynomial $\overline{g}(X)$, then the interleaved code $C(\lambda)$ is also cyclic. The generator polynomial of $C(\lambda)$ is $\overline{g}(X^\lambda)$. 
Error Correction Capability of an Interleaved Code

- A pattern of errors can be corrected for the whole array if and only if the pattern of errors in each row is a correctable pattern for the base code $C$.
- Suppose $C$ is a single-error-correcting code.
- Then a burst of length $\lambda$ or less, no matter where it starts, will affect no more than one digital in each row. This single bit error in each row will be corrected by the base code $C$.
- Hence the interleaved code $C(\lambda)$ is capable of correcting any error burst of length $\lambda$ or less.
Decoding of Interleaved Code

- At the receiving end, the received interleaved sequence is de-interleaved and rearranged back to a rectangular array of $\lambda$ rows.
- Then each row is decoded based on the base code $C$.
- Suppose the base code $C$ is capable of correcting any burst of length $l$ or less.
- Consider any burst of length $\lambda l$ or less. No matter where this burst starts in the interleaved code sequence, it will result a burst of length $l$ or less in each row of the corresponding code array as shown in Figure 7-5.
Figure 7-5  A burst of length $\lambda l$
As a result, the burst in each row will be corrected by the base code C.

Hence the interleaved code $C(\square)$ is capable of correcting any single error burst of length $\square l$ or less.

Interleaving is a very effective technique for constructing long powerful burst-error correcting codes from good short codes.

If the base code is an optimal burst-error-correcting code, the interleaved code is also optimal.
**Example 7-3**: Consider a (7,3) cyclic code $C$ generated by

$$g(X) = (X + 1)(X^3 + X + 1)$$

$$= 1 + X^2 + X^3 + X^4$$

This code is capable of correcting any burst of length $l = 2$ or less. It is optimal since

$$z = \frac{2l}{n-k} = \frac{2 \times 2}{7-3} = 1$$

- Suppose we interleave this code to a depth $\lambda = 10$.
- The interleaved code $C(10)$ is a (70,30) code which is capable of correcting any burst of length 20 or less.
- The burst-correcting efficiency of $C(10)$ is

$$z = \frac{2l}{n-k} = \frac{2 \times 20}{70-30} = 1$$

Hence $C(10)$ is also optimal.

- The generator polynomial of $C(10)$ is

$$g(X) = 1 + X^{20} + X^{30} + X^{40}$$
● Convolutional Interleaver:

A convolutional interleaver can be used in place of a block interleaver in much the same way.

Convolutional interleavers are better matched for use with the class of convolutional codes that will be described in next chapter.
7. Concatenated Coding Scheme

- **Concatenation** is a very effective method of constructing long powerful codes from shorter codes.
- It was devised by Forney in 1965.
- It is often used to achieve **high reliability** with **reduced decoding complexity**.
- A simple concatenated code is formed from two codes: an \((n_1,k_1)\) binary code \(C_1\) and an \((n_2,k_2)\) nonbinary code \(C_2\) with symbols from \(GF(2^{k_i})\), say a RS code.
- Concatenated codes are effective against a mixture of random errors and burst errors. Scattered random errors are corrected by \(C_1\). Bursts may affect relatively few bytes, but probably so badly that \(C_1\) cannot correct them. These few bytes can then be corrected by \(C_2\).
Figure 7-6  Concatenated coding
Encoding

• Encoding consists of two stages, the outer code encoding and the inner code encoding, as shown in Figure 7-6.
• First a message of $k_1 k_2$ bits are divided into $k_2$ bytes of $k_1$ bits each. Each $k_1$-bit byte is regarded as a symbol in GF($2^{k_1}$).
• This $k_2$-byte message is encoded into an $n_2$-byte codeword $\bar{v}$ in $C_2$.
• Each $k_1$-bit byte of $\bar{v}$ is then encoded into an $n_1$-bit codeword $\bar{w}$ in $C_2$.
• This results in a string of $n_2$ codewords in $C_2$, a total of $n_1 n_2$ bits.
• There are a total of $2^{k_1 k_2}$ such strings which form an $(n_1 n_2, k_1 k_2)$ binary linear code, called a concatenated code.
• $C_I$ is called the **inner code** and $C_2$ is called the **outer code**.
If the minimum distances of the inner and outer codes are $d_1$ and $d_2$ respectively, the minimum distance of their concatenation is at least $d_1 \cdot d_2$. 
Decoding

• Decoding of a concatenated code also consists of two stages, the inner code decoding and the outer code decoding, as shown in Figure 7-6.
• First, decoding is done for each inner codeword as it arrives, and the parity bits are removed. After $n_2$ inner codewords have been decoded, we obtain a sequence of $n_2 \ k_1$-bit bytes.
• This sequence of $n_2$ bytes is then decoded based on the outer code $C_2$ to give $k_1k_2$ decoded message bits.
• Decoding implementation is the straightforward combination of the implementations for the inner and outer codes.
Error Correction Capability

• Concatenated codes are effective against a mixture of random errors and bursts.
• In general, the inner code is a random-error-correcting code and the outer code is a RS code.
• Scattered random errors are corrected by the inner code, and bursts are then corrected by the outer code.
• Various forms of concatenated coding scheme are being used or proposed for error control in data communications, especially in space and satellite communications.
• In many applications, concatenated coding offers a way of obtaining the best of two worlds, performance and complexity.
8. Cascaded Coding Scheme : Product Code

- A simple generalization of the concatenated coding is shown in Fig. 7-7. The two-dimensional code is called the product code.
- The outer code $C_2$ is an $(n_2,k_2)$ RS code with symbols from $\text{GF}(2^m)$.
- The inner code $C_1$ is an $(n_1,k_1)$ binary linear code with $k_1 = \lambda m$ where $\lambda$ is a positive integer.
- The outer code $C_2$ is interleaved to a depth of $\lambda$.
- If the code $C_1$ has minimum weight $d_1$ and the code $C_2$ has minimum weight $d_2$, the minimum weight of the product code is exactly $d_1 \cdot d_2$. 
Figure 7-7 Code array for the product code $C_1 \square C_2$
**Encoding**

- A message of \( k_2 \) m-bit bytes (or \( k_2m \) bits) is first encoded into an \( n_2 \)-byte codeword in \( C_2 \).
- This codeword is then temporarily stored in a buffer as a row in an array as shown in Figure 7-7.
- After \( \lambda \) outer codewords have been formed, the buffer stores a \( \lambda \times n_2 \) array.
- Each column of the array consists of \( \lambda \) m-bit bytes (or \( \lambda m \) bits), and is encoded into an \( n_1 \)-bit codeword in \( C_1 \) and transmitted in serial manner.
- Note that the outer code is interleaved to a depth of \( \lambda \) and the inner code consists of \( \lambda \) bytes of message bits.