Principles of Data Communications

Reference Book: Data Communications and Networking by Behrouz A. Forouzan

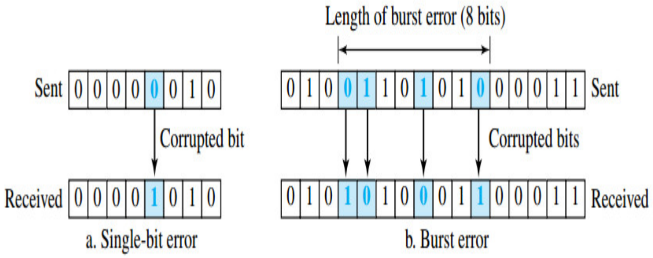
Error Detection and Correction

- Any time data are transmitted from one node to the next, they can become corrupted in passage.
- Many factors can alter one or more bits of a message.
- Some applications require a mechanism for detecting and correcting errors.

Types of Errors

- Single-bit error
    - The term single-bit error means that only 1 bit of a given data unit is changed from 1 to 0 or from 0 to 1.
- Burst error
    - The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

Single-bit and burst error



Redundancy

- The central concept in detecting or correcting errors is redundancy.
- To be able to detect or correct errors, we need to send some extra bits with our data.
- These redundant bits are added by the sender and removed by the receiver.
- Their presence allows the receiver to detect or correct corrupted bits.

Detection versus Correction

- The correction of errors is more difficult than the detection.
- In error detection, we are only looking to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of corrupted bits. A single-bit error is the same for us as a burst error.
- In error correction, we need to know the exact number of bits that are corrupted and, more importantly, their location in the message.

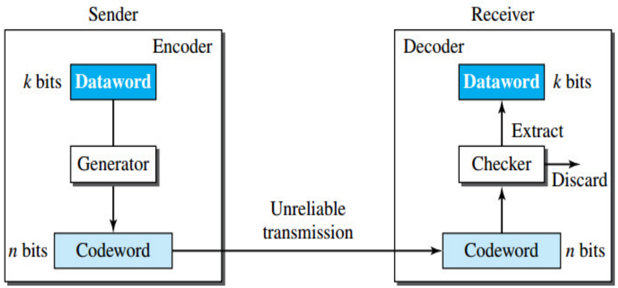
Coding

- Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.
- We can divide coding schemes into two broad categories:
    - Block Coding
    - Convolution Coding

BLOCK CODING

- In block coding, we divide our message into blocks, each of k bits, called datawords.
- We add r redundant bits to each block to make the length n = k + r.
- The resulting n-bit blocks are called codewords.

Process of error detection in block coding



Error Detection

- Figure shows the role of block coding in error detection.
- The sender creates codewords out of datawords by using a generator that applies the rules and procedures of encoding (discussed later).
- Each codeword sent to the receiver may change during transmission.

Example

Let us assume that $k = 2$ and $n = 3$. Table shows the list of datawords and codewords. Later, we will see how to derive a codeword from a dataword.

Table *A code for error detection in Example*

Dataword	*Codeword*	*Dataword*	*Codeword*
00	000	10	101
01	011	11	110

Hamming Distance

- One of the central concepts in coding for error control is the idea of the Hamming distance.
- The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits.
- We show the Hamming distance between two words x and y as d(x, y).
- We may wonder why Hamming distance is important for error detection. The reason is that the Hamming distance between the received codeword and the sent codeword is the number of bits that are corrupted during transmission.
- For example, if the codeword 00000 is sent and 01101 is received, 3 bits are in error and the Hamming distance between the two is $d(00000, 01101) = 3$. In other words, if the Hamming distance between the sent and the received codeword is not zero, the codeword has been corrupted during transmission.

- The Hamming distance can easily be found if we apply the XOR operation on the two words and count the number of 1s in the result. Note that the Hamming distance is a value greater than or equal to zero.
- d(000, 011)?
- d(10101, 11110)?
- Minimum Hamming distance is the smallest Hamming distance between all possible pairs of codewords.

Parity-Check Code

- In this code, a k-bit dataword is changed to an n-bit codeword where n = k + 1.
- The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even.
- Although some implementations specify an odd number of 1s, we discuss the even case.
- The code is a single-bit error-detecting code.

Dataword	Codeword	Dataword	Codeword
00	000	10	101
01	011	11	110

Table *Simple parity-check code C(5, 4)*

Dataword	*Codeword*	*Dataword*	*Codeword*
0000	**00000**	1000	**10001**
0001	**00011**	1001	**10010**
0010	**00101**	1010	**10100**
0011	**00110**	1011	**10111**
0100	**01001**	1100	**11000**
0101	**01010**	1101	**11011**
0110	**01100**	1110	**11101**
0111	**01111**	1111	**11110**

Example

Let us look at some transmission scenarios. Assume the sender sends the dataword 1011. The codeword created from this dataword is 10111, which is sent to the receiver. We examine five cases:

1. No error occurs; the received codeword is 10111. The syndrome is 0. The dataword 1011 is created.
2. One single-bit error changes a_1. The received codeword is 10011. The syndrome is 1. No dataword is created.
3. One single-bit error changes r_0. The received codeword is 10110. The syndrome is 1. No dataword is created. Note that although none of the dataword bits are corrupted, no dataword is created because the code is not sophisticated enough to show the position of the corrupted bit.
4. An error changes r_0 and a second error changes a_3. The received codeword is 00110. The syndrome is 0. The dataword 0011 is created at the receiver. Note that here the dataword is wrongly created due to the syndrome value. The simple parity-check decoder cannot detect an even number of errors. The errors cancel each other out and give the syndrome a value of 0.
5. Three bits—a_3, a_2, and a_1—are changed by errors. The received codeword is 01011. The syndrome is 1. The dataword is not created. This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.

A parity-check code can detect an odd number of errors.

- A parity-check code can detect an odd number of errors.

CYCLIC CODES

- In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.
- For example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.
- For example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword. In this case, if we call the bits in the first word a0 to a6, and the bits in the second word b0 to b6, we can shift the bits by using the following:
  - b1=a0; b2=a1; b3=a2; b4=a3; b5=a4; b6=a5; b0=a6
- The last bit of the first word is wrapped around and becomes the first bit of the second word.

Summary

- Hamming Distance
- Parity Check
- Cyclic Codes

THANK YOU