ERRORS, DETECTION & CORRECTION

OBJECTIVE

- ✓ Understand error classification
- ✓ Types of error
- ✓ Understand concept redundancy
- ✓ Hamming code concept
- ✓ CRC concept
- ✓ Checksum technic

7.1 INTRODUCTION: Errors in the data are basically caused due to the various impairments that occur during the process of transmission. When there is an imperfect medium or environment exists in the transmission it prone to errors in the original data. Errors can be classified as follows: Attenuation Noise Distortion

70

7.2 ERRORS CLASSIFICATION Following are the categories of the errors: **1. Attenuation:** As signal travels through the medium, its strength decreases as distance increases, as shown in the figure 7.1, the example is voice, it becomes weak over the distance and loses its contents beyond a certain distance. As the distance increases attenuation also increases.

Strength Distance Figure 7.1 **2. Noise:** Noise is defined as an unwanted data. When some electromagnetic signal gets inserted during the transmission, it is generally called as a Noise. Due to Noise it is difficult to retrieve the original data or information. **3. Distortion:** When there is an interference of the different frequencies who travel across the medium with the different speed, Distortion occurs. So it is important to have a space (guard space) between the different frequencies. **7.3 TYPES OF ERRORS:** If the signal comprises of binary data there can be two types of errors which are possible during the transmission: 1. Single bit errors 2. Burst Errors

71

Figure 7.2 1. Single-bit errors: In single-bit error, a bit value of 0 changes to bit value 1 or vice versa. Single bit errors are more likely to occur in parallel transmission. Figure 7.3 (a) **2. Burst errors:** In Burst error, multiple bits of the binary value changes. Burst error can change any two or more bits in a transmission. These bits need not be adjacent bits. Burst errors are more likely to occur in serial transmission. Figure 7.3 (b)

Original data Received data

0 1 0 0 1 1 0 1 1 1

0 0 0 0 1 1 0 1 1 1

Figure 7.3 (a) Single bit error

Original data Received data

0 1 0 0 1 1 0 1 1 1

0 0 0 1 1 1 0 0 1 1

Figure 7.3 (b) Burst error **7.4 REDUNDANCY** In order to detect and correct the errors in the data communication we add some extra bits to the original data. These extra bits are nothing but the redundant bits which will be removed by the receiver after receiving the data. Their presence allows the receiver to detect or correct corrupted bits. Instead of repeating the entire data stream, a short group of bits may be attached to the entire data stream.This technique is called redundancy because the extra bits are redundant to the information: they are discarded as soon as the accuracy of the transmission has been determined.

72

Receiver Sender

Figure 7.4 There are different techniques used for transmission error detection and correction. **Figure 7.5 2. Parity Check:**

0 1 1 0 1 1 0

0 1 1 0 1 1 0 0 1 1 0

Data and Redundancy

0 1 1 0 1 1 0

Rejected
Data

0 1 1 0 1 1 0 0 1 1 0

Data and Redundancy

Data

Data

Medium

OK ?

Yes

No

In this technique, a redundant bit called a parity bit is addedto every data unit so that the total number of 1‘s in the unit(including the parity bit) becomes even (or odd). Following

73

Figure shows this concept when transmit the binary data unit110101. Receiver Sender

Simple parity check can detect all single-bit errors. It canalso detect burst errors as long as the total number of bitschanged is odd. This method cannot detect errors where thetotal number of bits changed is even. **Two-Dimensional Parity Check:** A better approach is the two dimensional parity checks. In this method, a block of bits is organized in a table (rows and columns). First we calculate the parity bit for each data unit. Then we organize them into a table. We then calculate the parity bit for eachcolumn and create a new row of 8 bits. Consider the following example; we have four data units to send. They are organized in the tabular form as shown below.

Drop parity bit and accept the data

1 0 0 1 0 1

1 0 0 1 0 1 1

Data

Calculate

Parity bit

Medium

Calculate

Parity bit

Bits

Even ?

Yes

No

Rejected Data

Original Data

Data and Parity bits We then calculate the parity bit for eachcolumn and create a new row of 8 bits; they are the parity bits for the whole block. Note that the first parity bit in thefifth row is calculated based on all first bits: the secondparity bit is calculated based on all second bits: and so on.We then attach the 8 parity bits to the original data andsend them to the receiver. Two-dimensional parity check increases the likelihood ofdetecting burst errors.A burst error of more than _n‘bits is also detected by thismethod with a very high probability. **3. Cyclic Redundancy Check (CRC)** Most powerful of the redundancy checkingtechniques is the cyclic redundancy check (CRC). This method is based on the binary division. In CRC, the desired sequence of redundant bits are generated and is appended to the end of data unit. It is also called as CRC reminder. So that the resulting data unit becomes exactly divisible by a predetermined binary number.

0110110 1101001 1110011 0001110

0 1 1 0 1 1 0 0

1 1 0 1 0 0 1 0

1 1 1 0 0 1 1 1

0 0 0 1 1 1 0 1

0 1 0 0 0 1 0 0

Row parities

Column parities

01101100 11010010 11100111 00011101 01000100

At its destination, theincoming data unit is divided by the same number. If at thisstep there is no remainder then the data unit

is assumed to be correct and is therefore accepted.A remainder indicates thatthe data unit has been damaged in transit and thereforemust be rejected. The redundancy bits used by CRC are derived by dividingthe data unit by a predetermined divisor; the remainder isthe CRC.To be valid, a CRC must have two qualities: It musthave exactly one less bit than the divisor, and appending itto the end of the data string must make the resulting bitsequence exactly divisible by the divisor. The following figure shows the process: Receiver Sender

Data 00…0

n bits

n+1 bits

remainder
n bits
Divisor
CRC
Data CRC

Zero, accept
Non- zero, Reject
Divisor
Remainder
Data CRC

Step1: A string of 0's is appended to the data unit. It is n bits long. The number n is 1 less if-number of bits in the predetermined divisor which is n + 1 bits. Step 2: The newly generated data unit is divided by the divisor, using a process called as binary division. The remainder resulting from this division is the CRC. Step 3: the CRC of n bits derived in step 2 replaces the appended 0's at the data unit. Note that the CRC may consist of all 0's. The data unit arrives at the receiver data first, followed by the CRC. The receiver treats the whole string as a unit and divides it by the same divisor that was used the CRC remainder. If the string arrives without error, the CRC checker yields a remainder of zero, the data unit passes. If the string has been changed in transit, the division yields zero remainder and the data unit does not pass.
76
Following figure shows the process of generating CRC reminder: Figure : Quotient
___1 1 1 1 0 1________ Divisor 1 1 0 1 1 0 0 1 0 0 0 0 0 Extra bits 1 1 0 1 _________ 1 000 1 1 0 1 _______ 1 0 1 0 1 1 0 1 ______ 1 1 1 0 1 1 0 1 ________ 0 1 1 0 0 0 0 0 _______ 1 1 0 0 1 1 0 1 ________ 0 0 1 Remainder A CRC checker functions does exactly as the generator does. After receiving the data appended with the CRC, it does the samemodulo-2 division. If the remainder is all 0's, the CRC isdropped and the data is accepted: otherwise, the receivedstream of bits is discarded and data is resent.
77
Following Figure shows the same process of division in the receiver. Figure: Quotient
___1 1 1 1 0 1________ Divisor 1 1 0 1 1 0 0 1 0 0 0 0 1 CRC 1 1 0 1 _________ 1 000 1 1 0 1 _______ 1 0 1 0 1 1 0 1 ______ 1 1 1 0 1 1 0 1 ________ 0 1 1 0 0 0 0 0 _______ 1 1 0 1 1 1 0 1 ________ 0 0 0 Result Performance: CRC is a very effective error detection method. If the divisoris chosen according to the previously mentioned rules, 1.CRC can detect all burst errors that affect an odd numberof bits. 2.CRC can detect all burst errors of length less than or equalto the degree of the polynomial 3.CRC can detect, with a very high probability, burst errorsof length greater than the degree of the polynomial. **3. Checksum**

A checksum is fixed length data that is the result of performing certain operations on the data to be sent from sender to the receiver. The sender runs the appropriate checksum algorithm to compute the checksum of the data, appends it as a field in the