Error Detection

Dr. G. Omprakash

Assistant Professor, ECE, KLEF

Aim of the session

To familiarize students with the Error detection techniques

Learning Outcomes

At the end of this session, you should be able to:

- Understand the concept of error detection
- Implement error detection techniques, such as parity checking, checksums, and CRC, to detect errors in data transmission.

Overview

Error Control

- Error control is both error detection and error correction
- In the data-link layer, error correction is done using retransmission of the corrupted frame
- **Single-bit error** : Only 1 bit of a given data unit is changed from 1 to 0 or from 0 to 1
- **Burst error**: Two or more bits in the data unit have changed from 1 to 0 or from 0 to 1.
- If we are sending data at 1 kbps, a noise of 1/100 s duration can affect 10 bits

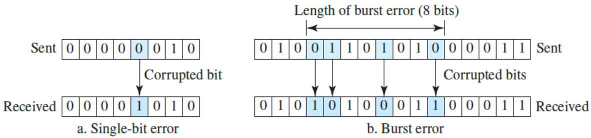


Figure: Single-bit error and burst error

Redundancy

- To detect or correct errors, we need to send some extra bits *(redundant bits)* with our data
- Their presence allows the receiver to detect or correct corrupted bits.
- *Error Detection*: Looking only to see if any error has occurred (Yes/No)
- *Error Correction*: Need to know the **exact number** of corrupted bits and their **locations**
- **Coding:** Redundancy is achieved through various coding schemes (**block coding** and convolution coding)

Block Coding

- **Datawords**: Divide the message into blocks, each consisting of k bits
- Add r redundant bits to each block to make the length $n = k + r$.
- (n, k) code, k/n -fraction of the codeword (**code rate**)
- Resulting n-bit blocks are called **codewords**:
- The number of possible codewords (2^n) is larger than the number of possible datawords (2^k)
- since the block-coding process is one-to-one, $2^n - 2^k$ codewords are not used (invalid/illegal)
- If the receiver receives an invalid codeword, this indicates that the data were corrupted during transmission.

Error Detection in Block coding

How can errors be detected by using block coding?

- Receiver has a list of valid codewords
- The original codeword has changed to an invalid one-receiver detects this change

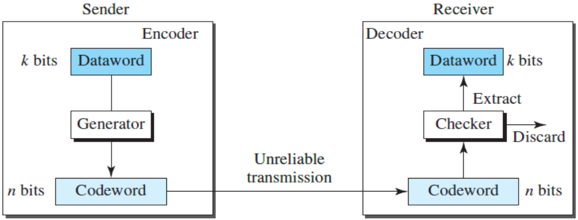


Figure: Error detection in block coding

Drawback: If the corrupted codeword matches a valid codeword, the error remains undetected

Parity-Check Code

- The parity bit is chosen so that the number of 1 bits in the codeword is even (or odd)
- The parity bit is appended to the data. $n = k + 1$
- Even parity
  - Eg: Data word:1011010 $\implies$ Codeword: 10110100
- Odd parity
  - Eg: Data word:1011010 $\implies$ Codeword: 10110101

Parity-Check Code

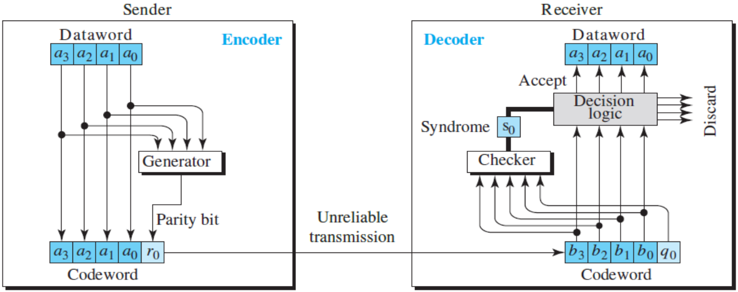


Figure: Encoder and decoder for simple parity-check code

- The encoder generates a parity bit
  $r_0 = a_0 + a_1 + a_2 + a_3 \quad (modulo - 2)$
- The syndrome is 0 when the number of 1s in the received codeword is even; otherwise, it is 1.
- Syndrome 0: no detectable error in the received codeword

Parity-Check Code

Dataword $(a_3 a_2 a_1 a_0)$ 1011 $\implies$ codeword created is 10111

- One single-bit error changes a_1
  - Received codeword is 10011 $\implies$ syndrome=1, Discard
- One single-bit error changes r_0
  - Received codeword is 10110 $\implies$ syndrome=1, Discard
    - Eventhough dataword is not corrupted, still it is discarded
- Two-bit error: r_0 and a_3
  - Received codeword is 00110 $\implies$ syndrome=0, create codeword
    - Parity-check decoder cannot detect an even number of errors
- Three bit error —a_3, a_2, a_1
  - Received codeword is 01011 $\implies$ syndrome=1, Discard

Two dimensional Even Parity scheme

$$\begin{array}{ccccc|c} 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

(b) No errors

$$\begin{array}{ccccc|cl} 0 & 1 & 1 & 1 & 0 & 1 & \\ 0 & 0 & 1 & 1 & 0 & 1 & \text{Row parity error} \\ 0 & 1 & 0 & 0 & 0 & 1 & \\ 0 & 1 & 0 & 1 & 1 & 1 & \\ \hline 0 & 0 & 0 & 1 & 1 & 0 & \end{array}$$

Column parity error

(c) Correctable single-bit error

$$\begin{array}{ccccccc|c} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

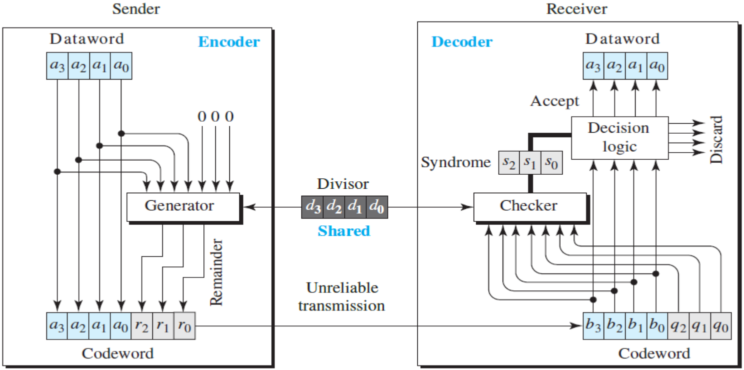
(d) Uncorrectable error pattern

- Error-detecting code consists of $i + j + 1$ parity bits
- Every bit participates in two parity checks
- Odd number of bit errors is detected

Cyclic codes

Cyclic codes are special linear block codes with property: if a codeword is cyclically shifted (rotated), the result is another codeword.

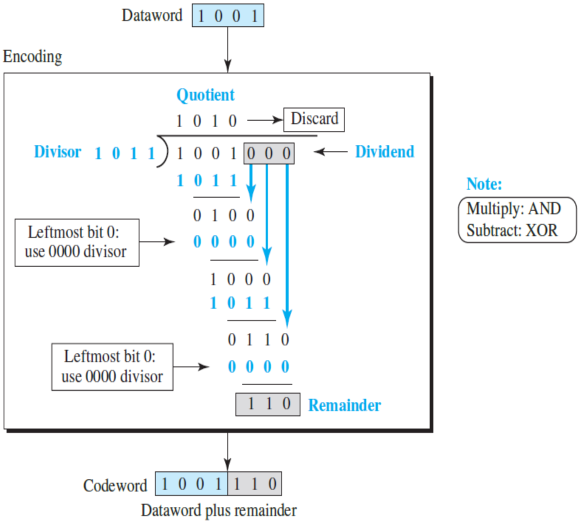
- **Cyclic Redundancy Check** (CRC) is a polynomial code
- Widely used in the data link layer



Cyclic Redundancy Check

- In the encoder, the dataword has k bits (4 here)
- The codeword has n bits (7 here).
- Dataword is augmented by adding n-k (3 here) 0s to the right hand side of the dataword
- n-bit result is fed into the generator
- The generator uses a divisor of size n-k+1 (4 here), predefined and agreed upon
- The generator divides the augmented dataword by the divisor (modulo-2 division)
- The quotient of the division is discarded; the remainder ($r_2 r_1 r_0$) is appended to the dataword to create the codeword.

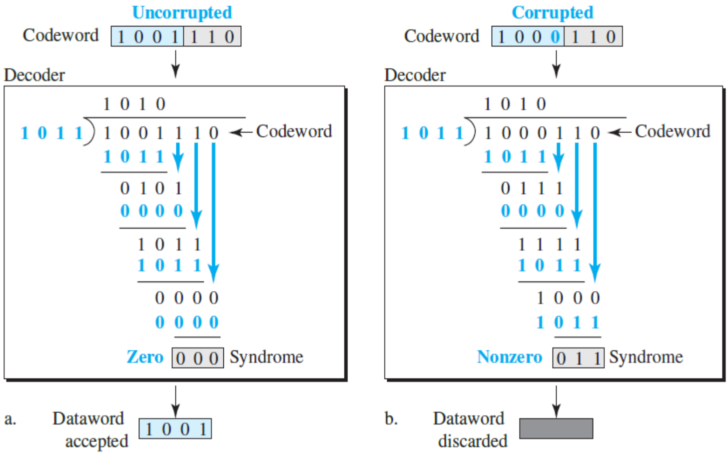
CRC Encoding



Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Figure: CRC code C(7,4)

CRC Decoding



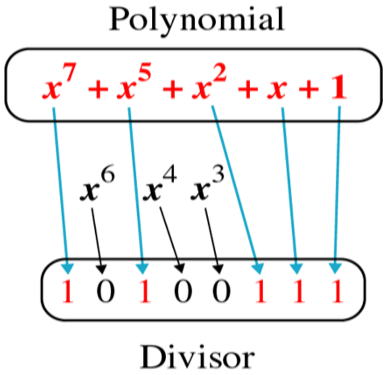


Figure: Polynomial representation of divisor

Problem

A bit stream 11001010 is transmitted using the standard CRC method. The generator polynomial is 1110. (1). Show the actual bit string transmitted. (2). Analyze the error if the third bit from the left is inverted during transmission.

Checksum

The word **checksum** is often used to mean a group of check bits associated with a message

- Checksum is based on the concept of redundancy
- checksum is usually placed at the end of the message, as the complement of the sum function
- To detect errors
  - Sum the entire received codeword, both data bits and checksum and take the complement.
    - Result=0 $\implies$ No Error

Checksum

Original data :
10101001 00111001

$$
\begin{array}{rr}
 & 10101001 \\
+ & 00111001 \\
\hline
Sum & 11100010 \\
Complement & 00011101
\end{array}
$$

Code transmitted:
10101001 00111001 00011101

Checksum

Received data :
10101001 00111001 00011101

$$
\begin{array}{rr}
 & 10101001 \\
 & 00111001 \\
+ & 00011101 \\
\hline
Sum & 11111111 \\
Complement & 00000000
\end{array}
$$

Original Data

10011001	11100010	00100100	10000100
1	2	3	4

k=4, m=8

Sender

```
1          10011001
2          11100010
          ----------
          1 01111011
          └──────→ 1
          ----------
            01111100
3           00100100
          ----------
            10100000
4           10000100
          ----------
          1 00100100
          └──────→ 1
          ----------
Sum:        00100101
          ----------
CheckSum:   11011010
```

Receiver

```
1          10011001
2          11100010
          ----------
          1 01111011
          └──────→ 1
          ----------
            01111100
3           00100100
          ----------
            10100000
4           10000100
          ----------
          1 00100100
          └──────→ 1
          ----------
            00100101
            11011010
          ----------
Sum:        11111111
Complement: 00000000
```

Conclusion : Accept Data

Bandwidth, Bit length

- The **bandwidth** of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.
  - If a composite signal contains frequencies between 1000Hz and 5000Hz, its bandwidth B=5000 – 1000= 4000Hz.
- If a signal has L levels, each level needs $log_2 L$ bits
- **Bit length** is the distance 1 bit occupies on the transmission medium; $Bit\ length = \frac{1}{bit\ \ rate}$

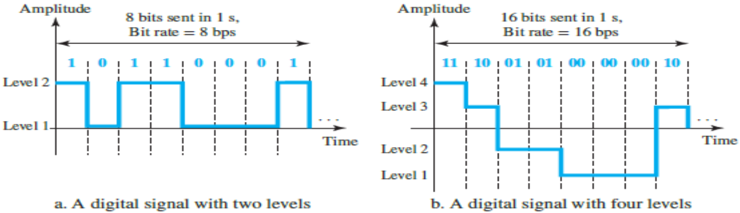


Figure: Signal level

- Assume we need to download text documents at the rate of 100 pages per minute. What is the required bit rate of the channel? A page is an average of 24 lines with 80 characters in each line. If we assume that one character requires 8 bits, what is the required bit rate?
  - $100 \times 24 \times 80 \times 8/60 = 2,56,000$ bps $= 256$ Kbps
- **Nyquist Bit rate** $=2 \times B \times log_2 L$
  - We need to send 265 kbps over a noiseless (ideal) channel with a bandwidth of 20 kHz. How many signal levels do we need?
    - $265,000 = 2 \times 20,000 \times log_2 L$

Shannon Capacity and Attenuation

- **Shannon Capacity** To determine the theoretical highest data rate for a noisy channel: $C = B \times log_2(1 + SNR)$
  - A telephone line normally has a bandwidth of 3000 Hz (300 to 3300 Hz) assigned for data communications. The signal-to-noise ratio is usually 3162. Calculate the theoretical highest bit rate of this telephone line
    - $C = B \times log_2(1+SNR) = 3000 log_2(1+3162) = 34,881 bps$
- Suppose a signal travels through a transmission medium and its power is reduced to one-half. This means that $P_2 = 0.5P_1$. In this case, the attenuation
  - the attenuation in $dB = 10 log_{10} \frac{P_2}{P_1} = 10 log_{10} \frac{0.5P_1}{P_1} = -3dB$

Bandwidth delay product

Bandwidth-delay product is the maximum number of bits that can fill the link

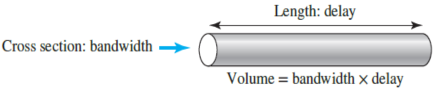


Figure: Bandwidth-delay product

- Given link with a bandwidth of 1 bps, assume that the delay of the link is 5s
- Link capacity=Bandwidth$\times$ delay=1$\times$ 5=5bits

Bandwidth delay product

Q. Calculate the (bandwidth $\times$ delay) product for 100Mbps Ethernet with a delay of 10μsec. Use one-way delay, measured from first bit sent to first bit received

A. $100 \times 10^6 \times 10 \times 10^{-6} = 1000$ *bits*

Latency

- The latency or delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source.
- Latency = propagation delay + transmission delay + queuing delay + processing delay

What is the total delay (Latency) for a frame of size 5 million bits that is being on the sent on a link with 10 routers each having a queuing time of 2 μsec and a processing time of 1μsec. The link has a bandwidth of 5Mbps. Which component if the total delay is dominant? Which one is negligible? Assume The length of the link is 2000 Km. The speed of light inside the link is 2 $\times 10^8$ m/s

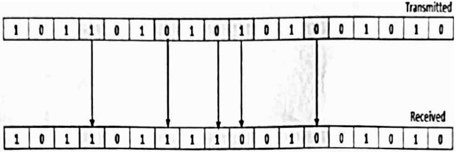
Latency

What is the total delay (Latency) for a frame of size 5 million bits that is being on the sent on a link with 10 routers each having a queuing time of 2 μsec and a processing time of 1μsec. The link has a bandwidth of 5Mbps. Which component if the total delay is dominant? Which one is negligible? Assume The length of the link is 2000 Km. The speed of light inside the link is 2 $\times 10^8$ m/s

- Propagation delay $= \frac{Distance}{propagation\ \ speed} = \frac{2000Km}{2 \times 10^8 m/s} =$ 10 ms
- Transmission delay $= \frac{Message\ \ size}{Bandwidth} = \frac{5 \times 10^6 bits}{5Mbps} =$ 1 s
- Queuing delay = 10 routers $\times 2\mu s$ = 20 μs
- Processing Delay = 10 routers $\times 1\mu s$ = 10 μs
- Latency = propagation delay + transmission delay + queuing delay + processing delay
- Latency= 10ms+ 1s+ 20 μs+ 10μs

Problems

- An image is of size 1024×768 pixels with 3bytes/pixel. Assume the image is uncompressed. How long does it take to transmit it over a
  - 56Kbps Modem channel
  - 1Mbps Cable Modem
  - 10Mbps Ethernet
- Find the length of the burst error



Acknowledge various sources for the images.

Thankyou