UNIVERSIDAD DEL VALLE DE GUATEMALA
CC3067 - REDES
Sección 11
MIGUEL NOVELLA LINARES

Laboratorio 2 - Primera parte

Esquemas de detección y corrección de errores

Isabella Miralles #22293

Ruth de León #22428

Guatemala, julio 2025

Corrección de errores

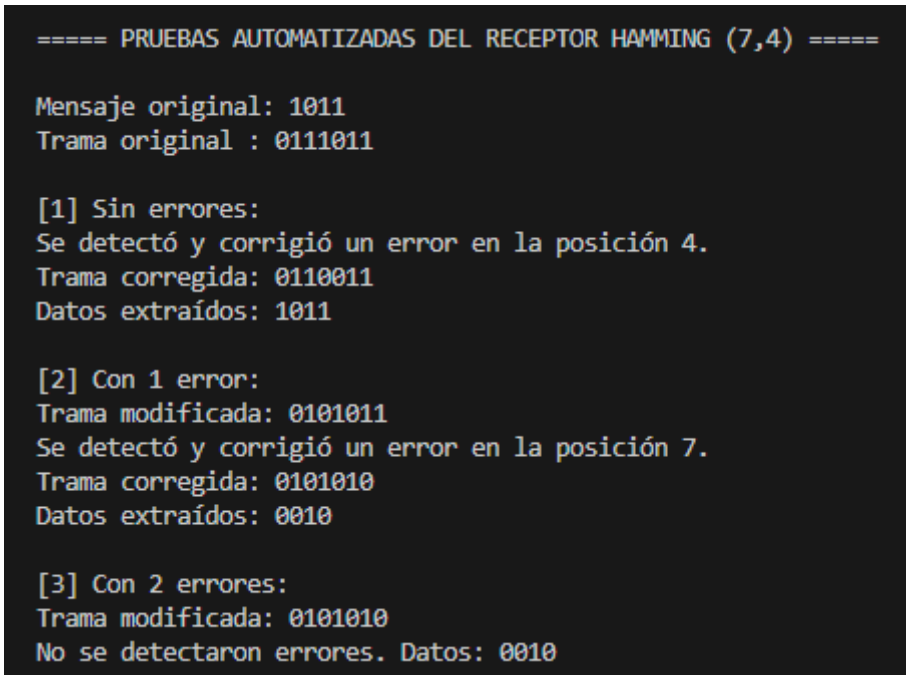
En este laboratorio, para la corrección de errores se implementó el algoritmo Hamming (7,4). Este permite codificar una trama de 4 bits en una de 7 bits añadiendo 3 bits de paridad para la detección y corrección de errores.

Por medio de pruebas controladas y utilizando el mismo conjunto de mensajes base, se implementó el emisor en c++ y el receptor en python.

Escenarios de prueba

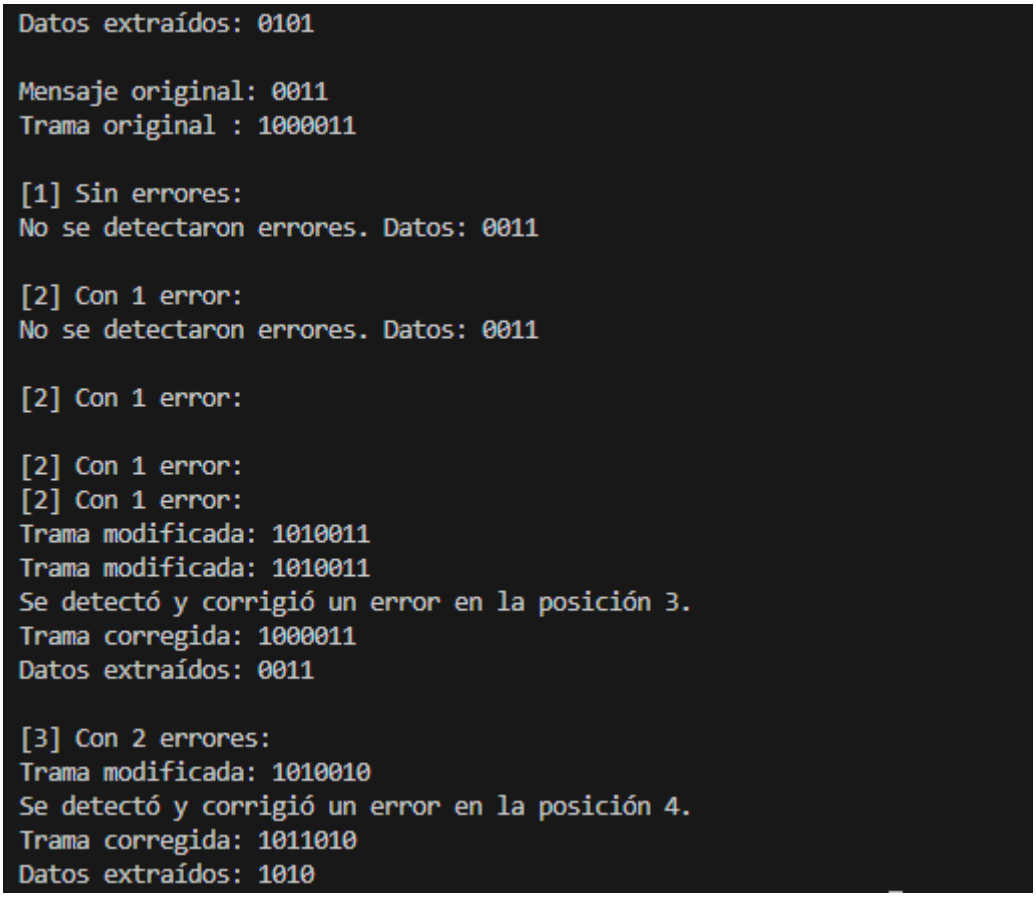
Cada mensaje se probó en tres escenarios

- Trama sin errores
- Trama con un solo bit alterado
- Trama con dos bits alterados



```
Mensaje original: 0001
Trama original : 1000001

[1] Sin errores:
Se detectó y corrigió un error en la posición 6.
Trama corregida: 1000011
Datos extraídos: 0011

[2] Con 1 error:
Trama modificada: 1010001
Se detectó y corrigió un error en la posición 5.
Trama corregida: 1010101
Datos extraídos: 1101

[3] Con 2 errores:
Trama modificada: 1010000
Se detectó y corrigió un error en la posición 2.
Trama corregida: 1110000
Datos extraídos: 1000

Mensaje original: 1111
Trama original : 0001111

[1] Sin errores:
No se detectaron errores. Datos: 0111

[2] Con 1 error:
Trama modificada: 0011111
Se detectó y corrigió un error en la posición 3.
Trama corregida: 0001111
Datos extraídos: 0111

[3] Con 2 errores:
Trama modificada: 0011110
Se detectó y corrigió un error en la posición 4.
Trama corregida: 0010110
```

```
Datos extraídos: 1110

Mensaje original: 0100
Trama original : 1001100

[1] Sin errores:
No se detectaron errores. Datos: 0100

[2] Con 1 error:
Trama modificada: 1011100
Se detectó y corrigió un error en la posición 3.
Trama corregida: 1001100
Datos extraídos: 0100

[3] Con 2 errores:
Trama modificada: 1011101
Se detectó y corrigió un error en la posición 4.
Trama corregida: 1010101
Datos extraídos: 1101

Mensaje original: 1100
Trama original : 0111100

[1] Sin errores:
No se detectaron errores. Datos: 1100

[2] Con 1 error:
Trama modificada: 0101100
Se detectó y corrigió un error en la posición 3.
Trama corregida: 0111100
Datos extraídos: 1100

[3] Con 2 errores:
Trama modificada: 0101101
Se detectó y corrigió un error en la posición 4.
Trama corregida: 0100101
```



Preguntas

¿Es posible manipular los bits de tal forma que el algoritmo seleccionado no sea capaz de detectar el error? ¿Por qué sí o por qué no?

Si, ya que el algoritmo solo corrige un error de bit. Al momento de alterar dos bits en posiciones específicas, es posible que el algoritmo no lo detecte o que detecte el error, pero corrija el bit incorrecto, esto genera datos corruptos sin advertencia.

Ventajas

- Corrige un error de bit con solo 3 bits extra
- Bajo costo computacional
- Implementación sencilla

Desventajas

- No detecta múltiples errores con certeza
- No es escalable (Solo 4 bits de datos por trama)
- Vulnerable a errores simétricos de 2 bits

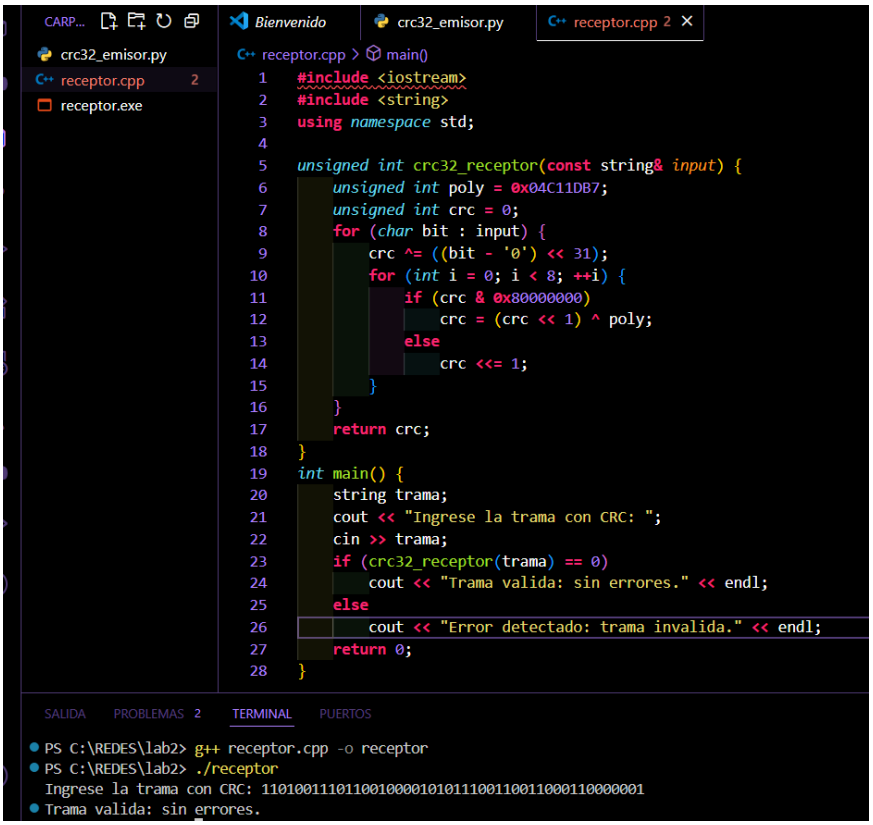
Detección de errores

En esta sección se implementó el algoritmo CRC-32 (Cyclic Redundancy Check) para detectar errores en tramas binarias. Este método permite verificar la integridad de los datos transmitidos usando un polinomio generador estándar de 32 bits: Polinomio estándar CRC-32: 0x04C11DB7 ($x^{32} + ^{6} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$)

Emisor

1. Solicita una trama binaria al usuario ("11010101").
2. Se calcula el CRC-32 de la trama, agregando 32 bits de ceros al final antes de aplicar el algoritmo.
3. El resultado del CRC se concatena al final de la trama original para formar el mensaje transmitido.

Receptor



Escenario de prueba:

Mensaje base utilizado: 11010011101100

Tipo de prueba	Trama entregada	Resultado
2 bits cambiados	11010011101100100001010111001100110011100000000	Error detectado
1 bit cambiado	11010011101100100001010111001100110000110000000	Error detectado
Sin cambios	11010011101100100001010111001100110000110000001	Trama válida

Pregunta

¿Es posible manipular los bits de tal forma que el algoritmo seleccionado no sea capaz de detectar el error?

Es muy poco probable que un error no sea detectado con CRC-32. Está diseñado para detectar: Todos los errores de 1 bit, todos los errores de 2 bits y todas las ráfagas de hasta 32 bits.
Básicamente pueden existir colisiones (mensajes diferentes con el mismo CRC), aunque su probabilidad es extremadamente baja.

Ventaja

- Alta capacidad de detección de errores múltiples.
- Uso extendido en redes y almacenamiento.
- Buena eficiencia en hardware/software optimizado

Desventaja

- Alta capacidad de detección de errores múltiples.
- Uso extendido en redes y almacenamiento.
- Buena eficiencia en hardware/software optimizado

Enlace del repositorio
https://github.com/Isabella-22293/Lab2-Redes.git