

Laboratorio 2

Esquemas de Detección y Corrección de Errores

Andy Fuentes 22944

Davis Roldan 22672

1. Antecedentes

Los errores de transmisión suceden en toda comunicación y requieren mecanismos de detección y corrección de errores.

2. Objetivos

- Analizar algoritmos de detección y corrección.
- Implementar un algoritmo de cada tipo.
- Identificar ventajas y desventajas.

3. Algoritmos implementados

Corrección Hamming (7,4). Emisor en C++, receptor en Python.

Detección CRC-32 (polinomio IEEE 802.3). Emisor en Python, receptor en C++.

4. Implementación

5. Pruebas y resultados

5.1 Captura de tests/results.log

5.2 Extractos clave

Escenario	Línea de salida
Hamming sin error	msg=1010 code=1011010 ->Sin errores. Datos: 1010
Hamming 1 error	msg=0111 ok=0001111 err=0101111 ->Error en bit 2. Código corregido: 0001111. Datos: 0111

Escenario	Línea de salida
Hamming 2+ errores	msg=0001 ok=1101001 err=1001011 ->Error en bit 4. Código corregido: 1000011. Datos: 0011
CRC-32 sin error	msg=0001 frame=000110100101000001011101111100011011 ->Sin errores. Trama OK.
CRC-32 1 error	err_frame=011101001100011001110111101000101110 ->Error detectado. Se descarta.
CRC-32 2+ errores	err_frame=000010100101000001011101011100011011 ->Error detectado. Se descarta.

6. Caso no detectado

Hamming: con dos errores (peso 2) el síndrome apunta a una posición incorrecta, de modo que la corrección produce datos erróneos. **CRC-32:** existen colisiones teóricas (errores cuyo polinomio es múltiplo del generador), pero encontrarlas es inviable en la práctica.

7. Ventajas y desventajas

- **Hamming (7,4)**
 - Overhead: +3 bits por cada 4 bits de datos.
 - Corrige un único bit sin retransmisión.
 - No corrige múltiples errores; puede “corregir mal”.
- **CRC-32**
 - Detecta casi todos los errores, incluso ráfagas largas.
 - Overhead fijo de 32 bits.
 - No corrige; requiere retransmisión/ARQ.

8. Conclusiones

Ambos algoritmos cumplen su propósito: Hamming corrige un bit con bajo overhead; CRC-32 ofrece robusta detección en tramas largas.

Repositorio: <https://github.com/Andyfer004/Lab2-Redes>

```

tests > results.log
1  === Hamming (7,4)  SIN error ===
2  msg=1010 code=1011010 -> Sin errores. Datos: 1010
3  msg=0111 code=0001111 -> Sin errores. Datos: 0111
4  msg=0001 code=1101001 -> Sin errores. Datos: 0001
5
6  === Hamming (7,4)  1 error ===
7  msg=1010 ok=1011010 err=1011000 -> Error en bit 6. Código corregido: 1011010. Datos: 1010
8  msg=0111 ok=0001111 err=0101111 -> Error en bit 2. Código corregido: 0001111. Datos: 0111
9  msg=0001 ok=1101001 err=1100001 -> Error en bit 4. Código corregido: 1101001. Datos: 0001
10
11 === Hamming (7,4)  2+ errores ===
12 msg=1010 ok=1011010 err=0011011 -> Error en bit 6. Código corregido: 0011001. Datos: 1001
13 msg=0111 ok=0001111 err=0011011 -> Error en bit 6. Código corregido: 0011001. Datos: 1001
14 msg=0001 ok=1101001 err=1001011 -> Error en bit 4. Código corregido: 1000011. Datos: 0011
15
16 === CRC-32  SIN error ===
17 msg=1010 frame=101000110010110101110000011010010011 -> Sin errores. Trama OK.
18 msg=0111 frame=011101001100011001100111101000101110 -> Sin errores. Trama OK.
19 msg=0001 frame=000110100101000001011101111100011011 -> Sin errores. Trama OK.
20
21 === CRC-32  1 error ===
22 err_frame=101010110010110101110000011010010011 -> Error detectado. Se descarta.
23 CRC rcv: 10110010110101110000011010010011 | CRC calc: 00110010110101110000011010010011
24 err_frame=011101001100011001110111101000101110 -> Error detectado. Se descarta.
25 CRC rcv: 01001100011001110111101000101110 | CRC calc: 0100110001100111101000101110
26 err_frame=00111010010100000101110111100011011 -> Error detectado. Se descarta.
27 CRC rcv: 10100101000001011101111100011011 | CRC calc: 01001011000010111011111000110111
28
29 === CRC-32  2+ errores ===
30 err_frame=001000110110110101110000011010010011 -> Error detectado. Se descarta.
31 CRC rcv: 00110110110101110000011010010011 | CRC calc: 00111100000011001000111010100001
32 err_frame=001101001100011001110111101000100110 -> Error detectado. Se descarta.
33 CRC rcv: 01001100011001100111101000100110 | CRC calc: 01001011000010111011111000110111
34 err_frame=000010100101000001011101011100011011 -> Error detectado. Se descarta.
35 CRC rcv: 10100101000001011101011100011011 | CRC calc: 1101001000000101110111110001101
36

```

Figura 1: Salida completa del script de pruebas (tests/run_all.sh).