

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3067 - REDES

Sección 11

MIGUEL NOVELLA LINARES



Laboratorio 2 – Segunda parte

Esquemas de detección y corrección de errores

Ruth de León, 22428
Isabella Miralles, 22293

GUATEMALA, julio del 2025

Corrección de errores

Resultados de forma manual

Receptor

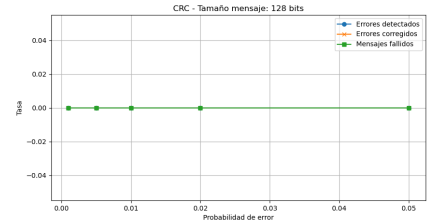
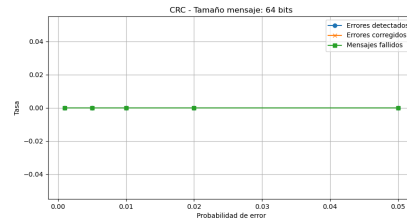
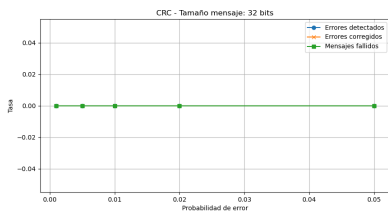
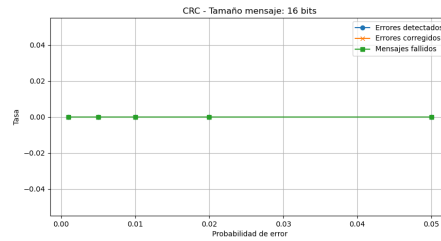
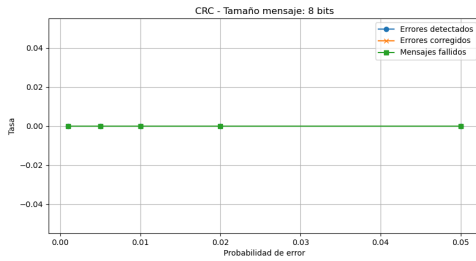
```
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ python3 receptor.py
Receptor escuchando en 127.0.0.1:9090
Trama recibida: hamming:1001100111000011001101111111110011001110011001101101001
Mensaje recibido sin errores: Hola
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ python3 receptor.py
Receptor escuchando en 127.0.0.1:9090
Trama recibida: crc:0100100001101111011011000110000110000000
Error múltiple detectado o no corregible. Mensaje perdido.
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ python3 receptor.py
Receptor escuchando en 127.0.0.1:9090
Trama recibida: crc:010000110110000101110010011100100110111100011110
Error múltiple detectado o no corregible. Mensaje perdido.
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ python3 receptor.py
Receptor escuchando en 127.0.0.1:9090
Trama recibida: hamming:10011001000011110011011010010001111010101000011110101011001101111111
Mensaje recibido sin errores: Carro
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ python3 receptor.py
Receptor escuchando en 127.0.0.1:9090
```

Emisor

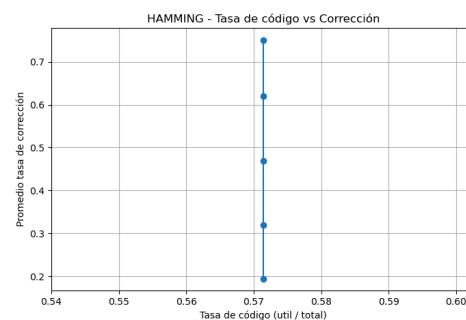
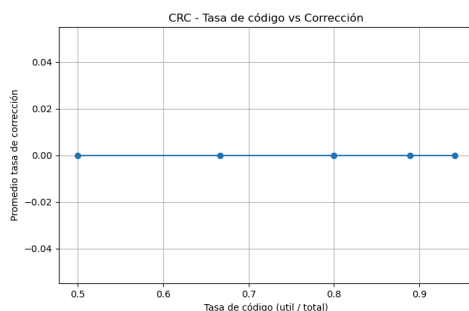
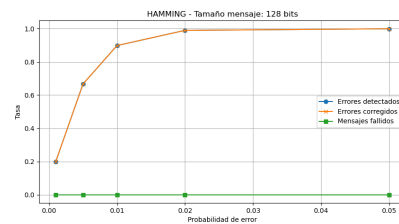
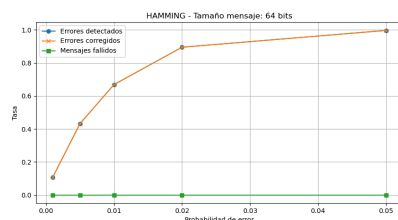
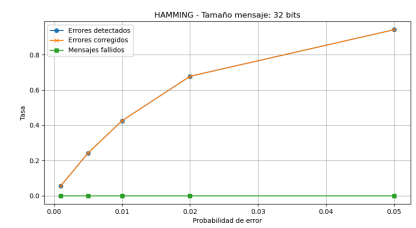
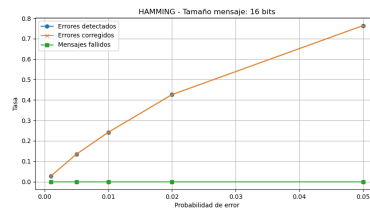
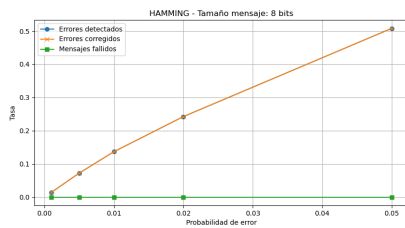
```
des/Correccion$ g++ emisor.cpp -o emisor
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ ./emisor
Ingrese mensaje a enviar: Hola
Seleccione algoritmo para comprobación de integridad (hamming/crc): hamming
Trama sin ruido: 1001100111000011001101111111110011001110011001101101001
Trama con ruido: 100110011100001100110111111111001100111001100110110101001
Mensaje enviado con hamming y posible ruido.
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ ./emisor
Ingrese mensaje a enviar: Hola
Seleccione algoritmo para comprobación de integridad (hamming/crc): crc
Trama sin ruido: 0100100001101111011011000110000110000000
Trama con ruido: 0100100001101111011011000110000110000000
Mensaje enviado con crc y posible ruido.
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ ./emisor
Ingrese mensaje a enviar: Carro
Seleccione algoritmo para comprobación de integridad (hamming/crc): crc
Trama sin ruido: 010000110110000101110010011100100110111100011110
Trama con ruido: 010000110110000101110010011100100110111100011110
Mensaje enviado con crc y posible ruido.
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ ./emisor
Ingrese mensaje a enviar: Carro
Seleccione algoritmo para comprobación de integridad (hamming/crc): hamming
Trama sin ruido: 1001100100001111001101101001000111101010100001111010101011011011111111
Trama con ruido: 1001100100001111001101101001000111101010100001111010101011011011111111
Mensaje enviado con hamming y posible ruido.
```

```
isabella@DESKTOP-MTPG2S8 X + v - □ X
isabella@DESKTOP-MTPG2S8:/mnt/d/Documentos/Octavo Semestre/Redes/Lab2-Redes/Correccion$ ./emisor
Ingrese mensaje a enviar: Carro
Seleccione algoritmo para comprobación de integridad (hamming/crc): hamming
Trama sin ruido: 1001100100001111001101101001000111101010100001111010101011011011111111
Trama con ruido: 10011001000011110011011010010001111010101000011110101010000111101010101101101111111
Mensaje enviado con hamming y posible ruido.
```

Resultado de las pruebas CRC



Hamming



Discusión

Hamming tuvo mejor funcionamiento.

- Detectó y corrigió todos los errores de 1 bit
- La tasa de mensajes fallidos fue de 0 en casi todas las pruebas, esto quiere decir que el receptor siempre recuperó el mensaje original.

CRC detecta los errores pero no corrige ninguno, así que si ocurre un error el mensaje se pierde.

El algoritmo Hamming tiene una mayor flexibilidad para tasas altas de error.

- Tolera errores de un bit por bloque y los corrige de forma automática.
- Mientras la tasa de error sea moderada, la mayoría de los errores serán de un bit por bloque y se pueden recuperar.

Por otro lado, CRC solo detecta los errores, esto quiere decir que si la tasa de error es alta, la mayoría de mensajes se perderán ya que no hay corrección.

Es mejor utilizar un algoritmo de detección cuando

- La tasa de error es baja ya que la retransmisión es rara.
- El sistema permite retransmisión.
- Se prioriza velocidad sobre complejidad, ya que es más rápido y requiere menos bits de redundancia que Hamming.

Al utilizar sólo detección se necesita un protocolo de retransmisión para recuperar mensajes perdidos, en canales con alta latencia o donde es costoso retransmitir, es preferible la corrección para evitar reenvíos. Al no haber retransmisión cada error implica pérdida total del mensaje,

DETECCIÓN

```
PS C:\REDES\Lab2-Redes\Deteccion> py crc32_emisor.py
Ingrese el mensaje a enviar: Te amo Josue

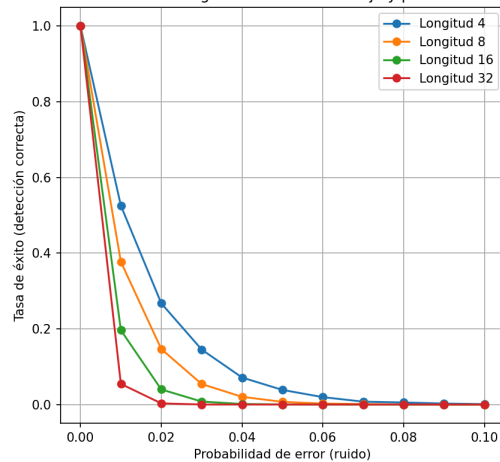
[EMISOR] Trama generada: 01010100011001010010000001100001011011010111100100000010010100110111101110011011101011100101000110111000111100100001110000
[EMISOR] Trama con ruido: 01010100011001010010001001100110101101110010000101001010011011101110110111010001100101000110111000111100100001110000
PS C:\REDES\Lab2-Redes\Deteccion> g++ receptor.cpp -o receptor
PS C:\REDES\Lab2-Redes\Deteccion> ./receptor
Ingrese la trama recibida: 0101010001100101001000000110000101101101011110010000001001010011011110111001101110101100101000110111000111100100001110000
[RECEPTOR] Trama valida. Mensaje: Te amo Josue
PS C:\REDES\Lab2-Redes\Deteccion>
```

Pruebas

```
Longitud 4, Prob. error 0.000, Tasa éxito 1.0000
Longitud 4, Prob. error 0.010, Tasa éxito 0.5255
Longitud 4, Prob. error 0.020, Tasa éxito 0.2674
Longitud 4, Prob. error 0.030, Tasa éxito 0.1453
Longitud 4, Prob. error 0.040, Tasa éxito 0.0714
Longitud 4, Prob. error 0.050, Tasa éxito 0.0386
Longitud 4, Prob. error 0.060, Tasa éxito 0.0196
Longitud 4, Prob. error 0.070, Tasa éxito 0.0076
Longitud 4, Prob. error 0.080, Tasa éxito 0.0055
Longitud 4, Prob. error 0.090, Tasa éxito 0.0027
Longitud 4, Prob. error 0.100, Tasa éxito 0.0011
Longitud 8, Prob. error 0.000, Tasa éxito 1.0000
Longitud 8, Prob. error 0.010, Tasa éxito 0.3770
Longitud 8, Prob. error 0.020, Tasa éxito 0.1467
Longitud 8, Prob. error 0.030, Tasa éxito 0.0542
Longitud 8, Prob. error 0.040, Tasa éxito 0.0204
Longitud 8, Prob. error 0.050, Tasa éxito 0.0071
Longitud 8, Prob. error 0.060, Tasa éxito 0.0026
Longitud 8, Prob. error 0.070, Tasa éxito 0.0013
Longitud 8, Prob. error 0.080, Tasa éxito 0.0003
Longitud 8, Prob. error 0.090, Tasa éxito 0.0000
Longitud 8, Prob. error 0.100, Tasa éxito 0.0000
Longitud 16, Prob. error 0.000, Tasa éxito 1.0000
Longitud 16, Prob. error 0.010, Tasa éxito 0.1963
Longitud 16, Prob. error 0.020, Tasa éxito 0.0399
Longitud 16, Prob. error 0.030, Tasa éxito 0.0079
Longitud 16, Prob. error 0.040, Tasa éxito 0.0015
Longitud 16, Prob. error 0.050, Tasa éxito 0.0003
Longitud 16, Prob. error 0.060, Tasa éxito 0.0000
Longitud 16, Prob. error 0.070, Tasa éxito 0.0000
Longitud 16, Prob. error 0.080, Tasa éxito 0.0000
Longitud 16, Prob. error 0.090, Tasa éxito 0.0000
Longitud 16, Prob. error 0.100, Tasa éxito 0.0000
Longitud 32, Prob. error 0.000, Tasa éxito 1.0000
Longitud 32, Prob. error 0.010, Tasa éxito 0.0545
Longitud 32, Prob. error 0.020, Tasa éxito 0.0031
Longitud 32, Prob. error 0.030, Tasa éxito 0.0001
Longitud 32, Prob. error 0.040, Tasa éxito 0.0000
Longitud 32, Prob. error 0.050, Tasa éxito 0.0000
Longitud 32, Prob. error 0.060, Tasa éxito 0.0000
Longitud 32, Prob. error 0.070, Tasa éxito 0.0000
Longitud 32, Prob. error 0.080, Tasa éxito 0.0000
```

2 0 Live Share

Rendimiento de CRC32 según tamaño de mensaje y probabilidad de error



Descripción

Se desarrolló una simulación para evaluar el rendimiento del algoritmo de detección de errores CRC32 bajo condiciones de transmisión con ruido. La simulación codifica mensajes en binario, les añade un código CRC32 y luego introduce errores aleatorios con una probabilidad configurable. Posteriormente, se verifica si el mensaje recibido mantiene su integridad. Se realizaron 10,000 pruebas por combinación de longitud de mensaje y nivel de ruido, generando datos estadísticos que muestran cómo la efectividad de CRC32 disminuye al aumentar el tamaño del mensaje o la probabilidad de error.