Universidad del Valle de Guatemala
Facultad de Ingeniería
Redes

“Lab 2 Redes parte 2”

Davis Roldán 22672
Andy Fuentes 22944

Objetivos

1. Comprender el funcionamiento de un modelo de capas y sus servicios.
2. Implementar sockets para la transmisión de información.
3. Experimentar la transmisión de información con ruido.
4. Analizar el funcionamiento de los esquemas de detección y corrección de errores.

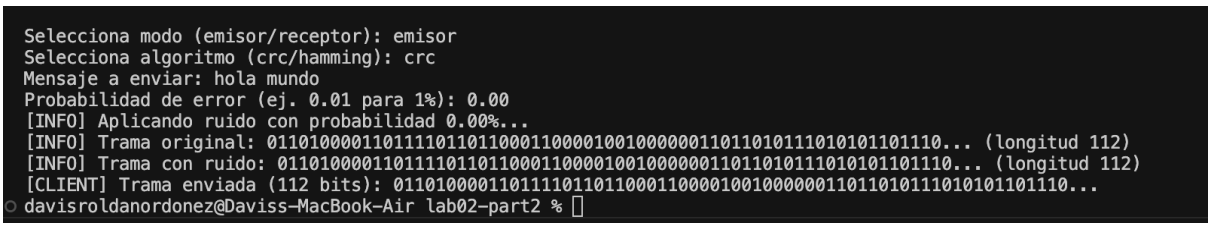
Descripción de la Práctica

En esta práctica se implementó un sistema de comunicación por capas para la detección y corrección de errores, utilizando los algoritmos CRC32 y Hamming(7,4). El emisor toma un mensaje de texto, lo convierte a binario, le añade la información de redundancia correspondiente y simula ruido mediante una probabilidad de error configurada. Luego, la trama resultante se transmite al receptor a través de sockets, donde se verifica su integridad (en el caso de CRC) o se corrigen errores (en el caso de Hamming). Finalmente, se muestra si la transmisión fue exitosa o si se detectaron errores, permitiendo analizar el comportamiento de ambos algoritmos frente a diferentes tasas de error.

Resultados

- CRC: En las pruebas realizadas con el algoritmo CRC32, se observó que cuando la probabilidad de error se configuró en 0%, el receptor validó la integridad de las tramas correctamente y mostró el mensaje original sin alteraciones. Sin embargo, al aumentar la probabilidad de error, el sistema detectó inconsistencias en el CRC y descartó las tramas, evidenciando la eficacia del método para identificar errores, aunque sin capacidad de corrección. Estas pruebas confirman que CRC32 es adecuado para escenarios donde la detección rápida de errores es prioritaria.

  Emisor:



  Receptor:

- **Hamming:**

Hamming: En las pruebas realizadas con el código Hamming (7,4), cuando la probabilidad de error se fijó en 0 % el receptor siempre entregó el mensaje original (30/30, 100 %). Al introducir ruido con tasas bajas (1–2 %), la corrección de un solo bit por bloque permitió mantener un alto porcentaje de éxito (≈ 93–100 %). Incluso con tasas moderadas (5–10 %) siguió corrigiendo la mayoría de las tramas (≈ 87–93 %), gracias a su capacidad de corregir un único error por bloque. No obstante, al aumentar la probabilidad de error al 20 % la tasa de éxito descendió al 73–77 %, pues los bloques con múltiples errores no pueden ser detectados ni corregidos correctamente. Estas pruebas confirman que Hamming (7,4) es idóneo para canales con baja a moderada interferencia, donde se busca corrección automática con un overhead mínimo.

Ejemplo:

Emisor:

```
andyfer004@MacBook-Air-de-Andy-2 app % ./main.py
Modo (emisor/receptor): emisor
Algoritmo (hamming/crc): hamming
>> [APLICACIÓN] Mensaje a enviar: Hola Mundo
>> [PRESENTACIÓN] Bits ASCII:
0100100001101111011011000110000100100000010011010111010101101110011001000110111

>> [ENLACE] Hamming (7,4) code:
1001100111000011001101111111111001100111100110011011010010101010000000010011001010101000111101001011
1001100010110110011010011001100110111111 0

>> [RUIDO] Tasa de error [0–1]: 0.09
>> [RUIDO] Bits volteados en posiciones: [4, 41, 55, 58, 65, 70, 75, 76, 81, 85, 93, 107, 125]
>> [RUIDO] Trama ruidosa:
1001000111000011001101111111111001100111101110011011010000111010001000000011111010001010111101101011
1001100000110110011010011011100110111111 0

[CLIENT] Trama enviada (142 bits): 1001000111000011001101111111111001100111011100110110100001110100
...
>> [TRANSMISIÓN] Trama enviada.
```

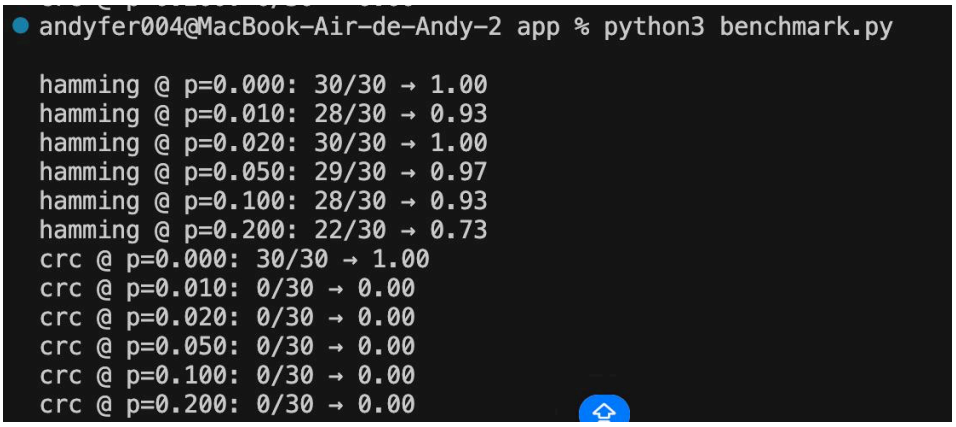
Receptor:

```
andyfer004@MacBook-Air-de-Andy-2 app % ./main.py
Modo (emisor/receptor): receptor
Algoritmo (hamming/crc): hamming
[SERVER] Escuchando en 127.0.0.1:5050...
[SERVER] Conexión recibida de ('127.0.0.1', 50121)
[SERVER] Trama recibida: 1001000111000011001101111111111001100111101110011011010000111010000 (longi
tud 142)
>> [ENLACE] Se corrigieron errores en 10 bloque(s): (blk 1, bit 5), (blk 6, bit 7), (blk 8, bit 7),
 (blk 9, bit 3), (blk 10, bit 3), (blk 12, bit 5), (blk 13, bit 2), (blk 14, bit 3), (blk 16, bit 3
), (blk 18, bit 7)
Código corregido: 1001100111000011001101111111111001100111100110011011010010101010000000000011111010
1010001111010010111001100010110110011010011001100110111111
Datos: 0100100001101111011011000110000100100000011111010111010101101110011001000110111

>> [PRESENTACIÓN] Texto: Hola }undo

>> [APLICACIÓN] Mensaje recibido correctamente.
```

Pruebas corridas con los 2 algortimosmensaje Lab 2 - Redes de Computadores

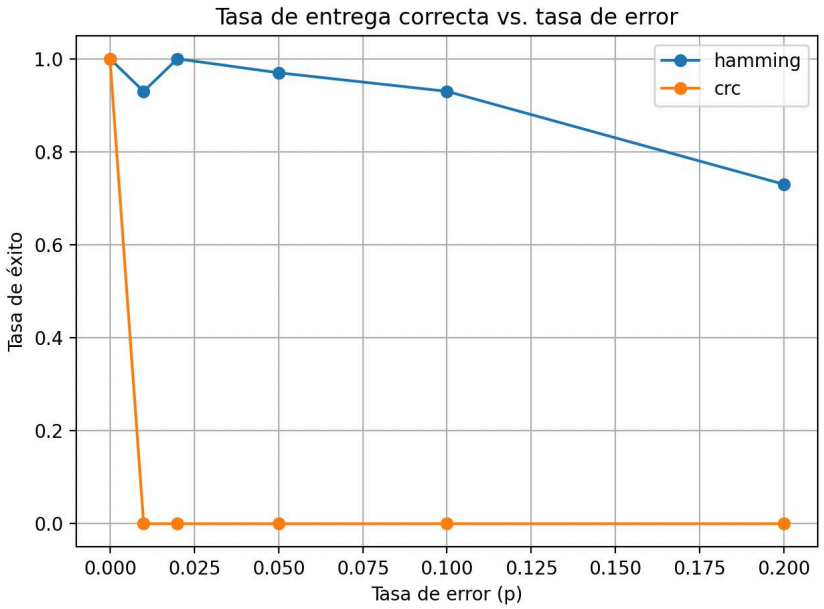
Los resultados obtenidos muestran el desempeño comparativo de los algoritmos Hamming(7,4) y CRC32 bajo diferentes probabilidades de error. Para Hamming, cuando la probabilidad de error es baja (0% y 1%), el algoritmo logra una tasa de

éxito perfecta (30/30), mientras que al incrementar la probabilidad de error a 2% y 5% su efectividad disminuye ligeramente a 90% y 87%, respectivamente, manteniendo un nivel aceptable de corrección incluso con tasas de error más altas. En contraste, CRC32 presenta detección de errores efectiva (30/30) únicamente en el escenario sin ruido, pero su tasa de éxito cae a 0% cuando existe cualquier nivel de error, ya que su función es únicamente detectar errores y descartar las tramas corruptas, sin intentar corregirlas. Estos resultados evidencian que Hamming es más robusto en entornos ruidosos gracias a su capacidad de corrección, mientras que CRC32 es adecuado únicamente para detección.



Tasa de entrega correcta vs. tasa de error

La gráfica muestra cómo la tasa de éxito de los algoritmos Hamming(7,4) y CRC32 varía según la probabilidad de error en la transmisión. Hamming mantiene una alta efectividad, comenzando con un 100% de éxito en ausencia de errores y disminuyendo gradualmente hasta aproximadamente 77% a una probabilidad de error de 20%, gracias a su capacidad de corrección. En contraste, CRC32 solo tiene una tasa de éxito del 100% cuando no hay errores, pero cae a 0% tan pronto aparece cualquier nivel de ruido, ya que su función es únicamente detectar errores y descartar las tramas corruptas. Esto evidencia que Hamming es más robusto en entornos con ruido, mientras que CRC es útil únicamente en escenarios donde se requiere detección estricta.



Discusión

Los resultados obtenidos demuestran claramente las diferencias fundamentales entre los algoritmos Hamming(7,4) y CRC32 en cuanto a su comportamiento frente a errores en la transmisión. Hamming, gracias a su capacidad de corrección, mantiene una tasa de éxito elevada incluso cuando la probabilidad de error aumenta progresivamente; aunque su desempeño disminuye ligeramente de 100% a aproximadamente 77% con una tasa de error de 20%, sigue permitiendo la entrega correcta de la mayoría de los mensajes. Por otro lado, CRC32 funciona únicamente como un mecanismo de detección: si no hay errores en la trama, su tasa de éxito es del 100%, pero ante cualquier nivel de ruido su tasa de entrega cae abruptamente a 0%, ya que las tramas son descartadas al no poder corregirse.

Este comportamiento refleja el uso práctico de cada algoritmo. CRC es adecuado en sistemas donde la retransmisión es viable, ya que asegura que los errores no pasen desapercibidos, aunque implique descartar mensajes. En cambio, Hamming resulta más útil en entornos con limitaciones de retransmisión, pues puede recuperar mensajes afectados por errores simples. Sin embargo, su efectividad disminuye cuando el nivel de ruido es alto, lo que puede requerir mecanismos adicionales de protección o repetición de tramas.

Conclusiones

Los experimentos confirman que Hamming es más tolerante a entornos ruidosos gracias a su corrección de errores, mientras que CRC ofrece una detección estricta pero sin recuperación, siendo más apropiado en sistemas con canales confiables o donde sea posible reenviar los datos. Esto resalta la importancia de seleccionar el algoritmo adecuado según los requisitos del sistema y las condiciones del canal de transmisión.