

Laboratorio 2: Parte 2

Esquemas de detección y corrección de errores

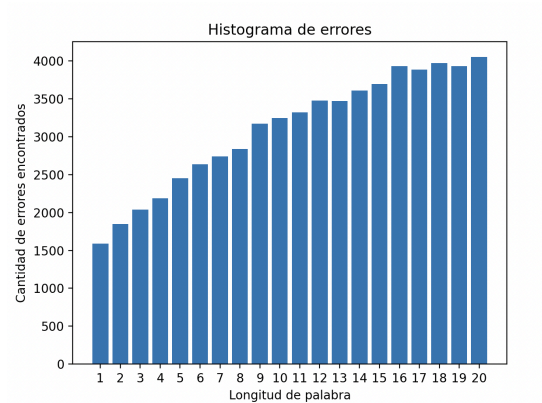
Descripción de la práctica

Con la finalidad de comprender el funcionamiento de un modelo de capas y sus servicios, esta práctica busca incorporar los algoritmos de detección y corrección, implementando su uso y funcionamiento mediante el modelo. Para ello, se crearon un emisor y un receptor con todas sus capas, los cuales se comunicaron mediante el uso de sockets. El programa del emisor y el receptor fueron escritos en lenguajes de programación distintos, para evidenciar la flexibilidad del uso de sockets. Además, se implementó una “capa” adicional que simuló el ruido o interferencia.

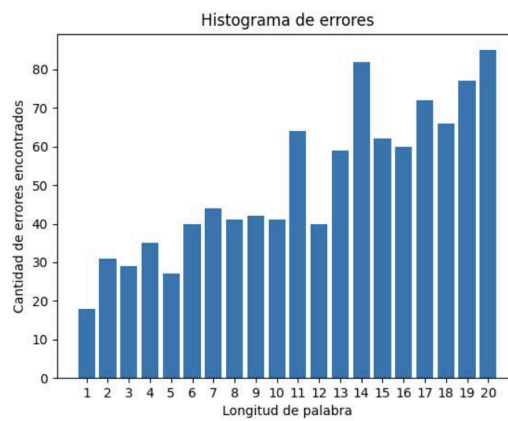
El emisor aplicará el ruido a la trama de datos antes de enviarla, y el receptor verificará y corregirá los errores, si el algoritmo seleccionado aplica. Una vez terminados el receptor y el emisor, se realizaron múltiples pruebas de emisión y recepción, considerando que la “capa” de ruido aplicaría una pequeña probabilidad de voltear un bit en cada bit de la trama. Para el envío de mensajes, se generaron 10 mil palabras que se enviaron una por una, y se almacenaron en un archivo de texto, para utilizarlas posteriormente. Durante este envío, el receptor tiene un módulo de estadísticas que se puede llamar al finalizar el envío para obtener un análisis de lo recibido.

Resultados

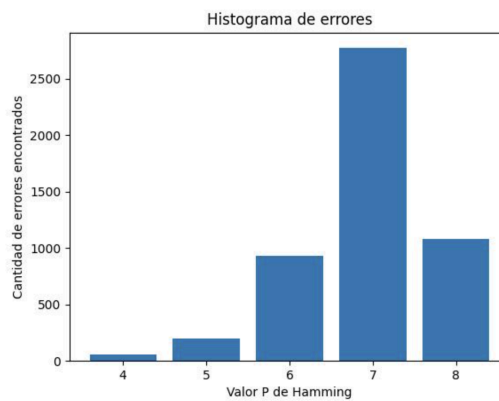
	CRC-32	Hamming
Porcentaje de palabras sin error	34.82%	53.22%
Porcentaje de palabras corregidas	No aplica	63.44%



Gráfica 1: Histograma CRC-32 de errores por longitud de palabra para 10,000 palabras, con el 0.01 de probabilidad de error



Gráfica 2: Histograma Hamming de errores por longitud de palabra para 10,000 palabras, con el 0.01 de probabilidad de error



Gráfica 2: Histograma Hamming de errores P para 10,000 palabras, con el 0.01 de probabilidad de error

Discusión

Con la finalidad de presentar y medir el rendimiento de los algoritmos, se implementaron medidas estadísticas sobre los resultados. Respecto a la medida de accuracy, CRC-32 obtuvo un 34.82% de tramas las cuales estaban posiblemente correctas. Por otro lado, Hamming obtuvo un 53.22%. En ambos casos, se utilizó el mismo archivo de texto con palabras generadas de manera aleatoria, dándonos a entender que, la exactitud depende directamente de la implementación del algoritmo. En el caso de CRC-32, tiene sentido que tenga un porcentaje menor de exactitud, ya que este agrega 32 bits adicionales, los cuales pueden sufrir de algún flip durante la ejecución.

Al observar los gráficos 1 y 2, los histogramas representan la frecuencia de palabras con errores según su longitud. CRC-32 demuestra 4000 errores en tramas de longitud 20, mientras que Hamming presenta solamente 80 errores, en tramas de la misma longitud. Asimismo, revisando la tendencia de las gráficas, podemos ver que la cantidad de errores depende de la longitud de la trama utilizada, ya que mientras más grande la trama, más espacio para hacer flip existe; y esto ocurre en ambos algoritmos.

Respecto a la cantidad de errores corregidos, podemos ver que Hamming corrige hasta un 63.44% de los casos. Este porcentaje representa para el emisor el no tener que enviar nuevamente el paquete de datos, siendo un gran beneficio para el algoritmo, ya que permite una transmisión de datos mucho más rápida. También hay que considerar que la probabilidad de error es baja, ya que solamente es del 0.01%, siendo 1 error por cada 100 bits.

Siguiendo con Hamming, el valor p para los bits de paridad, con más errores es de 7. Esto hace sentido ya que al ser un valor de p más alto, representa que habrá más bits de paridad, y por consecuencia, la trama será más larga. Y como se mencionó anteriormente, entre más larga la trama, mayor probabilidad de que existan errores de flip.

Conclusiones

- Hamming presenta un porcentaje de corrección de errores aceptable, para una tasa de error de uno sobre cien bits, haciéndolo confiable para ocasiones en donde ocurre solamente un flip en la trama.
- CRC-32 tiene una desventaja evidente al agregar 32 bits a la trama original. Haciendo el algoritmo altamente propenso a ocurrencia de flips.
- La baja probabilidad de error utilizada en las pruebas (0.01%) permitió observar cómo los algoritmos se comportan bajo condiciones controladas. A pesar de la baja probabilidad de error, la longitud de la trama siguió siendo un factor crucial en la ocurrencia de errores.

Referencias

- Grover, A., & Singh, S. (2015). Comparative Analysis of CRC-32 and SHA-1 Algorithms in WEP. Advanced Engineering Technology and Application, 4(1), 5-10.
- Griffiths, G., & Stones, G. C. (1987). The tea-leaf reader algorithm: an efficient implementation of CRC-16 and CRC-32. Communications of the ACM, 30(7), 617-620.

Anexos

Repositorio: <https://github.com/DanielMorales1103/Lab2Redes/tree/Parte2>