

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3067 Redes

Sección 10

Ing. Jorge Yass



Laboratorio 2 - Parte 2

Esquemas de detección y corrección de errores

SEBASTIAN ARISTONDO PEREZ 20880

JOSE DANIEL GONZALEZ CARRILLO 20293

GUATEMALA, 10 de agosto de 2023

Descripción y Metodología

Con el objetivo de comprender el funcionamiento de un modelo de capas y sus servicios esta práctica buscaba incorporar los algoritmos de detección y corrección, implementando su uso y funcionamiento por medio del modelo. Para ello se crearon un emisor y un receptor con todas sus capas, estos 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 la cual simuló el ruido o interferencia.

Una vez el receptor y el emisor estaban terminados era necesario realizar múltiples pruebas de emisión y recepción, tomando en cuenta que la “capa” de ruido aplicaría una pequeña probabilidad de flippear un bit a cada bit de la trama. Para el envío de mensajes se generaron 10 mil palabras las cuales fueron enviadas 1 por 1, las cuales se almacenaron en un archivo de texto como el conjunto de datos a utilizar. Durante este envío el receptor tiene un módulo de estadísticas el cual se puede llamar al finalizar el envío para obtener un análisis de lo que recibió.

La capa más alta fue la de aplicación, la cual solicitó los mensajes que se debían mandar del lado del emisor y mostró los mensajes recibidos del lado del receptor. Por debajo se usó una capa de presentación, la cual se encargó de codificar los mensajes de ASCII a una secuencia de bits del lado del emisor y de decodificar bits a ASCII del lado del receptor. Luego se realizó la capa de enlace, la cual implementó los algoritmos usados para corrección y detección de errores. Finalmente se usó una capa de transmisión para enviar los bits entre receptor y emisor. Además, se usó la capa de ruido, como se mencionó anteriormente.

Resultados

	Porcentaje de palabras sin error	Porcentaje de palabras bien corregidas
CRC	38.82%	NA
Hamming	52.66%	63.46%

Cuadro 1. Porcentajes de errores detectados y corregidos para CRC y Hamming con 10k palabras, con 0.01 de probabilidad de error.

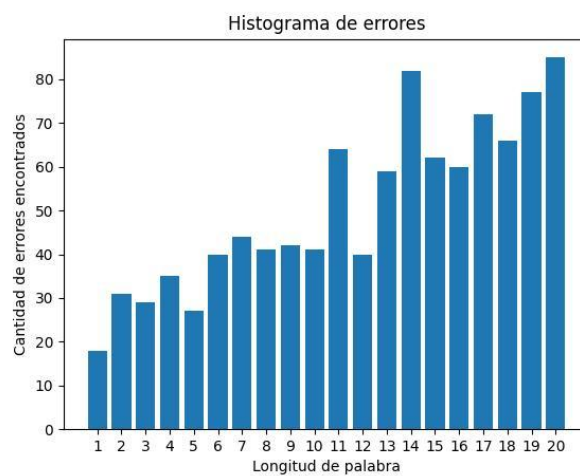


Gráfico 1. Histograma de errores por longitud de palabra para 10k palabras con 0.01 de probabilidad de error usando CRC.

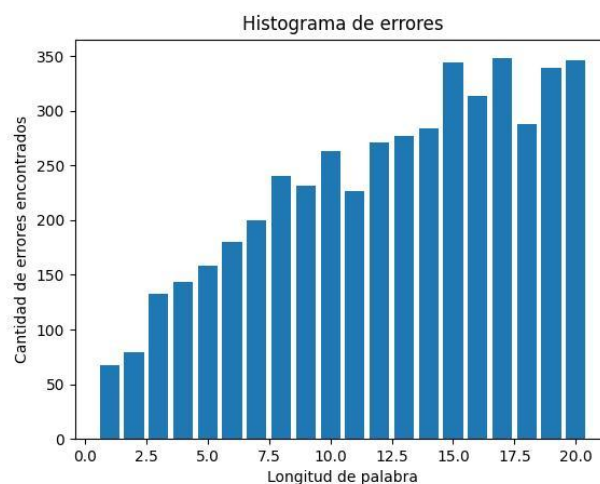


Gráfico 2. Histograma de errores por longitud de palabra para 10k palabras con 0.01 de probabilidad de error usando Hamming.

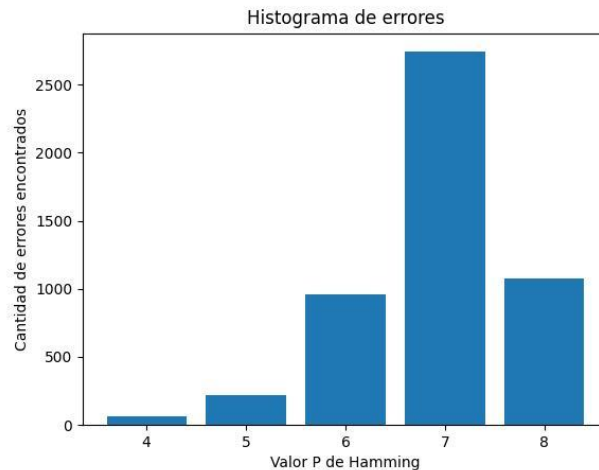


Gráfico 3. Histograma de errores por valor P con 10k palabras con 0.01 de probabilidad de error para Hamming.

Discusión

Con el objetivo de medir el rendimiento de los algoritmos de detección y corrección, se implementó un conjunto de parámetro para generar las estadísticas de los datos descifrados. Una de las estadísticas fue la exactitud, CRC-32 obtuvo 38.82% de tramas posiblemente íntegras, mientras que Hamming obtuvo 52.66%. El conjunto de datos a utilizar fue exactamente el mismo para ambos algoritmos, lo que nos permite concluir que la exactitud de cada uno se debe a su implementación. Ya que CRC-32 añade 32 bits adicionales al mensaje significa que nuestra “capa” de ruido tendrá 32 bits adicionales a los cuales aplicarles una pequeña probabilidad de flip. Mientras que en Hamming únicamente se añaden p bits. Claramente esta diferencia en longitud de caracteres tiene un gran impacto en la cantidad de tramas que son enviadas sin ser alteradas.

Estos datos se pueden ver respaldados en los gráficos 1 y 2 del apartado de resultados, estos histogramas nos dan la frecuencia de palabras con errores dada su longitud. CRC-32 presenta 350 errores en tramas de longitud 20 mientras que Hamming únicamente presenta 80. Por último para demostrar que la longitud de la trama tiene una alta influencia en la probabilidad de ser modificada, podemos comparar que CRC-32 en tramas de longitud 1 se reportaron 53 errores si mientras como se menciona anteriormente Hamming con tramas de longitud 20 presenta 80 errores, únicamente 30 errores de diferencia entre la menor longitud de CRC-32 y la mayor longitud de Hamming.

Al observar el cuadro 1, se puede observar que Hamming corrige el 63.46% de los errores de forma adecuada. Esto lo realiza con un 0.01% de probabilidad, la cual es relativamente baja para el ruido aplicado a cada bit enviado. En general el porcentaje es bastante bueno, lograr corregir errores en más del 60% de los datos enviados representa no tener que pedir al emisor que vuelva a enviar el paquete. Esto es un gran beneficio para lograr el principal objetivo de Hamming, que es poder agilizar la transmisión de datos. Para haber usado la implementación más básica de Hamming, dió un resultado bueno. Sin embargo, esto se podría mejorar usando una implementación más avanzada del algoritmo, como Hamming 7, 4, que segmenta el mensaje y agrega bits de paridad a cada segmento. (Invarato, 2016; Wright, 2022)

Se puede observar del gráfico 3, el valor p usado para los bits de paridad que más errores tiene es el 7. Esto es un resultado esperado, puesto que un p más alto indica que habrá más bits de paridad y que el mensaje enviado tiene una longitud más larga. Entonces, por esta longitud, existe mayor probabilidad de que algún bit se de vuelta. Es importante evidenciar que a pesar de esto, el valor p 8 tiene menos errores que el 7, pero esto simplemente se debe a que existen menos cadenas en el conjunto de datos con una longitud que requiera este valor p.

Finalmente, se puede decir del modelo de capas que fue bastante eficiente al momento de desarrollar los distintos módulos. Esto debido a que fue posible desarrollar los dos algoritmos, ocultando su implementación de las diferentes capas, y pudiéndolos intercambiar cuando fuera necesario. Si se hubieran desarrollado más algoritmos de detección y/o corrección de errores, sería sencillo comunicar las capas, por la abstracción del modelo, facilitando múltiples versiones de este. Además, si se quisieran realizar modificaciones en la transmisión, por ejemplo, no usar sockets si no que una pipe o message queue, esto podría ser realizado fácilmente sin afectar a todo el sistema.

Comentario grupal

Introducir la “capa” de ruido ayudó a dimensionar las dificultades que pueden tener los algoritmos de detección y corrección. Cuando introducimos los errores de forma manual no logramos entender en su totalidad la alta probabilidad de que un mensaje al llegar no fuera íntegro. En cambio ahora que el ruido puede o no ocasionar un flip, por muy baja que sea la probabilidad, mientras más larga sea la trama, es más probable que esta sufra de una alteración. Estas conclusiones se lograron alcanzar gracias a los requerimientos de estadísticas de parte del receptor, como parejas pensábamos que esta sección no era muy útil, pero tras evaluar y discutir lo que encontramos logramos asimilar de mejor manera la práctica y el contenido.

Conclusiones

- CRC-32 presentó un peor rendimiento debido que al tener una trama de mayor longitud, esta era más propensa a sufrir flips
- Se evidenció que el uso del algoritmo básico de Hamming da una tasa regular de corrección de errores, sin embargo, es recomendable usar una versión más compleja para mejorar el rendimiento como Hamming(7,4).<
- Un modelo de capas permite abstraer funcionalidades, haciendo que sea más sencilla y flexible la comunicación y mantenimiento del modelo utilizado en la comunicación de mensajes.

Citas y Referencias

Invarato, R. (2016, October 5). *Código de Hamming: Detección y Corrección de errores*. Jarroba.

<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/>

Wright, G. (2022, July). *What is Hamming code and how does it work?* WhatIs.com.

<https://www.techtarget.com/whatis/definition/Hamming-code#:~:text=Hamming%20code%20is%20an%20error>