

## **Laboratorio 2.2 - Esquemas de detección y corrección de errores**

### **Descripción de la práctica**

En esta práctica se desarrolló una aplicación que transmite y recibe mensajes o “tramas” utilizando como medio de transmisión un servidor de sockets. El objetivo es visualizar la importancia que tiene el ruido que existe en el medio de transporte. Las tramas fueron realizadas con los dos algoritmos que se utilizaron en la práctica anterior: *Hamming* y *Fletcher*. La metodología consistió en programar un cliente encargado de convertir 10,000 palabras (ASCII) a una trama utilizando el algoritmo seleccionado. Posteriormente se añadió una “capa de ruido” para simular las condiciones aleatorias que pueden suceder en una aplicación real. Por cada algoritmo se realizaron tres pruebas, modificando en cada una la probabilidad de que la trama sea afectada por ruido. Los valores utilizados fueron: 1/100 bits, 2/100 bits y 10/100 bits. Posteriormente se realiza un análisis gráfico de los resultados y se establecen conclusiones.

## Resultados

### Código Hamming

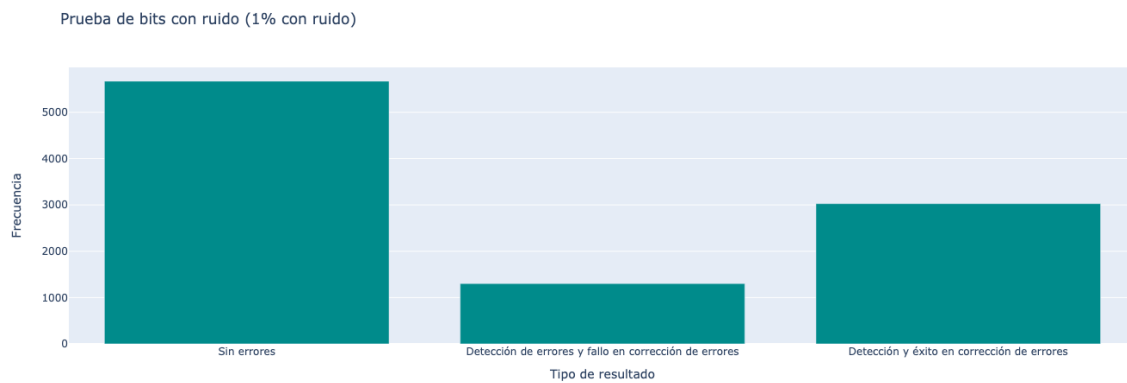


Figura 1 - Resultados algoritmo Hamming con 1% de ruido

La mayoría de las palabras fueron transmitidas sin errores, y de las que tuvieron errores aproximadamente un 70% tuvieron éxito en la corrección de errores.

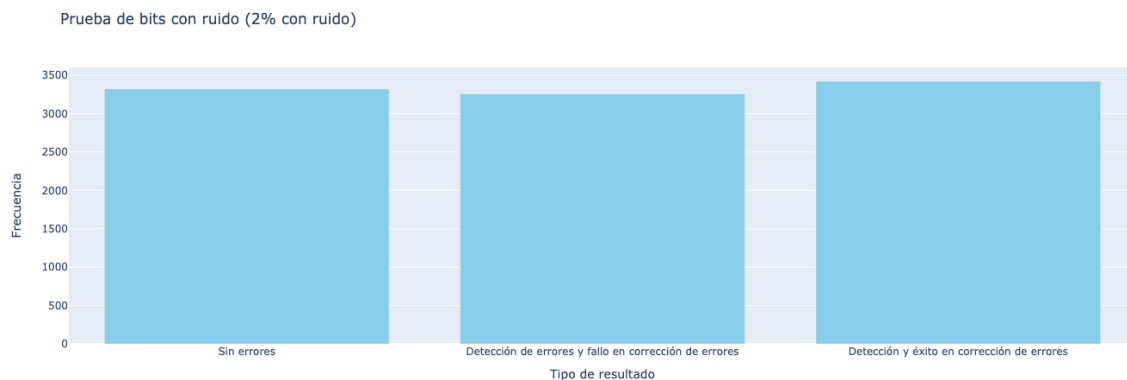


Figura 2 - Resultados algoritmo Hamming con 2% de ruido

La proporción de errores aumentó, provocando una reducción en las palabras sin errores a un 30%. Poco más del 50% de las palabras que tuvieron errores lograron ser corregidas por el algoritmo.

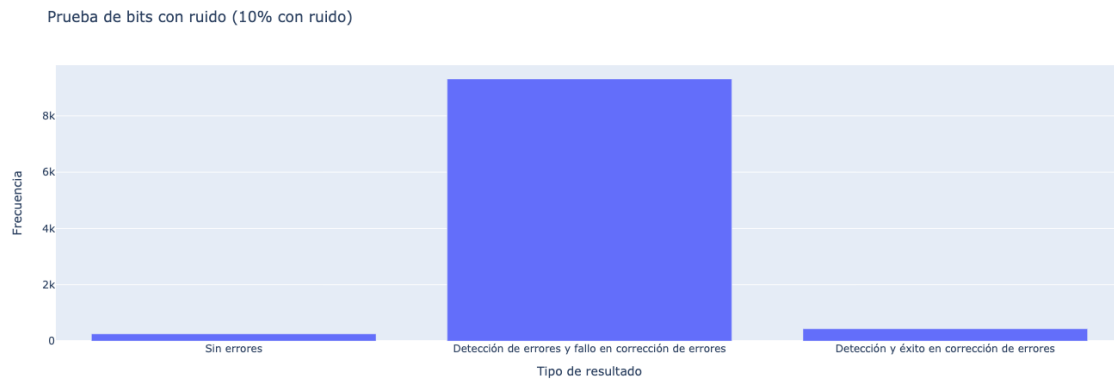


Figura 3 - Resultados algoritmo Hamming con 10% de ruido

La mayoría de palabras, casi un 95%, tuvo errores sin éxito de corrección.

### Algoritmo Fletcher

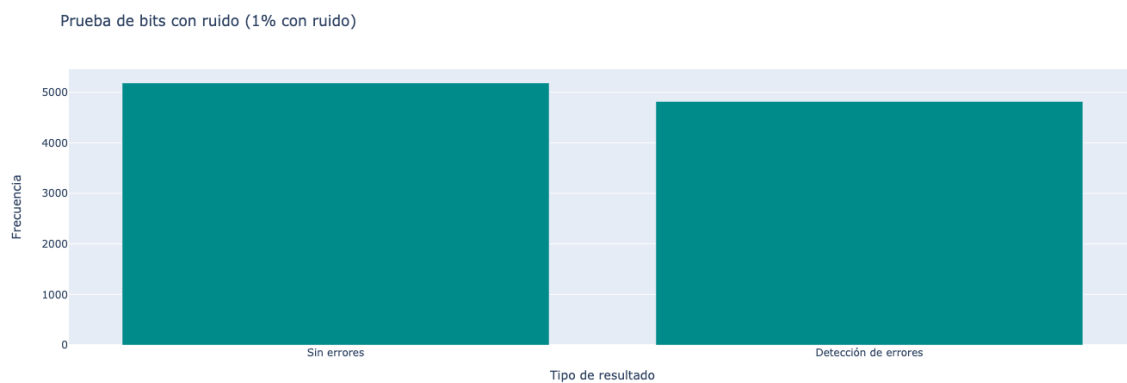


Figura 4 - Resultados algoritmo Fletcher16 con 10% de ruido

48% de las palabras fueron afectadas por el ruido.

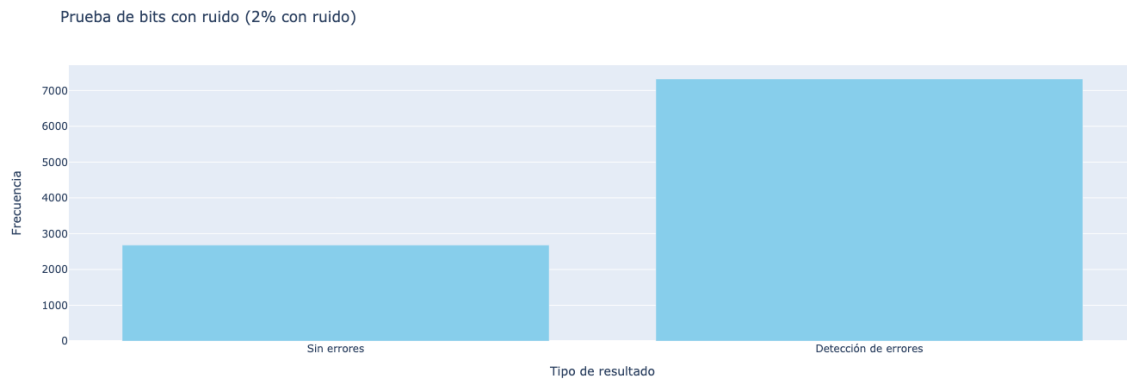


Figura 5 - Resultados algoritmo Fletcher16 con 2% de ruido

Únicamente un 27% de las palabras fueron enviadas satisfactoriamente.

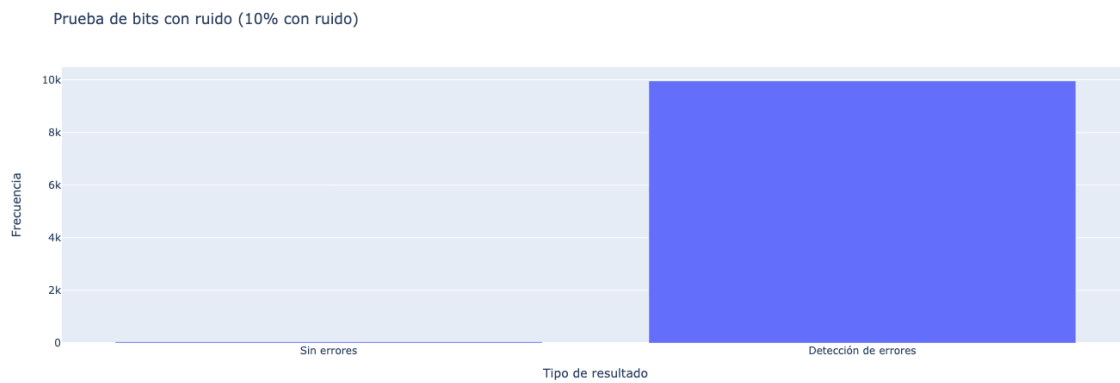


Figura 6 - Resultados algoritmo Fletcher16 con 10% de ruido

Menos del 1% de las palabras llegaron sin errores al servidor. Precisamente, solamente 45 de 10,000 llegaron sin error alguno.

## Discusión

El incremento de la probabilidad de ruido evidencia cómo el algoritmo de *Hamming* se vuelve cada vez menos efectivo para corregir errores dado que sólo puede detectar exitosamente un error específico. Sin embargo, puede seguir siendo utilizado como algoritmo de detección de errores. Asimismo se logró observar que, mientras más largo es el mensaje, mayor es la probabilidad de que este sea afectado por el ruido del medio. Por lo tanto, este algoritmo pierde su uso con estos mensajes y sería suficiente utilizar otro que sea únicamente para detectar errores, como lo es *Fletcher*. La versión utilizada para las pruebas fué *Fletcher 16*, la cuál agrega una *checksum* de 16 bits al final de la cadena. Estas cadenas fueron más propensas al ruido debido al tamaño de la trama una vez se agregaran los bits al final. Por ejemplo, en la primera prueba (Figura 4) se muestra que el número de palabras con errores y sin ellos es casi el mismo. Sin embargo, si el medio es suficientemente confiable este algoritmo puede ser utilizado únicamente para validar si una trama es correcta y de lo contrario solicitar que se vuelva a enviar.

## **Comentario Grupal**

El aprendizaje obtenido en esta práctica es valioso para nosotros dado que en la práctica siempre asumimos que el medio en donde enviamos nuestra información es “perfecto”. Sin embargo, esto se logra gracias al esfuerzo que realizan las capas de red en asegurar la integridad de un paquete que es enviado por el medio. Asimismo consideramos que es importante saber estos conceptos dado que nos podemos presentar con estas situaciones al realizar proyectos que requieran una seguridad de la información bastante fuerte o en industrias en donde es sumamente importante la integridad de los mensajes, por ejemplo: instituciones bancarias o públicas.

## **Conclusiones**

- Las capas deben de considerar los efectos que tiene el ruido del medio y llevar acciones requeridas para manejar esta situación.
- Los algoritmos de corrección son eficientes al tratarse de cadenas pequeñas dado que hay menos probabilidad de que un bit sea afectado por el ruido.
- El algoritmo de Hamming es más eficaz no solamente por ser capaz de corregir errores si no que también por lograr detectar más errores que el algoritmo de fletcher.

## **Referencias bibliográficas**

- Tanenbaum, A. S., & Wetherall, D. J. (2010). Computer Networks (5th ed.). Pearson.
- [https://www.youtube.com/watch?v=gQK9nROFX20&t=375s&ab\\_channel=Universitat\\_Polit%C3%A8cnica\\_de\\_Val%C3%A8ncia-UPV](https://www.youtube.com/watch?v=gQK9nROFX20&t=375s&ab_channel=Universitat_Polit%C3%A8cnica_de_Val%C3%A8ncia-UPV)
- [https://en.wikipedia.org/wiki/Fletcher%27s\\_checksum](https://en.wikipedia.org/wiki/Fletcher%27s_checksum)