

Esquemas de detección y corrección de errores

Descripción de la práctica

El objetivo principal de esta práctica era comprender el funcionamiento de un modelo de capas y sus servicios, implementando sockets para la comunicación. Para la elaboración de dicha práctica se utilizaron algoritmos realizados previamente para la detección y corrección de errores, los cuales eran CRC-32 para detección y Hamming para corrección. Con estos algoritmos ya preparados se estableció un modelo de capas las cuales eran: Aplicación, Enlace, Transmisión y Presentación, tanto para el receptor como para el emisor. La única diferencia existente era que el emisor contaba con una capa de ruido para modificar el mensaje.

Por parte del emisor la capa de aplicación solicitaba la cadena en texto plano que se quería enviar y con qué algoritmo se desea codificar, sea CRC-32 o Hamming. Esta información se enviaba a la capa de presentación donde el mensaje se codifica de texto plano a ASCII para tener la cadena en binario. El mensaje ya en binario pasaba a la capa de enlace donde se realizaba la codificación con el algoritmo establecido en la capa de aplicación. El mensaje codificado atravesaba la capa de ruido en donde cada uno de los bits era propenso a ser flipado para modificar el significado del mensaje original. Por último, el mensaje era enviado al receptor por la capa de transmisión a través de un socket apuntando al puerto establecido por el receptor.

En el receptor, la capa de transmisión recibía el mensaje del emisor y era enviado a la capa de enlace donde se verificaba si el mensaje contenía errores y de ser posible arreglos dichos errores. Si el mensaje no contiene errores se decodifica en la capa de presentación con el algoritmo que indica el emisor. Como último paso, en la capa de aplicación se muestra si el mensaje tenía un error en caso de presentarse o se imprimía el mensaje original.

Finalmente, se realizaron diversas pruebas con cadenas de mensajes generadas aleatoriamente, con el algoritmo seleccionado aleatoriamente para estudiar el rendimiento de dichos algoritmos cuando se transmiten grandes cantidades de información la cual puede contener errores.

Resultados

Figura 1. Comparación entre aciertos y errores de los algoritmos Hamming y CRC-32

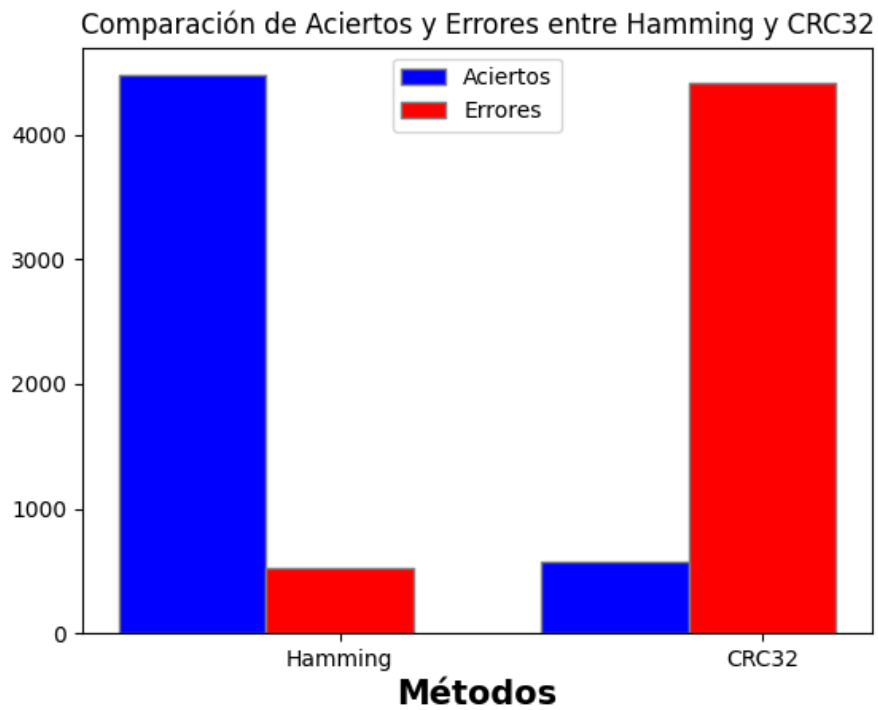


Figura 2. Errores en el algoritmo de Hamming vs Longitud de la cadena

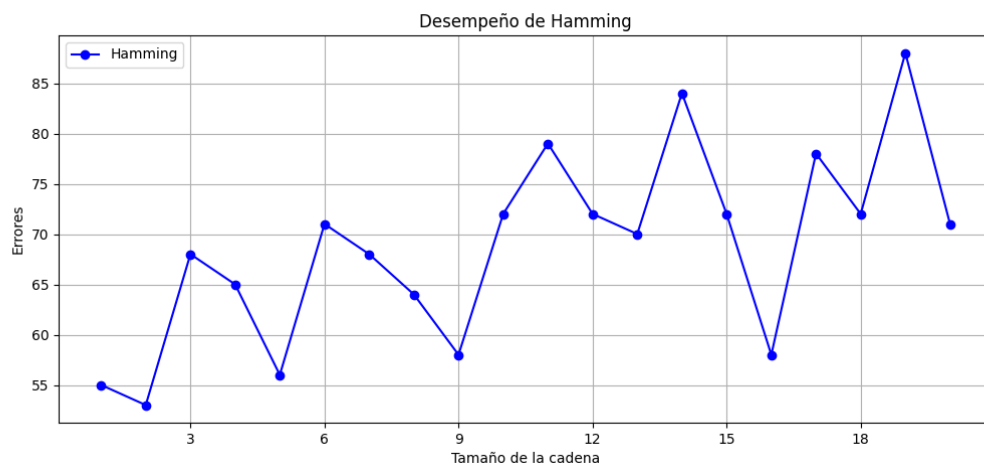
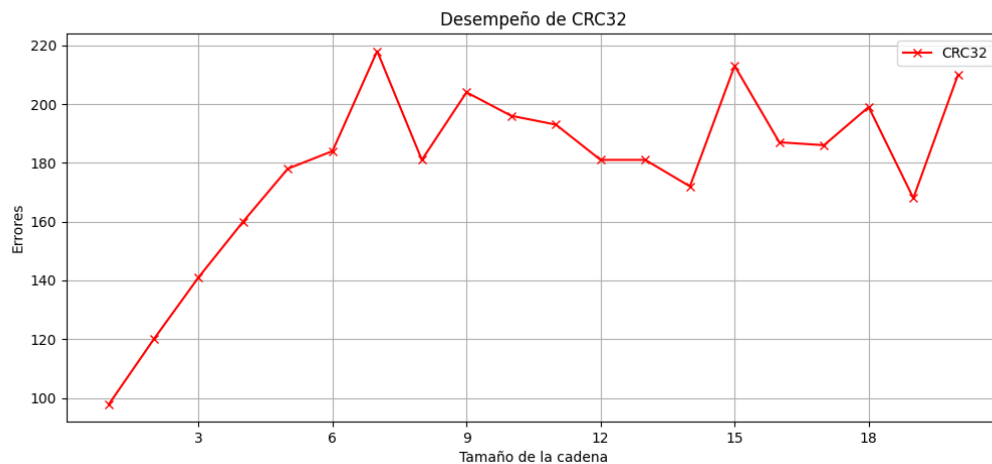


Figura 3. Errores en el algoritmo de Hamming vs Longitud de la cadena



Discusión

Al observar la figura 1, es posible determinar que el algoritmo que mejor desempeño tuvo fue el de Hamming. Este algoritmo destacó mucho más que CRC-32 debido a su capacidad de corregir errores. Esta ventaja se limita únicamente a un bit, sin embargo esto hace que pueda presentar menos errores. Por otro lado, CRC-32 es más susceptible a lanzar errores ya que es solo un algoritmo de detección. Como este algoritmo no puede realizar ningún cambio en sus bits, cualquier modificación causará un error. Es por esto que la cantidad de errores presentada por este algoritmo es bastante alta.

A pesar de que Hamming contenga la capacidad de reconocer y corregir errores, siempre está la posibilidad de aceptar falsos positivos. Estos falsos positivos pueden provocarse al cambiarse los bits de una manera que no sea posible detectar un error y codificar el mensaje de manera errónea. En este caso, donde todos los bits son susceptibles a cambios, incluyendo los de paridad, es posible que esto suceda. Esto llevó a tener que registrar que es lo que se enviaba y que es lo que se recibía para poder confirmar que los mensajes descifrados fueran los que en realidad fueron transmitidos.

Uno de los fenómenos vistos en ambos algoritmos es como la cantidad de errores cambia dependiendo de la longitud de la cadena. En ambos casos los errores empezaban a ser más comunes mientras más larga fuera la cadena. En ambas gráficas se puede ver esta tendencia de aumento. Por ejemplo, en la gráfica de Hamming donde más errores se presentaron es en las cadenas de 18 caracteres de longitud. De igual manera en CRC-32 uno de los valores donde más errores hubo fue en las cadenas de 18 de longitud. Entonces es posible decir que a medida que más caracteres presenta la cadena, mayor era la probabilidad de presentar un error.

En cuanto a cuándo es mejor utilizar algoritmos de detección y corrección de errores depende mucho de lo que se esté realizando. Por ejemplo, si se requiere exactitud en la transmisión de los datos será una mejor opción implementar un algoritmo de corrección de errores. En los casos donde la tasa de error detectada es muy baja podría ser bueno implementar un algoritmo de detección de errores. Esto ya que una implementación de un algoritmo de detección es más sencilla de implementar que un algoritmo de corrección de errores. (Rastersoft, 2022)

Comentario Grupal

Este laboratorio nos pareció una actividad muy interesante, pero más que todo retadora, porque iba a ser necesario regresar a Java, un lenguaje de programación que teníamos tiempo sin tocar. Sin embargo, nos gustó mucho el reto que esto presentaba. También nos gustó poder trabajar con sockets, ya que es algo que hemos aprendido en otras clases y que a ambos nos gustaría experimentar más. Estas actividades nos parecen muy entretenidas porque nos ponen a pensar más de lo normal y a solucionar los problemas de distintas que es la mejor parte de la carrera.

Conclusiones

- El algoritmo de Hamming debido a su capacidad de corregir demostró un mejor rendimiento a comparación del algoritmo de CRC-32
- A pesar del mejor rendimiento del algoritmo de Hamming es posible presentar errores indetectables ya que todos los bits son susceptibles a cambios, incluyendo los bits de paridad.
- La longitud del mensaje juega un papel importante en la presencia de errores en el intercambio de información, ya que mientras más información haya, más propenso es el mensaje a errores.

Citas

Rastersoft. (2022). *Detección y corrección de errores*. Rastersoft.com.

<https://www.rastersoft.com/articulos/errores.html#:~:text=Una%20de%20las%20grandes%20ventajas,poder%20proceder%20a%20su%20inspecci%C3%B3n> .