

Laboratorio 2 – Esquemas de Corrección y Detección

Parte 2

Integrantes:

Diego José Franco Pacay (20240)

Oscar Fernando López Barrios (20679)

Descripción de la Práctica

Para esta segunda parte del laboratorio se utilizaron los algoritmos que se desarrollaron en la parte # 1. Además de ello se tuvo que desarrollar una aplicación para la transmisión y recepción de mensajes. Esto con la finalidad de comunicar a los emisores que en nuestro caso fueron desarrollados en java con los receptores que fueron desarrollados en Python. Para hacer esta comunicación se hicieron a través de sockets mediante un puerto predeterminado, para que el receptor siempre estuviera “escuchando” en el puerto elegido a la espera de recibir datos.

Cabe mencionar que del lado del receptor se tiene la peculiaridad de recibir los mensajes en texto normal y este se encarga de codificar cada carácter individual en ASCII binario. Esto se hace con la finalidad de poder trabajar con los algoritmos CRC-32 y el de Hamming los cuales reciben tramas en binario.

Mientras que del lado del receptor que siempre va a estar pendiente de recibir los mensajes que se le transmitan, se tiene la peculiaridad de aplicar “ruido” a los mensajes antes de enviarlos a la decodificación. La forma de determinar si cada bit se voltea (bit = !bit) se basará en cierta probabilidad expresada en errores por bits transmitidos (por ejemplo, $1/100=0.01$ es un error por cada 100 bits).

Para este laboratorio se trabajará con una arquitectura por capas, esto con la finalidad de comprender el funcionamiento de un modelo de capas y sus servicios. Las capas implementadas fueron:

- APLICACIÓN
- PRESENTACIÓN
- ENLACE
- RUIDO
- TRANSMISIÓN

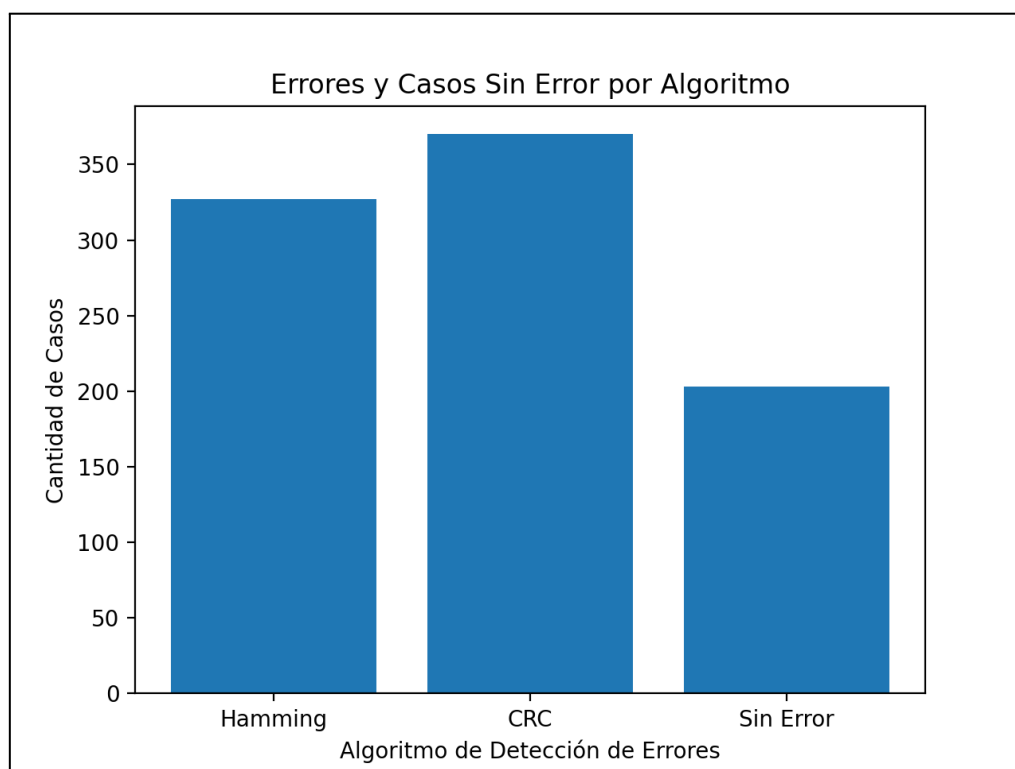
Resultados

En este laboratorio, se llevaron a cabo una serie de pruebas con el objetivo de evaluar el rendimiento de diferentes algoritmos de detección de errores. Estas pruebas abarcan una amplia gama de escenarios y parámetros para comprender en qué medida los algoritmos son capaces de identificar y corregir errores en las transmisiones de datos.

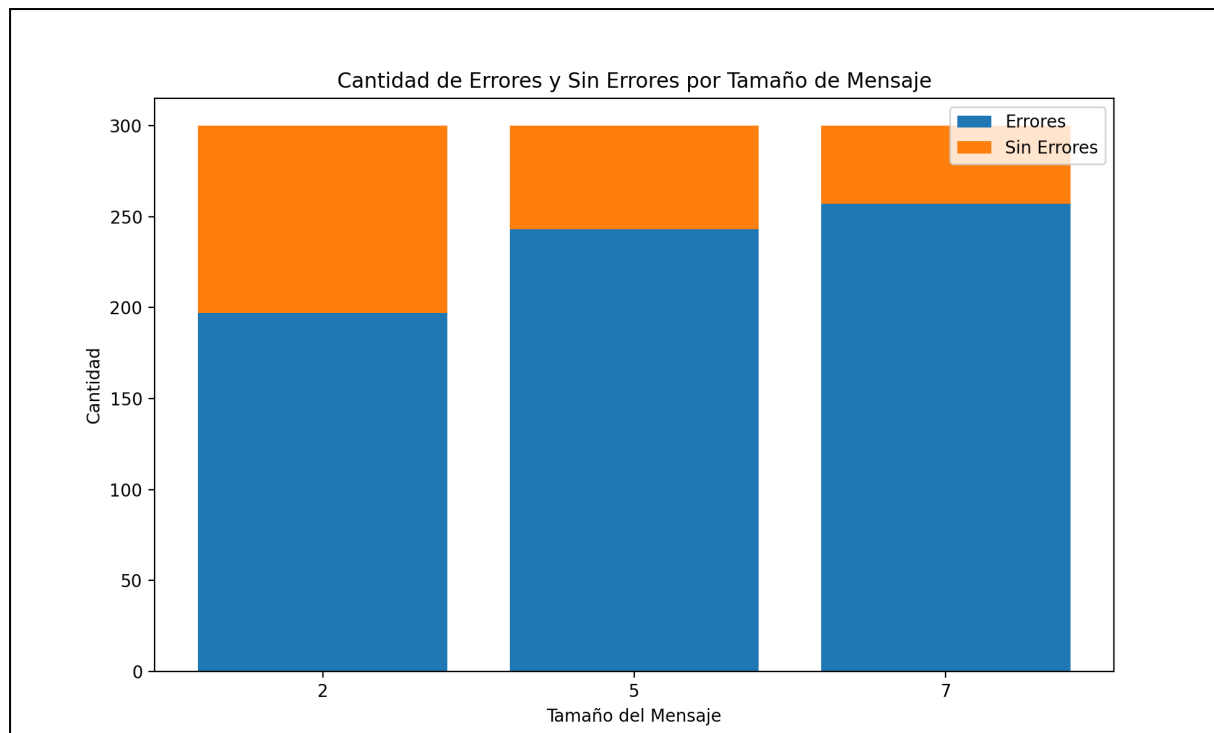
Para comenzar, las pruebas se realizaron utilizando tramas de distintos tamaños, concretamente de 2, 5 y 7 caracteres. Además, se incorporó el elemento de la probabilidad de error, lo que implicó simular la presencia de ruido en la transmisión. Se definieron probabilidades de error de 0.01, 0.05 y 0.1 para determinar cuándo sería necesario aplicar la técnica de "flip" de bits en la simulación de ruido. Cada combinación de tamaño de trama y probabilidad de error se sometió a 100 pruebas, lo que resultó en un total de 1000 pruebas individuales.

En cada una de estas pruebas, se eligió aleatoriamente qué algoritmo se aplicaría para analizar los datos. Esta elección aleatoria aseguró que los resultados fueran lo más imparciales posible y permitieron una comparación justa entre los algoritmos.

Los resultados de estas pruebas se colocaron en una gráfica que visualiza la cantidad de errores detectados por cada algoritmo. En una tercera columna adicional, se destacan los casos en los que ninguno de los dos algoritmos detectó errores. Esta gráfica tiene un propósito fundamental: determinar cuál de los dos algoritmos evaluados tiene una mayor capacidad para identificar y corregir errores en las transmisiones.



Adicionalmente, se generó otra gráfica que presenta la cantidad de errores y la cantidad de transmisiones sin errores acumuladas en función del tamaño inicial de la trama. Esta gráfica proporciona una perspectiva sobre si existe algún tipo de relación entre el tamaño de la trama y la presencia de errores. Es decir, si hay una tendencia que sugiera que ciertos tamaños de trama son más propensos a errores que otros.



Discusión

Como se mencionó anteriormente este laboratorio tuvo la finalidad de ver el rendimiento de los algoritmos de detección de errores cuando se ven sometidos a grandes cantidades de pruebas con distintos escenarios.

El algoritmo de Hamming de 405 veces que fue sometido a pruebas logró detectar una cantidad de 320 errores. Mientras que el algoritmo CRC-32 de 467 veces que fue sometido a pruebas logró detectar una cantidad de 352 errores.

Tras observar el primer gráfico y sabiendo lo anteriormente expuesto queda evidenciado que cada uno de los algoritmos tuvo una gran cantidad de detección de errores. Pero el que obtuvo mayor cantidad de detecciones fue el algoritmo CRC-32. Esto no significa que este algoritmo sea más eficiente que el Hamming. Esto lo único que refleja es que el algoritmo fue sometido a pruebas mayor cantidad de veces que el otro.

Además, una de las razones principales porque el algoritmo de CRC-32 sea más ineficiente al momento de formar parte de una metodología propensa a tener errores es debido a que el algoritmo agrega una gran cantidad de bits al momento de hacer el cálculo de la integridad del mensaje. Esto a comparación del algoritmo de Hamming que para hacer el cálculo de integridad hace una pequeña agregación de bits nuevos en la trama original, por lo tanto es menos propenso a que cambien los bits de la trama, debido a la cantidad de cada uno de los dos en distintos algoritmos.

En general, la metodología que se definió para la arquitectura de capas se trabajó de manera correcta pero a pesar de esto se tuvo una gran cantidad de errores y correcciones debido a la forma en la que estaba diseñado el funcionamiento del algoritmo, ya que al momento de realizar el envío de la información se tenía una probabilidad de tener ruido en cada uno de los bits de la trama.

Como se puede observar en la gráfica de cantidad de errores con respecto a la longitud del mensaje, se puede definir que para mensajes de mayor tamaño los errores son más grandes debido a la cantidad de caracteres aumenta la probabilidad de que se tenga un bit cambiado al momento de brindar la trama con ruido.

Además, se puede mencionar que en la mayoría de casos de los errores en las tramas de tamaño de 5 o 7 se tenían errores de más de 2 bits, por lo tanto el algoritmo de corrección de errores no era capaz de lograr de brindar la trama enviada correctamente, por lo tanto al momento de recibir el mensaje de parte del receptor, este aparecía de una manera incorrecta al que se había enviado al principio de la prueba.

La razón por la que se dieron los errores de envío de mensaje incorrecto fue debido a que el algoritmo de Hamming utilizado no estaba capacitado para realizar una descomposición de la trama que se enviaba de mensaje, por lo tanto al momento de haber más de un error en la trama, lo que pasaba era que esta se veía afectada por el mal funcionamiento del algoritmo de Hamming simple.

Comentario Personal

En general el laboratorio fue de gran importancia para la comprensión de los temas de arquitectura de capas y los esquemas de corrección, debido a que se puede visualizar de una manera más práctica la forma en la que puede funcionar el envío de información mediante una red y la forma en la que el ruido puede afectar de una manera muy mala la recepción de la información correcta.

El laboratorio ayudó a mejorar el entendimiento de tener algoritmos de integridad de información que funcionen de manera correcta en distintas condiciones en las que la información se puede ver comprometida de manera grande, siendo esta el envío de información totalmente distinta a la que el usuario quisiera recibir, por lo tanto brinda una visión correcta de la necesidad de mejorar los algoritmos para lograr estar atentos a cualquier error que pueda comprometer la integridad de la información de los usuarios.

Conclusiones

Se logró determinar que para la metodología que se utilizó para el laboratorio con el algoritmo y todas las características que se tomaron en cuenta para la arquitectura de capas los mensajes más pequeños son los que logran llegar de manera correcta al receptor. Por lo tanto se logró conocer que para mayor cantidad de caracteres existe una mayor cantidad de error que se pueda dar al momento de recibir el mensaje y al momento de traducirlo.

Debido a la forma en la que funciona cada uno de los algoritmos se puede determinar que el algoritmo de CRC al tener ruido aplicado es más probable que falle debido a la cantidad de bits extras que se le agregan a la trama, por lo tanto, esta misma aumenta el número de bits en una gran cantidad a comparación de lo que aumenta el número de bits extras que se agregan en el algoritmo de Hamming.

Para hacer uso de esta metodología se debe de mejorar el sistema en el que se corrigen los errores mediante un algoritmo más fuerte de Hamming, que logre detectar errores y corregirlos en una partición de los bits de la trama.

Los conceptos de arquitectura de capas y el ruido en el envío de información fueron comprendidos de manera correcta siendo brindados de una manera clara por las pruebas realizadas en el envío de información entre emisor y receptor por medio de sockets.

Citas y Referencias

- DEL, MARIA. *Sistemas Informáticos*. Ediciones Paraninfo, S.A., 30 June 2023.
- UNAM. *Ingeniería: Órgano Oficial de La Facultad de Ingeniería, Universidad Nacional Autónoma de México*. Google Books, La Facultad, 1975
- Izaskun Pellejero, et al. *Fundamentos Y Aplicaciones de Seguridad En Redes WLAN : De La Teoría a La Práctica*. Barcelona, Marcombo, 2006.