Universidad del Valle de Guatemala
Redes
Ciclo II, 2024
Fecha de entrega: 1 ago 2024 19:00

Adrián Ricardo Flores Trujillo	21500
Andrea Ximena Ramírez Recinos	21874

Laboratorio 2, Segunda Parte

Esquemas de Detección y Corrección.

[Parte 1] Descripción de la práctica

Esta práctica es una continuación del trabajo sobre esquemas de detección y corrección de errores. Anteriormente, se implementaron un algoritmo de corrección de errores (código de Hamming) y uno de detección de errores (checksum de Fletcher) en dos lenguajes de programación distintos. Los programas desarrollados no eran capaces de comunicarse entre sí. El objetivo de esta práctica es establecer un sistema de transmisión de mensajes entre los programas (uno de codificación y otro de decodificación) mediante el uso de sockets, y realizar múltiples pruebas con simulación de ruido en la transmisión de los mensajes para comparar el rendimiento de ambos algoritmos implementados.

[Parte 2] Resultados

Mensaje	Longitud
Sun	3
Euouae	6
Strengths	9
LoremIpsum	10
Unimaginatively	15
Subdermatoglyphic	17
Incomprehensibilities	21
Floccinaucinihilipilification	29
Pseudopseudohypoparathyroidism	30
Pneumonoultramicroscopicsilicovolcanoconiosis	45

Tabla 1. Mensajes de diferentes longitudes envíados durante las pruebas.

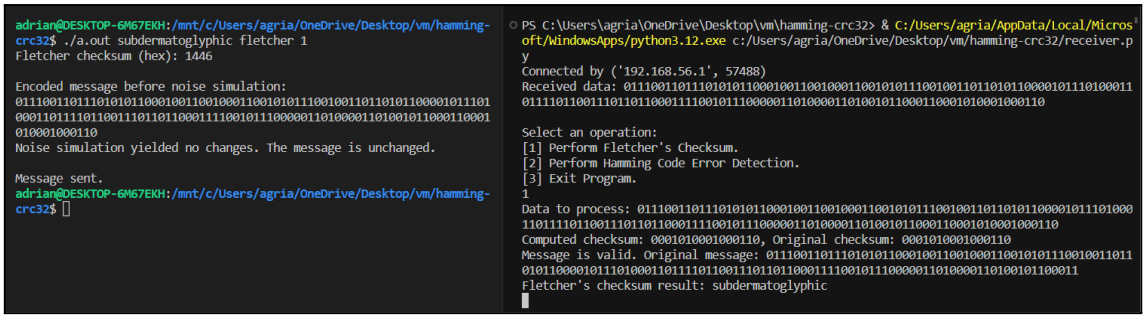

Figura 1. Ejemplo de envío de mensajes utilizando el algoritmo de Fletcher

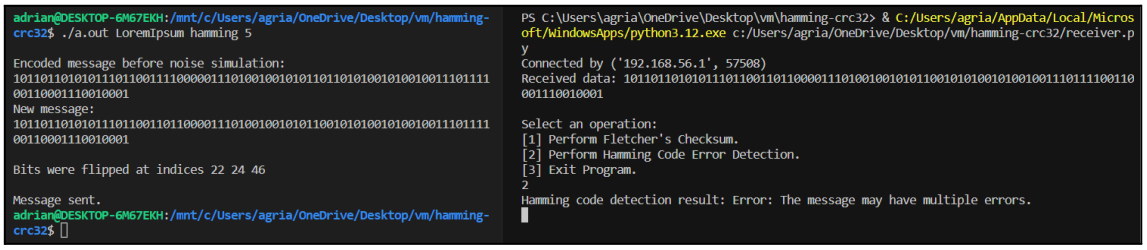

Figura 2. Ejemplo de envío de mensajes utilizando el código de Hamming

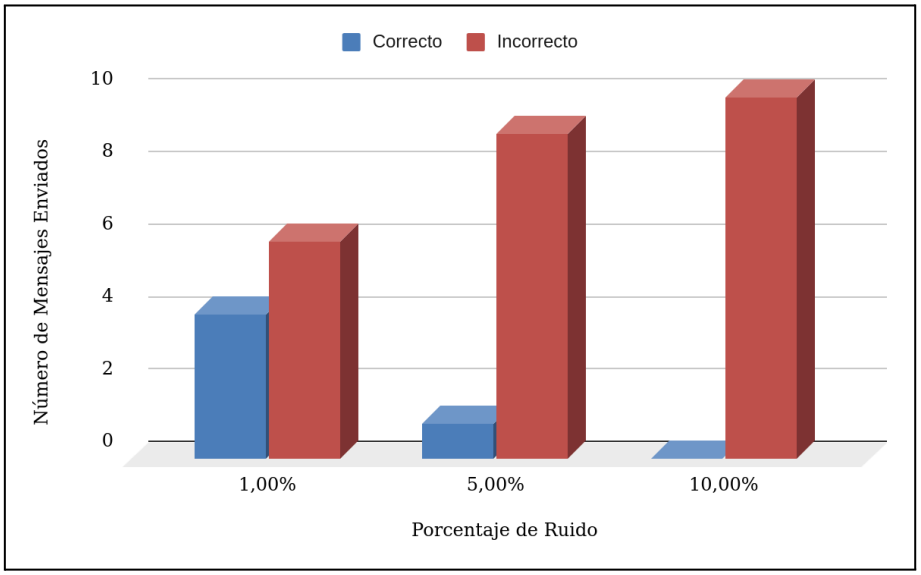

Figura 3. Resultados de pruebas realizadas con el algoritmo de Fletcher

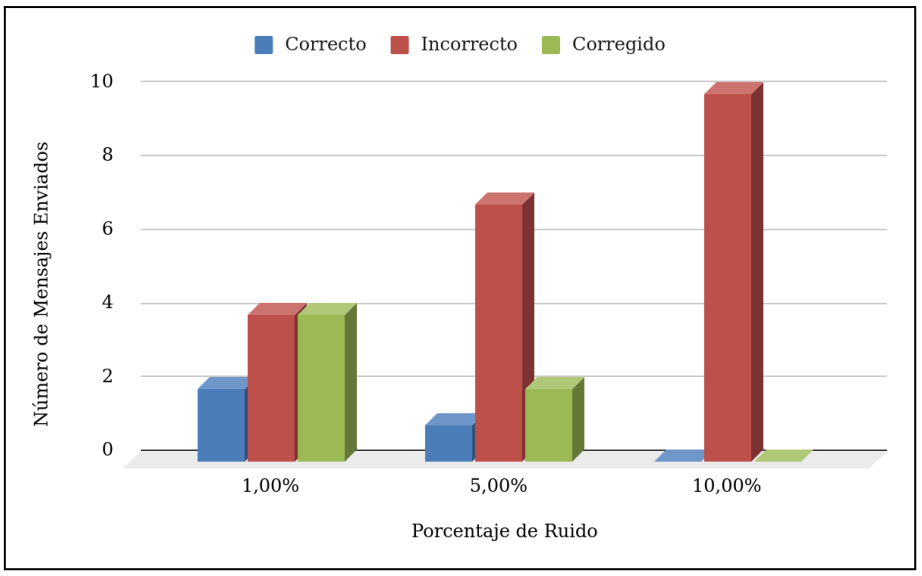

Figura 4. Resultados de pruebas realizadas con el código de Hamming

[Parte 3] Discusión

Al realizar múltiples pruebas con ambos algoritmos, variando solamente la probabilidad de encontrar ruido en cada bit y la longitud del mensaje, se observó que, en general, el algoritmo de codificación/decodificación de Hamming resultó superior en casos de prueba con poco ruido, ya que es capaz de corregir un solo error en el mensaje. En casos con más de un error, el algoritmo de Hamming obtuvo un rendimiento similar al del checksum de Fletcher-16, que no es capaz de corregir errores, sino solo de detectarlos.

Es importante destacar que ambos algoritmos cumplen propósitos ligeramente distintos: uno es un algoritmo de detección de errores y el otro es de corrección de errores. Por lo tanto, no es posible compararlos directamente y decir que uno es superior al otro, ya que la utilidad de los resultados depende del contexto en el que se aplique cada algoritmo. Por ejemplo, el código de Hamming no sería muy útil para verificar la integridad de un programa descargado de internet, ya que el propósito de verificar la integridad de una descarga es asegurar que los archivos no fueron modificados por un tercero o que no ocurrieron problemas significativos en la descarga (Digital Preservation Coalition, s.f.). En este caso, es mejor utilizar checksums para que cualquier persona pueda calcular el checksum del contenido y compararlo con el proporcionado por la fuente oficial. Por otro lado, el código de Hamming es mejor para la transmisión de mensajes cortos, ya que es capaz de corregir errores en escalas pequeñas, lo cual se evidenció en las pruebas realizadas para este laboratorio.

En base a lo anterior, es posible inferir que el código de Hamming es superior en cuanto al manejo de errores, por el simple hecho de poder corregirlos si solo hay un error. Cabe destacar que, en casos específicos, es posible obtener falsos positivos en la corrección de errores, ya que puede suceder que los bits alterados lleven a un mensaje matemáticamente íntegro, pero incorrecto respecto al mensaje original. Por otro lado, el algoritmo de Fletcher “falla” en todas las situaciones en las que haya cualquier error, y solo es capaz de decir si el mensaje está bien o no. Por lo tanto, para las pruebas realizadas en este laboratorio, el algoritmo de Fletcher resultó mejor para tratar con tasas de error significativas, ya que es improbable que éste obtenga falsos positivos.

[Parte 4] Conclusiones

- El algoritmo de Hamming es más apropiado para la transmisión de mensajes cortos gracias a su capacidad de corrección de errores.

- Utilizar checksums para la transmisión de mensajes cortos no es muy eficiente en comparación a los algoritmos de corrección de errores ya que estos solo son capaces de determinar si un mensaje está correcto o no.

- En situaciones con más de un error, el rendimiento del algoritmo de Hamming es similar al del checksum de Fletcher, que solo puede detectar errores, no corregirlos.

- El código de Hamming puede generar falsos positivos, corrigiendo errores de manera incorrecta en casos con más de un error.

- Para las pruebas realizadas en el laboratorio, el algoritmo de Fletcher demostró ser más efectivo que el código de Hamming para manejar tasas de error significativas.

[Parte 5] Referencias

1. Digital Preservation Coaltion (s.f.). Fixity and Checksums. Digital Preservation Handbook. https://www.dpconline.org/handbook/technical-solutions-and-tools/fixity-and-checksums