

Universidad Del Valle De Guatemala

Miguel Novella

Redes



Lab 2.1 Esquemas de detección y corrección de errores

Pablo Daniel Gonzalez 20362

Jose Andres Hernández Guerra 20053

Guatemala, 20 de julio del 2023

Descripción de la práctica

Esta práctica consiste en la implementación y evaluación de dos algoritmos clave para la detección y corrección de errores en la transmisión de datos: el Código de Hamming y CRC-32.

Código de Hamming

El Código de Hamming es un código de corrección de errores que utiliza bits de paridad en posiciones específicas para detectar y corregir errores en un solo bit. La eficacia de este código radica en su capacidad para identificar la ubicación exacta del error y corregirlo.

CRC-32

CRC-32 (Comprobación Redundante Cíclica) es un algoritmo de detección de errores que utiliza un polinomio generador para detectar alteraciones en los datos crudos. No corrige los errores pero es eficiente para detectar errores comunes en la transmisión de datos.

Resultados

Hamming con errores:

```
Pruebas con errores para el Código de Hamming:
Trama con error: 1110100101
Trama decodificada (Receptor): 110101
Mensaje: Se detectaron y corrigieron errores en la posición 7. Trama corregida: 1110101101

Trama con error: 1011001111
Trama decodificada (Receptor): 100110
Mensaje: Se detectaron y corrigieron errores en la posición 10. Trama corregida: 1011001110

Trama con error: 0010110010
Trama decodificada (Receptor): 111000
Mensaje: Se detectaron y corrigieron errores en la posición 9. Trama corregida: 0010110000
```

CRC-32 con errores:

```
Pruebas con errores para CRC-32:
Trama con error: 11010111101000100000001101110010001101
Trama decodificada (Receptor): 110101
Mensaje: Se detectaron errores, la trama se descarta

Trama con error: 10011000001110100101001110001001000111
Trama decodificada (Receptor): 100110
Mensaje: Se detectaron errores, la trama se descarta

Trama con error: 11100011001011001111000100001011111110
Trama decodificada (Receptor): 111000
Mensaje: Se detectaron errores, la trama se descarta
```

Pruebas con errores

Trama con error: 1110100101 (Hamming, error en posición 7)

- Mensaje del receptor: Se detectaron y corrigieron errores en la posición 7.
Trama corregida: 1110101101

Hamming sin errores:

```
Pruebas para el Código de Hamming:  
Trama original: 110101  
Trama codificada (Emisor): 1110101101  
Trama decodificada (Receptor): 110101  
Mensaje: No se detectaron errores
```

```
Trama original: 100110  
Trama codificada (Emisor): 1011001110  
Trama decodificada (Receptor): 100110  
Mensaje: No se detectaron errores
```

```
Trama original: 111000  
Trama codificada (Emisor): 0010110000  
Trama decodificada (Receptor): 111000  
Mensaje: No se detectaron errores
```

CRC-32 sin errores

```
Pruebas para CRC-32:  
Trama original: 110101  
Trama codificada (Emisor): 11010111101000100000001101110010001100  
Trama decodificada (Receptor): 110101  
Mensaje: No se detectaron errores
```

```
Trama original: 100110  
Trama codificada (Emisor): 10011000001110100101001110001001000110  
Trama decodificada (Receptor): 100110  
Mensaje: No se detectaron errores
```

```
Trama original: 111000  
Trama codificada (Emisor): 11100011001011001111000100001011111111  
Trama decodificada (Receptor): 111000  
Mensaje: No se detectaron errores
```

Pruebas sin errores

Trama original: 110101

- Emisor (Hamming): 1110101101
- Emisor (CRC-32): 11010111101000100000001101110010001100
- Mensaje del receptor (ambos): No se detectaron errores

Discusión

Implementación y Eficacia

Los resultados de las pruebas demuestran la eficiencia de ambos algoritmos en la detección y corrección de errores. El Código de Hamming está diseñado para corregir errores de un solo bit, este algoritmo mostró una buena capacidad para identificar y corregir estos errores en las tramas de prueba. Por otro lado, CRC-32 que se centra en la detección de errores, nos mostró que puede ser eficaz en identificar inconsistencias en las tramas, gracias a su comprobación redundante y que al mismo tiempo es cíclica.

Análisis de Errores

Se realizaron pruebas tanto con tramas correctas como con tramas que contenían errores. Los errores fueron introducidos manualmente en diferentes posiciones para ver cómo los algoritmos responden o se comportan en distintos casos. Esto incluyó la alteración de bits de datos y también en el caso del código de Hamming, la alteración de bits de paridad.

Casos no detectados

En el caso del Código de Hamming, hubo una situación en la que el error estaba en una posición de bit de paridad, lo que llevó a una detección incorrecta. Esta situación da como resultado una limitación incorrecta del algoritmo, que es incapaz de detectar múltiples errores o errores en las posiciones de paridad. No se observaron casos no detectados con CRC32 en las pruebas realizadas.

Comparación de Algoritmos

Ambos algoritmos tienen sus ventajas y desventajas. El Código de Hamming es especialmente útil cuando es esencial corregir errores, mientras que CRC-32 es más adecuado para situaciones donde la detección es suficiente y la corrección se maneja de otra manera como podría ser por ejemplo, transmitiendo la trama.

Comentario grupal

Los errores en la transmisión de datos son inevitables y pueden tener consecuencias graves al momento de estar interpretando o parseando la data que se recibe. La implementación y uso de algoritmos de detección y corrección de errores como los que programamos en esta parte del laboratorio son fundamentales para garantizar la integridad y confiabilidad de los datos con los que un programador puede estar manejando.

Conclusiones

- La práctica y la implementación (escritura de código tanto en Java como en Python) ofrecieron una comprensión profunda de cómo funcionan los algoritmos de detección y corrección de errores.
- El Código de Hamming y CRC-32 demostraron ser eficientes en sus roles, con limitaciones que se tenían contemplados en sus diseños.
- La gestión de errores es un aspecto crítico en la comunicación de datos, y la selección de los algoritmos adecuados es clave para la eficiencia y buen control del sistema.