



UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Ingeniería en Ciencia de la Computación y Tecnologías de la Información

Redes

Diego Ruiz, 18761 y Jose Jorge Pérez, 18364.

Laboratorio 2 - Primera Parte

Esquema de detección y corrección de errores

Descripción de la Práctica:

Para esta práctica, se implementaron dos algoritmos de detección y corrección de errores, el algoritmo de Hamming y CRC-32. El algoritmo de Hamming agrega bits de paridad a los datos para determinar si hay errores durante la transmisión. También es capaz de detectar errores de un solo bit y hacer la corrección pertinente. Utiliza los bits de paridad para hacer una matriz y realizar las operaciones para detección de errores con esto. Por otro lado, el algoritmo de detección de redundancia cíclica utiliza la operación matemática división cíclica para calcular un valor checksum a partir de los datos que se le dan. Si el valor que se calcula no es igual al valor recibido, se detecta un error. En esta práctica se implementaron las partes de emisor y receptor utilizando distintos lenguajes de programación.

Resultados:

Hamming

```
Mensaje original: 1000001
Mensaje codificado: 00100001001

../debugpy/launcher 531
Hamming Check: Passed
```

CRC-32

```
/usr/local/bin/node ./crc32.js
CRC-32: 2CB29632
Full Message: 011011002CB29632

../debugpy/launcher 53163
CRC-32 Check: Passed
```

Como se puede observar, ambos algoritmos lograban crear el mensaje codificado a partir del mensaje original, y pudieron realizar los chequeos pertinentes.

Discusión:

Sin embargo, es importante tener en cuenta que, en el caso de Hamming, el algoritmo sólo es capaz de corregir errores en un solo bit en algunas configuraciones específicas. Si se presentan más de un error en la trama, el algoritmo no podrá corregirlos y solo será capaz de detectar que ha ocurrido un error. Por otro lado, el algoritmo CRC-32 no puede corregir errores, solo puede detectarlos.

En cuanto a la tasa de detección de errores, ambos algoritmos demostraron ser efectivos en la mayoría de los casos. Sin embargo, es importante destacar que, en algunos casos extremos, donde se producen múltiples errores en la trama y estos afectan bits específicos, algunos errores pueden no ser detectados por los algoritmos.

Conclusiones:

- Ambos algoritmos cumplen con su tarea de codificación del mensaje y de detectar errores o de dar un chequeo como correcto con un mensaje generado por su contraparte.
- Hamming es adecuado para la corrección de errores de un bit, CRC-32 puede únicamente detectar errores.
- Hamming es útil para corregir errores en tiempo real, como para sistemas de comunicación en tiempo real.