

Universidad del Valle de Guatemala
Departamento de Ciencias de la Computación
CC3067 Redes
Jorge Andrés Yass Coy

Laboratorio No. 2

Esquemas de detección y corrección de errores

Ana Paola de León 203061
Gabriela Paola Contreras 20213

Guatemala, 03 de agosto de 2023

Introducción

El fin de este laboratorio fue conocer dos esquemas de detección y corrección de errores, para ello se utilizaron los algoritmos de Hamming y CRC los cuales fueron puestos en práctica mediante la implementación de estos en dos lenguajes de programación diferentes y de esta manera poder identificar las ventajas y desventajas de cada uno de los algoritmos.

Esta actividad fue realizada en parejas y quedó a discreción de estas que algoritmo desarrolló cada una. Para la realización de este se debía de implementar en cada algoritmo la versión de emisor para verificar la integridad del mensaje, y el receptor para poder utilizar la información generada por el emisor. Es importante destacar que el emisor debía ser realizado con un lenguaje de programación distinto al receptor y quedaba a discreción del desarrollador como conectar estos para el traspaso de información. Para poder verificar el correcto funcionamiento ambos algoritmos debían de ser probados con mínimo 4 tramas diferentes y en algunos casos modificar uno o más bits para ver si el resultado obtenido era el deseado. Al finalizar se discutieron diferentes aspectos por medio de los cuales se puede concluir que los esquemas de detección y corrección de errores son herramientas que ayudan a la integridad y transmisión de datos de manera confiable.

Resultados

- Algoritmo CRC

Cuadro No.01 Resultados utilizando tres tramas sin error

Emisor			
Trama	1001	1101	0010
Resultado	Trama ingresada: 0010 Resultado de CRC-32: 001000010011000001000111011011011110	Trama ingresada: 1101 Resultado de CRC-32: 110101100011100110110000110110101011	Trama ingresada: 0010 Resultado de CRC-32: 001000010011000001000111011011011110

Receptor			
Trama	1001100	1100110	0011001
Resultado	Mensaje sin errores. > Trama: 1001100	Mensaje sin errores. > Trama: 1100110	Mensaje sin errores. > Trama: 1100110

Cuadro No.02 Resultados utilizando las tres tramas anteriores con error

Receptor			
Trama	1001100	1100110	0011001
Cambio	1001101	1100100	0011101
Resultado	@! Hay errores en el mensaje, por lo tanto se descarta.	@! Hay errores en el mensaje, por lo tanto se descarta.	@! Hay errores en el mensaje, por lo tanto se descarta.

Cuadro No.03 Resultados utilizando las tres tramas anteriores con dos errores

Receptor			
Trama	1001100	1100110	0011001
Cambio	1001111	1100101	0011111
Resultado	@! Hay errores en el mensaje, por lo tanto se descarta.	@! Hay errores en el mensaje, por lo tanto se descarta.	@! Hay errores en el mensaje, por lo tanto se descarta.

Cuadro No.04 Resultados utilizando una trama sin error detectable

Receptor	
Trama	No se encontró ninguna trama en la que no se detectara el error.
Cambio	N/A
Resultado	N/A

- Algoritmo Hamming

Cuadro No.05 Resultados utilizando tres tramas sin error

Emisor			
Trama	1001	1101	0010
Resultado	-> La data ingresada es: 1001 -> La data codificada es: 1001100	-> La data ingresada es: 1101 -> La data codificada es: 1100110	-> La data ingresada es: 0010 -> La data codificada es: 0100101

Receptor			
Trama	1001100	1100110	0011001
Resultado	-> Se esta evaluando: 1001100 -- La cadena no cuenta con error --	-> Se esta evaluando: 1100110 -- La cadena no cuenta con error --	-> Se esta evaluando: 0100101 -- La cadena no cuenta con error --

Cuadro No.06 Resultados utilizando las tres tramas anteriores con error

Receptor			
Trama	1001100	1100110	0011001
Cambio	1011100	0100110	0011011

Resultado	-> Se esta evaluando: 1011100 -> El error se encuentra en la posición: 5 ... Corrigiendo ... -> El mensaje ha sido corregido, originalmente era: 1001100	-> Se esta evaluando: 0100110 -> El error se encuentra en la posición: 7 ... Corrigiendo ... -> El mensaje ha sido corregido, originalmente era: 1100110	-> Se esta evaluando: 0011011 -> El error se encuentra en la posición: 2 ... Corrigiendo ... -> El mensaje ha sido corregido, originalmente era: 0011001
-----------	--	--	--

Cuadro No.07 Resultados utilizando las tres tramas anteriores con dos errores

Receptor			
Trama	1001100	1100110	0011001
Cambio	1011101	1110010	1011011
Resultado	-> Se esta evaluando: 1011101 -> El error se encuentra en la posición: 1 ... Corrigiendo ... -> El mensaje ha sido corregido, originalmente era: 1011100	-> Se esta evaluando: 1110010 -> El error se encuentra en la posición: 3 ... Corrigiendo ... -> El mensaje ha sido corregido, originalmente era: 1110110	-> Se esta evaluando: 1011011 -> El error se encuentra en la posición: 5 ... Corrigiendo ... -> El mensaje ha sido corregido, originalmente era: 1001011

Cuadro No.08 Resultados utilizando una trama sin error detectable

Receptor	
Trama	1001100
Cambio	0110100
Resultado	-> Se esta evaluando: 0110100 -- La cadena no cuenta con error --

Discusión

El objetivo del laboratorio fue conocer esquemas de detección y corrección de errores, para cumplir dicho objetivo se trabajaron con los algoritmos de Hamming y CRC los cuales fueron implementados en su versión emisor y receptor cada uno en diferentes lenguajes programación.

El algoritmo CRC-32 es un algoritmo para la detección de errores y verificación de integridad de los datos. En este se utilizó un polinomio generados de 32 bits para luego realizar una división con una operación XOR. Al realizar el desarrollo del algoritmo y realizar las pruebas, se observó que el mismo tiene una excelente capacidad para detectar errores. En todos los casos evaluados, al introducir errores en el lado del receptor, el algoritmo logró identificarlos. Sin embargo, es importante recordar que, aunque sea una herramienta muy eficiente para detectar errores, este no es capaz de corregirlos.

El algoritmo de hamming es un sistema de corrección de errores el cual puede detectar y corregir errores el cual utiliza un mecanismo de paridad de bloques, el cual consiste en dividir los datos en bloques y agregarle paridad a el bloque que se esté evaluando. La cantidad de datos de paridad agregados al código de Hamming viene dada por la fórmula 2^p

$\geq i + p + 1$, donde p es el número de bits de paridad e i es el número de bits de datos (Wright, 2022). Para la implementación de esta se optó por utilizar hamming(7:4) en donde 4 bits son de data y 3 bits son de paridad, la parte del emisor es el encargado de procesar los 4 bits de data y calcular la paridad para de esta manera poder producir un mensaje con 7 bits el cual es el que el receptor será encargado de procesar para evaluar si este sufre algún cambio.

En base a los resultados obtenidos en el cuadro No. 5 se puede observar que el algoritmo está realizado correctamente, pues este fue capaz de dividir por bloque la trama y calcular su paridad con ayuda de la tabla de verdad y como resultado nos retorna una cadena de 7 bits, y debido a que no se realizó ningún cambio a dicha trama el receptor logra mostrar que dicha trama no cuenta con ningún error. Es evidente que en los cuadros 6 y 7 el receptor busca en qué posición se encuentra el bit que está generando un error y cambia este bit al correcto, cabe destacar que cuando la trama cuenta con únicamente un bit cambiado este es capaz de detectar el error correctamente, pero al momento de colocar más de un error este se limita a corregir solo un bit, esto se debe a que el código de Hamming estándar solo puede detectar y corregir un error de un solo bit, lo cual es una gran limitante para este. Finalmente en el cuadro 8 se puede ver que a pesar de que la trama que recibió el receptor es incorrecta esta fue detectada como correcta esto se puede deber la cantidad de errores, la redundancia de ciertos bits, es relevante mencionar que el modelo que se implementó cuenta con una efectividad del 57% puesto a que la detección del error recae en los bits de paridad que este calcula lo cual nos da la pauta de que este algoritmo puede detectar y corregir una parte significativa de los errores de bits que podrían ocurrir durante la transmisión.

Comentario grupal

Como grupo consideramos que a pesar de ser un laboratorio el cual requirió de investigación y varias vistas a la grabación de la clase fue una buena forma de hacernos comprender a profundidad cómo es que funcionan la detección y la corrección en base a diferentes algoritmos y nos desafío un poco pues al obligarnos a programar en dos lenguajes distintos nos hizo pensar en maneras de pasar los datos entre emisor y receptor. Por otra parte programar ambas partes y jugar un poco con las tramas que se iban a enviar nos permitió experimentar las limitaciones que cada uno de estos tiene. Al concluir esta nos pudimos dar cuenta de que los esquemas de detección y corrección de errores son importantes para garantizar la integridad y confiabilidad para la comunicación ya que estos se encargan de mitigar los errores y aumentar la precisión de mensaje lo cual puede ser de gran utilidad en distintos sistemas.

Conclusiones

- Se puede concluir que los esquemas de detección y corrección de errores son herramientas que ayudan a la integridad y transmisión de datos de manera confiable.
- Se recomienda que para futuras prácticas implementar el modelo de hamming extendido (15,11) puesto a que este puede llegar a ofrecer una mayor capacidad de detección y corrección de errores, lo cual asegura la integridad de los datos.
- En base a los resultados obtenidos, se demuestra que el algoritmo CRC-32 es un algoritmo con una alta tasa de detección de errores en donde se garantiza la integridad de los datos transmitidos.

Cita y referencias

- Coding Ninjas. (s. f.). *Coding Ninjas Studio*.
<https://www.codingninjas.com/studio/library/examples-for-hamming-code#:~:text=Limitations%20of%20Hamming%20code,-The%20disadvantages%20of&text=Hamming%20code%20cannot%20be%20used,high%20level%20of%20error%20correction>.
- Wright, G. (2022). Hamming code. *WhatIs.com*.
<https://www.techtarget.com/whatis/definition/Hamming-code#:~:text=Hamming%20code%20is%20an%20error,data%20is%20stored%20or%20transmitted>.