

Laboratorio 2 – Esquemas de Corrección y Detección

Integrantes:

Diego José Franco Pacay (20240)

Oscar Fernando López Barrios (20679)

Descripción de la Práctica

Para este laboratorio, se trabajaron conceptos sobre el manejo de errores en la transmisión de datos y los algoritmos que se utilizan ya sea para solo la detección de los mismos o su corrección.

Para familiarizarse más con estos algoritmos se utilizaron uno de cada tipo, es decir uno de detección y uno de corrección. Para cada familia de algoritmos se tuvo que desarrollar un algoritmo que simulara la parte del emisor, es decir que recibiera la trama original y la retornará con la información extra según el algoritmo utilizado para el envío del mensaje. Y un algoritmo que simulara la parte del receptor, es decir que recibiera la trama concatenada con la información adicional requerida por el algoritmo y que luego la detección/corrección de errores retorna si la trama ingresada fue aceptada sin problema o de detectar errores indicar que se descarta la trama por la misma razón. Y para el caso del algoritmo de corrección indicar la posición del bit corregido.

Los algoritmos utilizados fueron:

- Para la detección de errores: CRC-32

El algoritmo CRC-32 del lado del emisor recibe un mensaje en binario “trama”, y le agrega un polinomio de grado 32 al final de la trama. Luego de agregarlo se hacen divisiones binarias es decir diversos XOR hasta que los bits de la trama con la información extra se acaben. Por último se le agrega a la trama original los últimos 32 bits del residuo para que sea la información extra.

Del lado del receptor el algoritmo recibe la trama con la información agregada y hace divisiones binarias con el polinomio de 32 bits. Es decir diversos XOR hasta que los bits de la trama con la información extra se acaben. Si el residuo de la división entera contiene algún bit 1 significa que la trama no está buena. En cambio si no posee 1 la trama se acepta.

- Para la corrección de errores: Hamming

El algoritmo de Hamming por el lado del emisor debe de recibir una trama en binario, según la longitud de la trama se debe calcular el valor de p para $2^p \geq p + longitud + 1$. Al tener el valor de p se debe de calcular los bits de paridad según los valores de p y verificar cuales son los valores posee un número impar de 1 para completarlo o del mismo modo con los 0. Al tener esto se completa la trama con la información faltante y retorna la trama con estos bits de paridad adicionales.

En cuanto al lado del receptor recibe la trama con la información agregada y hace el mismo proceso de calcular la p . Al tener el valor de p se debe de calcular los bits de paridad según los valores de p y verificar cuales son los valores posee un número impar de 1 para completarlo o del mismo modo con los 0. Y si los bits de paridad quedan todos 0 es que la trama fue ingresada correctamente. Mientras que si existe algún 1 dentro del resultado significa que hubo un error en la trama y ese resultado es la representación en binario de la posición del bit erróneo y debe cambiarlo.

Resultados

Demostración de las pruebas con las tramas sin manipulación para el algoritmo CRC:

- Primera Trama: 1001

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> javac .\emisorCRC.java
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorCRC
Ingrese la trama en binario: 1001
Mensaje con CRC:
100100100010110010011111000000001111
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> 
```

Recepción de la trama con los datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Py
Ingrese la trama en binario concatenado con la información adicional: 100100100010110010011111000000001111
Mensaje original: 1001
Trama recibida sin errores: 100100100010110010011111000000001111
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> 
```

- Segunda Trama: 101010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorCRC
Ingrese la trama en binario: 101010
Mensaje con CRC:
10101010110111101010010110000000110110
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama con los datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-32/Scripts/python.exe" C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-32/Scripts/python.exe C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-32/Scripts/python.exe
Ingrese la trama en binario concatenado con la información adicional: 10101010110111101010010110000000110110
Mensaje original: 101010
Trama recibida sin errores: 10101010110111101010010110000000110110
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

- Tercera Trama: 10010010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorCRC
Ingrese la trama en binario: 10010010
Mensaje con CRC:
1001001000101100100111110000000011110000
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama con los datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-32/Scripts/python.exe" C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-32/Scripts/python.exe C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-32/Scripts/python.exe
Ingrese la trama en binario concatenado con la información adicional: 1001001000101100100111110000000011110000
Mensaje original: 10010010
Trama recibida sin errores: 1001001000101100100111110000000011110000
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Demostración de las pruebas con las tramas sin manipulación para el algoritmo Hamming:

- Primera Trama: 1001

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming
Ingrese una trama en binario: 1001
El mensaje en binario con la información adicional es:
0011001
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama con los datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando L...  
Ingrese el mensaje en binario con información adicional: 0011001  
No se detectaron errores. Trama recibida:  
0011001  
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

- Segunda Trama: 101010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming  
Ingrese una trama en binario: 101010  
El mensaje en binario con la informacion adicional es:  
0011010110  
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama con los datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando L...  
Ingrese el mensaje en binario con información adicional: 0011010110  
No se detectaron errores. Trama recibida:  
0011010110  
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

- Tercera Trama: 10010010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming  
Ingrese una trama en binario: 10010010  
El mensaje en binario con la informacion adicional es:  
111100110010  
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama con los datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando L...  
Ingrese el mensaje en binario con información adicional: 111100110010  
No se detectaron errores. Trama recibida:  
111100110010  
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Demostración de las pruebas con las tramas con manipulación para el algoritmo CRC:

- Primera Trama: 1001

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> javac .\emisorCRC.java
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorCRC
Ingrese la trama en binario: 1001
Mensaje con CRC:
100100100010110010011111000000001111
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

Recepción de la trama manipulada en el último bit con un 0

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Py
Ingrese la trama en binario concatenado con la información adicional: 100100100010110010011111000000001110
Mensaje original: 1001
Se detectaron errores: La trama se descarta.
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

- Segunda Trama: 101010

Emisión de la trama con sus datos extras

```
● PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorCRC
Ingrese la trama en binario: 101010
Mensaje con CRC:
10101010110111101010101100000000110110
○ PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

Recepción de la trama manipulada en el penúltimo bit con un 0

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Py
Ingrese la trama en binario concatenado con la información adicional: 10101010110111101010101100000000110100
○ Mensaje original: 101010
Se detectaron errores: La trama se descarta.
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

- Tercera Trama: 10010010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorCRC
Ingrese la trama en binario: 10010010
Mensaje con CRC:
1001001000101100100111110000000011110000
○ PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

Recepción de la trama manipulada en el antepenúltimo bit con un 1

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe" C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe
Ingrese la trama en binario concatenado con la información adicional: 1001001000101100100111110000000011110100
Mensaje original: 10010010
Se detectaron errores: La trama se descarta.
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex>
```

Demostración de las pruebas con las tramas con manipulación para el algoritmo Hamming:

- Primera Trama: 1001

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming
Ingrese una trama en binario: 1001
El mensaje en binario con la informacion adicional es:
0011001
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex>
```

Recepción de la trama manipulada en el último bit con un 0

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe" C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe
Ingrese el mensaje en binario con información adicional: 0011000
Se detectaron errores y se corrigieron.
Posición del bit erróneo: 7
Trama corregida:
0011001
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex>
```

- Segunda Trama: 101010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming
Ingrese una trama en binario: 101010
El mensaje en binario con la informacion adicional es:
0011010110
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex>
```

Recepción de la trama manipulada en el penúltimo bit con un 0

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe" C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe
Ingrese el mensaje en binario con información adicional: 0011010100
Se detectaron errores y se corrigieron.
Posición del bit erróneo: 9
Trama corregida:
0011010110
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex>
```

- Tercera Trama: 10010010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorHamming
Ingrese una trama en binario: 10010010
El mensaje en binario con la informacion adicional es:
000010001001
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

Recepción de la trama manipulada en el cuarto bit con un 1

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> & "C:/Users/Fernando L
Ingrese el mensaje en binario con información adicional: 00011001001
Se detectaron errores y se corrigieron.
Posición del bit erróneo: 4
Trama corregida:
00001001001
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

Demostración de las pruebas con las tramas con manipulación de 2 bits para el algoritmo CRC:

- Primera Trama: 100

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorCRC
Ingrese la trama en binario: 100
Mensaje con CRC:
1000001001100000100011101101101100
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

Recepción de la trama manipulada con 2 bits de 0s

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe" C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe
Ingrese la trama en binario concatenado con la información adicional: 100000100110000010001110110110110000
Mensaje original: 100
Se detectaron errores: La trama se descarta.
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

- Segunda Trama: 1000

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorCRC
Ingrese la trama en binario: 1000
Mensaje con CRC:
100000100110000010001110110110111000
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama manipulada en los últimos 2 bits con 1s

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Py
Ingrese la trama en binario concatenado con la información adicional: 100000100110000010001110110110111011
Mensaje original: 1000
Se detectaron errores: La trama se descarta.
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

- Tercera Trama: 101010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorCRC
Ingrese la trama en binario: 101010
Mensaje con CRC:
10101010110111101010010110000000110110
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama manipulada los 2 últimos bits por 0 y 1

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando Lopez/AppData/Local/Programs/Python/Py
Ingrese la trama en binario concatenado con la información adicional: 10101010110111101010010110000000110101
Mensaje original: 101010
Se detectaron errores: La trama se descarta.
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Demostración de las pruebas con las tramas con manipulación de 2 bits para el algoritmo Hamming:

- Primera Trama: 100

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming
Ingrese una trama en binario: 100
El mensaje en binario con la informacion adicional es:
010101
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```


Recepción de la trama manipulada en el los 2 últimos bits con 1 y 0

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> & "C:/Users/Fernan
Ingrese el mensaje en binario con información adicional: 010110
Se detectaron errores y se corrigieron.
Posición del bit erróneo: 3
Trama corregida:
011110
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

- Segunda Trama: 1000

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorHamming
Ingrese una trama en binario: 1000
El mensaje en binario con la informacion adicional es:
1101001
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

Recepción de la trama manipulada en el los primeros 2 bits por 0s

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> & "C:/Users/Fernan
Ingrese el mensaje en binario con información adicional: 0001001
Se detectaron errores y se corrigieron.
Posición del bit erróneo: 6
Trama corregida:
0001011
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

- Tercera Trama: 101010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorHamming
Ingrese una trama en binario: 101010
El mensaje en binario con la informacion adicional es:
0000101101
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> █
```

Recepción de la trama manipulada en el primer y último bit

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> & "C:/Users/Fernando
Ingrese el mensaje en binario con información adicional: 1000101100
Se detectaron errores y se corrigieron.
Posición del bit erróneo: 13
Trama corregida:
1000101100
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> |
```

Demostración de las pruebas con las tramas con manipulación para que no se detecte el error con el algoritmo CRC:

- Primera Trama: 100100

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorCRC
Ingrese la trama en binario: 100100
Mensaje con CRC:
10010010001011001001111100000000111100
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> |
```

Recepción de la trama manipulada con para no detectar el error

No fue posible encontrar una trama manipulada para que no encontrara el error

- Segunda Trama: 10100010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> java emisorCRC
Ingrese la trama en binario: 10100010
Mensaje con CRC:
101000101111100010101101101101101100000
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redes> |
```

Recepción de la trama manipulada con para no detectar el error

No fue posible encontrar una trama manipulada para que no encontrara el error

- Tercera Trama: 1001001010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorCRC
Ingrese la trama en binario: 1001001010
Mensaje con CRC:
100100101010111011111111100011100010101110
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama manipulada con para no detectar el error

No fue posible encontrar una trama manipulada para que no encontrara el error

Demostración de las pruebas con las tramas con manipulación para que no se detecte el error con el algoritmo Hamming:

- Primera Trama: 100100

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming
Ingrese una trama en binario: 100100
El mensaje en binario con la informacion adicional es:
0001010101
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama manipulada con para no detectar el error

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando
Ingrese el mensaje en binario con información adicional: 0100000101
No se detectaron errores. Trama recibida:
0100000101
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

- Segunda Trama: 10100010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming
Ingrese una trama en binario: 10100010
El mensaje en binario con la informacion adicional es:
110010000101
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama manipulada con para no detectar el error

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando L
Ingrese el mensaje en binario con información adicional: 100111000101
No se detectaron errores. Trama recibida:
100111000101
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

- Tercera Trama: 1001001010

Emisión de la trama con sus datos extras

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> java emisorHamming
Ingrese una trama en binario: 1001001010
El mensaje en binario con la informacion adicional es:
11011010001001
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Recepción de la trama manipulada con para no detectar el error

```
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> & "C:/Users/Fernando L
Ingrese el mensaje en binario con información adicional: 11000011101001
No se detectaron errores. Trama recibida:
11000011101001
PS E:\UNIVERSIDAD\8VO SEMESTRE\REDES\Lab2_Redex> █
```

Discusión

El laboratorio se enfocó en comprender los distintos tipos de algoritmos que existen para la detección y resolución de errores al enviar tramas de mensajes. Durante este laboratorio se implementaron dos algoritmos distintos en dos lenguajes de programación distintos. En este caso se hizo uso de los algoritmos de CRC-32 y Hamming en los lenguajes de programación Python y Java.

Los programas escritos en Java fueron los encargados de enviar la información, mientras que los programas escritos en Python fueron los encargados de recibir la información con los extras que se integraban por medio de cada uno de los algoritmos.

Al momento de realizar las pruebas con cada uno de los algoritmos se puede observar que la información que se agregaba mediante cada uno de los algoritmos en su versión de emisor fue correcta, debido a que cada uno de las implementaciones logró brindar las tramas de mensajes con los datos correspondientes que se deberían de agregar. Además, que en el caso de la detección del mensaje, el algoritmo de recepción es capaz de detectar que la información adicional de cada uno de los mensajes esté correctamente escrita, y en el caso de la implementación del CRC-32, puede indicar cuál es el mensaje que se recibe.

Para el caso de la detección de 1 bit de error, las dos implementaciones de los algoritmos son capaces de detectar y corregir el error que se tenga al momento de verificar la información que se tiene en la recepción de los mensajes.

En el caso de la detección de 2 bits modificados ocurre un fenómeno muy interesante, esto se produce debido a que en el caso del algoritmo de CRC-32 permite la detección de error de los mensajes a pesar que se hayan modificados 2 o más bits. El caso contrario pasa cuando se realiza la prueba con el algoritmo de Hamming debido a que este logra detectar error pero solo logra corregir uno de los errores que se puedan ingresar al momento de brindar la trama a la implementación de recepción. Esto se debe a que el algoritmo de Hamming normal está diseñado para lograr detectar 1 bit de error, por lo tanto al momento de realizar la detección de los errores brinda el valor de una trama corregida, la cuál no es la que se envió al principio. Esto se debe al diseño del algoritmo con respecto a tomar en cuenta los bits de paridad, por lo tanto en algunas ocasiones el algoritmo toma en cuenta otros bits cambiados como si fueran parte del mensaje (DEL, 2023).

Por último, se hizo la prueba sobre el cambio de bits de manera que cada una de las implementaciones logre pasar desapercibido el error, en este caso se tuvo resultados que demuestran la fortaleza del algoritmo de CRC-32, debido a que no se pudo encontrar una combinación fácil de modificar para lograr que el algoritmo no detectara errores, en este caso se debería de modificar la información de manera completa para lograr el cometido, aunque para el caso sería mejor utilizar una trama distinta. Esta fortaleza del algoritmo se debe a la forma en la que está diseñado sobre la premisa del uso del polinomio y las operaciones de división que se hacen entre los valores, lo cuál le brinda una mayor seguridad al momento de la detección de tramas que puedan tener errores entre ellas (Izaskun Pellejero et al.).

En el caso del algoritmo de Hamming se logró hacer que el mensaje pasara desapercibido pero esto implicó realizar varios cambios entre la trama que se recibía, la cuál en algunos casos se tenían que cambiar 3 bits distintos para lograr el cometido, y estos cambios se debían realizar en lugares estratégicos para lograr hacer que el algoritmo tuviera en cuenta los bits de paridad que este requiere.

Comentario grupal

Para este laboratorio se tuvo la oportunidad de adentrarnos en el funcionamiento de los algoritmos de detección y corrección de errores, y a partir de eso logramos entender mejor el funcionamiento de los algoritmos tanto de detección como de corrección.

Con el algoritmo CRC-32, nos ocurrió que a pesar de realizar los cálculos correctamente, a la hora de interpretar el resultado de la división binaria, se seguía corriendo hacia el bit más significativo. Esto llevó a obtener resultados incorrectos e hizo que analizáremos paso a paso cada cálculo y poder detectar el error.

Para el algoritmo de Hamming nos topamos con la situación de que para el algoritmo emisor llenamos las casillas de los bit de “adelante para atrás” mientras que para el receptor lo hacíamos de la manera contraria. Lo curioso fue que funcionaba bien con palíndromos, por obvias razones entonces nos fue un poco más difícil detectar este error.

Pero al final de cuentas, el habernos topado con estos pequeños detalles de los algoritmos hizo que pudiéramos entender con mejor detalle el funcionamiento de los algoritmos y cómo es que cosas tan pequeñas pueden influir tanto la detección de los errores a la hora del paso de los mensajes.

Conclusiones

En general, los dos algoritmos demostraron ser muy eficientes al momento de realizar las detecciones de los errores, en especial el algoritmo de CRC-32 que permite lograr detectar varios errores y cambios de datos entre las tramas de mensaje, lo cual para lograr la detección de errores en envíos lo vuelve un algoritmo muy confiable.

En el caso del Hamming también permite una detección segura de los errores pero al momento de corregir puede llegar a tener problemas, por lo tanto este solo se puede utilizar si se espera que solo se tenga un bit de error, en el caso de que se necesite corregir 2 bits o más se puede hacer uso del algoritmo de Hamming extendido, que permite brindar un diseño más robusto en la corrección de errores al momento de verificar mensajes.

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.