**Universidad Peruana de Ciencias Aplicadas**



PROYECTO

**TRABAJO FINAL**

CURSO DE REDES Y PROTOCOLO DE COMUNICACIONES

Profesora:

Daly Scaletti Corrado, Pablo Guillermo

Sección:

 CC61

Trabajo que presentan los alumnos:

Baldeon Albornoz,Braulio u201615681

Trujillano Figueroa,Alexander u201617272

Ysique Neciosup, José u201616544

Monterrico, noviembre de 2019

Contenido

[Introducción 3](#_Toc25079288)

[Objetivo del Estudiante 3](#_Toc25079289)

[Implementación de la solución 3](#_Toc25079290)

[- Consideraciones previas 3](#_Toc25079291)

[- Dataset 3](#_Toc25079292)

[- Búsqueda 3](#_Toc25079293)

[- CRC 4](#_Toc25079294)

[- Selección del mejor polinomio 4](#_Toc25079295)

[Conclusiones 4](#_Toc25079296)

[Anexos 5](#_Toc25079297)

[Bibliografía 6](#_Toc25079298)

# Introducción

CRC (Cyclic Redundancy Check) es un método que se usa comúnmente para detectar cambios accidentales en los datos transmitidos a través de redes de telecomunicaciones y dispositivos de almacenamiento. Desafortunadamente, la capacidad de detección de errores de los polinomios de CRC depende de la longitud de los datos. Por ese motivo, el objetivo de este trabajo es la elaboración de un programa que pueda determinar el polinomio de CRC adecuado para cada mensaje. Se toma como base la tabla 3 presentada en el paper (Anexo 1) de polinomios candidatos. Debido a que contiene un conjunto de 62 polinomios, los cuales proporcionan un buen rendimiento para longitudes de hasta 2048 bits con tamaños de CRC de 3 a 16 bits.

# Objetivo del Estudiante

* Implementar un algoritmo de CRC
* Comprobar la búsqueda de un polinomio CRC óptimo para una determinada trama según los polinomios candidatos.

# Implementación de la solución

## Consideraciones previas

La solución consiste en ingresar un mensaje cualquiera según el usuario, transformarla en su forma binaria y calcular su longitud en su forma binaria. En segundo lugar ingresar el HD deseado. Luego, buscar el mejor polinomio a través de la función *FindMinimumSizeCRC*, según la longitud del mensaje y el HD ingresado. Al conseguir el mejor polinomio posible se hará la comprobación del mismo. Se realizará la búsqueda de la forma binaria del polinomio seleccionado y se realizará el algoritmo de *CRC*. El output será la comprobación de errores en la transmisión de datos según el mensaje ingresado y el polinomio resultante.

En cuanto a la implementación de la solución, se utilizó el lenguaje Python, debido a la facilidad de incorporación de librerías tales como numpy y pandas. El primero ayuda al manejo de arreglos y el segundo, al manejo de dataset.

## Dataset

Se recopila, en un archivo .csv, los datos de la tabla 3 presentada en el paper (Anexo 1) de polinomios candidatos. Se ordena según se presenta en la tabla con sus respectivos atributos:

* nombre del polinomio
* tamaño de trama permitida
* tamaño de CRC (bits)
* máximo longitud de HD.

En segundo lugar se selecciona, en un archivo .csv, todos los posibles polinomios CRC candidatos y su forma binaria, según la tabla 4 presentada en el paper.

## Búsqueda

Se implementó dos algoritmos de búsqueda entre los polinomios candidatos. El primero de ellos (SearchGoodPolynomian) busca el polinomio que proporcionará la mejor HD posible según la longitud y tamaño de CRC requerido. De la lista de polinomios candidatos, el output será la máxima capacidad de longitud de trama para un tamaño CDC requerido. El segundo (FindMinimumSizeCRC), busca el CRC más pequeño que puede proporcionar el HD deseado con la longitud requerida. De la lista de polinomios candidatos, el output será el mínimo CRC posible.

## CRC

El CRC es un código de detección de errores comúnmente utilizado en redes digitales y dispositivos de almacenamiento para detectar cambios accidentales en datos sin procesar. Los bloques de datos que ingresan a estos sistemas obtienen un breve valor de verificación adjunto, basado en el resto de una división polinómica de sus contenidos.

Los bits representados de entrada son alineados en una fila y el divisor CRC, llamado polinomio, se coloca debajo de la parte izquierda del final de la fila. Luego se realiza una comparativa de bit a bit por el operador XOR (Anexo 2).

El divisor es entonces desplazado hacia la derecha, y el proceso se repite hasta que el divisor llega a la derecha, en la parte final de la fila de entrada. Este proceso lo realiza la función llamada CRC() (Anexo 3).

Todo este proceso lo hace el emisor de la trama. Luego el resto de esta división se agrega al final de la trama de datos, función encodeData() del programa, , y esta nueva trama se envía al receptor del mensaje que vuele a realizar el CRC con la función decodeData() (Anexo 4). Si el resto de este nuevo CRC está compuesto íntegramente de ceros, el mensaje enviado no obtuvo errores. En caso contrario la cantidad de 1 en el resto muestra la cantidad de bits que fueron cambiados durante el envió de la trama.

## Selección del mejor polinomio

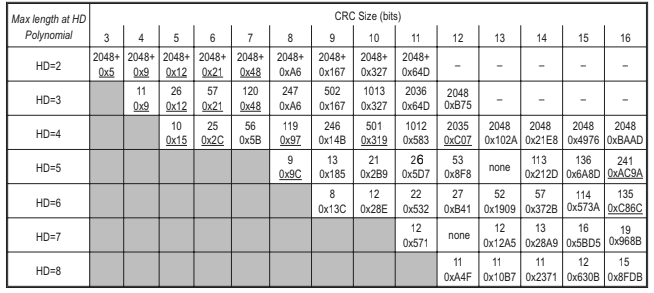
Se pide al usuario que ingrese un mensaje que desee enviar, el cual es convertido en su representación binaria. Luego decide si quiere buscar el polinomio por mejor HD posible o CRC SIze, dependiendo de su elección se ejecutan las funciones SearchGoodPolynomian o FindMinimumSizeCRC. Una vez obtenido el mejor polinomio posible, se hace una consulta al dataset Polinomio en búsqueda de la forma binaria del polinomio con la función binaryPoli().

Con el polinomio obtenido se lo pasa como parámetro a la función encodeData() junto a la trama de datos para codificar la imagen. Por último, se llama a la función decodeData() junto a la trama codificada y el polinomio para corroborar si no hubo error en el envió de la trama.

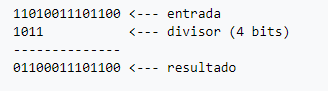
# Conclusiones

Como se menciona, el CRC es un código de detección de errores comúnmente utilizado en redes digitales y dispositivos de almacenamiento para detectar cambios accidentales en datos sin procesar. Esto para salvaguardar los datos que son enviados y así evitar la llegada de tramas de datos modificadas o incompletas, pero en la actualidad existe un uso no eficiente de este algoritmo ya que debido a la existencia de muchos polinomios a ser usados no siempre se usa uno que pueda responder adecuadamente a la tarea o especificaciones que el sistema demanda. Por ello, este trabajo desarrolla un software que permite determinar cuál es el mejor polinomio a partir de tabla 3 presentada en el paper (Anexo 1).

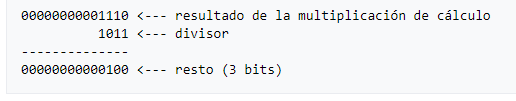
# Anexos



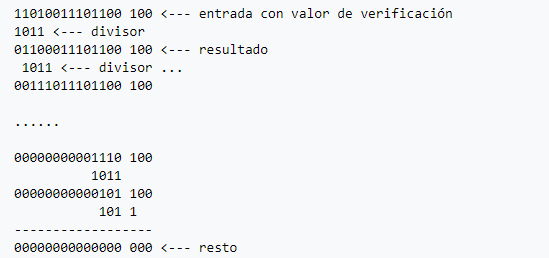
Anexo 1



Anexo 2



Anexo 3



Anexo 4

# Bibliografía

* Saleh Alrkiyan (2017) Qassim University.  “Error control coding techniques Cyclic Redundancy Check ”.  PowerPoint.  Recuperado de <https://www.researchgate.net/publication/312021156_Cyclic_Redundancy_Check_CRC>
* Koopman Philip(2004)“Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks” Recuperado de <https://users.ece.cmu.edu/~koopman/roses/dsn04/koopman04_crc_poly_embedded.pdf>