

**REPORTE 5. Algunas alternativas elegidas para invertir de manera adecuada un bit.**

## Contenido

Introducción: Detalles del proceso de inversión de bits.....	3
--------------------------------------------------------------	---

## Introducción: Detalles del proceso de inversión de bits

El proceso de inversión de bits que se plantea consiste en los siguientes pasos:

1. Primero, es necesario posicionarnos en el inicio del paquete RTP donde se va a aplicar el proceso de inversión de bits.
2. Ahora, vamos recorriendo en un ciclo el paquete RTP, para poder hacer este paso de manera adecuada, es necesario conocer el número de bits que conforman el paquete RTP. A partir del bit inicial del paquete RTP, se va eligiendo un punto de referencia para elegir en qué lugar se va a invertir el bit: si en el lado derecho o en el lado izquierdo. Esto significa que la posición de cambio es igual al bit de referencia más una elección de inversión, la siguiente fórmula es una representación más concisa:

$$\text{PosicionCambio} = \text{BitDeReferencia} + \text{ElecciónDeInversión}$$

En la ecuación anterior, la *ElecciónDeInversión* puede ser un valor positivo o negativo. Esto significa que nos podemos mover al lado derecho o izquierdo de *BitDeReferencia*. El valor de *BitDeReferencia* va aumentando porque se le suma un valor aleatorio que debe ser menor que el promedio de los códigos de Huffman y mayor que cero, esto debe ser así para asegurar que el valor de la *ElecciónDeInversión* pueda ocurrir en ambas direcciones.

3. A continuación, vamos a mostrar una lista de **algunos posibles** valores que *PosicionCambio* puede tomar. También, en dicha lista se especifica la posición correspondiente en el arreglo de caracteres que almacena la información que se está procesando. Hay que tener en cuenta que se utiliza un arreglo de caracteres para guardar la información a cifrar, cada elemento del arreglo tiene un tamaño de 8 bits.

PosicionCambio	Posición en el arreglo de datos.
Primer segmento de 50 bytes	
1	0
7	0
378	47
384	48
393	49
Segundo segmento de 50 bytes	
1	50
11	51
307	88
391	98
397	99

4. Pero, ¿Cómo se encontró la posición en el arreglo de datos donde está el bit indicado por la variable *PosicionCambio*? Esto se hace mediante la siguiente instrucción: **datosArchivo[refRTP + (posicionInversion/8)]**. En la expresión anterior, refRTP apunta al primer elemento de cada paquete RTP. Para el ejemplo que se muestra en este documento, cuando se trabaja con el primer segmento, entonces refRTP=0, cuando se aplica el proceso de inversión de bits al segundo segmento, entonces refRTP=50, recordemos que cada paquete RTP está formado por 50 bytes. Para encontrar la posición adecuada del bit de inversión a la variable refRTP se le suma el valor dado por (posicionInversion/8), ésta es una división entera que nos sirve para localizar la posición donde se encuentra el bit a invertir. Se divide entre 8 porque es el tamaño en bits del cada uno de los elementos del arreglo de datos. A continuación se vuelve a mostrar la tabla anterior, pero ahora especificando el resultado de la operación (posicionInversion/8).

PosicionCambio	Posición en el arreglo de datos.
Primer segmento de 50 bytes	
1	refRTP + (posicionInversion/8) = 0+(1/8)=0
7	refRTP + (posicionInversion/8) =0+(7/8)=0
378	refRTP + (posicionInversion/8) =0+(378/8)=47
384	refRTP + (posicionInversion/8) =0+(384/8)=48
393	refRTP + (posicionInversion/8) =0+(393/8)=49
Segundo segmento de 50 bytes	
1	refRTP + (posicionInversion/8) = 50+(1/8)=50
11	refRTP + (posicionInversion/8) = 50+(11/8)=51
307	refRTP + (posicionInversion/8) = 50+(307/8)=88
391	refRTP + (posicionInversion/8) = 50+(391/8)=98
397	refRTP + (posicionInversion/8) = 50+(397/8)=99

5. Ahora, una vez que hemos encontrado la posición en el arreglo de datos donde se encuentra el bit, es necesario invertir el bit indicado, esto se hace mediante la siguiente expresión:

**datosArchivo[refRTP + (posicionInversion/8)]=datosArchivo[refRTP + (posicionInversion/8)]^(1<(< 7 - (posicionInversion&7) )).**

A continuación, vamos a especificar algunos puntos importantes de la expresión anterior.

La expresión **posicionInversion&7** se utiliza para acotar el valor de posicionInversion a un valor entre 0 y 7, **este valor nos indica cuál es la posición del bit** a invertir en un grupo de ocho bits, el primer elemento tiene una posición cero, el segundo elemento tiene una posición uno y así sucesivamente hasta llegar al octavo elemento con una posición siete, el valor de posición depende de los bits que estén prendidos en los tres dígitos menos significativos del

valor de *posicionInversion*; a continuación, mostramos una tabla con algunos ejemplos:

posicionInversion	posicionInversion&7
1	1
7	7
11	3
307	3
378	2
384	0
391	7
393	1
397	5
399	7

La expresión 7 nos sirve para posicionar el valor 1 en el lugar correcto para que coincida con el bit de los datos que se va a invertir; por ejemplo, si *posicionInversion* =7, significa que vamos a invertir el bit menos significativo de un código cuya longitud es de 8, esto ocurre porque  $7 \& 7 = 7$ , y **significa que el valor de 1 se va a recorrer cero lugares a izquierda**, por tal motivo, la expresión completa: **( 7 - (*posicionInversion*&7) ) nos devuelve un valor igual a cero**. la operación XOR es la que invertirá el bit elegido, esta operación hace lo siguiente: se hace el cálculo bit a bit entre la posición del arreglo que contiene el bit a invertir y el valor que se obtiene al hacer el desplazamiento a la izquierda del valor de 1, éste valor está formado por un bit “prendido” en el mismo lugar donde está el bit a invertir y los demás bits “apagados”, esta configuración respeta a los bits que están en las posiciones donde no se va a invertir nada y sólo se invierte el bit elegido. Un ejemplo lo ilustra mejor:

Bit a cambiar en fondo azul:

1	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---

Desplazamiento de 1 necesario:

0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---

Aplicación de operación XOR con los dos bytes anteriores y resultado:

1	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---

6. **El paso 5 es una manera obsoleta de invertir bits**, esto es así por la cantidad de operaciones que se tienen que aplicar a la hora de hacer una inversión de un bit: restas, y desplazamientos de bits en cada momento, y si tomamos en cuenta que se hacen miles de inversiones de bits, entonces también se ejecutarán miles de operaciones. Para evitar lo anterior, existe una manera muy rápida de ahorrarse las operaciones innecesarias: declarar un arreglo con un número de elementos al tamaño en bits de un carácter, cada elemento tiene un número formado por un bit encendido en alguna de las posibles posiciones de un código y los demás bits apagados. Por ejemplo, si tenemos una longitud promedio de 8 bits, se tendrá un arreglo de la siguiente manera:

**unsigned char flip[8];**

**flip[0]=128; flip[1]=64; flip[2]=32; flip[3]=16;**

**flip[4]=8; flip[5]=4; flip[6]=2; flip[7]=1;**

Al final, la operación de inversión de bits queda de una manera muy sencilla y con menor complejidad de cálculo:

**datosArchivo[refRTP +(posicionInversion/8)]^=flip[posicionInversion&7];**