ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 22 – SEPTIEMBRE DE 2009

"¿QUIERES COMPROBAR CÓMO LAS REDES DETECTAN Y CORRIGEN ERRORES?"

AUTORÍA **MARÍA CATALÁ CARBONERO**
TEMÁTICA **DETECCIÓN Y CORRECCIÓN DE ERRORES**
ETAPA **CICLO MEDIO Y SUPERIOR DE INFORMÁTICA**

Resumen

Hoy en día las redes constituyen uno de los ejes más importantes de la sociedad, ya que alrededor de ellas trabajan miles de personas. Uno de los puntos en los que más se está trabajando es que el paso de información a través de las mismas se realice con toda seguridad. Por ello existen una serie de estrategias que sirven para detectar y corregir estos posibles errores que se pueden ocasionar. En el siguiente artículo se especificarán los métodos más importantes a través de ejemplos y ejercicios sencillos.

Palabras clave

Redes, seguridad, detección, corrección, errores

1. INTRODUCCIÓN

Hoy en día existen dos estrategias para la detección y corrección de errores, estas son las siguientes:

- *Códigos de corrección de errores:* en los canales que causan muchos errores es mejor agregar la redundancia suficiente a cada bloque, para que el receptor lo corrija sobre la marcha, canales inalámbricos (técnicas FEC, corrección de errores en el destino, no precisan retorno, ni control de flujo.
- *Códigos de detección de errores*: canales como la fibra o cobre altamente confiables, es más económico detectar y retransmitir (técnicas ARQ, petición automática de retransmisión).

A continuación vamos a especificar una serie de ejemplos sencillos que mostrarán al alumnado la forma en el que las redes realizan esta corrección y detección de errores.

REVISTA DIGITAL INNOVACIÓN Y EXPERIENCIAS EDUCATIVAS

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 22 – SEPTIEMBRE DE 2009

2. CÓDIGOS DE CORRECCIÓN DE ERRORES

Para entender la manera en que pueden manejarse los errores, es necesario estudiar de cerca lo que es en realidad un error. Por lo general, una trama consiste en *m* bits de datos y *r* bits redundantes o de verificación. Si la longitud total *n(n=m+r)*, a una unidad de *n* bits que contiene datos y bits de verificación se le conoce como *palabra codificada de n bits*.

2.1. Ejercicio1

Dadas dos palabras codificadas cualesquiera, digamos 10001001 y 10110001, ¿es posible determinar cuántos bits correspondientes difieren?

Solución

En este caso, difieren tres bits. Para determinar la cantidad de bits diferentes, basta con aplicar una operación XOR (iguales=0, distintos=1) a las dos palabras codificadas y contar la cantidad de bits 1 en el resultado.

La cantidad de posiciones de bits en la que difieren dos palabras codificadas se llama *distancia de Hamming (d).*

2.2. Ejercicio2

El significado del código de Hamming es que si dos palabras codificadas están separadas una distancia *d*, se requerirán *d* errores de 1 bit para convertir una palabra en la otra. En la mayoría de las aplicaciones de transmisión de datos, todos los 2^m mensajes de datos posibles son legales, pero debido a la manera en que se calculan los bits de verificación no se usan todas las 2^n palabras codificadas posibles.

Dado el algoritmo de cálculo de los bits de verificación, es posible construir una lista completa de palabras codificadas legales y encontrar, en esta lista, las dos palabras codificadas cuya distancia de Hamming es mínima. Especifica esta lista para m=2 y n=5, como muestra la siguiente tabla:

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 22 – SEPTIEMBRE DE 2009

Bloque de datos	Palabras código
00	00000
01	00111
10	11001
11	11110

Solución

En primer lugar calcularemos la distancia mínima entre todas las palabras código:

- d(1,2)= 3;
- d(1,3)= 3;
- d(1,4)= 4;
- d(2,3)= 4;
- d(2,4)= 3;
- d(3,4)= 3;

Como podemos observar la dmin=3

Si recibiéramos una palabra 00100, para calcular la distancia mínima con respecto a las palabras código haríamos:

- d(00000,00100)= 1;
- d(00111,00100)= 2;
- d(11001,00100)= 4;
- d(11110,00100)= 3;

Por lo tanto la distancia mínima sería la de la primera palabra código (dmin=1) y de esto podríamos sacra que en realidad lo que se mando fue un 00000 y no un 00100, hubo un error en la entrega!.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 22 – SEPTIEMBRE DE 2009

2.3. Ejercicio3

Un código con un solo bit de paridad tiene distancia de 2, pues cualquier error de un bit produce una palabra codificada con la paridad equivocada. Este sistema de agregar un solo bit de paridad a los datos (ya sea par o impar) puede utilizarse solo para detectar errores individuales.

Si quisiéramos diseñar un código con *m* bits de mensaje y *r* bits de verificación que permitiera la corrección de todos los errores individuales, cada uno de los 2^m mensajes legales tendrían *n* palabras codificadas ilegales a una distancia 1 de él. Éstas se forman invirtiendo en forma sistemática cada uno de los *n* bits de la palabra codificada de *n* bits que la forman.

Por lo tanto, cada uno de los 2^m mensajes legales requieren n+1 patrones de bits dedicados a él. Dado que la cantidad de patrones de bits es 2^n, debemos tener $(n+1)2^m <= 2^n$. Usando n=m+r, este requisito se vuelve $(m+r+1) <= 2^r$. Dado m, esto impone un límite inferior a la cantidad de bits de verificación necesarios para corregir errores individuales.

Para lograr este límite inferior teórico, se usa un método de Hamming, donde los bits de la palabra codificada se numeran en forma consecutiva. Los bits que son potencia de 2 (1,2,4,8..) son bits de verificación, el resto (3,5,7,9..) se rellenan con los *m* bits de datos.

Cada bit de verificación obliga a que la paridad de un grupo de bits, incluyéndolo a él, sea par (o impar). Un bit puede estar incluido en varios cálculos de paridad.

El ejercicio consiste en si me llega una palabra de código 00110010000, especificar como realizaría la paridad par, y como codificaría el error si el bit número 6 cambiara a 1, es decir, que la palabra de código que recibiría fuera 00110110000

Solución

- En primer lugar, se identifican los bits de verificación de los bits de datos:

bits	1	2	3	4	5	6	7	8	9	10	11
palabra	0	0	1	1	0	0	1	0	0	0	0

Los bits que están sombreados (1, 2, 4, y 8) son los bits de verificación, y el resto (3, 5, 6, 7, 9, 10, y 11) son los bits de datos.

INNOVACIÓN Y EXPERIENCIAS EDUCATIVAS
REVISTA DIGITAL

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 22 – SEPTIEMBRE DE 2009

- En segundo lugar, y para conseguir la paridad par se tendrán que ver los bits de verificación, y para ello se hace una operación XOR con los bits de datos que influyen es ese bit de verificación.
- Para la realización de este punto sería bueno especificar una tabla en la que las *columnas* representen a los bits de verificación y las *filas* correspondan a los bits de datos y así realizar la XOR. :

	1	2	4	8
3	1	2		
5	1		4	
6		2	4	
7	1	2	4	
9	1			8
10		2		8
11	1	2		8

1 → 3, 5, 7, 9, 11, cuyos valores son: 10100 → (XOR) bit de paridad =0

2 → 3, 6, 7, 10, 11, cuyos valores son: 10100 → (XOR) bit de paridad=0

4 → 5, 6, 7, cuyos valores son: 001→ (XOR) bit de paridad=1

8 → 9, 10, 11, cuyos valores son 000 → (XOR) bit de paridad=0

- Llegado a este punto, la pregunta es ¿Cómo codifico el error si me llega la cadena 00110110000? Pues los pasos a seguir serían los siguientes:
  - Se inicializa un contador, k=0. Como en el bit 6 participan los bits de verificación 2 y 4 entonces:

  1 → 3,5,7,9,11, aquí el bit 6 no está presente, por lo que es correcto, y como no hay error, k se queda como estaba, en este caso k=0.

  2 → 3, 6, 7 ,10, 11, en este caso el bits 6 si participa, por lo que hay un error, entonces k+=2.

  4 → 5, 6 ,7, habrá error, k+=4 → k=6

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 22 – SEPTIEMBRE DE 2009

8 → 9, 10, 11 no habrá error y k=6.

Por lo tanto como k=6, el error a ocurrido en el bit 6, tal y como habíamos especificado en el enunciado.

3. CÓDIGOS DE DETECCIÓN DE ERRORES

Los códigos de corrección d errores se utilizan de manera amplia en los enlaces inalámbricos, que son notoriamente más ruidosos y propensos a errores que el alambre o la fibra óptica. Sin embargo, a través del cable de cobre o fibra óptica, la tasa de error es mucho más baja, por lo que la detección de errores y retransmisión es más eficiente que enviar códigos redundantes en cada trama.

En la práctica, para detectar este tipo de errores se usa un método muy definido: el código polinomial (también conocido como código de redundancia cíclica CRC) que se basa en el tratamiento de cadenas de bits como representaciones de polinomios con coeficientes 0 y 1 solamente.

Un ejemplo sería el siguiente: si tengo la cadena 110001 de 6 bits, se representará por un polinomio de seis términos con coeficientes 1,1,0,0,0,1 y el polinomio quedaría así $x^5 + x^4 + x^0$

Cuando se emplea este método de código polinomial, se deben seguir los siguientes pasos:

- el emisor y el receptor deben acordar por adelantado un polinomio generador G(x). Tanto los bits de orden mayor y menor del generador deben ser 1.
- Para calcular la suma de verificación, para una trama con m bits, correspondiente al polinomio M(x), la trama debe ser más larga que el polinomio generador.
- La idea es incluir una suma de verificación al final de la trama de tal manera que el polinomio resultante sea divisible entre G(x).
- Cuando el receptor recibe la trama con la suma de verificación intenta dividirla entre G(x), y si hay un residuo habrá habido un error de transmisión, y en caso contrario no lo ha habido.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 22 – SEPTIEMBRE DE 2009

3.1. Ejercicio1

Calcular el método de código polinomial para la trama 1101011011, si sabemos que el código generador en el que se han puesto de acuerdo el emisor y receptor es el 10011.

Solución

- En primer lugar tendremos que saber el grado de G(x) y anexar tantos ceros al final de la cadena a transmitir como el grado de G(x).

G(x) = 10011 -> grado 4

Cadena = 11010110110000

- En segundo lugar, voy dividiendo la cadena de bits (el dividendo), entre G(x) (el divisor).

```
1 0 0 1 1 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0
            1 0 0 1 1
              1 0 0 1 1
              1 0 0 1 1
                0 0 0 0 1
                0 0 0 0 0
                  0 0 0 1 0
                  0 0 0 0 0
                    0 0 1 0 1
                    0 0 0 0 0
                      0 1 0 1 1
                      0 0 0 0 0
                        1 0 1 1 0
                        1 0 0 1 1
                          0 1 0 1 0
                          0 0 0 0 0
                            1 0 1 0 0
                            1 0 0 1 1
                              0 1 1 1 0
                              0 0 0 0 0
                                1 1 1 0   Resto
```

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 22 – SEPTIEMBRE DE 2009

- Por lo tanto, la trama transmitida final será la cadena de bits original, anexándole el resto d ela operación que se realizó anteriormente. Así como el receptor sabe el polinomio generador, tendrá que realizar la operación inversa, y así sabrá si ha habido un error o no.

Cadena original = 1 1 0 1 0 1 1 0 1 1

Resto= 1 1 1 0

Cadena transmitida= Cadena Original + Resto

1 1 0 1 0 1 1 0 1 1 | 1 1 1 0

4. BIBLIOGRAFÍA

- Tanembaum, A. (2003). *Redes de Computadoras.* Madrid: Prentice Hall
- Stallings, W. (2004). *Redes e Internet de Alta Velocidad. Rendimiento y Calidad de Servicio*. Madrid: Prentice-Hall

Autoría

- María Catalá Carbonero
- IES Florencio Pintado, Peñarroya-Pueblonuevo, Córdoba
- mcata44@hotmail.com