



Universidad Abierta Interamericana

Maestría en Tecnología Informática

Redes y Sistemas Distribuidos

Investigación sobre los Turbo Codes aplicados a las telecomunicaciones.

Ing. Víctor M. Acuña

2004

Introducción

Presentados originalmente en 1993 durante la Conferencia Internacional de Comunicaciones, en Ginebra, por los ingenieros franceses Claude Berrou y Alain Glavieux, los Turbo Codes son una nueva forma de codificación de la información que permiten mejorar el rendimiento de un canal de comunicaciones. Permitiendo obtener comunicaciones sin error, como nunca antes. Ya que con su aplicación se puede duplicar la tasa de transferencia de datos sin aumentar la potencia del transmisor. Fue algo increíble a primera vista, aunque una vez replicados los resultados por otros investigadores, los Turbo Codes revolucionaron la corrección de errores. Saltando del campo de la investigación al de la implementación; grandes compañías como, France Telecom., NTT DoCoMo, Sony, NEC, Lucent, Samsung, Ericsson, Nokia, Motorola y Qualcomm tienen grupos de trabajo implementando la próxima generación de equipos de comunicaciones con esta codificación de errores.

Basamento teórico

Para entender su significación hay que remontarse al Ing. Claude Shannon (1916 – 2001) que en 1948 publicó un trabajo titulado “La teoría matemática de las comunicaciones”.

En éste Shannon definió lo que era en ese entonces un concepto difuso, lo que significa “la información” para los ingenieros en comunicaciones y propuso una manera precisa para cuantificarla, en esta teoría la unidad de información fundamental es el bit.

Shannon mostró que cada canal de comunicaciones tiene una tasa máxima de transferencia para una comunicación confiable, lo que se llama capacidad del canal, medida en bits por segundo. Y demostró que utilizando ciertos esquemas de codificación se podían transmitir datos hasta alcanzar la capacidad máxima del canal, virtualmente sin errores (lo que fue una idea asombrosa), siendo un salto intelectual en su tiempo.

La capacidad del canal comenzó a ser un parámetro fundamental para la ingeniería de las comunicaciones, una medida de lo que un sistema puede o no transmitir. Expresada en muchos casos por la famosa fórmula,

$$C = W \log_2 (1 + P/N)$$

Donde **C** es la capacidad en bits por segundo, **W** es el ancho de banda en Hertz, **P** es la potencia de transmisión en Watts y **N** es la potencia de ruido también en Watts.

Esta teoría está embebida de manera invisible en todas las comunicaciones, desde las sondas espaciales a los CD players, desde los teléfonos celulares a las redes de computadoras, en todo canal de comunicaciones, para la codificación de la corrección de error, se toman como parámetros, los conceptos de Shannon.

Los Turbo Codes, desde el nicho tecnológico de los enlaces satelitales y las sondas espaciales estarán incorporados a la próxima generación de teléfonos celulares, permitiendo que estos dispositivos portátiles puedan manejar datos multimedia como ser video e imágenes, dentro de los muy ruidosos canales de las comunicaciones celulares.

Y se esta desarrollando el uso de los Turbo Codes para la radiodifusión de audio y video digital, así como para incrementar la velocidad de datos en las próximas versiones mejoradas de las redes Wi-Fi y WiMax.

Shannon básicamente planteó que no es necesario gastar mucha energía y tiempo, si se tienen los esquemas de codificación correctos. Antes de su trabajo, los ingenieros, para reducir la tasa de errores en comunicaciones, necesitaban aumentar la potencia de transmisión ó enviar el mensaje repetidas veces.

Luego de él, el campo de la teoría de errores cambió y los desarrolladores pudieron diseñar códigos de buen desempeño, aunque, aún los mejores diseños, usualmente requieren más del doble de potencia de transmisión de lo que la Ley de Shannon predice para alcanzar cierto grado de confiabilidad (lo que es un gran gasto de energía). La diferencia entre el valor ideal y el práctico es de alrededor de 3,5 dB, siendo ésta la proporción entre el nivel de señal y el nivel de ruido, en una escala logarítmica. Para mejorar mas esta relación se necesitan códigos de corrección mas elaborados. Esta fue la meta que los desarrolladores buscaron durante casi 40 años, hasta la aparición de los Turbo Codes de Berrou y Glaveux. Con ellos mostraron que es posible alcanzar increíblemente hasta un 0,5 dB del limite de Shannon para una tasa de error de uno en 100.000. Y se espera que los Turbo Codes puedan superar esta pequeña brecha.

Shannon definió el codeword (clave) como la secuencia de bits correspondientes a un símbolo dentro de un determinado código.

Y esta formado por los bits de la información más un conjunto de bits extras llamados bits de paridad que ayudan a identificar y corregir los errores en el terminal receptor. El conjunto de bits resultantes (los datos mas los bits de paridad) es lo que se llama codeword ó codificación y típicamente representa: ó un bloque de caracteres, ó algunos píxeles de una imagen, ó unas muestra de voz ó alguna pieza de información. Con la colección correcta de codewords (en otras palabras, con el código correcto) es posible alcanzar la capacidad del canal.

Estrategias de corrección de errores

La respuesta de que código es el correcto quedo sin respuesta de parte de Shannon y debieron inventarse los códigos que pudieran cumplir con esta premisa.

Se han desarrollado dos estrategias fundamentales para el manejo de los errores que pueden presentarse en la transmisión de datos. Una de ellas consiste en incluir redundancia para que el receptor pueda detectar la existencia de un error (pero no de su tipo) y solicitar una retransmisión. La otra estrategia consiste en agregar información redundante en cada bloque de datos enviado, para que el receptor pueda deducir cuál fue el carácter que se envió. En el primer caso se utilizan *códigos detectores de error* y en el segundo caso *códigos correctores de error*, que son mas sofisticados.

Algoritmo de Chequeo de Redundancia Cíclica (CRC)

Este es un esquema de detección de errores que se basa en la transmisión de mensajes acompañados de un código de redundancia al final de la trama. Este mensaje se construye de forma tal que pueda ser dividido al momento de su recepción por un polinomio específico (representado por una serie de bits). El receptor conoce los

criterios que le permitirán determinar si el mensaje es el original o si ha sido alterado por algún error de transmisión. En ese caso, se descarta el mensaje.

El algoritmo utilizado es el siguiente:

- 1) Representar los mensajes de n bits como polinomios de grado $n-1$.
- 2) Agregar k bits redundantes en el extremo de orden inferior de la trama correspondiente a un mensaje de n bits. Donde k es el grado de un polinomio divisor $G(x)$, conocido por el emisor y el receptor.
- 3) Dividir la serie de bits resultantes del punto 2 entre la serie de bits correspondientes a $G(x)$, utilizando división en módulo 2.
- 4) Restar a la serie de bits resultantes del punto 2 el residuo (que debe tener longitud menor o igual a k) de la división definida en el punto 3. El resultado de esta operación es $T(x)$, la trama a transmitir.

Ejemplo

Trama: 1 1 0 1 0 1 1 0 1 1
 Generador: 1 0 0 1 1
 Mensaje después de añadir 4 bits 0: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

Resto \rightarrow 1 1 1 0

Trama transmitida: 1 1 0 1 0 1 1 0 1 1 1 1 0

Cálculo del código de redundancia cíclica (CRC)

En este ejemplo se utiliza como polinomio generador:

$$G(x) = x^4 + x + 1, \text{ por lo que } k = 4.$$

Una vez que se han representados los mensajes como polinomios, el receptor efectúa la división de, la trama original + los 4 bits de redundancia, entre el polinomio generador. El residuo de esta división se resta al mensaje (después de agregar los bits) para obtener la trama a transmitir.

Por otra parte, el receptor divide $T(x) + E(x)$ (la trama que recibe) entre $G(x)$. Si el resto es cero se debe a una de estas dos razones: $E(x)$ era cero (no hubo errores) o, era

exactamente divisible por $G(x)$. Este último caso es el que quiere evitarse, y para ello debe escogerse un polinomio generador adecuado.

Con este método, se pueden detectar los siguientes tipos de errores:

- * Errores de 1 bit, cuando los coeficientes de x^k y x^0 en $G(x)$ son diferentes de cero. En otras palabras, si $G(x)$ tiene dos o más términos, nunca podrá dividir a $E(x)$, por lo que se detectarán todos los errores de 1 bit. Nótese que en este caso, $E(x)$ es de la forma x^i , donde i determina el bit que contiene el error.
- * Errores de 2 bits, si $G(x)$ contiene al menos tres términos. Cuando ocurren dos errores aislados de 1 bit, $E(x)$ puede expresarse como: $x^i + x^j$, donde $i > j$ o, de forma equivalente, $x^j(x^{i-j} + 1)$. Luego, si se supone que $G(x)$ no es divisible por x , será una condición suficiente para la detección, el que no divida a $x^k + 1$, para cualquier k . (k acotado superiormente por la longitud máxima de la trama).
- * Un número impar de errores, cuando $G(x) = (x+1) C'(x)$. En este caso, $E(x)$ contiene un número impar de términos. Por otra parte, se ha demostrado que en álgebra módulo 2, ningún polinomio de este tipo, contiene como factor a $x+1$. Luego, si se agrega $x+1$ como factor al polinomio, se detectarán los errores que contengan un número impar de bits invertidos.
- * Ráfagas de error de longitud menor o igual que k , para un código polinómico con r bits de redundancia. Definiéndose una ráfaga como $x^i(x^{k-1} + \dots + 1)$, donde i indica qué tan lejos del extremo derecho de la trama, está ubicada la ráfaga. Si $G(x)$ contiene x^0 , no tendrá a x^i como factor, por lo tanto, si el grado es mayor que el grado de la expresión entre paréntesis, el resto no podrá ser cero. De esta manera se detectarán ráfagas de longitud menor o igual que k .
- * Ráfagas de más de k bits consecutivos, un gran porcentaje de veces. Se ha demostrado que la probabilidad de que una trama incorrecta pase inadvertida es $1/2^k$, suponiendo que los conjuntos de bits son semejantes.

Algunos de los polinomios generadores más comunes

CRC	$G(x)$
CRC-8	$X^8 + X^2 + X^1 + 1$
CRC-10	$X^{10} + X^9 + X^5 + X^4 + X^1 + 1$
CRC-12	$X^{12} + X^{11} + X^3 + X^2 + 1$
CRC-16	$X^{16} + X^{15} + X^2 + 1$
CRC-CCITT	$X^{16} + X^{12} + X^5 + 1$
CRC-32	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Por ejemplo el Polinomio $G(x) = X^{16} + X^{12} + X^5 + 1$ es el utilizado en los disquetes y discos rígidos.

podrán detectar d errores, de otra forma no podrían cambiarse los d errores de bit por una palabra válida.

Usando este método, el receptor podrá detectar y corregir errores producidos en la transmisión. Cuando reciba una palabra de código inválida, detectará la existencia de un error. Cuando se quieran corregir d errores, se necesitará una distancia de $2d+1$, porque de ese modo, las tramas legales estarán lo suficientemente separadas como para que inclusive con d cambios, la palabra original esté más cercana que cualquier otra y podrá ser determinada en forma única.

Algoritmo del método:

1. Se enumeran los bits en forma consecutiva comenzando por el bit 1, localizado en el extremo izquierdo.
2. Los bits que son potencias de 2 (1,2,4,8,...) son utilizados como bits de redundancia. El resto se llenan con los m bits de datos.
3. Cada bit de redundancia debe forzar la paridad de alguna colección de bits, incluyéndose a si mismo para que sea par (o impar).
4. Se rescribe k como una suma de potencias de 2, para saber a qué bits de redundancia contribuye el bit de datos de la posición k . Por ejemplo $13 = 1 + 4 + 8$.
5. El paso anterior se hace para comprobar el bit en base a aquellos que aparecen en su expansión. En el ejemplo anterior, 13 se comprobaría mediante el uso de los bits 1, 4 y 8.
6. El receptor inicia un contador en cero, cuando recibe una trama. Después, revisa cada bit de redundancia para ver si tiene la paridad correcta.
7. Si la paridad no es la esperada, el receptor suma el valor k al contador.
8. Si el contador es 0, después de haber revisado todos los bits de redundancia, la trama se acepta como válida. Sino, el contador tiene el número del bit incorrecto.

Ejemplo:

Caracter	ASCII	Bits de verificación
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	11111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	00101011111
d	1100100	11111001100
e	1100101	00111000101

Orden de transmisión de los bits

Uso del Código de Hamming para corregir errores a ráfagas.

En esta figura se muestran caracteres ASCII de 7 bits codificados como palabras de 11 bits, utilizando el código Hamming.

Este método permite corregir errores individuales y puede ser modificado para corregir errores por ráfagas

El límite práctico de los códigos de error

Shannon definió las tramas como puntos en el espacio. De esta manera el código 011 puede ser considerado como un punto en el espacio tridimensional, con coordenadas $x=0$, $y=1$, y $z=1$.

Las tramas con mas de tres bits son puntos en el hiperespacio. El ruido puede alterar los bits en las tramas y por lo tanto sus coordenadas en el espacio. Si dos puntos están cercanos entre si y el ruido alteró uno, puede suceder que caiga justo encima de otro resultando en un error en la decodificación. Entonces cuanto mas grandes sean las diferencias entre las tramas (cuan apartadas estén unas de otras en términos de codificación de bits) será mas dificultoso que el ruido cause errores.

Para mejorar la capacidad Shannon demostró que uno puede elegir aleatoriamente una cantidad de tramas infinitas. En otras palabras, yendo a la analogía espacial, si uno hace las tramas aleatorias y tan largas como quiera podrá poner los puntos arbitrariamente lejos entre si en el espacio.

De esta manera, esencialmente, no habrá posibilidad de que un punto caiga encima de otro.

Desgraciadamente los códigos aleatorios muy largos no son prácticos, porque tendrán un numero astronómico de tramas y este código será extremadamente lento ya que requiere un numero grande de bits para transmitir cada trama.

Los expertos se han concentrado en desarrollar códigos prácticos que puedan ser implementados en sistemas reales. Eligiendo en el desarrollo de códigos, bits de paridad que constriñan a las tramas a ciertos valores, las hace difícil de confundir unas con otras.

Por ejemplo suponiendo una trama de 8 bits (siete datos mas un bit de paridad), y que se desea que todas las tramas tengan una cantidad par de unos, haciendo a este bit de paridad uno si es necesario para cumplir con este requerimiento. Ahora si alguno de los ocho bits es modificado por el ruido (incluyendo al bit de paridad) el receptor sabrá que hubo un error, porque la paridad no se cumple, esto es la cantidad de unos es impar. Este esquema básico puede detectar un error pero no lo puede corregir, porque no sabe cual cambio. Como hemos visto anteriormente en la definición de códigos detectores de error.

Para corregir errores se necesitan mas bits de paridad. Aquí los expertos en códigos han desarrollado numerosas y cada vez mas sofisticadas maneras de generar bits de paridad. De allí los mencionados códigos correctores de error, como los códigos de bloque, códigos de Hamming, los Reed-Solomon y los convolutorios que son ampliamente usados para alcanzar tasas de error bajas.

El problema de la complejidad de un código se presenta en términos del costo de un código a la hora de ver la cantidad de computación requerida para decodificar los datos. Cuanto mas cerca se esté del límite de Shannon más complicado se volverá el proceso, ya que se necesitarán mas bits y las tramas se harán muy largas.

Para tramas de solo 3 bits, por ejemplo, hay un total de 2^3 o sea 8 tramas posibles. Pero, para alcanzar una capacidad practica se podrán necesitar tramas de unos 1000 bits y el decodificador necesitara buscar entonces en la inimaginable cantidad de 2^{1000} (aproximadamente 10^{301}) tramas. Para comparar se me ocurre mencionar que el numero estimado de átomos en el universo visible es de alrededor de 10^{80} .

Como conclusión de esto si se desea llegar lo mas cerca posible del limite de Shannon la complejidad de calculo se vuelve astronómica ya que estos códigos no tienen la capacidad para lograrlo. No había esperanza de ir mas allá hasta la aparición de los Turbo Codes.

Los Turbo Codes

Los Turbo Codes resuelven el complejo problema separándolo en componentes mas manejables.

En lugar de un solo codificador en el transmisor y un único decodificador en el receptor estos utilizan dos codificadores y dos decodificadores en cada extremo.

Alrededor de 1970 se descubrió que pasando los datos por dos codificadores en serie se mejoraba la capacidad de soportar errores, de manera que “el todo es mas que la suma de las partes”. Los Turbo Codes emplean dos codificadores pero que trabajan sinérgicamente, no en serie, sino en paralelo.

El proceso de los turbo codes comienza con tres copias del bloque de datos a ser transmitido. La primera copia va a uno de los codificadores donde un código convolucional toma los bits de datos y computa los correspondientes bits de paridad. La segunda copia va al segundo codificador que contiene un código convolucional idéntico. Este segundo codificador no toma los bits originales sino que los toma en otro orden, intermezclados por un sistema llamado *interleaver*. Este encoder entonces toma los bits entremezclados y computa la paridad a partir de ellos. Finalmente el transmisor toma la tercera copia de los datos originales y la envía junto a las dos cadenas de bits de paridad por el canal de comunicación.

Así la trama compuesta por los tres datos codificados se transmite por el canal.

Este re-arreglo de los bits por medio del *interleaver* es la llave de todo el éxito del sistema.

Básicamente esta permutación es la que brinda mas diversidad a los códigos, en la analogía espacial esto pone a los puntos mas alejados entre si en el espacio.

De manera que el rol de la permutación es introducir un factor aleatorio en el código. Dándoles un personalidad mas parecida a los códigos descriptos por Shannon para la transmisión de información.

Los Turbo Codes como cualquier otro código con gran cantidad de claves deberían estar limitados por la complejidad computacional. De hecho los turbo codes usualmente trabajan con claves que tienen alrededor de mil bits, un numero bastante alto. Esto es desalentador si se trabaja con un solo decodificador en el receptor. Pero los Turbo Codes utilizan dos decodificadores que trabajan juntos para resolver el problema de la complejidad.

En el receptor la trama recibida se separa de manera que el rol de cada decodificador es tomar los datos que pueden estar corruptos por el ruido del canal y decide cual es el valor mas apropiado, 0 o 1, para cada bit individual. Para tomar esta decisión cuenta sobremanera la interrelación que existe entre los dos, no es una elección aleatoria sino consensuada.

Cada decodificador cuenta también los pistas que le ayudan en la decisión de si un bit es 0 o 1. primero inspecciona la amplitud de la señal analógica de los bits recibidos.

Mientras la mayoría de los esquemas de decodificadores transforman la señal recibida en unos o ceros, dejan afuera una valiosa información sobre las fluctuaciones de la señal que nos puede decir mas acerca del estado de cada bit. Los turbo decodificadores

transforman la señal en números enteros que miden cuán confiables podemos estar de que un bit sea uno o cero. Además el decodificador analiza los bits de paridad para ver si los bits recibidos están íntegros o no.

El resultado de este análisis es esencialmente informar el estado de cada bit. Los que turbo codes realizan internamente es, además de trabajar con las decisiones del estado de cada bit, trabajar con la confiabilidad de que las decisiones fueron correctas. Esta confiabilidad del estado de cada bit se expresa en números llamados *relación de chance* – likelihood- o probabilidad que puede variar entre +127 y –127. Por ejemplo si el sistema trabaja con +7 y –7 recibir un rango de +7 significa que el decodificador esta seguro del valor del bit es 1, mientras un –5 significa que el decodificador piensa que el valor del bit es 0 pero no esta totalmente seguro.

Aún con el manejo de los niveles de la señal y la paridad como indicadores no son suficientes. Un decodificador simple aún no sería siempre capaz de hacer las decisiones correctas sobre los bits transmitidos y como comúnmente deberá trabajar con varias cadenas de bits erróneos se perderá en el universo de los códigos y la clave elegida no será siempre la correcta. Por eso un decodificador solo no puede hacer el trabajo. Pero la confiabilidad de un decodificador puede ser potenciada por otro y viceversa ya que las dos cadenas de bits de paridad se refieren a los mismos datos, pero arreglados de distinto orden. De manera que los dos decodificadores tratan de resolver el mismo problema visto desde perspectivas distintas.

Entonces los dos decodificadores intercambiarán la información de confiabilidad en un trabajo interactivo para mejorar su propia descodificación. Las cadenas de decisión se re-arreglarán de la manera que cada decodificador necesite. Así un bit que fue detectado con una fuerte probabilidad en un decodificador, influenciará la opinión del otro sobre ese bit. Cada decodificador tendrá entonces su propia opinión y una opinión externa para tomar la decisión sobre el estado de cada bit.

El corazón de la codificación turbo es este proceso de interacción donde cada uno toma ventaja del trabajo del otro. Luego de un cierto número de interacciones, típicamente entre 4 y 10, ambos decodificadores estarán de acuerdo en todos los bits. Esto significa que no se perdieron en el universo de los códigos sino que pudieron pasar la barrera de la complejidad de la descodificación.

Esta solución de “dividir y conquistar” consiste entonces en dividir el problema en varias pequeñas partes, resolverlas, y luego volver a juntarlas.

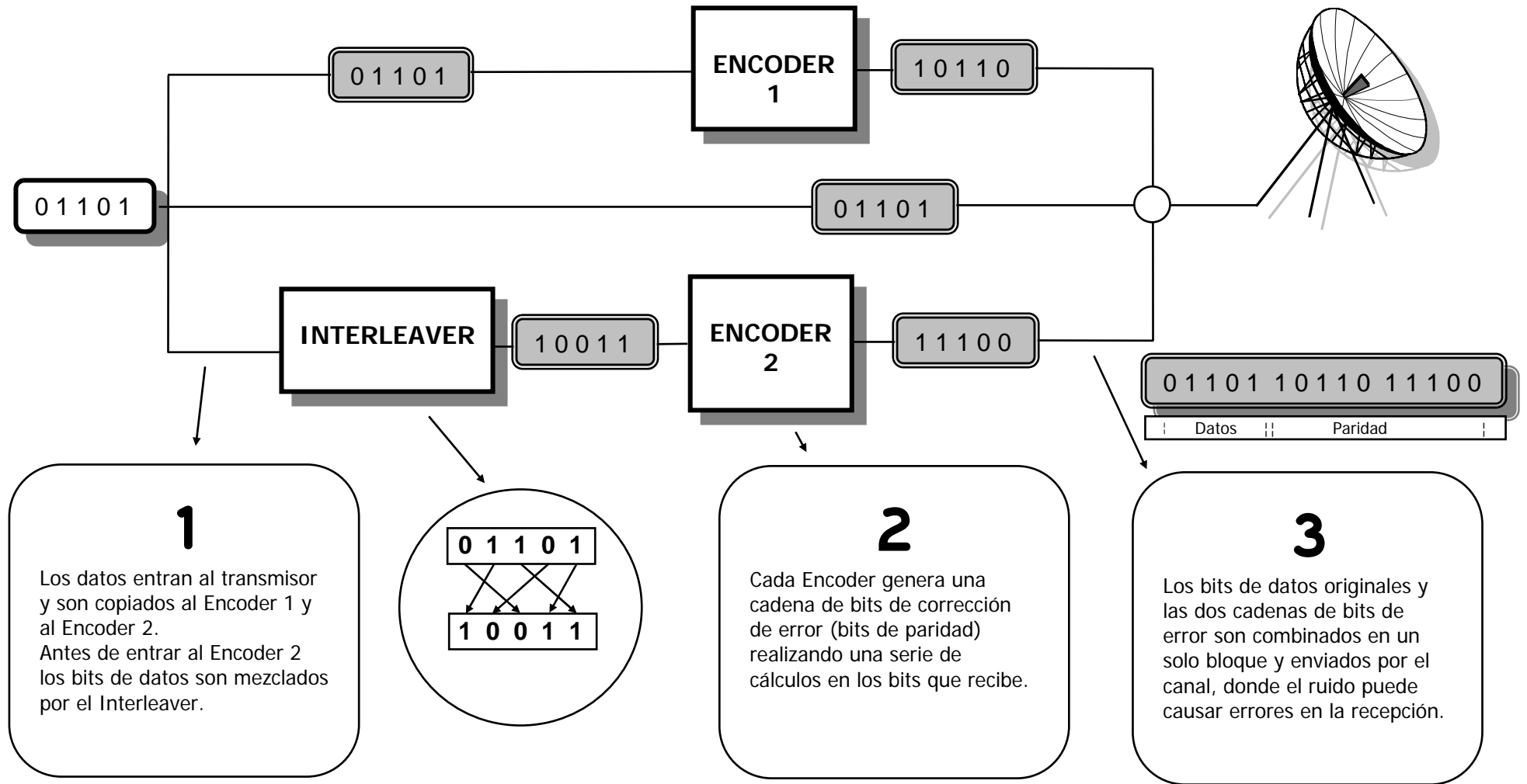
Esto conlleva la transmisión de información redundante pero el resultado final es la descodificación certera de los códigos transmitidos.

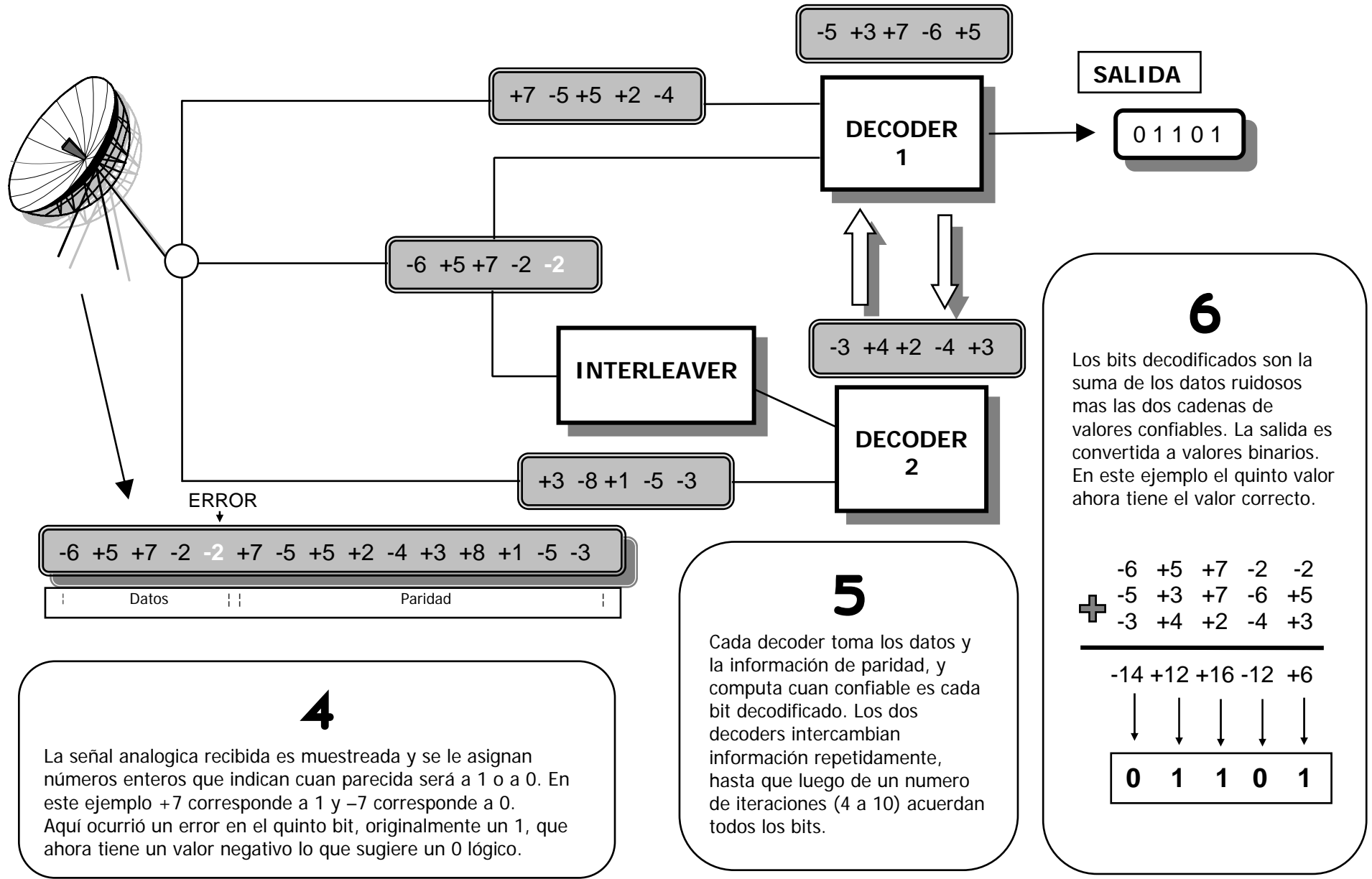
Se puede hacer la analogía de utilizar una matriz donde la solución de las columnas puede ayudar a la resolución de las filas y viceversa, como cada decodificador ayuda a l otro en el esquema de los turbo codes.

Este concepto también puede asimilarse al esquema muy utilizado en electrónica que es la realimentación, donde parte de la salida se realimenta a la entrada para mejorar la performance de un amplificador.

Tanto Berrou como Glavieux, se preguntaron por qué esto no podía funcionar también para la codificación de la información. Y cuando corrieron sus primeros cómputos de simulación y vieron los resultados, se preguntaron durante varios días dónde estaría el error en el programa, ya que no podían creer los resultados obtenidos.

Como Trabajan los Turbo Codes





Aplicaciones en la tecnología

Como France Telecom fue el sponsor de su trabajo de investigación, es la propietaria de las patentes que protegen sus algoritmos. Cuando se les solicitó a Berrou y a Glavieux un nombre comercial para su invento se basaron en el concepto del súpercargador utilizado en los autos de carrera. De manera que si los gases de escape fuerzan la entrada de aire para reforzar la combustión, voilà: “turbo codes” fue la respuesta de Berrou.

Actualmente se utilizan en la tercera generación de celulares y se los conoce oficialmente como UMTS – Universal Mobile Telecommunications System. Estos se utilizan para la transmisión de video, imágenes y email. Para la voz sin embargo se utilizan los códigos convolucionales porque su retardo de decodificación es menor que con los turbo codes.

De hecho, el retraso en la decodificación es el mayor contratiempo en la implementación de los turbo codes. Las necesarias interacciones requeridas por la decodificación turbo hacen que el retraso sea inaceptable para su utilización en tiempo real como la voz y otras comunicaciones que requieran procesamiento instantáneo de datos como ser almacenamiento en discos rígidos y en transmisiones ópticas.

Para sistemas que toleran los retrasos como las comunicaciones espaciales hacia el espacio profundo estos son una atractiva solución, que la NASA y la ESA (European Space Administration) están implementando. Este año se lanzará la sonda Rosetta que se encontrará con un cometa y dependerá de los turbo codes para hacer mas confiables sus comunicaciones. La NASA por su lado las implementará en el Mars Reconnaissance Orbiter and Messenger – MROM - que el Ingeniero argentino Alejandro M. San Martín esta desarrollando en el Jet Propulsion Laboratory en Pasadena, California.

La radiodifusión de audio digital que provee calidad CD en los programas de radio, los enlaces satelitales y el nuevo sistema de redes de área global (Global Area Network) de Inmarsat Ltd. incorporan los turbo codes en sus sistemas.

Los turbo codes han disparado lo que se llama los *turbo principios*, ya que ayudaron a los ingenieros a resolver un número de problemas en las comunicaciones aportando una gran cantidad de otras ideas. En la universidad de Southampton en el Reino Unido el profesor Lajos Hanzo de la Escuela de Electrónica y Ciencias de la Computación esta trabajando para mitigar los efectos de la propagación por caminos múltiples (multipath), lo que representa la distorsión de la señal cuando se reciben replicas que han rebotado en diferentes superficies. Por lo tanto los turbo codes eventualmente ayudaran a que los dispositivos portátiles superen su mayor limitación en la telefonía móvil.

Además están dando aliento a varios esquemas que se fueron olvidados, así los códigos LDPC (Low Density Parity Check) de chequeo de paridad, inventados por Robert Gallager en el MIT en los sesentas y que eran muy complicados para la tecnología de ese momento, utilizan también la interacción, aunque son considerablemente diferentes a los turbo codes. Ahora los investigadores están implementando codificaciones LDPC que podrán superar a los turbo codes acercándose mas al limite de Shannon.

Es más se cree que serán un serio competidor de los turbo codes para la próxima generación de sistemas de redes inalámbricas, como los estándares IEEE 802.11b y IEEE 802.16. Si bien estos utilizan las mismas ideas generales de los turbo codes su implementación y análisis es mas sencillo y además su patente ha expirado lo que habilita a su uso sin pagar derechos intelectuales de copyright.

Conclusiones

Los turbo codes han terminado con una búsqueda que duró 40 años, su invención es revolucionaria y llegaron en el momento preciso donde la tecnología puede implementarlos prácticamente. Si no podemos acercarnos mas a la capacidad de canal impuesta por Shannon o si esta es 0,1dB o 0,0001 dB no importa lo que ahora hay que trabajar es la velocidad del proceso.

La teoría de los turbo codes son el resultado de un trabajo empírico, de una construcción laboriosa, de un esquema de codificación / decodificación, utilizando herramientas existentes, que nunca antes fueron combinadas de esa manera. No es necesario conocer mucho sobre los limites teóricos para alcanzarlos, como comentó una vez Berrou “como el inocente no sabía que la tarea era imposible, la hizo”.

La posesión de una patente sobre ellos impide su uso comercial fuera del entorno licenciado, lo que limitará su uso masivo. Pero han abierto una nueva manera de atacar el problema, como hemos visto.

Referencias

Computer Networks. Andrew S. Tanenbaum. Tercera Edición. Prentice Hall. 1996. 813 páginas.

Codeword: From Algorithms and Theory of Computation Handbook, page 12-21, Copyright © 1999 by CRC Press LLC. Appearing in the Dictionary of Computer Science, Engineering and Technology, Copyright © 2000 CRC Press LLC.

IEEE Spectrum. ISSN 0018-9235 + Communications Closing in the perfect code. page. 28-34 March 2004

<http://www.icc2004.org/>.

<http://people.myoffice.net.au/~abarbulescu/>.

Turbo Codes: Desirable and Designable, de A. Giulietti, B. Bougard y L. Van der Perre. Copyright Kluwer Academic, Dordrecht, Netherlands © 2004.