

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 386 232**

51 Int. Cl.:

**H04L 1/00** (2006.01)

**H04L 1/18** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **09001024 .0**

96 Fecha de presentación: **21.12.2005**

97 Número de publicación de la solicitud: **2058971**

97 Fecha de publicación de la solicitud: **13.05.2009**

54 Título: **Procedimiento y aparato para utilizar distintos esquemas de modulación para retransmisiones de un paquete**

30 Prioridad:  
**22.12.2004 US 22538**

45 Fecha de publicación de la mención BOPI:  
**14.08.2012**

45 Fecha de la publicación del folleto de la patente:  
**14.08.2012**

73 Titular/es:  
**QUALCOMM INCORPORATED  
5775 MOREHOUSE DRIVE  
SAN DIEGO, CA 92121-1714, US**

72 Inventor/es:  
**Khandekar, Aamod y  
Agrawal, Avneesh**

74 Agente/Representante:  
**Fàbrega Sabaté, Xavier**

ES 2 386 232 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

**DESCRIPCIÓN**

Procedimiento y aparato para utilizar distintos esquemas de modulación para retransmisiones de un paquete.

**5 ANTECEDENTES****I. Campo**

10 La presente invención se refiere generalmente a la comunicación y más específicamente a la transmisión de datos en un sistema de comunicación.

**II. Antecedentes**

15 En un sistema de comunicación, un transmisor codifica, entrelaza y modula típicamente (es decir hace corresponder con símbolos) datos de tráfico para obtener símbolos de datos, que son símbolos de modulación. Para un sistema coherente, el transmisor se multiplexa en símbolos piloto con los símbolos de datos. Los símbolos piloto son símbolos de modulación para piloto, que es el dato conocido *a priori* tanto por el transmisor como por el receptor. El transmisor procesa, además, los datos multiplexados y los símbolos piloto para generar una señal modulada y transmite esta señal por un canal de comunicación. El canal distorsiona la señal transmitida con una respuesta de canal y degrada también la señal con ruido e interferencia.

25 El receptor recibe la señal transmitida y procesa la señal recibida para obtener símbolos de datos recibidos y símbolos piloto recibidos. Para un sistema coherente, el receptor evalúa la respuesta de canal con los símbolos piloto recibidos y lleva a cabo una detección coherente sobre los símbolos de datos recibidos con las estimaciones de respuesta de canal para obtener símbolos de datos detectados, que son estimaciones de los símbolos de datos enviados por el transmisor. El receptor desentrelaza y descodifica entonces los símbolos de datos detectados para obtener datos descodificados, que son una estimación de los datos de tráfico enviados por el transmisor.

30 La codificación y la modulación llevadas a cabo por el transmisor tienen un gran impacto sobre la prestación de la transmisión de datos.

35 El artículo científico *Medium access control and radio resource management for packet data services over IS-136 channels* de Kristina Balachandran y otros, Conferencia de Tecnología Vehicular, 1999 IEEE 49° Houston, Tejas EE.UU., describe un canal de datos diseñado para proporcionar modulación adaptativa de ranuras de tiempo y para permitir modo de redundancia incremental.

Existe por lo tanto una necesidad en el ramo de técnicas para llevar a cabo codificación y modulación de forma que se alcancen buenas prestaciones.

**40 RESUMEN**

45 La invención se define en las reivindicaciones independientes. En este documento se describen técnicas para usar múltiples esquemas de modulación (es decir, formatos de modulación) para dos únicos paquetes para alcanzar buenas prestaciones.

50 Cada paquete de datos se puede procesar y transmitir en hasta T bloques, donde  $T > 1$ . Cada bloque lleva bits de código para solamente una parte del paquete y se modula basándose en un esquema de modulación usado para ese bloque. Se usan múltiples esquemas diferentes de modulación para los T bloques del paquete para conseguir una buena prestación. Los esquemas de modulación para los T bloques se pueden seleccionar como se describe en lo que sigue.

55 Un transmisor puede codificar un paquete de datos según un código de base corrección de error en recepción (FEC) (por ejemplo un código turbo o convolucional) para generar bits de código para el paquete. El transmisor forma entonces un bloque de bits de código con los bits de código generados para el paquete, por ejemplo basándose en un modelo de perforación. El transmisor determina el esquema de modulación a usar para el bloque (por ejemplo, basado en un modo/velocidad seleccionado para el paquete de datos) y mapeo los bits de código para el bloque basado en este esquema de modulación para obtener símbolos de datos para el bloque. El transmisor, además, procesa y transmite el bloque de símbolos de datos a un receptor. El transmisor genera y transmite otro bloque de símbolos de datos de manera similar si el paquete de datos no se descodifica correctamente por el receptor y si los T bloques no se han transmitido para el paquete. El receptor lleva a cabo el procesamiento complementario para recibir y descodificar el paquete.

Se describen diversos aspectos y realizaciones de la invención más en detalle en lo que sigue.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

Las características y la naturaleza de la presente invención se harán más evidentes a partir de la descripción detallada expuesta en lo que sigue realizada en combinación con los dibujos en los cuales se identifican a lo largo de los mismos con los números de referencia idénticos correspondientemente y en los cuales:

- 5 La FIG. 1 ilustra una transmisión por redundancia incremental;  
 La FIG. 2 muestra un diagrama de bloques de un transmisor y de un receptor;  
 La FIG. 3 muestra un diagrama de bloques de un procesador de datos de transmisión (TX);  
 La FIG. 4 ilustra el procesamiento de un paquete de datos por el procesador de datos TX;  
 10 La FIG. 5A muestra la formación de bloques de bits de código para un paquete con un único esquema de modulación;  
 Las FIGS. 5B y 5C muestran la formación de bloques de bits de código para un paquete con múltiples esquemas de modulación para dos modos diferentes;  
 La FIG. 6 muestra un diagrama de bloques de un detector y un procesador de datos de recepción (RX);  
 15 Las FIGS. 7 y 8 muestran procedimientos para transmitir y recibir respectivamente un paquete de datos con múltiples esquemas de modulación.

**DESCRIPCIÓN DETALLADA**

- 20 El término "ejemplar" se usa en el presente documento para significar "que sirve de ejemplo, muestra o ilustración".  
 Cualquier realización o diseño descrito en la presente memoria como "ejemplar" no se ha de interpretar necesariamente como preferido o ventajoso respecto de otras realizaciones o diseños.

- 25 Las técnicas descritas en el presente documento para el uso de múltiples esquemas de modulación para un único paquete se pueden usar para diversos sistemas de modulación como un sistema de Acceso Múltiple por división de Frecuencia Ortogonal (OFDMA), un sistema de Accesos Múltiple por División de Código (TDFMA), un sistema basado en el multiplexado por división de frecuencia ortogonal (OFDM), un sistema de entrada única y salida única (SISO), un sistema de entrada múltiple y salida múltiple (MIMO), etc. Estas técnicas se pueden usar para sistemas que utilizan redundancia incremental (IR) y sistemas que no utilizan IR (por ejemplo, sistemas que simplemente repiten datos). Por motivos de claridad, estas técnicas se describen para un sistema de comunicación que utiliza IR.

- 30 La FIG. 1 ilustra una transmisión IR entre un transmisor y un receptor en un sistema de comunicación. La secuencia cronológica para la transmisión de datos se divide en tramas, teniendo cada trama una duración temporal particular. Para la realización de transmisión IR mostrada en la FIG. 1, el receptor estima inicialmente el canal de comunicación, selecciona un "modo" basado en la condición del canal, y envía el modo seleccionado al transmisor en la trama 0. Alternativamente, el receptor devuelve una estimación de la calidad del canal, y el transmisor selecciona el modo basado en la estimación de calidad de canal. En cualquier caso, el modo puede indicar la dimensión del paquete, la tasa de transmisión de código, el esquema de modulación etc. par el paquete. El transmisor procesa un paquete de datos (Paquete 1) según el modo seleccionado, y genera hasta T bloques de símbolos de datos para el paquete. T es el número máximo de bloques para un paquete dado de datos y es superior a uno ( $T > 1$ ) para IR. El primer bloque contiene típicamente suficiente información para permitir al receptor descodificar el paquete bajo unas condiciones óptimas del canal. Cada bloque subsiguiente contiene típicamente información adicional sobre paridad/redundancia no contenida en los bloques anteriores. El transmisor transmite entonces el primer bloque de símbolos de datos (bloque 1) para el paquete 1 en la trama 1. El receptor recibe, detecta y descodifica el primer bloque de símbolos de datos, determina que el paquete 1 se descodifica como error (es decir "eliminado"), y devuelve un acuse de recibo negativo (NAK) en la trama 2. El transmisor recibe el NAK y transmite el segundo bloque de símbolos de datos (Bloque 2) para el paquete 1 en la trama 3. El receptor recibe y detecta el bloque 2, descodifica los bloques 1 y 2 y determina que el paquete 1 siga descodificado como error, y devuelve otro NAK en la trama 4. La transmisión de bloques y la respuesta NAK se pueden repetir tantas veces como se quiera.

- 50 Para el ejemplo mostrado en la FIG. 1, el transmisor recibe un NAK para el bloque N-1 de símbolos de datos y transmite el bloque N de símbolos de datos (Bloque N) para el paquete 1 en la trama n, donde  $N \leq T$ . El receptor recibe y detecta el bloque N, descodifica los bloques 1 a través de N, determina que el paquete se descodifique correctamente, y devuelve un acuse de recibo (ACK) en la trama n + 1. El receptor también evalúa el canal de comunicación, selecciona un modo para el siguiente paquete de datos, y envía el modo seleccionado al transmisor en la trama n + 1. El transmisor recibe el ACK para el bloque N y termina la transmisión del paquete 1. El transmisor también procesa el siguiente paquete de datos (Paquete 2) según el modo seleccionado, y transmite el primer bloque de símbolos de datos (Bloque 1) para el paquete 2 en la trama n + 2. El procesamiento en el transmisor y el receptor sigue de la misma manera para cada paquete de datos transmitido por el canal de comunicación.

- 60 Como se muestra en la FIG. 1, con la redundancia incremental, el transmisor envía cada paquete de datos en una serie de transmisiones de bloques, llevando cada transmisión de bloques una parte del paquete. El receptor puede intentar descodificar el paquete después de cada transmisión de bloques basada en todos los bloques recibidos. El transmisor termina la transmisión del paquete después de la descodificación con éxito por parte del receptor.

Para el ejemplo mostrado en la FIG. 1, hay un retardo de una trama para la respuesta ACK/NAK desde el receptor para cada transmisión de bloques. En general, este retardo puede ser una o múltiples tramas. Para mejorar la utilización del canal, los paquetes de datos se pueden transmitir de una manera entrelazada. Por ejemplo, los paquetes de datos para un canal de tráfico se pueden transmitir en tramas de números impares y paquetes de datos para otro canal de tráfico se pueden transmitir en tramas de números pares. Igualmente se pueden entrelazar más de dos canales de tráfico, por ejemplo, si el retardo ACK/NAK es superior a una trama.

El sistema puede estar destinado a soportar un conjunto de modos, que también se pueden denominar velocidades, formatos de paquetes, configuraciones de radio, o de algún otro modo. Cada modo se puede asociar a una velocidad de código o esquema de codificación particular, un esquema de modulación particular, una eficiencia espectral particular, y una relación mínima de señal de ruido e interferencia (SINR) requerida para conseguir un nivel diana de prestación, por ejemplo una velocidad de error de paquete del 1% (PER). La eficiencia espectral se refiere a la velocidad de datos (o la velocidad binaria de información) normalizada por el ancho de banda del sistema, y se da en unidad de bits por segundo por Hertzio (bps/Hz). En general se necesitan mayores SINR para mayores eficiencias espectrales. El conjunto de modos soportados abarca un intervalo de eficiencias espectrales, típicamente en incrementos que están espaciados de manera regular. Para una condición de canal dada y una SINR recibida, se puede seleccionar el modo con la mayor eficiencia espectral soportada por la SINR recibida y se usa para la transmisión de datos.

La eficiencia espectral se determina mediante la velocidad de código y el esquema de modulación. La velocidad de código es la relación del número de bits de entrada dentro de un codificador respecto del número de bits de código generados por el codificador y transmitidos. Por ejemplo, una velocidad de código de  $2/9$  (o  $T = 2/9$ ) genera nueve bits de código para cada dos bits de entrada. Una menor velocidad de código (por ejemplo  $T = 1/4$  o  $1/5$ ) tiene mayor redundancia y de este modo mayor capacidad de corrección de errores. Sin embargo, cuantos más bits de código de transmiten para una velocidad de código menor, menor es también la eficiencia espectral.

Se pueden usar diversos esquemas de modulación para la transmisión de datos. Cada esquema de modulación se asocia a una constelación de señales que contiene  $M$  puntos de señal, donde  $M > 1$ . Cada punto de señal se define por un valor de complejo y se identifica por un valor binario de bit  $B$ , donde  $B \geq 1$  y  $2^B = M$ . Para el mapeo de los símbolos, los bits de código a transmitir se agrupan primero en conjuntos de  $B$  bits de código. Cada conjunto de bits de código  $B$  forma un valor  $B$  binario de bits que se hacen corresponder en un punto de señal específico, que se transmite entonces como símbolo de modulación para ese grupo de  $B$  bits de código. Cada símbolo de modulación lleva de este modo información para  $B$  bits. Algunos esquemas de modulación habitualmente usados incluyen Modulación por desplazamiento binario de fase (BPSK), Modulación por desplazamiento en cuadratura de fase (QPSK), Modulación por desplazamiento M-aria de fase (M-PSK) y Modulación de amplitud en cuadratura M-aria (M-QAM). El número de bits de código por símbolo de modulación ( $B$ ) se puede dar como:  $B = 1$  para BPSK,  $B = 2$  para QPSK,  $B = 3$  para 8-PSK,  $B = 4$  para 16-QAM,  $B = 6$  para 64-QAM etc.  $B$  es indicativo del orden de un esquema de modulación y se pueden enviar más bits de código por símbolo de modulación para esquemas de modulación de mayor orden.

La eficiencia espectral ( $S$ ) para una velocidad de código data y un esquema de modulación dada se puede calcular como el producto de la velocidad de código ( $R$ ) y el número de bits de código por símbolo de modulación ( $B$ ) para el esquema de modulación, o  $s = R \times B$ . Se puede obtener una eficiencia espectral dada con diversas combinaciones diferentes de velocidad de código y esquema de modulación (o pares de codificación y modulación). Por ejemplo, una eficiencia espectral de  $S = 4/3$  con las siguientes combinaciones de velocidad de código y esquema de modulación:

QPSK ( $B = 2$ ) y velocidad de código  $R = 2/3$ ;  
 8-PSK ( $B = 3$ ) y velocidad de código  $R = 4/9$ ;  
 16-QAM ( $B = 4$ ) y velocidad de código  $R = 1/3$ ; y  
 64-QAM ( $B = 6$ ) y velocidad de código  $R = 2/9$ ;

Las diferentes combinaciones de velocidad de código y esquema de modulación para una eficiencia espectral dada pueden tener diferente prestación, lo cual se puede cuantificar mediante la SINR requerida para conseguir el PER diana. Estas diferentes combinaciones de velocidad de código y esquema de modulación se pueden evaluar (por ejemplo por simulación informática, mediciones empíricas etc.) para diferentes condiciones de canal y posiblemente diferentes modelos de canal. Se puede seleccionar la combinación de velocidad de código y esquema de modulación con la mejor prestación e incluirla en el conjunto de modos soportados por el sistema.

Para una eficiencia espectral dada, una combinación de un esquema de modulación de orden superior y una velocidad de código inferior pueden conseguir típicamente una mejor capacidad que una combinación con un esquema de modulación de orden inferior y una velocidad de código superior. Sin embargo, una combinación dada de esquema de modulación y velocidad de código que consigue una mejor capacidad no puede proporcionar una mejor prestación a causa de los desafíos prácticos, y hay típicamente un compromiso entre el esquema de modulación y la velocidad de código. Hay una combinación "óptima" del esquema de modulación y la velocidad de código para este compromiso con el fin de optimizar la prestación.

La ventaja de prestación de un esquema de modulación de orden elevado y una combinación de baja velocidad de codificación tiende a incrementarse por un canal de selección de frecuencia o de tiempo. Un canal de selección de frecuencia es un canal de comunicación que tiene una respuesta de frecuencia que varía a lo largo del ancho de banda del sistema (es decir, no plana). Un canal de selección de tiempo es un canal de comunicación que tiene una respuesta que varía a lo largo del tiempo (es decir, no estacionaria). Se ha llevado a cabo un estudio para determinar las mejores combinaciones de velocidad de codificación y esquema de modulación para diferentes eficiencias espectrales. A partir del estudio, una regla general resultante fue que las velocidades de código de  $R = 1/2$  o inferiores se deberían usar para un canal de selección de frecuencia/tiempo. De este modo, incluso si se pueden usar velocidades de código superiores a la velocidad  $1/2$  para una eficiencia espectral dada, se debería usar una velocidad de codificación cercana a o por debajo de la velocidad  $1/2$  en combinación con un esquema de modulación de mayor orden. Este estudio se llevó a cabo para un modelo de canal específico, y la prestación puede diferir para algunos modelos de canal.

Para un sistema que no utiliza redundancia incremental, se codificó cada paquete de datos y se moduló basándose en una velocidad de código específica y un esquema de modulación específico para un modo seleccionado para ese paquete. Se transmite entonces todo el paquete y se consigue la frecuencia espectral asociada al modo seleccionado. Se puede determinar la velocidad de código y el esquema de modulación para cada modo soportado por el sistema por simulación informática, mediciones empíricas etc.

Para un sistema que utiliza redundancia incremental, se transmite cada paquete de datos en uno o más bloques hasta que el paquete se descodifica correctamente por el receptor, como se ha descrito anteriormente. Cada bloque adicional transmitido para un paquete de datos reduce la eficiencia espectral del paquete. La eficiencia espectral para cada paquete de datos se conoce de este modo *a priori* y cambia con cada transmisión de bloques para el paquete.

La Tabla 1 muestra un conjunto ejemplar de siete modos soportados por el sistema. Cada modo  $m$  se asocia a un esquema de modulación específico y una velocidad de código "base" específica  $R_{base,m}$ , que es la velocidad de código utilizada para cada bloque. Por ejemplo, QPSK y una velocidad de código base de  $R = 2/3$  se usan para el modo 3. Para un paquete de datos enviado como el modo 3. Cada bloque del paquete se transmite de este modo usando QPSK y la velocidad de código  $2/3$ . La Tabla 1 supone que  $T = 4$  y un paquete de datos se termina después de cuatro transmisiones de bloques sin tener en cuenta si el paquete se descodificó correctamente o no.

Cada modo se asocia también a diferentes velocidades de código "efectivas" para diferentes números de transmisiones de bloques. Para un paquete de datos enviado con el modo  $m$ , la velocidad de código efectiva  $R_{eff,m}(\ell)$  para el paquete de datos después de la  $\ell$ -ésima transmisión de bloques donde  $\ell = 1, 2, \dots, T$  es igual a la velocidad de código base dividida por  $\ell$ , o  $R_{eff,m}(\ell) = R_{base,m}/\ell$ . Esto es porque  $\ell$  veces el número de bits de código se han enviado para el paquete de datos en  $\ell$  transmisiones de bloques. Por ejemplo, un paquete de datos enviado con el modo 3 tiene una velocidad de código efectiva de  $R = 2/3$  y una eficiencia espectral de  $S = 4/3$  si solamente se transmite un bloque para el paquete, una velocidad de código efectiva de  $R = 1/3$  y una eficiencia espectral de  $S = 2/3$  si se transmiten dos bloques para el paquete, una velocidad de código efectiva de  $R = 2/9$  y una eficiencia espectral de  $S = 4/9$  si se transmiten tres bloques para el paquete, y una velocidad de código efectiva de  $R = 1/6$  y una eficiencia espectral de  $S = 1/3$  si se transmiten cuatro bloques para el paquete.

Tabla 1

Modo	Esquema de modulación	Velocidad de código base	Eficacia espectral después de...			
			Una transmisión de bloques	Dos transmisiones de bloques	Tres transmisiones de bloques	Cuatro transmisiones de bloques
1	QPSK	$R = 1/5$	$2 \times (1/5) = 2/5$	$2 \times (1/10) = 1/5$	$2 \times (1/15) = 2/15$	$2 \times (1/20) = 1/10$
2	QPSK	$R = 1/3$	$2 \times (1/3) = 2/3$	$2 \times (1/6) = 1/3$	$2 \times (1/9) = 2/9$	$2 \times (1/12) = 1/6$
3	QPSK	$R = 2/3$	$2 \times (2/3) = 4/3$	$2 \times (2/6) = 2/3$	$2 \times (2/9) = 4/9$	$2 \times (2/12) = 1/3$
4	8-PSK	$R = 2/3$	$3 \times (2/3) = 2$	$3 \times (2/6) = 1$	$3 \times (2/9) = 2/3$	$3 \times (2/12) = 1/2$
5	16-QAM	$R = 2/3$	$4 \times (2/3) = 8/3$	$4 \times (2/6) = 4/3$	$4 \times (2/9) = 8/9$	$4 \times (2/12) = 2/3$
6	64-QAM	$R = 5/9$	$6 \times (5/9) = 10/3$	$6 \times (5/18) = 5/3$	$6 \times (5/27) = 10/9$	$6 \times (5/36) = 5/6$
7	64-QAM	$R = 2/3$	$6 \times (1/5) = 2/5$	$6 \times (2/6) = 2$	$6 \times (2/9) = 4/3$	$5 \times (2/12) = 1$

Como se muestra en la Tabla 1, la velocidad de código efectiva (que se muestra entre paréntesis en las columnas 4 a 7) y la eficiencia espectral (que se muestra después del signo igual en las columnas 4 a 7) para cada modo varían dependiendo del número de bloques transmitidos para un paquete. De este modo se pueden usar cada modo para

múltiples eficiencias espectrales. Para cada modo, la combinación de velocidad de código y esquema de modulación que consigue la mejor prestación a una eficiencia espectral particular (por ejemplo, con dos transmisiones de bloques) se puede seleccionar y usar para ese modo. Sin embargo, esta combinación de velocidad de código y esquema de modulación puede no conseguir buen rendimiento en otras eficiencias espectrales (por ejemplo, con una, tres y cuatro transmisiones de bloques).

Para mejorar el rendimiento, se pueden usar diferentes combinaciones de velocidad de código y esquema de modulación para diferentes transmisiones de bloques para un modo dado. Las diferentes combinaciones de velocidad de código y esquema de modulación para cada modo se pueden determinar de diversas maneras.

En una primera realización, el esquema de modulación a usar para cada transmisión de bloques se selecciona basándose en la eficiencia espectral conseguida después de esa transmisión de bloques. Para cada modo  $m$ , se puede seleccionar el esquema de modulación  $M_m(\ell)$  para la transmisión de la  $\ell$ -ésima transmisión de bloques como sigue. La eficiencia espectral  $S_m(\ell)$  después de la  $\ell$ -ésima transmisión de bloques para el modo  $m$  se determina en primer lugar a partir de la Tabla 1. El esquema de modulación de la  $M_m(\ell)$  se establece entonces en el esquema de modulación de orden inferior que, en combinación con una velocidad de código efectiva  $R'_{eff,m}(\ell)$  de  $1/2$  o menor, consigue la eficiencia espectral  $S_m(\ell)$ . La velocidad de código efectiva  $R'_{eff,m}(\ell)$  es la velocidad de código para todas las  $\ell$  transmisiones de bloques si se usó el esquema de modulación  $M_m(\ell)$  para cada uno de las  $\ell$  transmisiones de bloque. La relación entre la velocidad de código efectiva, el esquema de modulación, y la eficiencia espectral se pueden expresar como  $S_m(\ell) = R'_{eff,m}(\ell) \times B_m(\ell)$ , donde  $B_m(\ell)$  es el número de bits de código por símbolo de modulación para el esquema de modulación  $M_m(\ell)$ . El esquema de modulación  $M_m(\ell)$  se puede seleccionar entonces basándose en la eficiencia espectral  $S_m(\ell)$  como sigue:

- Se usa QPSK para eficiencias espectrales de 1,0 bps/Hz e inferiores;
- Se usa 8-PSK para eficiencias espectrales entre 1,0 y 1,5 bps/Hz;
- Se usa 16-QAM para eficiencias espectrales entre 1,5 y 2,0 bps/Hz; y
- Se usa 64-QAM para eficiencias espectrales superiores a 2,0 bps/Hz.

La correspondencia anterior supone que solamente son soportados por el sistema QPSK, 8-PSK, 16-QAM y 64-QAM. También se pueden usar otras correspondencias para diferentes conjuntos de esquemas soportados de modulación.

La Tabla 2 muestra el esquema de modulación basado en la primera realización para cada transmisión de bloques para los siete modos listados en la Tabla 1. La segunda columna de la Tabla 2 muestra la eficiencia espectral base  $S_{base,m}$  que es la eficiencia espectral después de una transmisión de bloques. Para cada modo  $m$ , se determina el esquema de modulación para cada transmisión de bloques basándose en la eficiencia espectral después de esa transmisión de bloques y usando la correspondencia de la eficiencia espectral-esquema de modulación anteriormente descrita. A modo de ejemplo, para el modo 3, se usa 8-PSK para la primera transmisión de bloques ya que la eficiencia espectral después de esta transmisión es  $S = 4/3$ , se usa QPSK para la segunda transmisión de bloques ya que la eficiencia espectral después de esta transmisión es  $S = 2/3$ , etc. A modo de otro ejemplo, para el modo 6 se usa 64-QAM para la primera transmisión de bloques ya que la eficiencia espectral después de esta transmisión es  $S = 10/3$ , se usa 16-QAM para la segunda transmisión de bloques ya que la eficiencia espectral después de esta transmisión es  $S = 5/3$ , se usa 8-PSK para la tercera transmisión de bloques ya que la eficiencia espectral después de esta transmisión es  $S = 10/9$ , y QPSK se usa para la cuarta transmisión de bloques ya que la eficiencia espectral después de esta transmisión es  $S = 5/6$ .

Tabla 2

Modo $m$	$S_{base,m}$	Primera transmisión de bloques	Segunda transmisión de bloques	Tercera transmisión de bloques	Cuarta transmisión de bloques
1	2/5	QPSK	QPSK	QPSK	QPSK
2	2/3	QPSK	QPSK	QPSK	QPSK
3	4/3	8-PSK	QPSK	QPSK	QPSK
4	2	16-QAM	QPSK	QPSK	QPSK
5	8/3	64-QAM	8-PSK	QPSK	QPSK
6	10/3	64-QAM	16-QAM	8-PSK	QPSK
7	4	64-QAM	16-QAM	8-PSK	QPSK

En una segunda realización, para cada modo, la combinación especial de velocidad de código y esquema de modulación a usar para cada bloque de transmisión se selecciona independientemente para conseguir una buena prestación. Para la primera transmisión de bloques con el modo  $m$ , se evalúan diversas combinaciones de velocidad de código y esquema de modulación con eficiencia espectral de  $S_{base,m}$  (por ejemplo, basándose en simulación

informática, mediciones empíricas, etc), y se selecciona la combinación de la velocidad de código  $R_m(1)$  y el esquema de modulación  $M_m(1)$  con la mejor prestación. Para la segunda transmisión de bloques con modo  $m$ , se evalúan de nuevo diversas combinaciones de velocidad de código y esquema de modulación con eficiencia espectral de  $S_{base,m}$ , aunque bajo la premisa de que la primera transmisión de bloques se envió usando  $M_m(1)$  y  $R_m(1)$  y el paquete descodificado se elimina. Se selecciona la combinación de velocidad de código  $R_m(2)$  y el esquema de modulación  $M_m(2)$  con la mejor prestación para la segunda transmisión de bloques. Para la tercera transmisión de bloques con el modo  $m$ , se evalúan de nuevo diversas combinaciones de velocidad de código y esquema de modulación con eficiencia espectral de  $S_{base,m}$ , aunque bajo la premisa de que la primera transmisión de bloques se envió usando  $M_m(1)$  y  $R_m(1)$ , la segunda transmisión de bloques se envió usando  $M_m(2)$  y  $R_m(2)$ , y el paquete descodificado con ambas transmisiones de bloques se elimina. Se vuelve a seleccionar la combinación de la velocidad de código  $R_m(3)$  y el esquema de modulación  $M_m(3)$  con la mejor prestación. Para la cuarta transmisión de bloques con modo  $m$ , se vuelven a evaluar diversas combinaciones de velocidad de código y esquema de modulación con eficiencia espectral de  $S_{base,m}$ , aunque bajo la premisa de que se envió la primera transmisión de bloques usando  $M_m(1)$  y  $R_m(1)$ , se envió la segunda transmisión de bloques usando  $M_m(2)$  y  $R_m(2)$ , se envió la tercera transmisión de bloques usando  $M_m(3)$  y  $R_m(3)$ , y se eliminó el paquete descodificado con las tres transmisiones. Se vuelve a seleccionar la combinación de velocidad de código  $R_m(4)$  y el esquema de modulación  $M_m(4)$  con la mejor prestación. Se puede, de este modo, seleccionar una combinación diferente de la velocidad de código  $R_m(\ell)$  y el esquema de modulación  $M_m(\ell)$  para cada transmisión de bloques en modo  $m$ .

En una tercera realización, se selecciona la combinación de velocidad de código y esquema de modulación que proporciona la mejor prestación para una eficiencia espectral designada para cada modo. Esta eficiencia espectral designada es la eficiencia espectral después de un número predeterminado de (por ejemplo dos) transmisiones de bloque y se supone que los paquetes se decodifican correctamente después de esta transmisión de muchos bloques. El mejor esquema de modulación para cada modo con terminación temprana después de la segunda transmisión de bloques se muestra para la cuarta columna en la Tabla 2. Para cada modo, se selecciona entonces un esquema de modulación de orden superior para la primera transmisión de bloques si la velocidad de código resultante es de  $\frac{1}{2}$  o inferior. La Tabla 3 muestra los esquemas de modulación seleccionados para cada modo para la tercera transmisión. Esta realización puede reducir la complejidad en el transmisor y el receptor mientras proporciona una prestación mejorada para la primera transmisión de bloques, la cual es la más frecuente de todas las transmisiones de  $T$  bloques.

TABLA 3

Modo $m$	$S_{base,m}$	Primera transmisión de bloques	Segunda transmisión de bloques	Tercera transmisión de bloques	Cuarta transmisión de bloques
1	2/5	QPSK	QPSK	QPSK	QPSK
2	2/3	QPSK	QPSK	QPSK	QPSK
3	4/3	8-PSK	QPSK	QPSK	QPSK
4	2	16-QAM	8-PSK	8-PSK	8-PSK
5	8/3	64-QAM	16-QAM	16-QAM	16-QAM
6	10/3	64-QAM	64-QAM	64-QAM	64-QAM
7	4	64-QAM	64-QAM	64-QAM	64-QAM

Anteriormente, se han descrito diversas realizaciones para seleccionar velocidad de código y esquema de modulación para cada transmisión de bloques en cada modo. Igualmente se puede seleccionar la velocidad de código y el esquema de modulación para cada transmisión de bloques de otras maneras, y esto dentro del ámbito de la invención.

La FIG. 2 muestra un diagrama de bloques de un transmisor 210 y un receptor 250 en un sistema de comunicación inalámbrica 200 que utiliza transmisión IR. En el transmisor 210, un procesador de datos TX 220 recibe paquetes de datos desde una fuente de datos 212. El procesador de datos TX 220 procesa (por ejemplo, formatea, codifica, particiona, entrelaza y modula) cada paquete de datos según un modo seleccionado para ese paquete y genera hasta  $T$  bloques de símbolos de datos para el paquete. El modo seleccionado para cada paquete de datos puede indicar (1) la dimensión del paquete (es decir, el número de bits de información para el paquete) y (2) la combinación particular de velocidad de datos y esquema de modulación a usar para cada bloque de símbolos de datos de ese paquete. Un controlador 230 proporciona diversos controles a la fuente de datos 212 y el procesador de datos TX 220 para cada paquete de datos basándose en el modo seleccionado así como la realimentación (ACK/NAK) recibida para el paquete. El procesador de datos TX 220 proporciona un flujo de bloques de símbolos de datos (por ejemplo, un bloque para cada trama), donde los bloques para cada paquete se pueden entrelazar con los bloques para uno u otros paquetes más.

Una unidad de transmisor (TMTR) 222 recibe el flujo de bloques de símbolo de datos a partir del procesador de datos TX 220 y genera una señal modulada. La unidad de transmisor 222 se multiplexa en símbolos piloto con los símbolos de datos (por ejemplo, multiplexado del tiempo de uso, frecuencia, y/o división de código) y obtiene un flujo de símbolos de transmisión. Cada símbolo de transmisión puede ser un símbolo de datos, un símbolo piloto, o un símbolo nulo con un valor de señal de cero. La unidad transmisora 222 puede llevar a cabo modulación OFDM si el sistema usa OFDM. La unidad transmisora 222 genera un flujo de muestra de dominio-tiempo y otras condiciones (por ejemplo convierte a analógico, escala en frecuencia, filtra y amplifica) el flujo de muestra para generar la señal modulada. La señal modulada se transmite entonces a partir de una antena 224 y por un canal de comunicación al receptor 250.

En el receptor 250, la señal transmitida es recibida por una antena 252, y la señal recibida se proporciona a una unidad receptor (RCVR) 254. La unidad receptor 254 acondiciona, digitaliza y preprocesa (por ejemplo desmodula en OFDM) la señal recibida para obtener símbolos de datos recibidos y símbolos piloto recibidos. La unidad receptora 254 proporciona los símbolos de datos recibidos a un detector 256 y los símbolos piloto recibidos a un estimador de canal 258. El estimador de canal 258 procesa los símbolos de piloto y proporciona estimaciones de canal (por ejemplo, estimaciones de ganancia de canal y estimaciones SINR) para el canal de comunicación. El detector 256 lleva a cabo la detección en los símbolos de datos recibidos con las estimaciones de canal y proporciona símbolos de datos detectados a un procesador de datos RX 260. Los símbolos de datos detectados se pueden representar mediante relaciones de probabilidad de registro (LLRS) para los bits de código usados para formar los símbolos de datos (como se describe en lo sucesivo) o por otras representaciones. Cuando se obtiene un nuevo bloque de símbolos de datos detectados para un paquete de datos dado, el procesador de datos RX 260 procesa (por ejemplo desentrelaza y descodifica) todos los símbolos de datos detectados obtenidos para ese paquete y proporciona un paquete a un receptor de datos 262. El procesador de datos RX también verifica el paquete descodificado y proporciona el estado de paquete, que indica si el paquete se descodifica correctamente o erróneamente.

Un controlador 270 recibe las estimaciones de canal a partir del estimador de canal 258 y el estado de paquete a partir del procesador de datos RX 260. El controlador 270 selecciona un modo para el siguiente paquete de datos a transmitir al receptor 250 basado en las estimaciones de canal. El controlador 270 ensambla también la información de realimentación, que puede incluir el modo seleccionado para el siguiente paquete, un ACK o un NAK para el paquete recién descodificado, y así sucesivamente. La información de realimentación se procesa por un procesador de datos TX 282, se acondiciona además por una unidad transmisor 284, y se transmite por antena 252 al transmisor 210.

En el transmisor 210, la señal transmitida desde el receptor 250 se recibe por antena 224, se acondiciona por una unidad receptora 242, y se procesa, además, por un procesador de datos RX 244 para recuperar la información de realimentación enviada por el receptor 250. El controlador 230 obtiene la información de realimentación recibida, usa el ACK/NAK para controlar la transmisión IR del paquete que se envía al receptor 250, y usa el modo seleccionado para procesar el siguiente paquete de datos para enviar al receptor 250.

Los controladores 230 y 270 dirigen la operación en el transmisor 210 y el receptor 250, respectivamente. Las unidades de memoria 232 y 272 proporcionan almacenamiento para los códigos de programa y los datos usados por los controladores 230 y 270, respectivamente.

La FIG. 3 muestra un diagrama de bloques de una realización del procesador de datos TX 220 en el transmisor 210. El procesador de datos TX 220 recibe paquetes de datos, procesa cada paquete basado en el modo seleccionado, y proporciona hasta T bloques de símbolos de datos para el paquete. La FIG. 4 ilustra el procesamiento para un paquete de datos por el procesador de datos TX 220.

Dentro del procesador de datos TX 220, un generador 312 de verificación de redundancia cíclica (CRC) recibe un paquete de datos, genera un valor CRC para el paquete de datos, y añade el valor CRC al paquete de datos para formar un paquete formateado. El valor CRC se usa por el receptor para verificar si el paquete se descodifica correctamente o erróneamente. Igualmente se pueden usar otros códigos de detección de error en lugar de CRC. Un codificador de corrección de error en recepción (FEC) 314 codifica el paquete formateado según un esquema de codificación base y proporciona un paquete codificado o "palabra código". La codificación incrementa la fiabilidad de la transmisión de datos. El codificador FEC 314 puede ejecutar un código Turbo, un código convolucional, un código de verificación de paridad de baja densidad (LDPC), o algún otro código. Por ejemplo, el codificador FEC 314 puede llevar a cabo un código Turbo de velocidad 1/5 y generar bits de código 5K para cada paquete formateado con K bits de entrada, donde K es la dimensión de paquete y pueden ser dependientes sobre el modo seleccionado. Un código Turbo ejemplar de velocidad 1/5 se define por IS-2000 estándar y se describe en un documento 3GPP2 C.S0024, titulada "cdma2000 High Rate Packet Data Air Interface Specification" que está públicamente disponible.

Una unidad de particionamiento 320 recibe los bits de código para cada paquete y proporciona un número suficiente de bits de código para cada bloque basado en el esquema de modulación usado para ese bloque, como se indica mediante un control de codificación a partir del controlador 230. Las memorias intermedias 322a a 332t reciben y almacenan los bits de código para los bloques 1 a T, respectivamente, de cada paquete. Cada memoria intermedia

322 puede también entrelazar (por ejemplo, reordenar) los bits de código para su bloque según un esquema de entrelazación. La entrelazación proporciona diversidad de tiempo y/o frecuencia para los bits de código. Un multiplexor (MUX) 324 se acopla a todas las T memorias intermedias 322a a 322t y proporciona T bloques de bits de código, un bloque a la vez, y si se dirige mediante un control de transmisión IR de un controlador 230. El multiplexor 324 proporciona los bits de código de la memoria intermedia 322a para la primera transmisión de bloques, los bits de código de la memoria intermedia 322b (no mostrada en la FIG. 3) para la segunda transmisión de bloques, etc, y los bits de código de la memoria intermedia 322t para la última transmisión de bloques. El multiplexor 324 proporciona el siguiente bloque de bits de código se recibe un NAK para el paquete de datos. Todas las T memorias intermedias 322a a 322t se pueden purgar siempre que se reciba un ACK.

Una unidad de mapeo de símbolos 326 recibe los bits de código para cada bloque y hace corresponder los bits de código con los símbolos de modulación. El mapeo de símbolos se lleva a cabo según el esquema de modulación usado para el bloque, como se indica mediante un control de modulación del controlador 230. El mapeo de símbolos se puede llevar a cabo por (1) agrupando conjuntos de B bits para formar B valores binarios de bits, donde  $B = 2$  para QPSK,  $B = 3$  para 8-PSK,  $B = 4$  para 16-QAM y  $B = 6$  para 64-QAM, y (2) hacer corresponder cada valor binario de B bits con un punto en una constelación de señales para el esquema de modulación usado para el bloque. La unidad de correspondencia de símbolos 326 proporciona un bloque de símbolos de datos para cada bloque de bits de código.

Por razones de claridad, se describe en lo sucesivo el particionamiento de los bits de código de un paquete codificado en múltiples bloques para un diseño ejemplar. Para este diseño, el código FEC es un código Turbo de velocidad 1/5, el número máximo de transmisiones de bloque es cuatro (es decir  $T=4$ ), la dimensión de paquete es K bits de entradas para todos los modos, y cada bloque contiene F/S símbolos de modulación para una eficiencia espectral de S. El uso de la misma dimensión de paquetes para todos los modos ilustra claramente el procesamiento descrito más adelante para los diferentes modos. En muchos sistemas, el número de símbolos de modulación es fijo para todos los modos, y la dimensión de paquete varía para diferentes modos. De este modo se pueden usar también diferentes dimensiones de paquete para diferentes modos, y también se puede usar una dimensión de bloque fija para todos los modos.

La FIG. 5A muestra un diagrama de bloques de una unidad de particionamiento 320a para el esquema mostrado en la Tabla 1 con el mismo esquema de modulación que se usa para todas las transmisiones de T bloques para un modo dado. Se añade un paquete de datos con un valor CRC para formar un paquete formateado con K bits de entrada, que se codifica entonces para generar un paquete codificado con 5K bits de código. Para el código Turbo de velocidad 1/5, los primeros K bits de código y se denominan bits sistemáticos, y los 4K bits de código restantes se generan mediante el codificador Turbo y se denominan bits de paridad.

La FIG. 5A muestra el particionamiento para el modo 3 en la Tabla 1, que usa QPSK para cada transmisión de bloques. Para el diseño ejemplar, cada bloque contiene  $3K/4$  símbolos de modulación para el modo 3, se pueden enviar y  $3K/2$  bits de código en un bloque usando QPSK. Dentro de la unidad de particionamiento 320a, una unidad de perforación 510a recibe los 5K bits de código para el paquete codificado, proporciona  $3K/2$  bits de código para la primera transmisión de bloques a la memoria intermedia 322a, y proporciona los  $7K/2$  bits de código restantes a una unidad de perforación 510b. Para la transmisión IR, se envían típicamente los K bits sistemáticos y tanto bits de paridad como sean necesarios en la primera transmisión de bloques. Esto permite que el receptor recupere el paquete de datos con solamente una transmisión de bloques en condiciones de canal favorables. Los bits de paridad enviados en cada transmisión de bloques se pueden tomar de todo el paquete codificado basado en un modelo particular de perforación. Se puede conseguir una prestación de descodificación mejorada por dispersión pseudoaleatoria de los bits de paridad para el paquete codificado a través de las múltiples transmisiones de bloques.

La unidad de perforación 510b recibe los  $7K/2$  bits de código de la unidad 510a, selecciona  $3K/2$  bits de código de entre los  $7K/2$  bits de código recibidos basados en un modelo de perforación usado para el segundo bloque, proporciona los  $3K/2$  bits de código seleccionados a la memoria intermedia 322b, y proporciona los 2K bits de código restantes a una unidad de perforación 510c. La unidad 510c selecciona  $3K/2$  bits de código de entre los 2K bits de código recibidos basados en un modelo de perforación usado para una unidad de perforación 510d. Estos  $K/2$  bits de código no son suficientes para otro bloque. Ya que todo el paquete codificado se ha transmitido, el mismo paquete codificado se repite como se muestra en la FIG. 4. En general, el paquete codificado se puede repetir tantas veces como sea necesario para las T transmisiones de bloques para el paquete. La unidad 510d recibe también bits de código del codificador FEC 314, selecciona los  $K/2$  bits de código de la unidad 5120c así como los 1K bits sistemáticos del codificador FEC 314, y proporciona los  $3K/2$  bits de código seleccionados a una memoria intermedia 322d. Los  $3K/2$  bits de código de cada una de las memorias intermedias 322a a 322d son después símbolos correspondidos para obtener  $3K/4$  símbolos de modulación QPSK.

La FIG. 5B muestra un diagrama de bloques de una unidad de particionamiento 320b para el modo 3 de la realización mostrada en la Tabla 2 con múltiples esquemas de modulación usados para un único paquete. Para el modo 3 en la Tabla 2, se usa 8-PSK para el primer bloque y se usa QPSK para cada bloque posterior. Dentro de la unidad de particionamiento 320b, una unidad de perforación 520a recibe los 5K bits de código para el paquete codificado, proporciona  $9K/4$  bits de código para el primer bloque a la memoria intermedia 322a, y proporciona los

11K/4 bits de código restantes a una unidad de perforación 520b. La unidad 520b, selecciona 3K/2 bits de código de entre los 11K/4 bits de código recibidos basados en un modelo de perforación usado para el segundo bloque, proporciona los 3K/2 bits de código seleccionados a la memoria intermedia 322b, y proporciona los 5K/4 bits de código restantes a una unidad de perforación 520c. La unidad 520c también recibe los 5K bits de código del codificador FEC 314, selecciona los 5K/4 bits de código de la unidad 520b así como los primeros K/4 bits sistemáticos del codificador FEC 314, proporciona los 3K/2 bits de código seleccionados a una memoria intermedia 322c, y proporciona los 19K/4 bits de código restantes a una unidad de perforación 520d. La unidad 520d selecciona los 3K/4 bits sistemáticos restantes, selecciona otros 3K/4 bits de paridad basados en un modelo de perforación, y proporciona los 3K/2 bits de código seleccionados a una memoria intermedia 322d. Los bits de código de cada una de las memorias intermedias 322a a 322d son después símbolos correspondidos para obtener 3K/4 símbolos de modulación.

La FIG. 5C muestra un diagrama de bloques de una unidad de particionamiento 320c para el modo 7 de la realización mostrada en la Tabla 2 que usa 64-QAM, 16-QAM, 8-PSK y QPSK para el primer, segundo, tercer y cuarto bloque, respectivamente. Para una dimensión de bloque de K/S, cada bloque contiene K/4 símbolos de modulación para el modo 7 con S = 4, y 3K/2 bits de código se pueden enviar en un bloque usando 64-QAM. Dentro de la unidad de particionamiento 320c, una unidad de perforación 530a recibe los 5K bits de código para el paquete codificado, proporciona 3K/2 bits de código para el primer bloque a la memoria intermedia 322a, y proporciona los 7K/2 bits de código restantes a una unidad de perforación 530b. La unidad 530b selecciona K bits de código seleccionado de entre los 7K/2 bits de código recibidos basados en un modelo de perforación usado para el segundo bloque, proporciona los K bits de código seleccionado a la memoria intermedia 322b, y proporciona los 5K/2 bits de código restantes a una unidad de perforación 530c. La unidad 530c selecciona 3K/4 bits de código de entre los 5K/2 bits de código recibidos basados en un modelo de perforación usado para el tercer bloque, proporciona los 3K/4 bits de código seleccionado a la memoria intermedia 322c, y proporciona los 7K/4 bits de código restantes a una unidad de perforación 530d. La unidad 530d selecciona K/2 bits de código de entre los 7K/4 bits de código recibidos basados en un modelo de perforación usado para el cuarto bloque, y proporciona los K/2 bits de código seleccionados a la memoria intermedia 322d. Los bits de código de cada una de las memorias intermedias 322a a 322d son después símbolos correspondidos para obtener K/4 símbolos de modulación.

Las figuras 5A a 5C muestran el particionamiento y la perforación de un diseño ejemplar y para diversos modos. El particionamiento y la perforación para los T bloques para cada modo se pueden llevar a cabo como se describen anteriormente o de alguna otra manera. Por ejemplo, los bits sistemáticos no se pueden transmitir en primer lugar para el paquete, los bits de código para cada paquete se pueden seleccionar de una manera pseudoaleatoria, etc. El código FEC (por ejemplo convolucional) también pueden no generar bits sistemáticos, en cuyo caso los bits de código para cada bloque se pueden seleccionar de manera pseudoaleatoria desde el paquete codificado. Igualmente, la estructura de trama puede ser diferente de la estructura descrita anteriormente.

La FIG. 6 muestra un diagrama de bloques de una realización del detector 256 y el procesador de datos RX 260 en el receptor 250. Dentro del detector 256 una unidad de cálculo de LLR 610 obtiene los símbolos de datos recibidos procedentes de la unidad receptor 254 y las estimaciones de canal del estimador de canal 258 y calcula las LLR para los símbolos de datos recibidos. Cada símbolo de datos recibidos se puede expresar como:

$$\hat{s}_i = h_i \cdot s_i + \eta_i , \tag{Eq (1)}$$

donde  $s_i$  es el i-ésimo símbolo de datos enviado para un paquete de datos;  
 $h_i$  es una ganancia canal complejo observada por el símbolo de datos  $s_i$ ;  
 $\eta_i$  es el ruido y la interferencia observada por el símbolo de datos  $s_i$ ; y  
 $\hat{s}_i$  es el i-ésimo símbolo de datos recibido para el paquete de datos.

La ecuación (1) supone un canal de comunicación en el cual cada símbolo de datos  $s_i$  observa una sola ganancia de canal  $h_i$ . Este puede ser el caso, por ejemplo, si cada símbolo de datos se envía sobre una subbanda usando OFDM, o si el canal de comunicación tiene una única derivación de canal para una única trayectoria de propagación. Se puede suponer que el ruido es un ruido blanco gaussiano complejo blanco aditivo (AGO) con media cero y una variancia de  $\sigma_v$ .

Cada símbolo de datos recibido  $\hat{s}_i$  es una estimación de un símbolo de datos transmitido  $s_i$ , que se obtiene haciendo corresponder B bits de código  $b_i = [b_{i,1} \ b_{i,2} \dots \ b_{i,B}]$  con un punto en una constelación de señales para el esquema de modulación usado para ese símbolo de datos  $s_i$ . El IR del j-ésimo bit de código para símbolo de datos recibido  $\hat{s}_i$  se puede expresar como:

$$LLR_{i,j} = \log \left[ \frac{\Pr(\hat{s}_i | b_{i,j} = 1)}{\Pr(\hat{s}_i | b_{i,j} = -1)} \right] , \quad \text{for } j = 1, \dots, B , \tag{Eq (2)}$$

en la que

$b_{i,j}$  es el j-ésimo bit de código para símbolo de datos recibido  $\hat{s}_i$ ;

$\Pr(\hat{s}_i | b_{i,j} = 1)$  es la probabilidad de símbolo de datos recibido  $\hat{s}_i$ , siendo el bit  $b_{i,j} = 1$ ;

$\Pr(\hat{s}_i | b_{i,j} = -1)$  es la probabilidad de símbolo de datos recibidos  $\hat{s}_i$ , siendo el bit  $b_{i,j} = -1$  (es decir, '0'); y

$LLR_{i,j}$  es la LLR del bit de código  $b_{i,j}$ .

Una LLR es un valor bipolar, con un mayor valor positivo que corresponde a una mayor probabilidad de que el bit de código sea a +1 y un mayor valor negativo que corresponde a una mayor probabilidad de que el bit de código sea a -1. Una LLR de cero indica que el bit de código es igualmente probable que sea +1 o -1.

Si los B bits de código para cada símbolo de datos recibido  $\hat{s}_i$  son independientes, lo cual se puede conseguir con entrelazación apropiada, entonces se puede expresar la ecuación (2) como:

$$LLR_{i,j} = \log \left[ \frac{\sum_{s \in \Omega_{j,1}} \exp \left[ \frac{-1}{v_i} |\hat{s}_i - \hat{h}_i \cdot s|^2 \right]}{\sum_{s \in \Omega_{j,-1}} \exp \left[ \frac{-1}{v_i} |\hat{s}_i - \hat{h}_i \cdot s|^2 \right]} \right], \quad \text{Eq (3)}$$

en la que

$\Omega_{j,q}$  es un conjunto de puntos en la constelación de señales cuyo j-ésimo bit de código es igual a q;

s es un símbolo de modulación o punto de señal en el conjunto  $\Omega_{j,q}$  que se evalúa, y

$\hat{h}_i$  es una estimación de la ganancia de canal para símbolo de datos recibido  $\hat{s}_i$ .

El conjunto de señales  $\Omega_{j,1}$  para  $q=1$ , el conjunto de señales  $\Omega_{j,-1}$  para  $q = -1$ , y el parámetro B son dependientes del esquema de modulación usado para el símbolo de datos recibido  $\hat{s}_i$ . Se pueden usar diferentes esquemas de modulación para diferentes bloques de un paquete y  $\Omega_{j,1}$ ,  $\Omega_{j,-1}$ , y B pueden ser diferentes para bloques diferentes.

La ecuación (3) se puede evaluar de diversas maneras, como se conoce en la técnica. La unidad 610 calcula B LLR, designados como  $\{LLR_{i,j}\}$ , para los B bits de código de cada símbolo de datos recibido  $\hat{s}_i$ . La unidad 610 también puede combinar LLR calculados para múltiples transmisiones del mismo símbolo de datos  $s_i$  de manera que solamente una LLR se almacene para cada bit de código de un paquete codificado, lo cual puede reducir el requisito de memoria. La unidad 610 también puede cuantificar el LLR para cada bit de código a un número predeterminado de bits para facilitar el almacenamiento. El número de bits a usar para las LLR es dependiente de diversos factores tales como los requisitos del descodificador, la SINR de los símbolos de datos recibidos, etc. La unidad 610 proporciona LLR para los bits de código de cada bloque de datos recibido al procesador de datos RX 260.

Dentro del procesador de datos RX 260, una memoria intermedia de paquete 620 almacena las LLR para los bits de código de cada paquete de datos. Antes de recibir un nuevo paquete de datos, se inicializa o llena la memoria intermedia 620 con borrados, que son valores LLR de cero. Un borrado es un valor que sustituye un bit de código que falta (uno que aun no se ha recibido o no se ha transmitido) y se le da un peso apropiado dato en el proceso de descodificación. Un generador de direcciones 622 genera una dirección apropiada para cada LLR recibido de la unidad 610, de manera que el LLR se almacena en el propio emplazamiento para el paquete. La dirección para el LLR para cada bit de código se puede generar basado en (1) el modo seleccionado para el paquete de datos, (2) el bloque particular en el cual el bit de código se recibe, y (3) el modelo de perforación usado para este bloque, todo lo cual se puede indicar mediante un control de transmisión IR.

Cuando se recibe un nuevo bloque de símbolos de datos procedente del transmisor 210 para un paquete de datos, la descodificación se puede llevar a cabo de nuevo sobre las LLR para todos los bloques recibidos para ese paquete. La memoria intermedia de paquete 620 proporciona una secuencia de LLR y borrados (es decir, un paquete reensamblado) a un descodificador FEC 630 para descodificar. Esta secuencia contiene LLR para todos los símbolos de datos recibidos para el paquete y borrados para todos los símbolos de datos no recibidos para el paquete. Después de recibir el primer bloque, la secuencia contiene LLR para los bits de código llevado en el bloque 1 y borrados para todos los otros bits de código. Después de recibir el segundo bloque, la secuencia contiene LLR para los bits de código llevados en los bloques 1 y 2 y borrados para todos los otros bits de código. El descodificador FEC 630 descodifica la secuencia de LLR y borrados de una manera complementaria a la descodificación FEC llevada a cabo en el transmisor 210, como se indica mediante un control de descodificación del controlador 270. Por ejemplo, se puede usar un descodificador Turbo o un descodificador Viterbi para el descodificador FEC 630 si se lleva a cabo una codificación Turbo o convolucional, respectivamente, en el transmisor 210. El descodificador 630 proporciona un paquete descodificado. Un verificador CRC 632 verifica entonces el paquete descodificado para determinar si el paquete se descodifica correctamente o erróneamente y proporciona el estado del paquete descodificado.

El receptor 250 también puede descodificar un paquete usando una detección iterativa y un esquema de descodificación (IDD). El esquema IDD explota las capacidades de corrección de error del código FEC para proporcionar una prestación mejorada. Esto se lleva a cabo pasando iterativamente una información previa entre la unidad de cálculo LLR 610 y el descodificador 630 para múltiples iteraciones. La información previa indica la probabilidad de los bits de código transmitidos para los símbolos de datos recibidos. Para cada iteración, la unidad de cálculo LLR calcula las LLR para los bits de código basados en los símbolos de datos recibidos, las estimaciones de canal, y las LLR de descodificador procedentes del descodificador FEC 630. La ecuación (2) se puede revisar para tener en cuenta las LLR de descodificador. El descodificador FEC 630 descodifica entonces las LLR actualizados de la unidad 610 para obtener nuevos LLR de descodificador, los cuales pueden ser proporcionados de vuelta a la unidad 610. Durante la detección iterativa y el proceso de descodificación, la fiabilidad de los símbolos de datos detectados mejora con cada iteración de detección/descodificación.

En general, el receptor 250 puede llevar a cabo la detección y descodificación de diversas maneras. La generación de LLR es una implementación específica de descodificación que se usa típicamente para códigos Turbo y convolucionales. El receptor 250 puede usar cualquier técnica de descodificación general aplicable a la técnica de codificación usada en el transmisor 210.

La FIG. 7 muestra un diagrama de flujo de un procedimiento 700 lleva a cabo por el transmisor 210 para transmitir un paquete de datos. El transmisor codifica en primer lugar el paquete de datos (por ejemplo, con un código FEC base) para generar bits de código (bloque 712). Un índice  $l$  para el número de bloques se inicia en 1 para el primer bloque (bloque 714). Para transmitir el  $l$ -ésimo bloque, se forma un bloque de bits de código con los bits de código generados para el paquete de datos y teniendo en cuenta los bits de código ya enviados para el paquete (bloque 722). El esquema de modulación a usar para el  $l$ -ésimo bloque se determina basado en el modo seleccionado para el paquete de datos (bloque 724). Los bits de código para el  $l$ -ésimo bloque se hacen corresponder entonces a los símbolos de datos basados en el esquema de modulación para este bloque (bloque 726). Además, se procesa y transite el  $l$ -ésimo bloque de símbolos de datos (bloque 728). Si el paquete de datos no se descodifica correctamente basado en  $l$  transmisiones de bloques (como se determina en el bloque 730) y si el número máximo de bloques no se ha transmitido (como se determina en el bloque 732), entonces el índice  $l$  se incrementa (bloque 734), y el proceso vuelve al bloque 722 para generar y transmitir el siguiente bloque de símbolos de datos. De otro modo, el proceso termina.

La FIG. 8 muestra un diagrama de flujo de un proceso 800 llevado a cabo por el receptor 250 para recibir un paquete de datos. El receptor inicia en primer lugar la memoria de paquete con borrados para todos los bits de código del paquete de datos (bloque 812). El índice  $l$  para el número de bloques se inicia en 1 para el primer bloque (814). Para el  $l$ -ésimo bloque, se obtiene inicialmente un bloque de símbolos de datos recibidos (bloque 824). El esquema de modulación usado para el  $l$ -ésimo bloque se determina basándose en el modo seleccionado para el paquete de datos (bloque 824). El receptor lleva a cabo entonces la detección en el  $l$ -ésimo bloque se determina basándose en el modo seleccionado para el paquete de datos (bloque 824). El receptor lleva a cabo entonces la detección sobre el  $l$ -ésimo bloque de símbolos de datos recibidos según el esquema de modulación usado para el bloque para obtener LLR para los bits de código enviados en este bloque (bloque 826). Las LLR para el  $l$ -ésimo bloque se pueden combinar con LLR previamente calculadas para los bits de código en este bloque (bloque 828). En cualquier caso las LLR para el  $l$ -ésimo bloque se almacenan en los emplazamientos propios en la memoria intermedia de paquete (también bloque 828). Las LLR y borrados para el paquete de datos se recuperan entonces a partir de la memoria de paquete y se descodifican según el código FEC base para obtener un paquete descodificado (bloque 830), que se verifica, además, para determinar si el paquete se descodificó correctamente o erróneamente (bloque 832). Si el paquete de datos no se descodificó correctamente basándose en  $l$  bloques de símbolos de datos recibidos (como se determina en el bloque 840) y si no se ha obtenido el número máximo de bloques (como se determina en el bloque 842), entonces el índice  $l$  se incrementa (bloque 844), y el proceso vuelve al bloque 822 para obtener y procesar el siguiente bloque de símbolos de datos recibidos. De otro modo, el proceso termina.

El uso de múltiples esquemas de modulación para un único paquete de datos puede proporcionar una prestación mejorada. El uso de un esquema de modulación de orden superior (en combinación con una velocidad de código correspondientemente inferior) para la primera transmisión de bloques se puede proporcionar ganancias considerables (por ejemplo 1 a 2,5 dB) para esta transmisión de bloques para algunos modelos de canal. El uso de esquemas de modulación de orden inferior (y velocidades de código correspondientemente superiores) para posteriores transmisiones de bloques evita o reduce la repetición el paquete codificado, lo cual también puede mejorar la prestación. Por ejemplo, un paquete codificado se repite parcialmente si se usa 64-QAM para las cuatro transmisiones de bloques en el modo 7 y no se repite si se usan 64-QAM, 16-QAM, 8-PSK y QPSK para las cuatro transmisiones de bloques.

La técnica descrita en la presente memoria descriptiva para el uso de múltiples esquemas de modulación para un único paquete se puede implementar por diversos medios. Por ejemplo, estas técnicas se pueden implementar en hardware, software o una combinación de los mismos. Para una implementación de hardware, se pueden implementar las unidades de procesamiento en un transmisor (por ejemplo procesador de datos TX 220) dentro de uno o más circuitos integrados específicos de aplicación (ASIC), procesadores de señales digitales (DSP), dispositivos de procesamiento de señales digitales (DSPD), dispositivos lógicos programables (PLD), sistemas de

compuerta programable de campo (FPGA), procesadores, controladores, microcontroladores, microprocesadores, otras unidades electrónicas, destinadas a llevar a cabo las funciones descritas en el presente documento o una combinación de las mismas. Las unidades de procesamiento en un receptor (por ejemplo detector 256 y procesador de datos RX 260) se pueden también implementar dentro de uno o más ASIC, DSP, etc.

5 Para una implementación de software, las técnicas se pueden implementar con módulos (por ejemplo, procedimientos, funciones, etc.) que llevan a cabo las funciones descritas en el presente documento. Los códigos de software se pueden almacenar en una unidad de memoria (por ejemplo, unidades de memoria 232 y 272 en la FIG. 2) y ejecutar por un procesador (por ejemplo los controladores 230 y 270). La unidad de memoria se puede  
10 implementar dentro del procesador o externamente al procesador.

15 La descripción anterior de las realizaciones reveladas se proporciona para permitir que cualquier persona experta en la técnica realice o use la presente invención. Diversas modificaciones a estas realizaciones serán evidentes para los expertos en la técnica, y los principios genéricos definidos en el presente documento se pueden aplicar a otras realizaciones sin salirse del alcance de la invención. De este modo, la presente invención no está concebida a estar limitada a las realizaciones mostradas en la presente memoria descriptiva sino que se le concede el mayor alcance en coherencia con los principios y las nuevas características reveladas en el presente documento.

**REIVINDICACIONES**

- 5 1. Un procedimiento para recibir datos en un sistema de comunicación, que comprende:
  - estimar en el receptor una condición de un canal de comunicación;
  - seleccionar en el receptor un modo en base a las condiciones del canal;
  - enviar a un transmisor el modo seleccionado;
  - 10 recibir (822, 270) del transmisor una pluralidad de bloques de símbolos de datos para un paquete de datos procesado en el transmisor usando una combinación de una tasa de código y un esquema de modulación que proporciona el mejor rendimiento para un eficiencia espectral designada para el modo seleccionado;
  - determinar (824, 270) en el receptor el esquema de modulación utilizado para cada uno de la pluralidad de bloques en base al modo seleccionado para el paquete de datos;
  - 15 llevar a cabo (826, 256) en el receptor detección en cada uno de la pluralidad de bloques de símbolos de datos recibidos según el esquema de modulación utilizado para el bloque para generar símbolos de datos detectados para el bloque; y
  - decodificar (803, 260) en el receptor símbolos de datos detectados generados a partir de la pluralidad de bloques de símbolos de datos recibidos para obtener un paquete decodificado.
- 20 2. El procedimiento según la reivindicación 1, en el que el modo seleccionado indica un esquema de modulación y una eficiencia espectral para cada uno de la pluralidad de bloques.
3. El procedimiento según cualquiera de las reivindicaciones anteriores, que comprende además:
  - 25 seleccionar un siguiente modo para un siguiente paquete en base a una estimación de canal;
  - enviar el siguiente modo al transmisor; en donde el siguiente modo utiliza un siguiente esquema de modulación para transmitir el siguiente paquete de datos.
- 30 4. El procedimiento según cualquiera de las reivindicaciones anteriores, en el que llevar a cabo detección en cada uno de la pluralidad de bloques de símbolos de datos recibidos comprende generar un conjunto de tasas de log-verosimilitud (LLRs) para un conjunto de bits de código para cada símbolo de datos recibido.
- 35 5. El procedimiento según cualquiera de las reivindicaciones anteriores, en el que el modo indica al menos dos esquemas de modulación.
6. El procedimiento según cualquiera de las reivindicaciones anteriores, en el que llevar a cabo detección en cada uno de la pluralidad de bloques de símbolos de datos recibidos comprende
  - 40 llevar a cabo detección en un primer bloque de símbolos de datos recibidos, obtenido primero, según un primer esquema de modulación que tiene un orden mayor entre al menos dos esquemas de modulación diferentes.
- 45 7. El procedimiento según la reivindicación 1, en el que llevar a cabo detección en cada uno de la pluralidad de bloques de símbolos de datos recibidos comprende además
  - llevar a cabo detección en un segundo bloque de símbolos de datos recibidos, obtenido segundo, según un segundo esquema de modulación que tiene un segundo mayor orden entre los al menos dos esquemas de modulación diferentes.
- 50 8. El procedimiento según cualquiera de las reivindicaciones anteriores, en el que la decodificación de los símbolos de datos detectados comprende después de obtener cada uno de la pluralidad de bloques de símbolos de datos recibidos, decodificar símbolos de datos detectados generados para todos los bloques de símbolos de datos recibidos para obtener el paquete decodificado.
- 55 9. El procedimiento según cualquiera de las reivindicaciones anteriores, que comprende además:
  - combinar (828, 260) múltiples símbolos de datos detectados generados para cada símbolo de modulación.
- 60 10. Un medio legible por ordenador que comprende código para hacer que un ordenador lleve a cabo el método según cualquiera de las reivindicaciones anteriores,
- 65 11. Un aparato para recibir datos para un sistema de comunicaciones, adaptado para
  - estimar una condición de un canal de comunicación;
  - seleccionar un modo en base a la condición de canal;
  - enviar al transmisor el modo seleccionado; comprendiendo medios (254) para recibir del transmisor un pluralidad de bloques de símbolos de datos para un paquete de datos procesado en el transmisor utilizando

- una combinación de una tasa de código y un esquema de modulación que proporciona el mejor rendimiento para una eficiencia espectral designada para el modo seleccionado;
- 5 medios (270) para determinar el esquema de modulación utilizado para cada uno de la pluralidad de bloques de símbolos de datos recibidos en base al modo seleccionado para el paquete de datos;
- medios (256) para llevar a cabo detección en cada uno de la pluralidad de bloques de símbolos de datos recibidos según el esquema de modulación utilizado para el bloque para generar símbolos de datos detectados para el bloque; y
- 10 medios (260) para decodificar los símbolos de datos detectados generados a partir de la pluralidad de bloques de símbolos de datos recibidos para obtener un paquete decodificado.
12. El aparato según la reivindicación 11, en el que el modo seleccionado indica un esquema de modulación y una eficiencia espectral para cada uno de la pluralidad de bloques.
- 15 13. El aparato según cualquiera de las reivindicaciones 11 ó 12, adaptado además para:
- seleccionar un siguiente modo para un siguiente paquete en base a una estimación de canal;
- 20 enviar el siguiente modo al transmisor;
- en el que el transmisor es operable para utilizar un siguiente esquema de modulación para transmitir un siguiente paquete de datos.
- 25 14. El aparato según cualquiera de las reivindicaciones 11 a 13, en el que los medios para llevar a cabo detección en cada uno de la pluralidad de bloques de símbolos de datos recibidos está además adaptado para generar un conjunto de tasas de log-verosimilitud (LLRs) para un conjunto de bits de código para cada símbolo de datos recibido.
- 30 15. El aparato según cualquiera de las reivindicaciones 11 a 14, en el que los medios para llevar a cabo detección en cada uno de la pluralidad de bloques de símbolos de datos recibidos está además adaptado para llevar a cabo detección en un primer bloque de símbolos de datos recibidos, obtenido primero, según un esquema de modulación que tiene un orden mayor entre al menos dos esquemas de modulación diferentes.



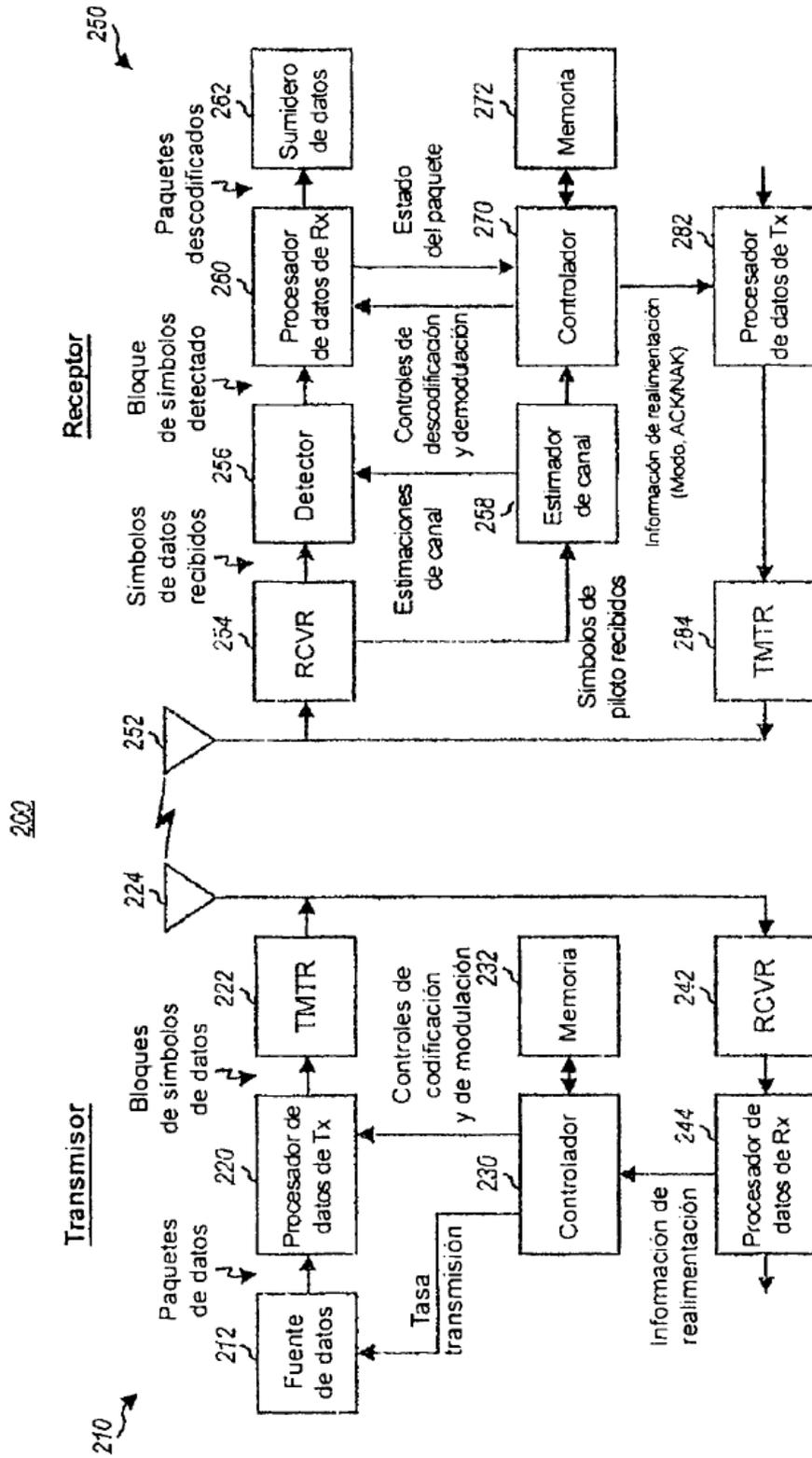


FIG. 2

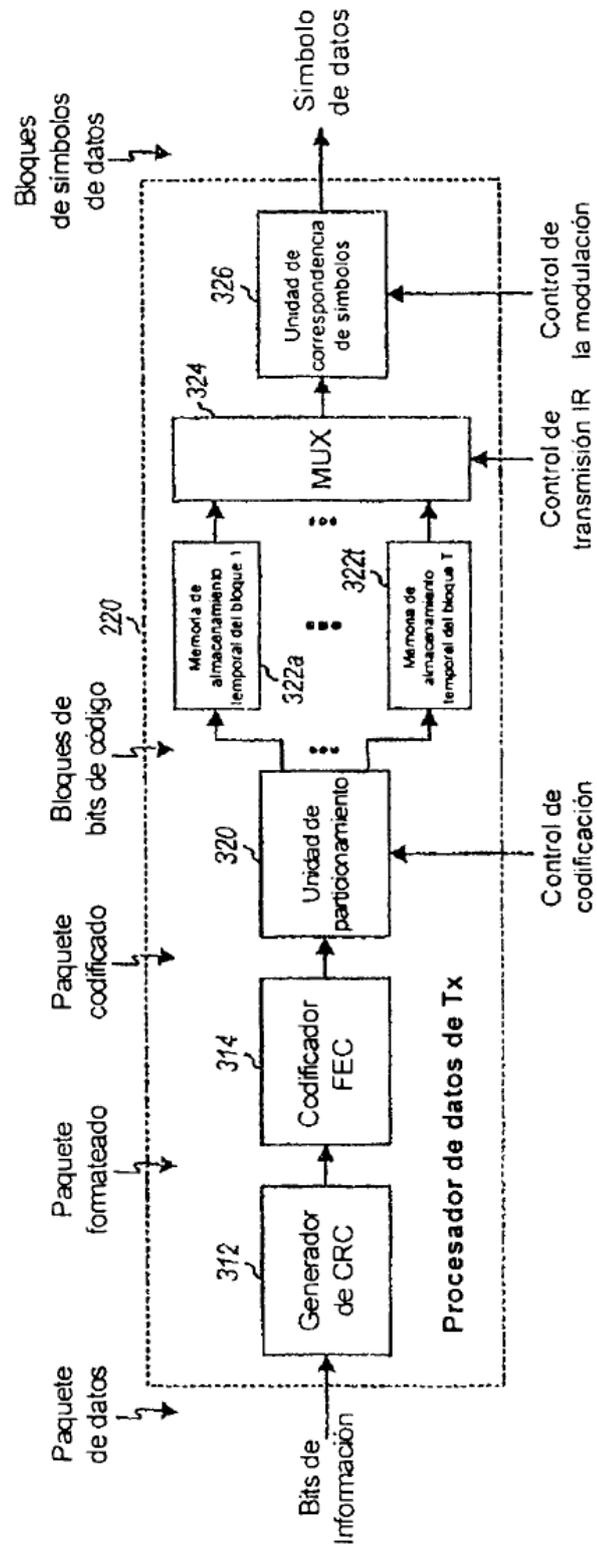


FIG. 3

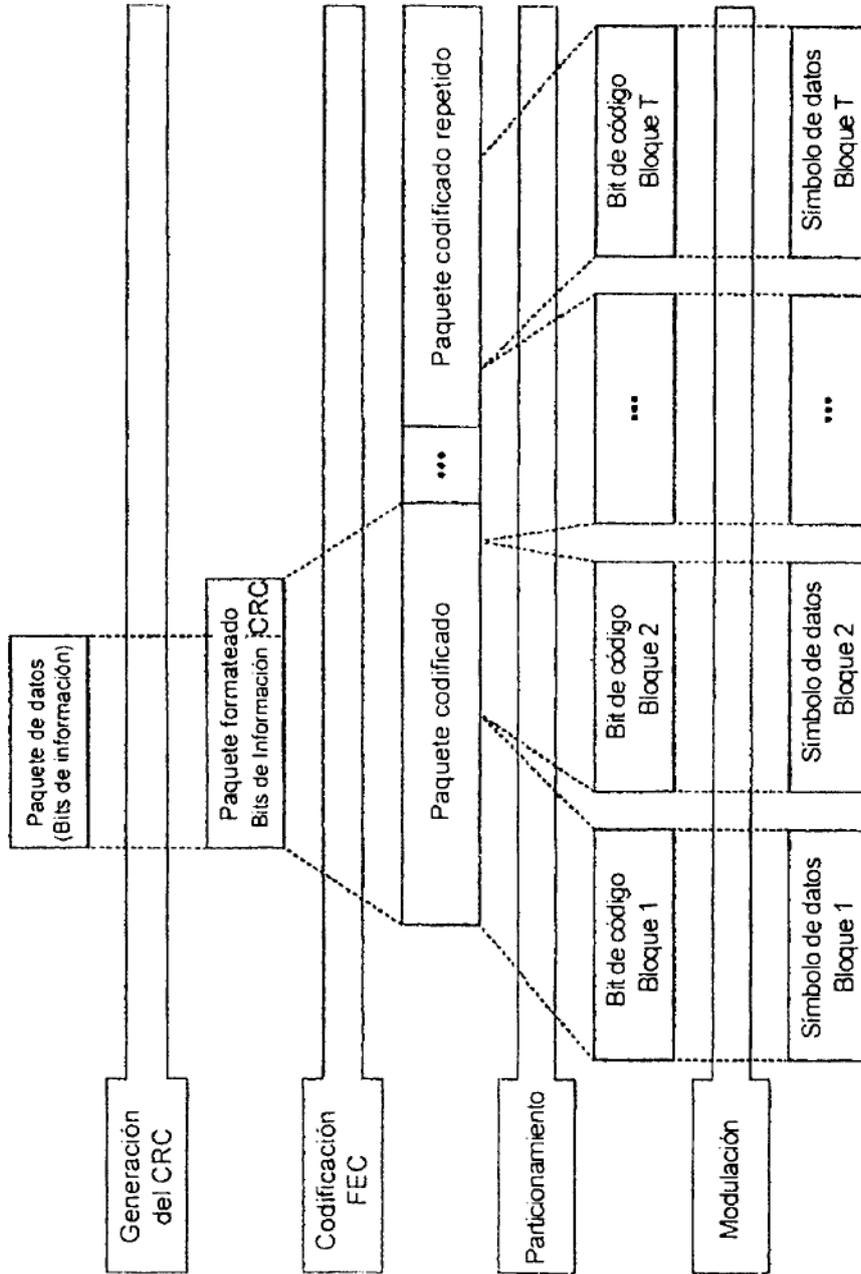


FIG. 4

Modo 3 - Esquema de modulación simple para un paquete

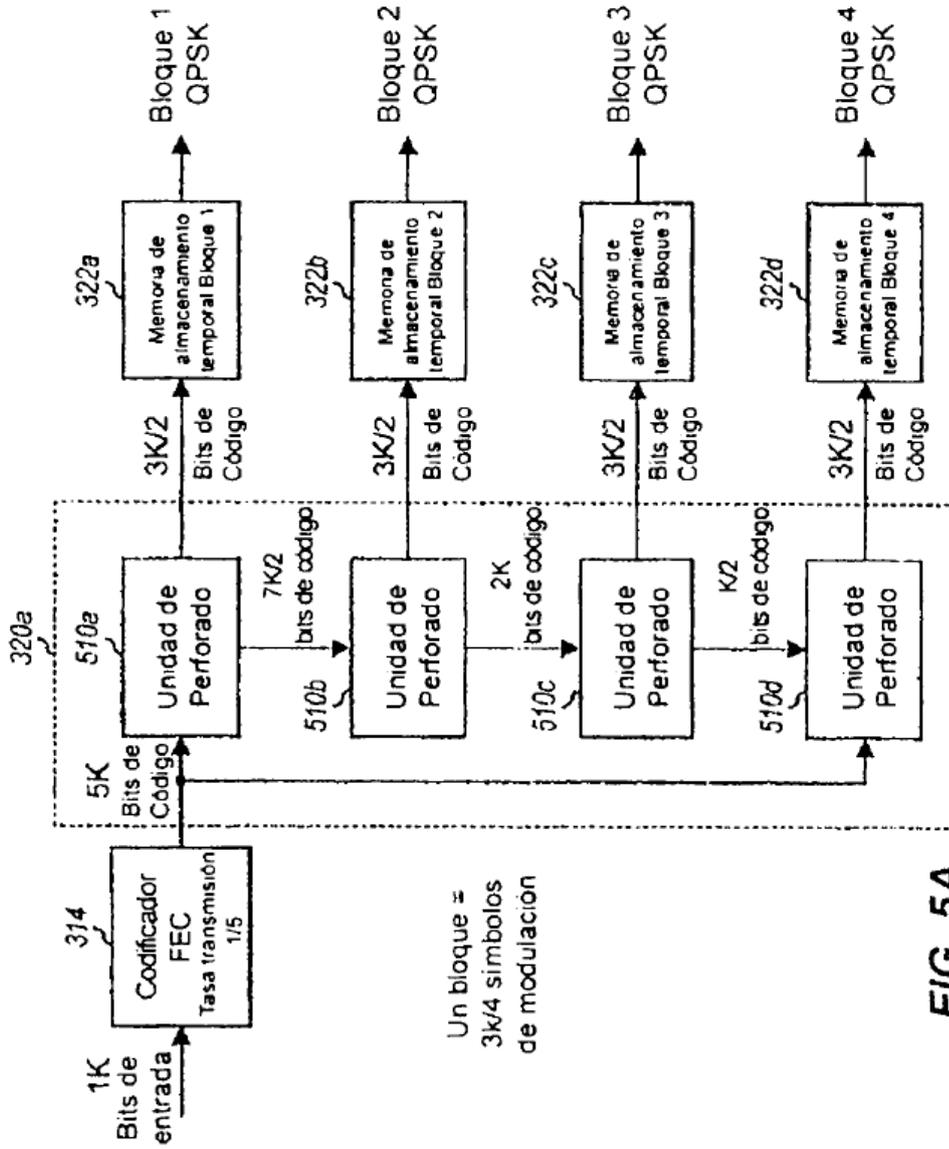


FIG. 5A

Esquemas de modulación múltiple para un paquete

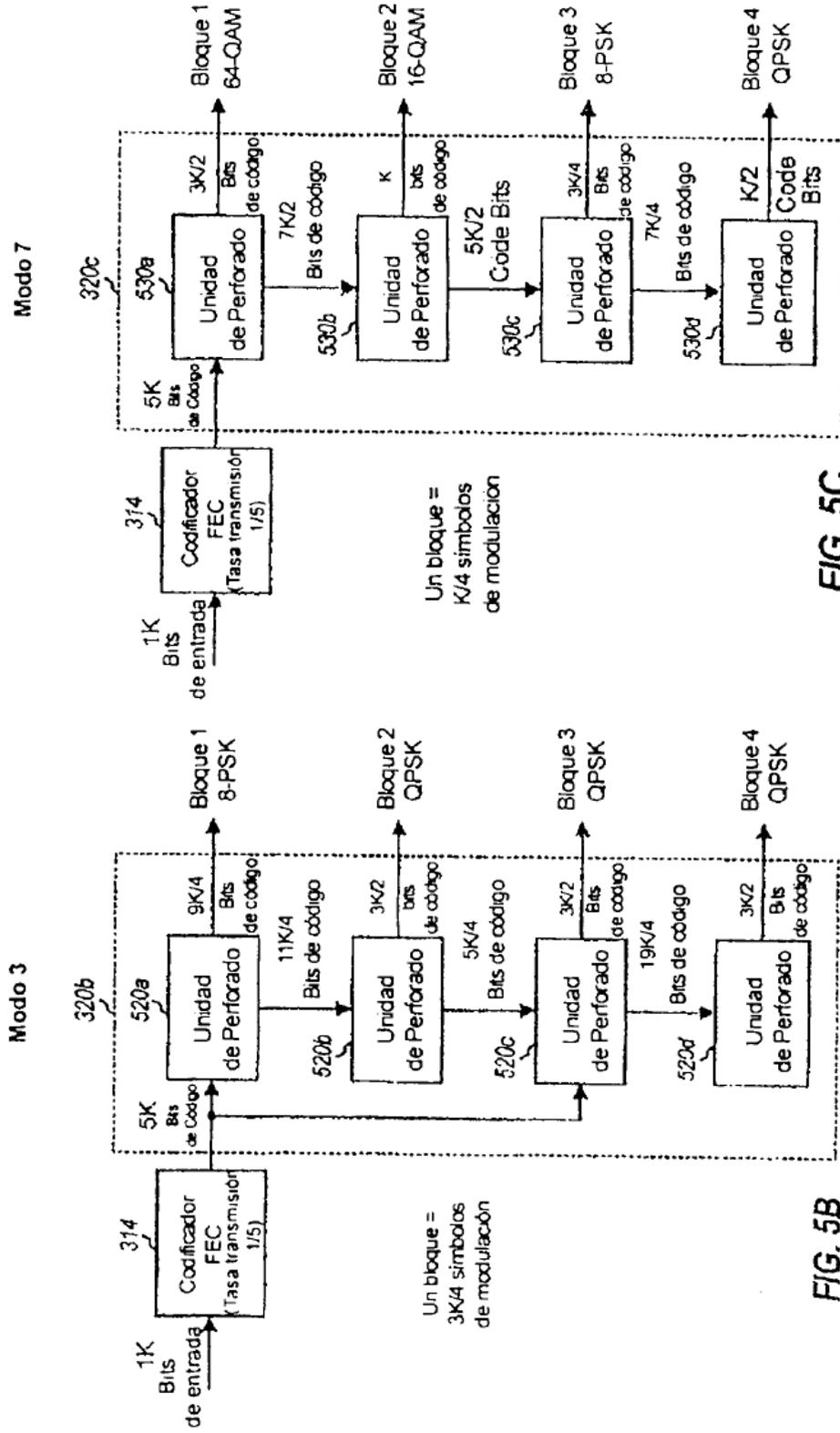


FIG. 5C

FIG. 5B

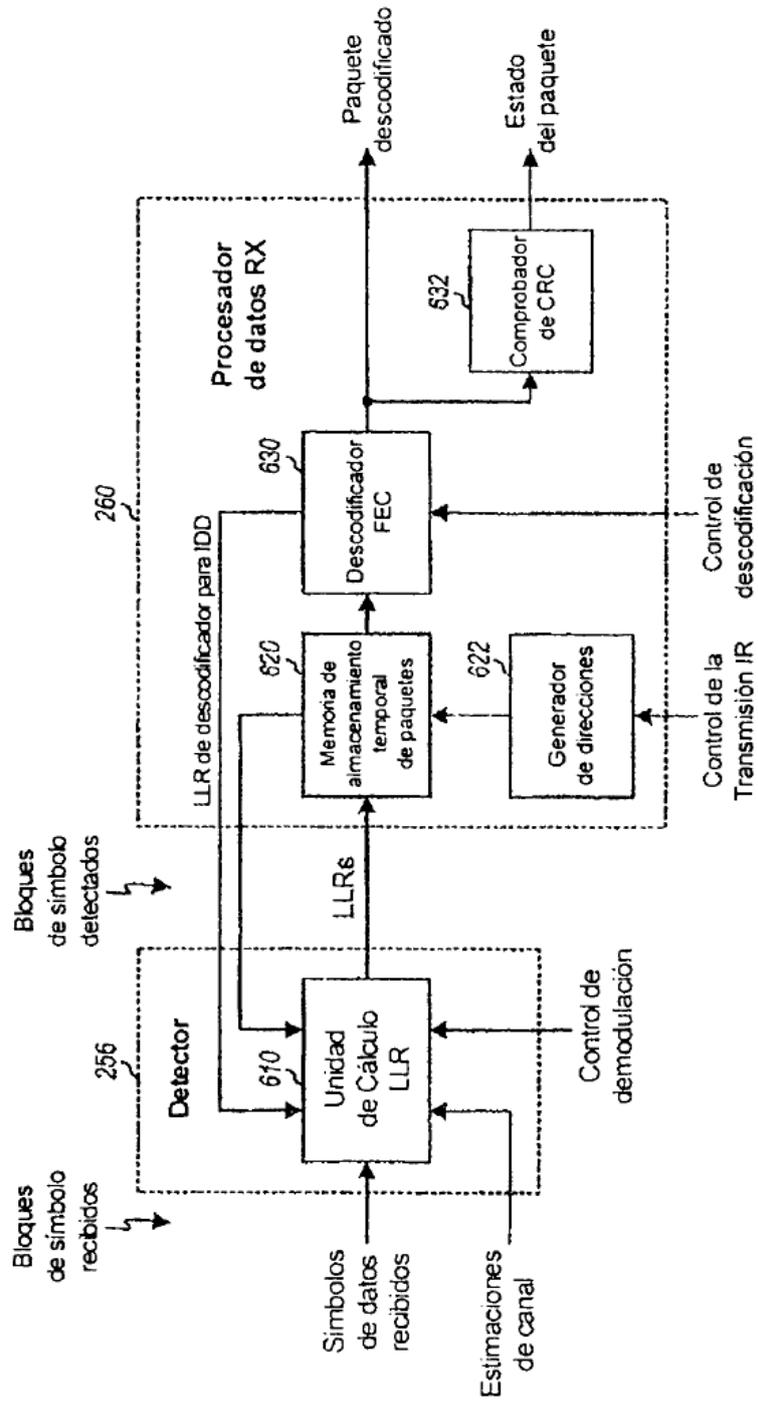


FIG. 6

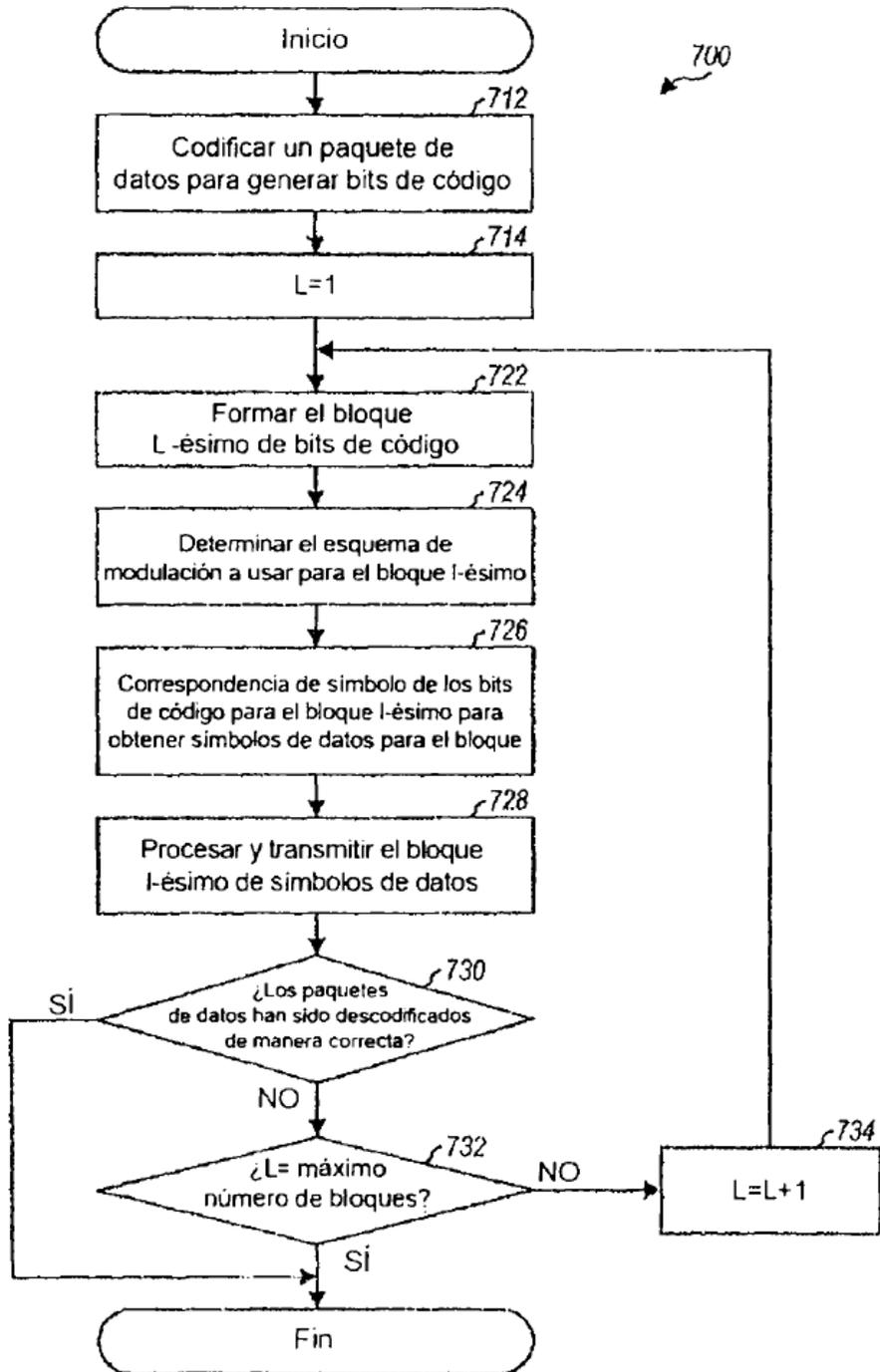


FIG. 7

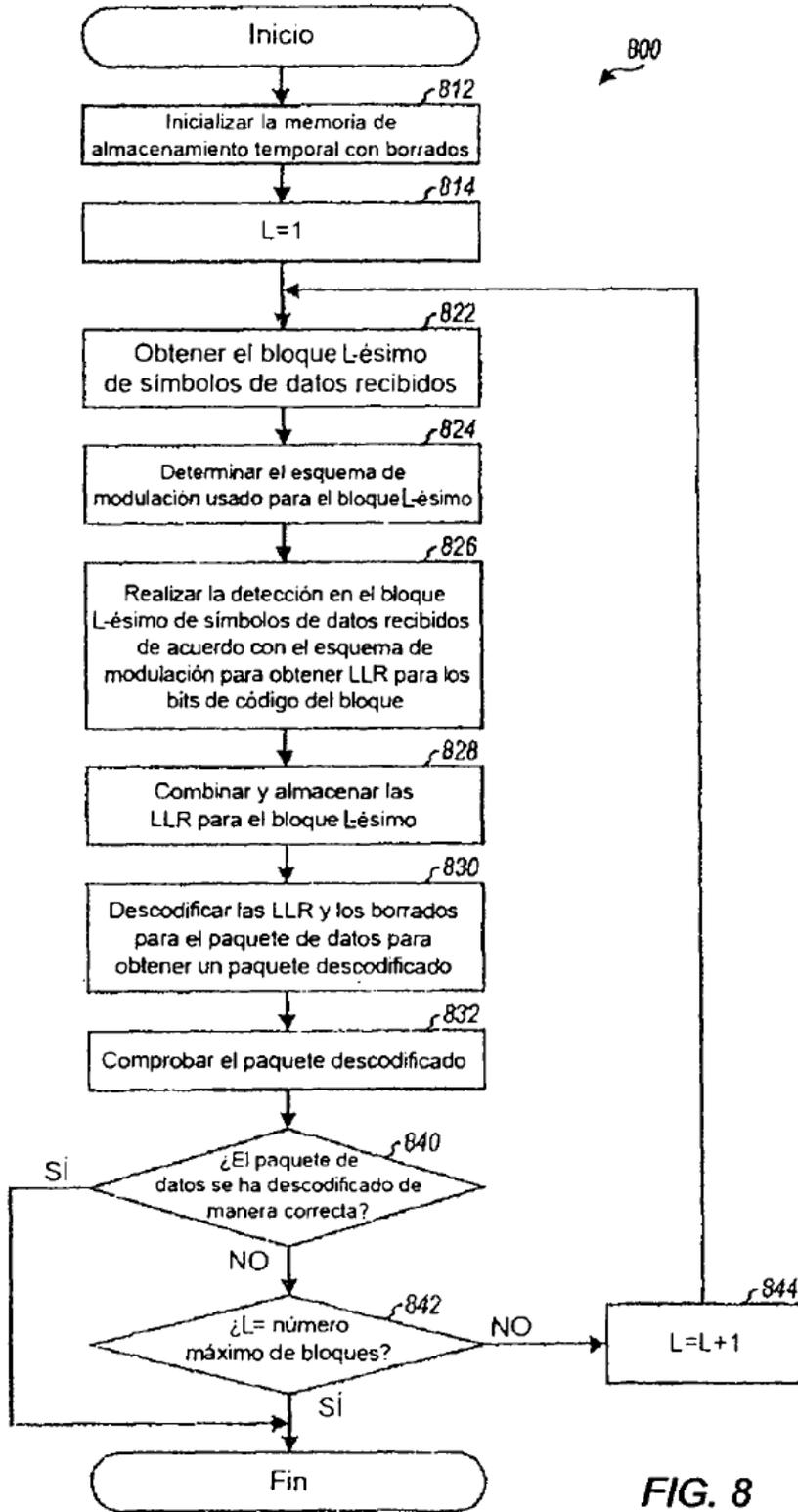


FIG. 8