(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 545 601**

(51) Int. CI.:

H04W 72/12 (2009.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **07.10.2011 E 11306305 (1)**

(97) Fecha y número de publicación de la concesión europea: **13.05.2015 EP 2440001**

(54) Título: **Procedimiento y dispositivo de transmisión de informaciones en contención en intervalos de tiempo entre nodos emisores-receptores de una red ad hoc**

(30) Prioridad:

08.10.2010 FR 1003993

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
14.09.2015

(73) Titular/es:

THALES (100.0%)
45, rue de Villiers
92200 Neuilly Sur Seine, FR

(72) Inventor/es:

FOUILLOT, PASCALE MARIE ADELINE y
MASSIN, RAPHAËL ANDRÉ MICHEL

(74) Agente/Representante:

PONTI SALES, Adelaida

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 dispositivo de transmisión de informaciones en contención en intervalos de tiempo entre nodos emisores-receptores de una red ad hoc

[0001] La presente invención se refiere a un procedimiento de transmisión de informaciones en una red ad hoc que comprende al menos dos nodos emisores-receptores adecuados para recibir y emitir informaciones y que se comunican entre sí mediante el envío de informaciones en intervalos de tiempo en acceso aleatorio organizados en tramas.

[0002] Una red ad hoc es una red inalámbrica capaz de organizarse sin infraestructura definida previamente. No existe ningún elemento central o punto de acceso que permita gestionar las comunicaciones entre las diferentes entidades de la red. Cualquier nodo de dicha red es a la vez nodo terminal y nodo de retransmisión. En dicha red un nodo se considera vecino de otro cuando están suficientemente cerca para que sea posible una comunicación. Dos nodos demasiado alejados para que sea posible una comunicación entre los dos pero suficientemente cerca para que cada uno genere interferencias destructivas en el otro, son nodos interferentes.

[0003] Una red ad hoc puede estar constituida usando la tecnología WiFi descrita en el documento IEEE Std 802.11™-2007, IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007. Un nodo de esta red que desee transmitir informaciones a un nodo destinatario de esta misma red difunde primero a todos los nodos vecinos un mensaje que comprende datos de señalización que indican el identificador del nodo destinatario. Existe una colisión en un nodo cuando se reciben simultáneamente al menos dos mensajes en un mismo intervalo de tiempo. Si no hay colisión con otro mensaje, el nodo recibe un mensaje de confirmación del nodo destinatario y le envía entonces los datos útiles. En caso contrario debe efectuar una retransmisión de datos de señalización. La retransmisión sólo se realiza transcurrido un tiempo de espera cuya duración se calcula mediante un algoritmo, por ejemplo un algoritmo de retroceso exponencial. Esta duración se expresa a menudo en un múltiplo de una cierta cantidad de tiempo, que puede variar de una unidad a varias decenas, o bien centenas. Este mecanismo permite limitar las colisiones pero conduce a tiempos de espera largos entre dos intentos de transmisión de datos de señalización si las cantidades de tiempo tienen una duración por ejemplo del orden de 5 ms.

[0004] El documento “Improving Quality-of-Service in Wireless Sensor Networks by Mitigating ’Hidden-Node Colisión’” Koubâa, IEEE 2009 desvela un procedimiento de distribución de nodos en red ad hoc en grupos, de manera que cada grupo emite en una ventana de tiempo.

[0005] El objeto de la invención es proponer un mecanismo de reducción de colisiones que no imponga un retardo importante.

[0006] Para este efecto, la invención tiene por objeto un procedimiento según la reivindicación 1.

[0007] La invención tiene asimismo por objeto un procedimiento según las reivindicaciones dependientes 2 a 11.

[0008] La invención tiene asimismo por objeto una red ad hoc según la reivindicación 12.

[0009] Estas características y ventajas de la invención serán evidentes a partir de la lectura de la descripción que se ofrece a continuación, proporcionada únicamente a modo de ejemplo, y hecha con referencia a los dibujos adjuntos, en los que:

- la figura 1 es una vista esquemática de una red ad hoc en un instante temporal dado;

- la figura 2 es un organigrama del algoritmo de reducción de colisiones del procedimiento según la invención;

- la figura 3 es una tabla que muestra diferentes situaciones durante el envío de mensajes en una misma trama;

- las figuras 4, 5 y 6 son tablas que representan ejemplos de transmisión de mensajes en una trama;

- las figuras 7, 8, 9 y 10 son vistas esquemáticas de un ejemplo de aplicación del algoritmo de reducción de

colisiones del procedimiento según la invención;

- las figuras 11, 12, 13, 14, 15, 16, 17, 18, y 19 son esquemas que representan ejemplos de implementación del algoritmo del procedimiento según la invención.

[0010] La red ilustrada en la figura 1 es por ejemplo una red ad hoc de comunicación entre individuos en un terreno de operaciones. En esta figura los alcances útiles de los nodos de la red se representan mediante trazo continuo. Esta red ad hoc usa intervalos de tiempo de acceso aleatorios para transmitir mensajes. Estos intervalos de tiempo designados en inglés habitualmente por slot están distribuidos en tramas con un número de intervalos de tiempo en cada trama constante de una trama a otra

[0011] Esta red responde por ejemplo a la norma WiMax móvil 802.16m.

[0012] Cada individuo está equipado con un emisor-receptor que constituye un nodo de la red. Cada nodo incluye medios de comunicación para recibir mensajes y emitirlos en tramas divididas en intervalos de tiempo. Un intervalo de tiempo permite la transmisión de un mensaje cuya dimensión es compatible con la duración de los intervalos de tiempo. Después, en cada trama, se transmite un mensaje durante un intervalo de tiempo. La elección de este intervalo de tiempo se realiza de manera aleatoria. Este intervalo de tiempo se usa para una transmisión en difusión.

[0013] La red representada en la figura 1 incluye once nodos, numerados de 1 a 11, y muestra cuatro situaciones de colisiones en los nodos 1, 3, 8 y 10. En el nodo 1, existe una colisión entre los mensajes emitidos por los nodos 2 y 4, en el nodo 3, existe una colisión entre los mensajes emitidos por los nodos 2 y 5, en el nodo 8, existe una colisión entre los mensajes emitidos por los nodos 4 y 5, y en el nodo 10, existe una colisión entre los mensajes emitidos por los nodos 4 y 5.

[0014] Cada nodo es adecuado para implementar de forma continua el algoritmo de la figura 2. Este algoritmo se implementa para cada nueva trama recibida por el nodo considerado.

[0015] Con el fin de permitir la implementación del algoritmo, con vistas a reducir el número de colisiones, las variables contenidas en dos campos siguientes se añaden a cada mensaje dirigido en un intervalo de tiempo del mensaje transmitido por el nodo:

- *msgConstraint* que representa una restricción impuesta por el nodo emisor a cada uno de sus nodos vecinos; y

- *ownConstraint* que representa la restricción que el nodo emisor se impone en función de las restricciones que le han sido impuestas por los nodos vecinos.

[0016] Un nodo sometido a una restricción *ownConstraint* tiene el derecho de emitir todas las *ownConstraint* tramas, y debe renunciar a emitir en las otras tramas. El algoritmo asegura así la definición de la restricción adecuada para cada nodo, lo que llega a distribuir los nodos en grupos, cada uno asociado a tramas, teniendo sólo los nodos del grupo asociado a una trama dada el derecho de emitir en esta trama. El algoritmo reduce así los riesgos de colisión.

[0017] Esta restricción *ownConstraint* es determinada por cada nodo para sí mismo en función de las restricciones m*sgConstraint* impuestas por los nodos vecinos.

[0018] Así, cada nodo:

- Estima el número de vecinos o interferentes que emiten en su vecindad en función del número de slots recibidos correctamente, en colisión y libres;

- Calcula la probabilidad de colisión en su vecindad;

- Calcula una restricción que se impondrá a sus vecinos para alcanzar una probabilidad de colisión umbral;

- Se somete él mismo a las restricciones impuestas por sus vecinos;

[0019] Una fase de inicialización 10 antecede a la ejecución del algoritmo. Durante esta fase se inicializan

algunas variables de estado:

- *neighborConstraint* es un número entero que representa la restricción impuesta por el nodo considerado a cada uno de sus nodos vecinos, inicializada al valor 1,

- *nodeToWait* es el identificador del nodo emisor de la restricción a la que se somete el nodo considerado, inicializada al valor “indefinido”,

- *frameNumber* es el número de la trama actual, inicializado a cero.

[0020] Más exactamente, el algoritmo incluye cuatro fases principales 20, 22, 24 y 26 ilustradas en la figura 2 que corresponden respectivamente a una estimación del número de vecinos e interferentes que emiten en una trama durante la fase 20, un registro de las restricciones enviadas por los vecinos en el curso de una trama, durante la fase 22, una actualización, en ciertas condiciones, de la restricción que se impondrá a sus vecinos, durante la fase 24 y una actualización, en ciertas condiciones, de la restricción a la que el nodo se somete en el curso de una trama durante la fase 26.

[0021] Las dos primeras fases 20, 22 se efectúan en cada nuevo intervalo de tiempo mientras que las dos últimas fases 24, 26 se efectúan en cada fin de trama.

[0022] La fase 20 consiste en una actualización de las variables que contienen el número de intervalos libres, legibles y en colisión del nodo considerado desde que recibe las informaciones de un nuevo intervalo de tiempo. Estas variables le permitirán posteriormente estimar el número de nodos que ha emitido en su vecindad en fin de trama.

[0023] La fase 22 sólo se realiza si el intervalo de tiempo en curso puede leerse. Consiste en un registro de las restricciones enviadas por el nodo emisor vecino del nodo considerado.

[0024] La fase 24 consiste en un cálculo del número total de nodos emisores, vecinos o interferentes del nodo considerado, a partir de las variables actualizadas durante la fase 20 y en ciertas condiciones, una determinación de la regla de distribución que se impondrá en los nodos vecinos del nodo considerado, teniendo en cuenta el número de nodos emisores, vecinos o interferentes estimados.

[0025] Finalmente, la fase 26 consiste en una actualización de la regla de distribución a la que debe someterse el nodo considerado.

[0026] El mismo algoritmo es implementado de manera permanente por cada nodo de la red.

[0027] En este caso se considera la aplicación del algoritmo de la figura 2 para un nodo de una red ad hoc.

[0028] Este algoritmo se implementa para cada nueva trama cuya recepción comienza en la etapa 27 en el curso de la cual la variable *frameNumber* se incrementa en 1.

[0029] Las variables usadas por el algoritmo se inicializan primero para cada nueva trama durante una etapa 28, en la forma:

nbReadableSlot = 0, *nbCollidedSlot* = 0 y *nbIdleSlot* = 0, *ownConstraintTemp* = 1 y *changeConstraint* = “falso” para el nodo considerado, respondiendo estas variables a las definiciones siguientes:

- *nbReadableSlot* es el número de intervalos de tiempo legibles en una trama. Un intervalo de tiempo puede leerse si se ha recibido un mensaje, es decir, si exactamente un nodo vecino del nodo considerado ha intentado acceder al intervalo de tiempo.

- *nbCollidedSlot* es el número de intervalos de tiempo en el curso de una trama en los que han tenido lugar colisiones. Un intervalo de tiempo está en colisión si al menos dos nodos vecinos o interferentes del nodo considerado han enviado un mensaje en este intervalo.

- *nbIdleSlot* es el número de intervalos de tiempo libres en una trama. Un intervalo de tiempo está libre si ningún nodo vecino o interferente ha enviado mensajes en este intervalo.

- *ownConstraintTemp* es la restricción más elevada recibida de los nodos vecinos del nodo considerado en el curso de la trama en curso de recepción. Una regla de distribución impone a los nodos que se someten a ella que emitan una sola vez todas las q tramas, siendo q denominada restricción. *ownConstraint* es la restricción q para la que el nodo considerado se somete en un instante dado, que puede ser diferente de *ownConstraintTemp,* y

- *changeConstraint* es una variable booleana que toma el valor "verdadero" si la restricción *ownConstraint* debe ser actualizada.

[0030] Las fases 20 y 22 son implementadas para cada nuevo intervalo de tiempo de la trama recibida en la etapa 29.

[0031] La fase 20 comienza con la recepción del mensaje contenido en el nuevo intervalo de tiempo por una primera etapa 30 que consiste en determinar si este intervalo de tiempo está libre o no.

[0032] Si el intervalo está libre, la variable *nbldleSlot* se incrementa en una unidad en el curso de una etapa 32 y se pasa directamente al final de la fase 22, en el que se determina durante una etapa 34 si la trama determinada a la etapa 27 se termina o no. Si no se termina, el nodo considerado debe tratar las informaciones contenidas en un nuevo intervalo de tiempo detectado en la etapa 29. En caso contrario se implementa la fase 24.

[0033] Si el intervalo no está libre durante la prueba de la etapa 30, se determina en una etapa 36 si el intervalo puede leerse o no.

[0034] Si el intervalo no puede leerse la variable *nbCollidedSlot* se incrementa en una unidad en el curso de una etapa 38 y se pasa directamente a la etapa 34 al final de la fase 22.

[0035] Si el intervalo puede leerse la variable *nbReadableSlot* se incrementa en una unidad en el curso de una etapa 40 y a continuación se pasa a la fase 22.

[0036] La fase 22 sólo se realiza así si el intervalo de tiempo en curso puede leerse.

[0037] Esta fase comienza con una etapa 42 en el curso de la cual el nodo compara la restricción recibida, representada por la variable *msgConstraint,* con la variable *ownConstraintTemp* que es la restricción más elevada recibida de los nodos vecinos del nodo considerado en el curso de la trama en curso de recepción. La restricción más elevada entre la restricción recibida *msgConstraint* y la variable *ownConstraintTemp,* hasta la más elevada, se registra en la variable *own-ConstraintTemp.*

[0038] A la etapa 44 siguiente, el identificador del nodo emisor de esta restricción la más elevada se registra en una variable *nodeToWaitTemp.*

[0039] La etapa 46 consiste en verificar si el nodo *nodeToWait* que ha transmitido en el curso de las tramas precedentes la restricción *msgConstraint* a la que se somete el nodo considerado ha reemitido una restricción más débil. Si no sucede así, se pasa a la etapa 34, y en caso contrario la variable *changeConstraint* toma el valor "verdadero" en el curso de una etapa 48 y a continuación se pasa a la etapa 34.

[0040] La fase 24 comienza por una etapa 50 de cálculo del número de nodos vecinos o interferentes que ha emitido en la trama que acaba de ser recibida, este número se memoriza en una tabla *nbNeighborPerFrame* indexada por el número de la trama actual módulo *neighborConstraint.* En lo sucesivo, para facilitar la notación se omite el índice. Para efectuar el cálculo, se usa la relación:

$$nbNeighborPerFrame = nbReadableSlot + k \cdot nbCollidedSlot \text{ con } k \geq 2$$

[0041] El coeficiente k se fija primero en la inicialización de la red en función del número medio de nodos vecinos o interferentes por nodo y del número de intervalos de tiempo en una trama. A continuación su valor puede cambiarse dinámicamente para reflejar mejor la topología local de la red alrededor del nodo considerado.

[0042] Los ejemplos que se ofrecen a continuación ilustran la elección del coeficiente k.

[0043] En la figura 3, una red ad hoc está compuesta por ocho nodos numerados de 1 a 8. Un nodo 8 está en

escucha de los mensajes enviados por nodos en su vecindad y busca estimar su número de nodos emisores, vecinos o interferentes. Una tabla 90 representa una trama dividida en cuatro intervalos de tiempo 92, 94, 96 y 98, durante los cuales los nodos numerados de 1 a 7 emiten mensajes. Analizando los mensajes recibidos en esta trama, el nodo ocho detecta una colisión durante los intervalos de tiempo 94 y 98; los dos intervalos de tiempo 92 y 96 son de por sí legibles. Una estimación del número de nodos vecinos o interferentes que ha emitido en la trama consiste en aplicar la relación (1). En el ejemplo de la figura 3, las variables *nbReadableSlot* y *nbCollidedSlot* valen dos. Eligiendo k=2, el nodo 8 estima que tiene seis vecinos.

[0044] En la figura 4, una trama 100 se divide en cinco intervalos de tiempo 101, 102, 103, 104 y 105. Siete nodos, numerados de 1 a 7, emiten mensajes en esta trama. La relación entre el número de nodos y el número de intervalos de tiempo vale entonces 1,4. Se calcula un k ideal igual al número de mensajes en colisiones dividido por el número de intervalos de tiempo en los que se producen las colisiones. En la figura 4, hay cuatro mensajes 106, 107, 108 y 109 que entran en colisión, produciéndose estas colisiones en el curso de los dos intervalos de tiempo 102 y 105. El k ideal toma como valor 2.

[0045] En la figura 5, una trama 120 se divide en cuatro intervalos de tiempo 121, 122, 123 y 124. Siete nodos, numerados de 1 a 7, emiten mensajes en esta trama. La relación entre el número de nodos y el número de intervalos de tiempo vale entonces 1,75. En la figura 5, hay cinco mensajes 125, 126, 127, 128 y 129 que entran en colisión, produciéndose estas colisiones en el curso de los dos intervalos de tiempo 122 y 124. El k ideal toma como valor 2,5.

[0046] En la figura 6, una trama 140 se divide en cinco intervalos de tiempo 141, 142, 143, 144 y 145. Diez nodos, numerados de 1 a 10, emiten mensajes en esta trama. La relación entre el número de nodos y el número de intervalos de tiempo vale entonces 2. En la figura 6, hay cinco mensajes 146, 147, 148, 149 y 150 que entran en colisión, produciéndose estas colisiones en el curso de los dos intervalos de tiempo 142 y 145. El k ideal toma como valor 2,6.

[0047] Las figuras 4, 5 y 6 ponen de relieve el hecho de que cuanto más elevada es la relación entre el número de nodos y el número de intervalos de tiempo en una trama, mayor es el factor k ideal.

[0048] Durante el inicio de la fase 24, el nodo considerado impone para el momento en los nodos vecinos una regla de distribución cuyo valor de la restricción se memoriza en una variable *neighborConstraint.*

[0049] La estimación del número total de nodos emisores, vecinos o interferentes del nodo considerado que permite determinar a continuación la nueva restricción que se impondrá a los nodos vecinos se realiza en las últimas Q = *neighborConstraint* tramas, sumando las vecindades estimadas en cada una de las Q tramas.

[0050] Para este efecto, durante una etapa 52, el nodo determina si se trata de la última estimación antes de poder estimar el número total de nodos emisores, vecinos o interferentes y de calcular la nueva restricción que se impondrá a los nodos vecinos.

[0051] Si no se trata de la última estimación y si el nodo tiene el derecho de emitir en la siguiente trama, lo que se verifica durante una etapa 53, el nodo considerado elige de manera aleatoria, en la etapa 54, un intervalo de tiempo para emitir en la siguiente trama las informaciones útiles así como las restricciones *msgConstraint,* que toma el valor de *neighborConstraint* y *ownConstraint.* Durante la etapa 53 el nodo determina si tiene el derecho de emitir verificando si pertenece al grupo que tiene este derecho, distribuyéndose los diferentes nodos de la red en grupos según una regla de distribución. Por ejemplo, un nodo sometido a una restricción *ownConstraint* tiene el derecho de emitir una restricción para su vecindad en la trama siguiente si y solamente si su dirección MAC módulo *ownConstraint* es igual al número de la trama siguiente módulo *ownConstraint,* es decir mod(@ *MAC,ownConstraint*) = *mod*(*frameNumber,ownConstraint*).

[0052] Si no se trata de la última estimación y que el nodo considerado no tiene el derecho de emitir, se pasa directamente al final de la fase 24.

[0053] Si se trata de la última estimación, en una etapa 56 se memoriza en una variable *nbNeighborMeasurement* la suma de los números de nodos emisores, vecinos o interferentes estimados en cada una de las Q tramas es decir la suma de las *nbNeighborPerFrame.* En una etapa siguiente 58, se efectúa un alisado de la estimación del número total de nodos vecinos *nbNeighborEstim* ponderando el valor estimado en la trama actual por el estimado en la precedente según la relación:

$$nbNeighborEstim = \alpha * nbNeighborEstim + (1 - \alpha) * nbNeighborMeasurement \quad (2)$$

en la que *nbNeighborEstim* es una variable inicializada al inicio de la ejecución del algoritmo a cero o en otro valor si el número de vecinos total en la red ad hoc es conocido al inicio, y α un coeficiente de ponderación por ejemplo igual a 0,8.

[0054] El alisado de la estimación de la vecindad permite evitar cambios de restricción en cada error.

[0055] Durante esta misma etapa 58, el nodo calcula la nueva restricción *neighborConstraint* que se impondrá a los nodos vecinos.

[0056] El valor de la restricción *neighborConstraint* que se impondrá a los nodos vecinos en función de la vecindad estimada se establece del modo siguiente.

[0057] Sea n el número de intervalos de tiempo en una trama. Sea N el número de nodos emisores en la vecindad a un salto del nodo en escucha (vecinos más interferentes).

[0058] En el ejemplo propuesto, la hipótesis es que todos los nodos de la vecindad del nodo en escucha deben transmitir un mensaje a cada trama. La probabilidad de que el nodo en escucha reciba, sin que exista colisión, el mensaje de uno de sus vecinos en el curso de una trama se escribe como

$$\overline{p}_{colisión}(S) = \left(1 - 1/n\right)^{N-1} \quad (3)$$

[0059] Se fija una probabilidad P_{Umbral} por ejemplo igual a 0,8 que corresponde a la probabilidad mínima de que los vecinos del nodo en escucha tengan que hacerse oír por este nodo en escucha.

[0060] El número máximo de vecinos e interferentes que emiten en una misma trama y que pueden rodear al nodo en escucha para no sobrepasar P_{Umbral} viene dado por la relación:

$$M = 1 + \frac{\ln\left(p_{umbral}\right)}{\ln\left(1 - \frac{1}{n}\right)} \quad (4)$$

[0061] En el caso en que el número de nodos vecinos e interferentes, denotado P, estimado por el nodo en escucha es superior a M, mientras que los nodos vecinos deben dividirse en Q grupos emitiendo una vez todas las Q tramas, con:

$$Q = \left[P / M\right] + 1 \quad (5)$$

en la que $[x]$ representa la parte entera de x.

[0062] El valor de P es igual a *nbNeighborMeasurement.*

[0063] Q se denomina restricción y el valor de *neighborConstraint* se toma como igual a Q.

[0064] El quinto grupo (con $1 < q < Q$) está formado por nodos vecinos cuyas direcciones MAC módulo Q son iguales a q-1 y emiten a cada trama cuyo número módulo Q es igual a q-1. La estimación del número de nodos vecinos e interferentes debe así hacerse en todas las Q tramas, sumando las vecindades estimadas durante cada una de las Q tramas.

[0065] En el caso en que el número de nodos vecinos e interferentes estimado por el nodo en escucha es inferior a M, entonces los nodos vecinos pueden seguir emitiendo a cada trama. Entonces se fija $Q=1$.

[0066] En la figura 7, una red ad hoc está compuesta por ocho nodos numerados de 1 a 8. Un nodo 8 está en escucha de los mensajes enviados por los nodos en su vecindad o por los interferentes. En esta figura, una trama

160 se divide en cuatro intervalos de tiempo 161, 162, 163 y 164. Se fija k=2 y p_{Umbral} = 0,70 . Aplicando la fórmula de la ecuación (4), se obtiene M=2,23. El nodo 8, al no conocer el número de nodos vecinos que lo rodean, impone la restricción Q=1 a sus nodos vecinos. Siendo estos últimos 7, tienen una probabilidad de identificación de 0,18, de acuerdo con la ecuación (3). Se supone que los nodos vecinos del nodo 8 han emitido mensajes como los representados en la tabla 160. Aplicando la fórmula de la ecuación (1), sabiendo que hay dos intervalos de tiempo legibles, dos intervalos de tiempo en colisión y que k=2, el nodo 8 estima seis vecinos.

[0067] El nodo 8 calcula la restricción que enviará a sus vecinos de acuerdo con la ecuación (5), que vale Q=3. Suponiendo que el número de la primera trama en el curso de la cual los nodos deben someterse a la restricción impuesta por el nodo 8 es 1, los nodos 1, 4 y 7 podrán emitir durante la trama 1, los nodos 2 y 5 emitirán durante la segunda trama y los nodos 3 y 6 emitirán en el curso de la tercera trama.

[0068] En la figura 8, sólo los nodos 1, 4 y 7 emiten durante la trama 1, representada por la tabla 170. La probabilidad de que no haya ninguna colisión es entonces de 0,56, de acuerdo con la ecuación (3).

[0069] En la figura 9, sólo los nodos 2 y 5 emiten durante la trama 2, representada por la tabla 180. La probabilidad de que no haya ninguna colisión es entonces de 0,75, de acuerdo con la ecuación (3).

[0070] En la figura 10, sólo los nodos 3 y 6 emiten durante la trama 3, representada por la tabla 190. La probabilidad de que no haya ninguna colisión es entonces de 0,75, de acuerdo con la ecuación (3). Siendo la restricción impuesta por el nodo 8 igual a 3, la siguiente estimación del número de vecinos se realiza al final de la trama 3 en la que estime: $nbNeighbourMeasurement = \underbrace{1+k*1}_{Trama1} + \underbrace{2+k*0}_{Trama2} + \underbrace{2+k*0}_{Trama3} = 7$ vecinos. La nueva restricción *neighborConstraint* que se enviará a los nodos vecinos se calcula gracias a la ecuación (5), vale Q=4. A continuación se sigue con el algoritmo con esta nueva restricción. En una etapa 60, el nodo determina si tiene el derecho de emitir la nueva restricción en la trama siguiente de acuerdo con el mismo tratamiento que la etapa 53.

[0071] Si el nodo considerado tiene el derecho de emitir en la trama siguiente, se pasa a una etapa 64, que consiste en calcular el número de la trama al final de la cual el nodo considerado debe determinar la siguiente restricción *neighbor-Constraint* que impondrá a sus vecinos: es la trama siguiente (*frameNumber* + 1).

[0072] En una etapa 66 el nodo considerado elige de manera aleatoria un intervalo de tiempo para emitir en la siguiente trama las informaciones útiles así como las nuevas restricciones *msgConstraint* (que toma el valor de *neighborConstraint*) y *ownConstraint.*

[0073] Si el nodo considerado no tiene el derecho de emitir en la trama siguiente, en una etapa 68, el nodo considerado calcula el número de la trama al final de la cual debe determinar la siguiente restricción *neighborConstraint* que impondrá a sus vecinos. Existen dos casos: (1) si la restricción acaba de ser transmitida mientras el siguiente cálculo de *neighbor-Constraint* se realizará Q tramas más tarde: *frameNumber + Q;* (2) en caso contrario, el siguiente cálculo de *neighborConstraint* se realizará en la trama siguiente: *frameNumber* + 1.

[0074] La fase 24 se consigue así mediante la etapa 66 ó 68.

[0075] En la fase 26 se determina la restricción a la que debe someterse el nodo considerado. El mecanismo de actualización de *ownConstraintTemp* descrito en la etapa 42 asegura la aplicación del principio de base según el cual un nodo que recibe varias restricciones de sus vecinos se somete a la más intensa, lo que se traduce en retener la frecuencia de emisión más baja.

[0076] En la figura 11, una red ad hoc está compuesta por ocho nodos numerados de 1 a 8. Se supone que los nodos vecinos 2 y 3 del nodo 1 han emitido restricciones en el curso de una trama tal como la representada en la tabla 191. Según este principio de base, el nodo 1 se someterá a la restricción emitida por el nodo 3, es decir Q=4.

[0077] El vecino del nodo considerado que envía la restricción más intensa puede ser sometido él mismo a una restricción por uno de sus vecinos. Así pues, no podrá enviar su restricción a cada trama. Los nodos que se someten a su restricción no deben someterse a restricciones más débiles cuando no reciben su restricción.

[0078] En la figura 12, una red ad hoc está compuesta por diez nodos numerados de 1 a 10, estando para cada nodo las restricciones impuestas a los vecinos. Una tabla 192 representa las restricciones recibidas por el nodo 1 en el curso de siete tramas consecutivas, numeradas de 15 a 21. En este ejemplo, el nodo 3 envía la restricción

Q=4 en el curso de la trama 15. Estando él mismo sometido a la restricción q=3, sólo podrá emitir a las tramas 18 y 21. De acuerdo con el principio de base, en el curso de las tramas 16 a 21 el nodo 1 debe someterse a la restricción q=4. Si no se respeta este principio de base, el número de vecinos será subestimado por el nodo 3, emisor de la restricción máxima, y el algoritmo divergirá con una estimación del número de vecinos infinita.

[0079] La fase 26 comienza así por una etapa 69 en el curso de la cual el nodo considerado determina si *ownConstraintemp* es superior a la restricción a la que se somete *ownConstraint.*

[0080] Si así sucediera, se pasa a una etapa 70, en el curso de la cual el nodo determina su propia restricción que afecta al valor *ownConstraintTemp* a la variable *ownConstraint* y el valor *nodeToWaitTemp* a la variable *nodeToWait.* Simultáneamente, se reinicializa el contador *timeToLive* a un cierto valor, por ejemplo el doble de la restricción a la que se somete el nodo considerado. Este contador permite definir si un nodo no ha emitido desde hace tiempo.

[0081] Si, en la etapa 69, la variable *ownConstraintemp* no es superior a la restricción a la que se somete o*wnConstraint,* se compara la variable *changeConstraint* con “verdadero” en el curso de una etapa 72.

[0082] Si es igual a “verdadero”, caso en el que el nodo emisor de la restricción a la que se somete actualmente el nodo considerado ha emitido una restricción más débil, se pasa a la etapa 70. En caso contrario se pasa a una etapa 74 en el curso de la que se examina el valor de *timeToLive*. Si *timeToLive* alcanza el valor 0, entonces se considera que el nodo no está ya probablemente en la vecindad del nodo considerado. En la etapa 74 se comprueba si el contador *timeToLive* del nodo que ha emitido la restricción a la que se somete el nodo considerado es nulo o no. Si *timeToLive* vale cero entonces se pasa a la etapa 70, en caso contrario se pasa a una etapa 76 en el curso de la cual se decrementa el contador *TimeToLive* en una unidad.

[0083] Las etapas 69, 72 y 74 evitan la divergencia del algoritmo en la que se impone al nodo considerado que se someta a la restricción máxima que ha recibido en el curso de una trama si y solamente si esta restricción máxima es superior a la restricción a la que se somete o el nodo al que se somete ha reemitido una restricción más débil o el nodo al que se somete no está ya probablemente en su vecindad.

[0084] En las figuras 13 a 15 se explicitan los desfases temporales entre la transmisión y la aplicación de las restricciones respectivamente por un nodo emisor y por nodos receptores.

[0085] En la figura 13, un nodo calcula al final de una trama t la restricción Q que se impondrá a su vecindad. Suponiendo que la restricción q a la que el nodo considerado se somete lo permita, transmite la restricción Q a los nodos vecinos en la trama siguiente t+1. Los nodos vecinos que reciben esta restricción se someterán a ella a partir de la trama t+2.

[0086] En la figura 14, un nodo calcula al final de una trama t la restricción Q que se impondrá a su vecindad. En el ejemplo de la figura 14, la restricción vale tres. Suponiendo que la restricción q a la que el nodo considerado se somete lo permita, este último emite la restricción a la trama t+1, los nodos vecinos se someten a la trama t+2. El nodo considerado comienza sus estimaciones de la vecindad a partir del final de la trama t+2. El siguiente cálculo de la restricción se efectúa al final de la trama t+2+Q-1, ya sea al final de la trama t+4 en el ejemplo de la figura 14, a partir de las estimaciones de la vecindad efectuadas en las tramas t+2, t+3, ..., t+2+Q-1, o en las tramas t+2, t+3 y t+4 en el ejemplo de la figura 14.

[0087] En la figura 15, un nodo 2 calcula al final de una trama 10 la restricción Q que se impondrá a su vecindad. En el ejemplo de la figura 15, la restricción vale tres. El nodo considerado es sometido a una restricción, que en este ejemplo le impide emitir la restricción a la trama siguiente 11. Existe un informe de emisión de la restricción que se impondrá a los nodos vecinos. Sea v la trama a la que el nodo en escucha tiene el derecho de emitir la restricción calculada anteriormente. Entonces el nodo en escucha debe informar de su próximo cálculo de restricción a la trama v+Q a partir de las estimaciones realizadas al final de las tramas v+1, v+2, ..., v+Q. En el ejemplo de la figura 15, v tiene como valor 12, el nodo en escucha realiza el cálculo de la nueva restricción al final de la trama 15, a partir de las estimaciones realizadas al final de las tramas 13, 14 y 15. En este ejemplo de aplicación del algoritmo, no se realiza ninguna estimación de la vecindad al final de la trama de transmisión de la restricción, sea al final de la trama 12, y al final de la trama de informe de emisión de la restricción, por ejemplo al final de la trama 11.

[0088] En la figura 16, una tabla 200 representa una sucesión de tramas numeradas de 14 a 22, y tres nodos N1, N3 y N7. El nodo N1 impone a sus vecinos una restricción Q=2 y se somete él mismo a una restricción q=4. El

nodo N3 impone a sus vecinos una restricción Q=4 y se somete él mismo a una restricción q=3. El nodo N7 impone a sus vecinos una restricción Q=3 y se somete él mismo a una restricción q=4. Este ejemplo ilustra la manera en que el algoritmo del procedimiento obtiene parte de los slots del informe de transmisión para estimar su vecindad y calcular la restricción. Después de un informe de transmisión, la restricción Q calculada por un nodo denominado S no es enviada a sus vecinos. Los vecinos de S siguen así usando la misma restricción que antes. Los nodos que transmiten un mensaje en el curso de una trama de informe de transmisión T son, salvo error debido por ejemplo a una colisión, los mismos que los que transmiten en la trama T-Q. El algoritmo del procedimiento usa la estimación del número de nodos vecinos e interferentes que transmiten en el curso de la trama T para recalcular la vecindad del nodo S. Los diferentes cálculos de vecindad se consideran de acuerdo con la relación (2). En el ejemplo de la figura 16, el nodo N1 calcula al final de la trama 15 su vecindad total V(1) El nodo N1 debe esperar a la trama 17 para poder transmitir la nueva restricción. En final de trama 16, denominada trama de informe de transmisión, tiene lugar un segundo cálculo de la vecindad V'(1), sobre la base de las estimaciones efectuadas en la trama 15 y 16 y de V(1). V'(1) se usa para el cálculo de la nueva restricción que se transmitirá en la trama 17.

[0089] La figura 17 ilustra uno de los beneficios del alisado de la estimación del número de vecinos e interferentes en el curso de las slots de transmisión. En esta figura, una tabla 230 representa una sucesión de seis tramas numeradas de 16 a 21, y un nodo Nx tal que mod(x,4)=mod(15,4). El nodo Nx impone una restricción Q=3 a sus vecinos y él mismo se somete a una restricción q=4. Se supone que en el curso de la trama 16 comete un error al estimar que tiene un vecino en lugar de tres, que en el curso de la trama 17 estima que tiene dos vecinos, y que en el curso de la trama 18 estima que tiene dos vecinos. Se supone que no comete error en las estimaciones del número de vecinos que ha emitido en las tramas 17 y 18. al final de la trama 18 estima su número de vecinos total en cinco en lugar de siete, lo que le conduce a imponer una restricción Q=2 en lugar de Q=3 a sus vecinos en la trama 19, llamada trama de transmisión. En el ejemplo de la figura 17, el alisado de la estimación se implementa en el algoritmo del procedimiento según la invención. La restricción calculada en la trama 19 no se tendrá en cuenta ya que la restricción calculada en la trama 18 ya ha sido enviada. Por el contrario, el número de vecinos calculados en la trama 19 se tiene en cuenta en el cálculo de la vecindad de las tramas siguientes gracias al alisado de la estimación. Así no hay pérdida de informaciones relacionada con la transmisión de una restricción en una trama. En este ejemplo, un alisado de la estimación y un uso del conjunto de las tramas para la estimación son implementados por el algoritmo, con lo que se mejora la estimación del número de vecinos por un nodo.

[0090] A causa de las colisiones, un nodo no recibe forzosamente la restricción máxima a la que debe someterse lo que puede conllevar una fuerte variación del número de vecinos estimado.

[0091] En una primera solución, un nodo envía la restricción que impone a sus vecinos así como la restricción a la que se somete él mismo, de tal manera que sus vecinos pueden saber si su restricción ha sido recibida.

[0092] En la figura 18, una red ad hoc está compuesta por nueve nodos numerados de 1 a 9. Se supone que los nodos emiten mensajes en tramas que incluyen cuatro intervalos de tiempo indicados como "slot" en la figura. En el curso de una trama 11, se supone que todos los nodos se someten a una restricción q = 1. Al final de la trama 11, cada nodo de la red calcula la restricción Q, que va a imponer a sus vecinos. Estas restricciones son emitidas en la trama 12 y representadas en la figura 18. Se usarán a partir de la trama 13. Una tabla 193 representa los mensajes recibidos por el nodo 1 en el curso de la trama 12. En los mensajes enviados por un nodo, q designa la restricción a la que se somete y Q la restricción que impone a sus vecinos. En el curso del intervalo de tiempo 194, existe colisión entre los mensajes enviados por los nodos 4 y 5. El nodo 1 se somete así a la restricción máxima que recibe, es decir Q=2, emitida por el nodo 3. En el curso de la trama 13, se interesa en los mensajes recibidos por el nodo 5, nodo que ha emitido la restricción más fuerte en el curso de la trama precedente. El nodo 1 transmite a la vez la restricción que impone a sus vecinos y la restricción a la que se somete. Así, el nodo 5 sabe que el nodo 1 se somete a una restricción más débil que aquella que ha impuesto. El nodo 5 es así capaz de detectar el error. Esta solución presenta la ventaja de que un nodo puede saber en algunos casos que su estimación de vecinos será deficiente y no actualizará entonces esta estimación.

[0093] En una segunda solución, un nodo conoce la identidad del emisor del mensaje que recibe.

[0094] En la figura 19 se considera la misma red ad hoc que la de la figura 18. Tiene lugar la misma colisión en el curso de la trama 12. El nodo 1 se somete así a una restricción q=2. El nodo 5 que ha enviado la restricción Q=3 en la trama 12, estima su vecindad en las tramas 13 a 15. La tabla 195 representa los mensajes recibidos por el nodo 5 en el curso de estas tres tramas 13, 14 y 15. En la trama 15, el nodo 1 emite un segundo mensaje en el periodo de estimación de la vecindad del nodo 5. Por comparación de los identificadores de los nodos emisores, el nodo 5 detecta este doble envío y no lo tiene en cuenta.

[0095] Esta solución presenta la ventaja de que un nodo es capaz en algunos casos de detectar si un nodo vecino transmite más de un mensaje en el curso de un periodo de estimación, y de contar entonces una única vez este nodo vecino. Al contrario de la primera solución que sólo permite la detección de un error, esta solución permite, si se detecta el error, corregirlo. A diferencia de la primera solución presenta la ventaja de detectar el error a partir del momento en el que habiéndose equivocado uno de los mensajes del nodo es recibido correctamente mientras que la segunda solución sólo puede detectar el error si estando equivocados los dos mensajes del nodo son recibidos correctamente. Es posible implementar en el algoritmo del procedimiento según la invención una u otra de estas soluciones o bien combinar las dos para beneficiarse a la vez de un máximo de errores detectados y de la posibilidad de corregir algunos.

[0096] El procedimiento según la invención permite así reducir el número de colisiones sin ajustar el tamaño de las tramas, en la práctica puede usarse con una trama de tamaño fijo. El procedimiento no necesita ya el envío de mensajes suplementarios de señalización por ejemplo de mensajes baliza. Sólo se añaden dos campos suplementarios a los mensajes transmitidos por los nodos.

REIVINDICACIONES

1. Procedimiento de transmisión de informaciones en una red ad hoc que comprende al menos dos nodos emisores/receptores adecuados para recibir y emitir informaciones y que se comunican entre sí mediante el envío de informaciones en intervalos de tiempo en acceso aleatorio organizados en tramas,

en el que los nodos de la red están distribuidos en grupos según una regla de distribución, de manera que los nodos de cada grupo tienen el derecho de emitir informaciones solamente en un subconjunto de tramas predefinido específico del grupo,

caracterizado porque la regla de distribución de un nodo en un grupo tiene en cuenta el número de nodos emisores vecinos o interferentes que tienen los nodos vecinos del nodo considerado, y **porque:**

- cada nodo define (fase 24) una restricción *neighborConstraint* para los nodos vecinos,

- cada nodo transmite (fase 24) la restricción definida *neighborConstraint* a los nodos vecinos, y

- cada nodo (fase 26) define el grupo al que pertenece según la regla de distribución en función de las restricciones *neighborConstraint* recibidas de los nodos vecinos,

la etapa de definición de una restricción *neighborConstraint* por un nodo para los nodos vecinos que comprende una etapa (56) de estimación del número total de nodos emisores vecinos o interferentes y una etapa (58) de establecimiento de la restricción *neighborConstraint* en función del número total de nodos vecinos o interferentes estimado y de una probabilidad p_{Umbral} mínima objeto de que los vecinos del nodo puedan emitir informaciones sin colisión.

2. Procedimiento según la reivindicación 1, en el que el nodo define (fase 26) el grupo al que pertenece según la regla de distribución en función de la restricción *neighborConstraint* recibida de los nodos vecinos que conduce a la frecuencia de emisión más baja.

3. Procedimiento según la reivindicación 2, en el que la etapa de definición de una restricción *neighborConstraint* por un nodo para los nodos vecinos incluye además una etapa (50) de estimación del número de nodos vecinos e interferentes que emiten para una trama dada, y una etapa (58) de alisado del número total de nodos emisores vecinos o interferentes, siendo la restricción *neighborConstraint* función del número total de nodos vecinos o interferentes alisado y de la probabilidad p_{Umbral} mínima objeto de que los vecinos del nodo puedan emitir informaciones sin colisión.

4. Procedimiento según la reivindicación 3, en el que la etapa (50) de estimación del número de nodos vecinos o interferentes que emiten para una trama dada se basa en una medida del número de intervalos de tiempo en los que: no se ha detectado ninguna transmisión, se ha recibido una transmisión única y se han detectado más de dos transmisiones que generan una colisión.

5. Procedimiento según la reivindicación 2 ó 3, en el que la etapa (56) de estimación del número total de nodos emisores vecinos o interferentes incluye una suma (56) del número de nodos vecinos o interferentes que han emitido mensajes en cada una de las tramas que corresponden al conjunto de grupos de distribución de nodos según la restricción *neighborConstraint* definida por el nodo y transmitida a sus nodos vecinos.

6. Procedimiento según la reivindicación 5, en el que la etapa de cálculo del número total de nodos emisores vecinos o interferentes por un nodo incluye una ponderación (58) del número de nodos total de nodos emisores o interferentes actual calculado, por el número total de nodos emisores vecinos o interferentes calculado anteriormente para ese nodo.

7. Procedimiento según una cualquiera de las reivindicaciones precedentes, en el que cada nodo define el grupo al que pertenece según la regla de distribución sometiéndose a una restricción *ownConstraint* entre las restricciones de los nodos vecinos *neighborConstraint* y en el que el cálculo de una restricción *ownConstraint* específico de un nodo sólo se realiza si se aplica al menos una de las condiciones siguientes:

- no hay restricciones *neighborConstraint* más elevadas recibidas de un nodo vecino (69),

- el nodo vecino que ha definido la restricción *ownConstraint a* la que se somete el nodo considerado, ha transmitido una restricción menos elevada (72),

- la expiración de un periodo de tiempo predeterminado (*TimeToLive, 74).*

8. Procedimiento según una cualquiera de las reivindicaciones precedentes, en el que cada nodo define el grupo al que pertenece según la regla de distribución sometiéndose a una restricción *ownConstraint* entre las restricciones de los nodos vecinos *neighborConstraint* y en el que cada nodo transmite (fase 24) la restricción *ownConstraint a* la que se somete *msgConstraint.*

9. Procedimiento según la reivindicación 8, **caracterizado porque** un nodo que ha emitido una restricción *neighborConstraint* a una trama precedente determina si la restricción a la que se somete un nodo vecino *ownConstraint* y que ha recibido de este nodo vecino, es inferior o no a la restricción *neighborConstraint* que ha emitido y si la restricción *own-Constraint* a la que se somete el nodo vecino es inferior a la restricción *neighborConstraint* que ha emitido, mientras que el nodo no tiene en cuenta los elementos de estimación de su vecindad calculados durante la trama actual.

10. Procedimiento según la reivindicación 8 ó 9, **caracterizado porque** un nodo que impone a sus vecinos una restricción n*eighborConstraint* determina si un mismo nodo vecino emite al menos dos veces durante n tramas sucesivas, siendo n la restricción *neighborConstraint* impuesta al nodo, y si el nodo vecino emite al menos dos veces durante n tramas sucesivas, el nodo sólo tiene en cuenta una única vez el nodo vecino para la definición de su número de nodos emisores vecinos o interferentes.

11. Procedimiento según una cualquiera de las reivindicaciones precedentes, **caracterizado porque** las transmisiones entre los nodos carecen de mensajes dedicados a la señalización para reducir el número de colisión en la red.

12. Red ad hoc que incluye un conjunto de nodos emisores-receptores adecuados para recibir y emitir informaciones y que se comunican entre sí mediante el envío de informaciones en intervalos de tiempo en acceso aleatorio organizados en tramas, **caracterizado porque** cada nodo incluye medios para la implementación del procedimiento según una cualquiera de las reivindicaciones precedentes.

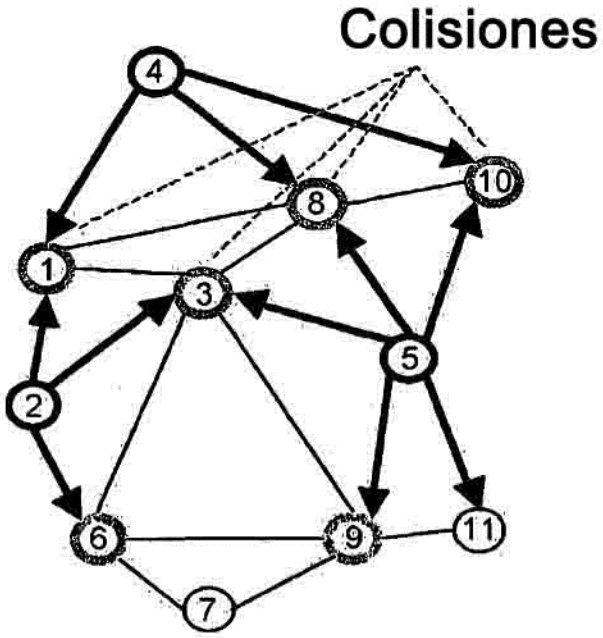


FIG.1

FIG.2 (inicio)

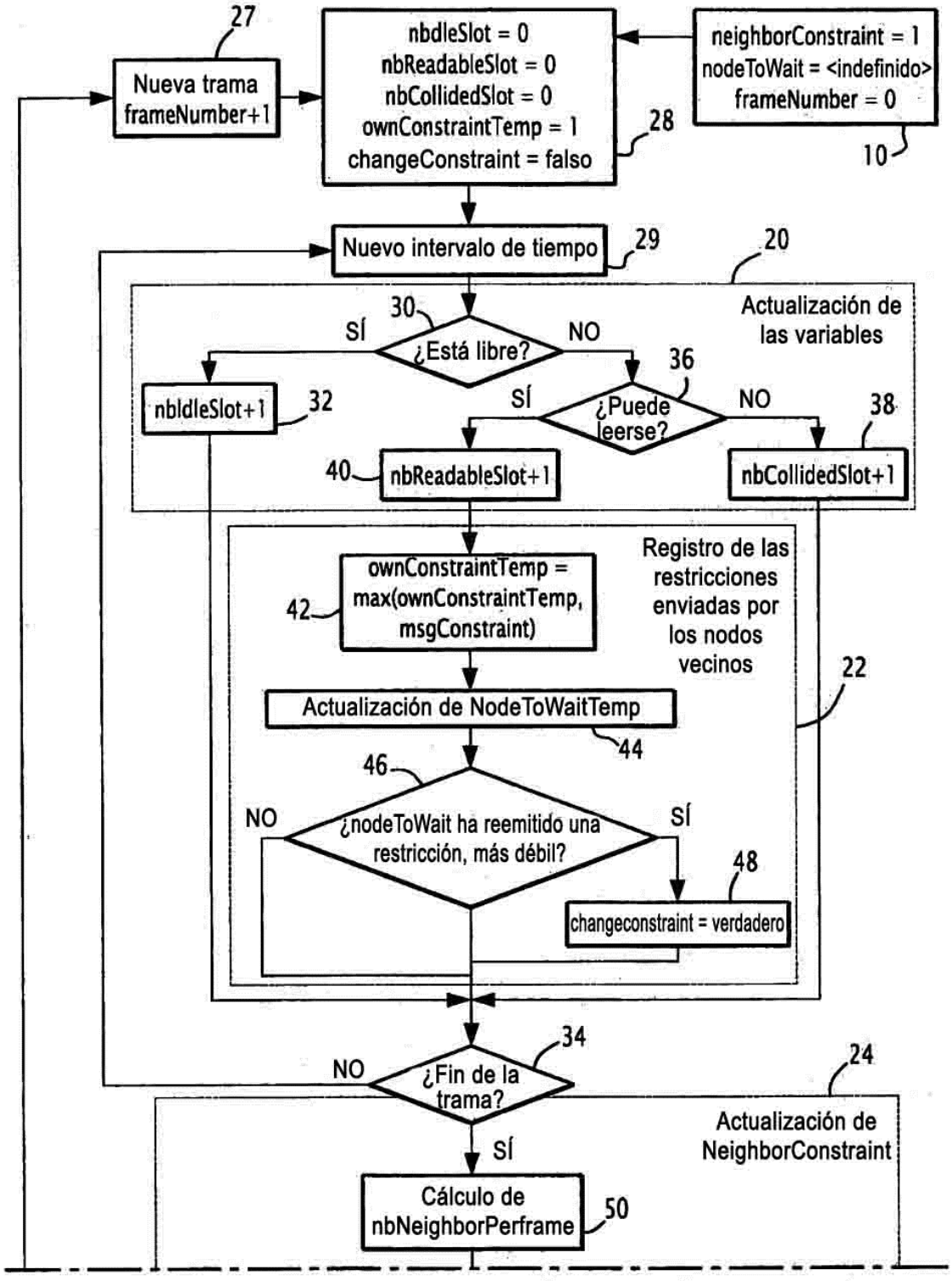
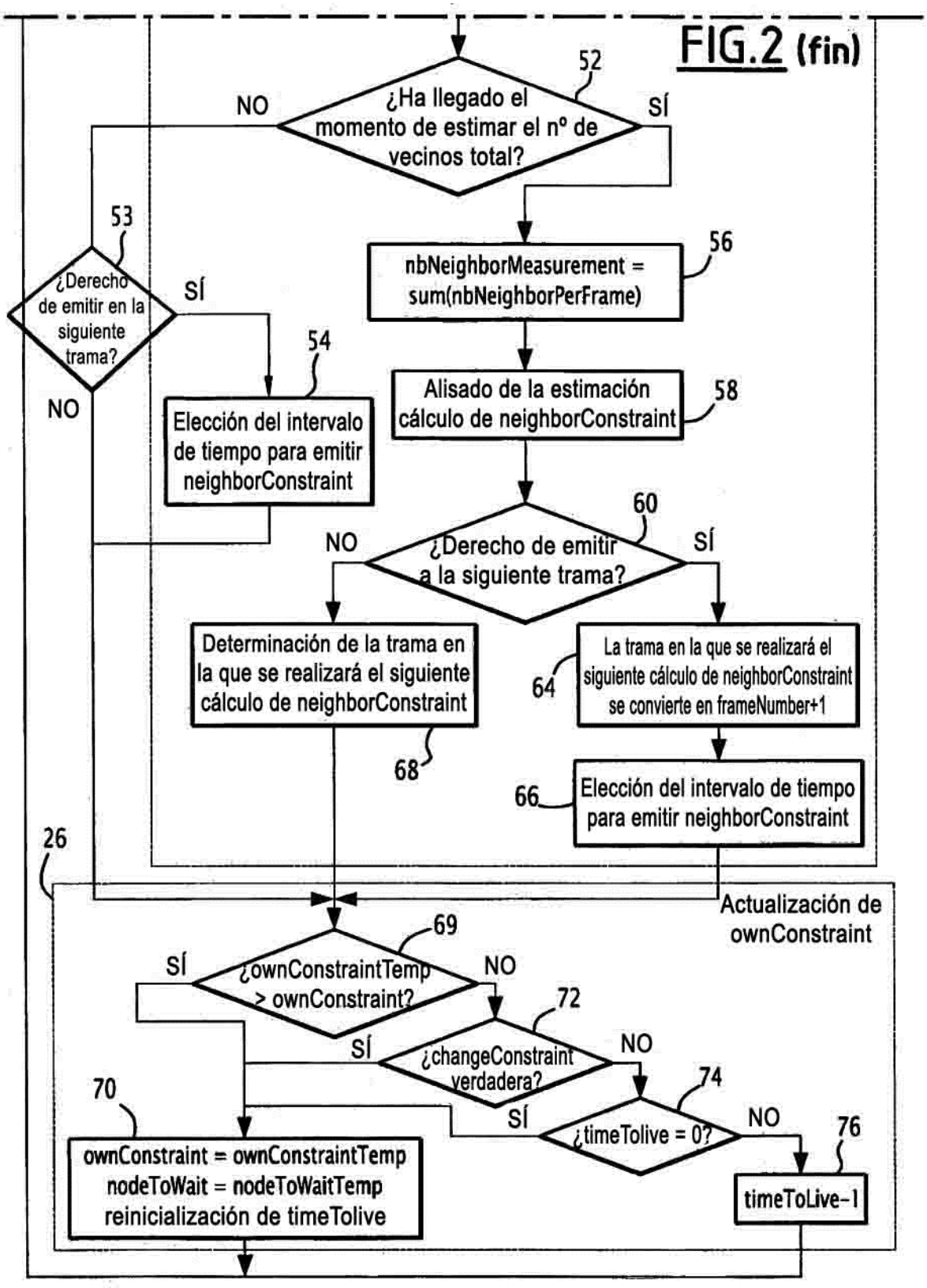

27
Nueva trama
frameNumber+1
nbdleSlot = 0
nbReadableSlot = 0
nbCollidedSlot = 0
ownConstraintTemp = 1
changeConstraint = falso
28
neighborConstraint = 1
nodeToWait = <indefinido>
frameNumber = 0
10
Nuevo intervalo de tiempo
29
20
Actualización de las variables
30
SÍ
¿Está libre?
NO
36
nbIdleSlot+1
32
SÍ
¿Puede leerse?
NO
38
40
nbReadableSlot+1
nbCollidedSlot+1
Registro de las restricciones enviadas por los nodos vecinos
ownConstraintTemp = max(ownConstraintTemp, msgConstraint)
42
22
Actualización de NodeToWaitTemp
44
46
NO
¿nodeToWait ha reemitido una restricción, más débil?
SÍ
48
changeconstraint = verdadero
34
NO
¿Fin de la trama?
24
SÍ
Actualización de NeighborConstraint
Cálculo de nbNeighborPerframe
50

FIG.2 (fin)


52
¿Ha llegado el momento de estimar el nº de vecinos total?
NO
SÍ
53
¿Derecho de emitir en la siguiente trama?
SÍ
NO
54
Elección del intervalo de tiempo para emitir neighborConstraint
56
nbNeighborMeasurement = sum(nbNeighborPerFrame)
58
Alisado de la estimación cálculo de neighborConstraint
60
¿Derecho de emitir a la siguiente trama?
NO
SÍ
Determinación de la trama en la que se realizará el siguiente cálculo de neighborConstraint
68
64
La trama en la que se realizará el siguiente cálculo de neighborConstraint se convierte en frameNumber+1
66
Elección del intervalo de tiempo para emitir neighborConstraint
26
Actualización de ownConstraint
69
¿ownConstraintTemp > ownConstraint?
SÍ
NO
72
¿changeConstraint verdadera?
SÍ
NO
74
¿timeTolive = 0?
SÍ
NO
70
ownConstraint = ownConstraintTemp
nodeToWait = nodeToWaitTemp
reinicialización de timeTolive
76
timeToLive-1

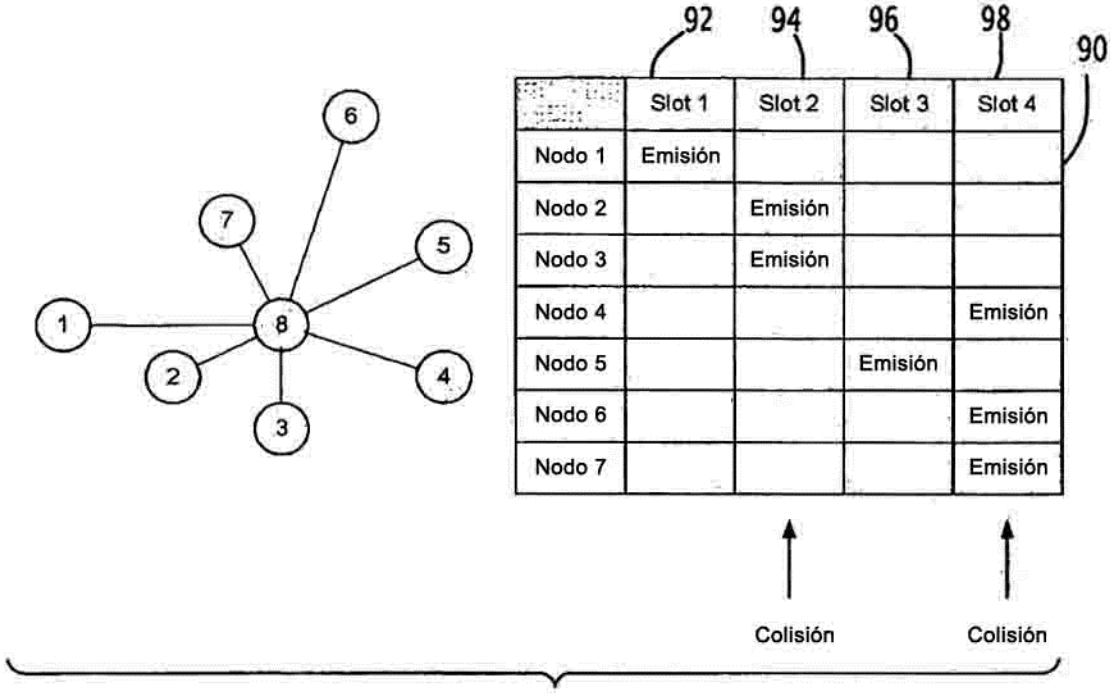

92
94
96
98
90
Slot 1
Slot 2
Slot 3
Slot 4
Nodo 1
Emisión
Nodo 2
Emisión
Nodo 3
Emisión
Nodo 4
Emisión
Nodo 5
Emisión
Nodo 6
Emisión
Nodo 7
Emisión
Colisión
Colisión
1
2
3
4
5
6
7
8

FIG.3

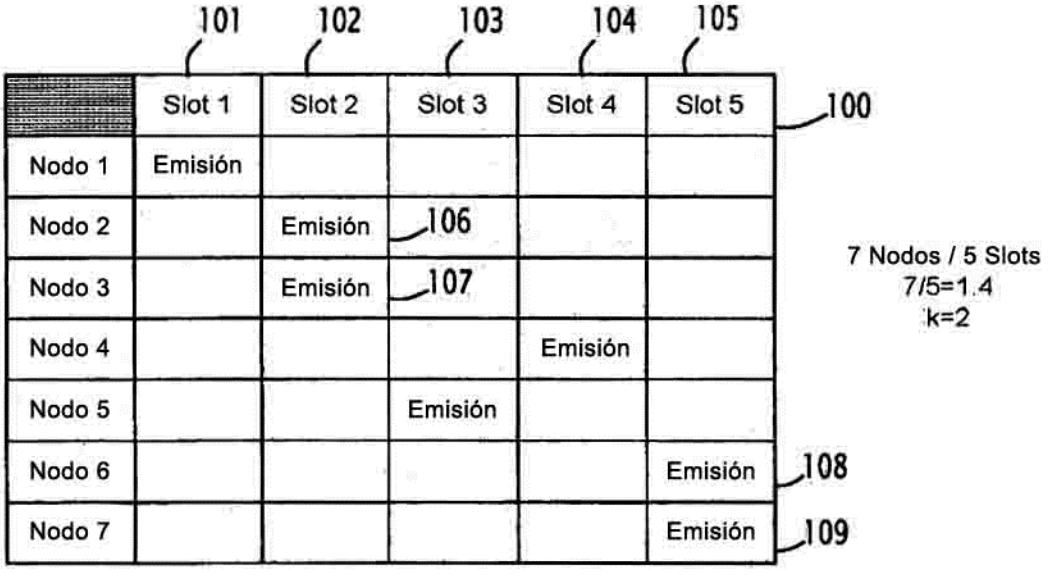

101
102
103
104
105
Slot 1
Slot 2
Slot 3
Slot 4
Slot 5
100
Nodo 1
Emisión
Nodo 2
Emisión
106
Nodo 3
Emisión
107
Nodo 4
Emisión
Nodo 5
Emisión
Nodo 6
Emisión
108
Nodo 7
Emisión
109
7 Nodos / 5 Slots
7/5=1.4
k=2

FIG.4

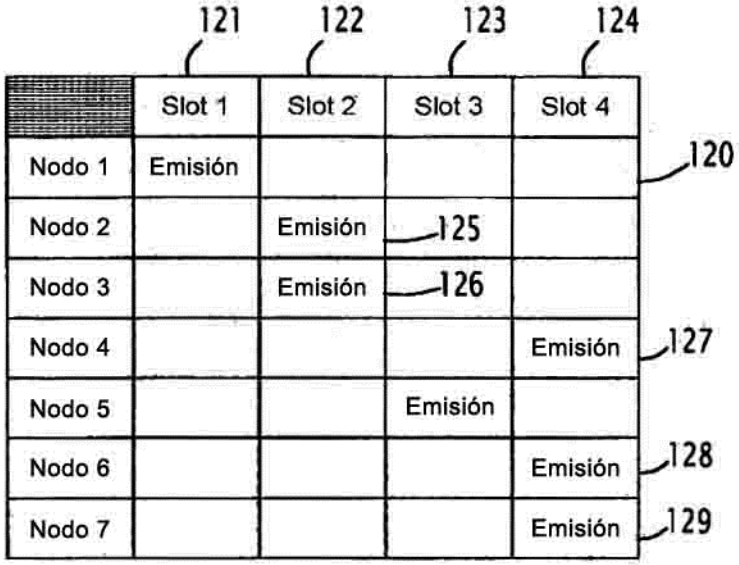

121
122
123
124
Slot 1
Slot 2
Slot 3
Slot 4
Nodo 1
Emisión
Nodo 2
Emisión
Nodo 3
Emisión
Nodo 4
Emisión
Nodo 5
Emisión
Nodo 6
Emisión
Nodo 7
Emisión
120
125
126
127
128
129
7 Nodos / 4 Slots
7/4 = 1.75
k=2.5

FIG.5

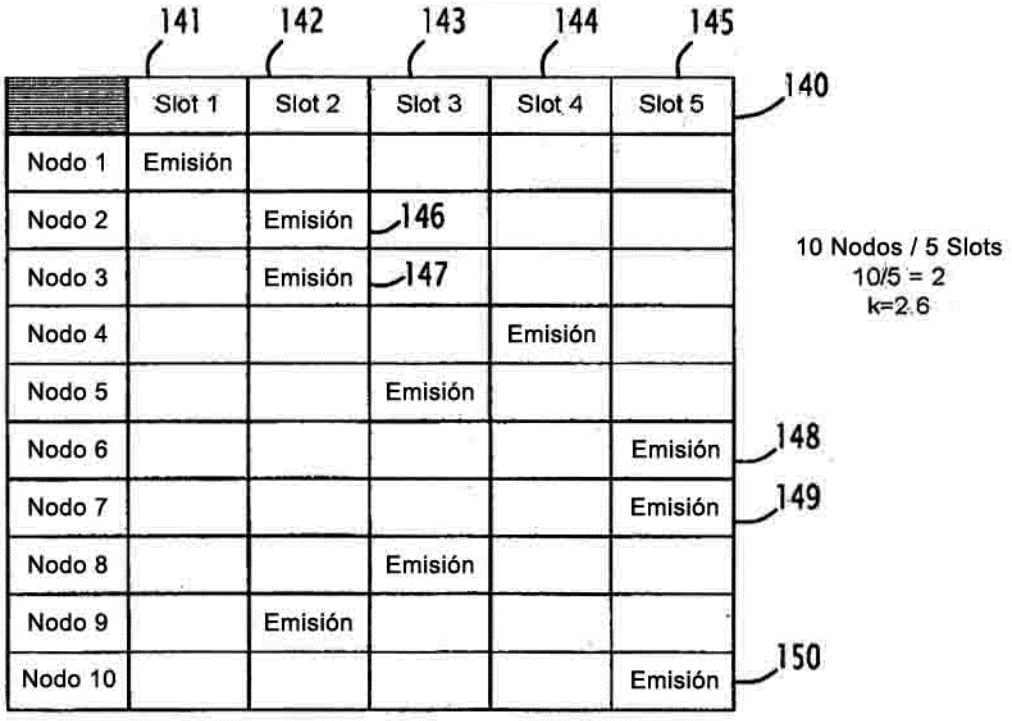

141
142
143
144
145
Slot 1
Slot 2
Slot 3
Slot 4
Slot 5
Nodo 1
Emisión
Nodo 2
Emisión
Nodo 3
Emisión
Nodo 4
Emisión
Nodo 5
Emisión
Nodo 6
Emisión
Nodo 7
Emisión
Nodo 8
Emisión
Nodo 9
Emisión
Nodo 10
Emisión
140
146
147
148
149
150
10 Nodos / 5 Slots
10/5 = 2
k=2.6

FIG.6

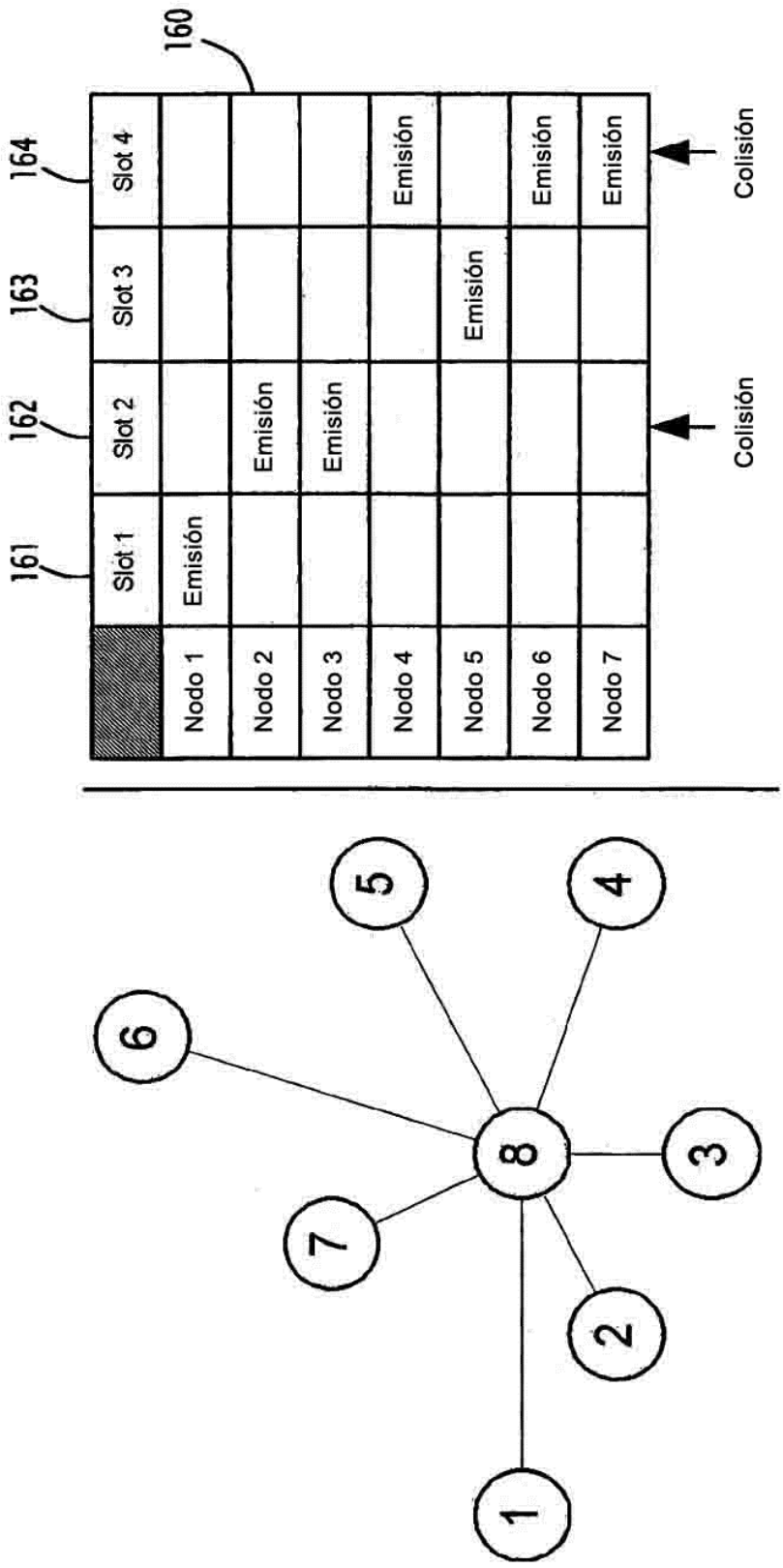

160
161
162
163
164
Slot 1
Slot 2
Slot 3
Slot 4
Nodo 1
Nodo 2
Nodo 3
Nodo 4
Nodo 5
Nodo 6
Nodo 7
Emisión
Colisión
Colisión
1
2
3
4
5
6
7
8

FIG.7

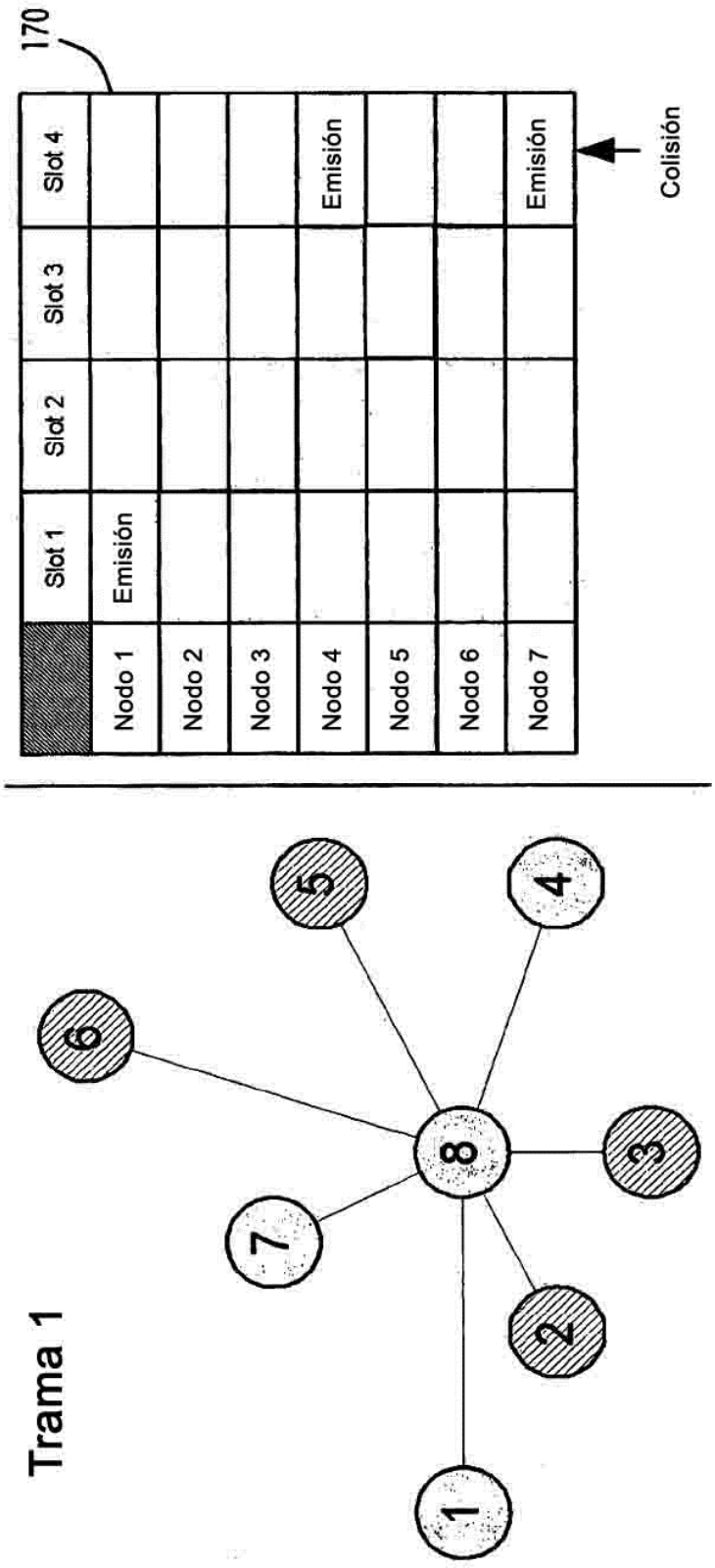

170
Slot 1
Slot 2
Slot 3
Slot 4
Nodo 1
Nodo 2
Nodo 3
Nodo 4
Nodo 5
Nodo 6
Nodo 7
Emisión
Emisión
Emisión
Colisión
Trama 1
1
2
3
4
5
6
7
8

FIG.8

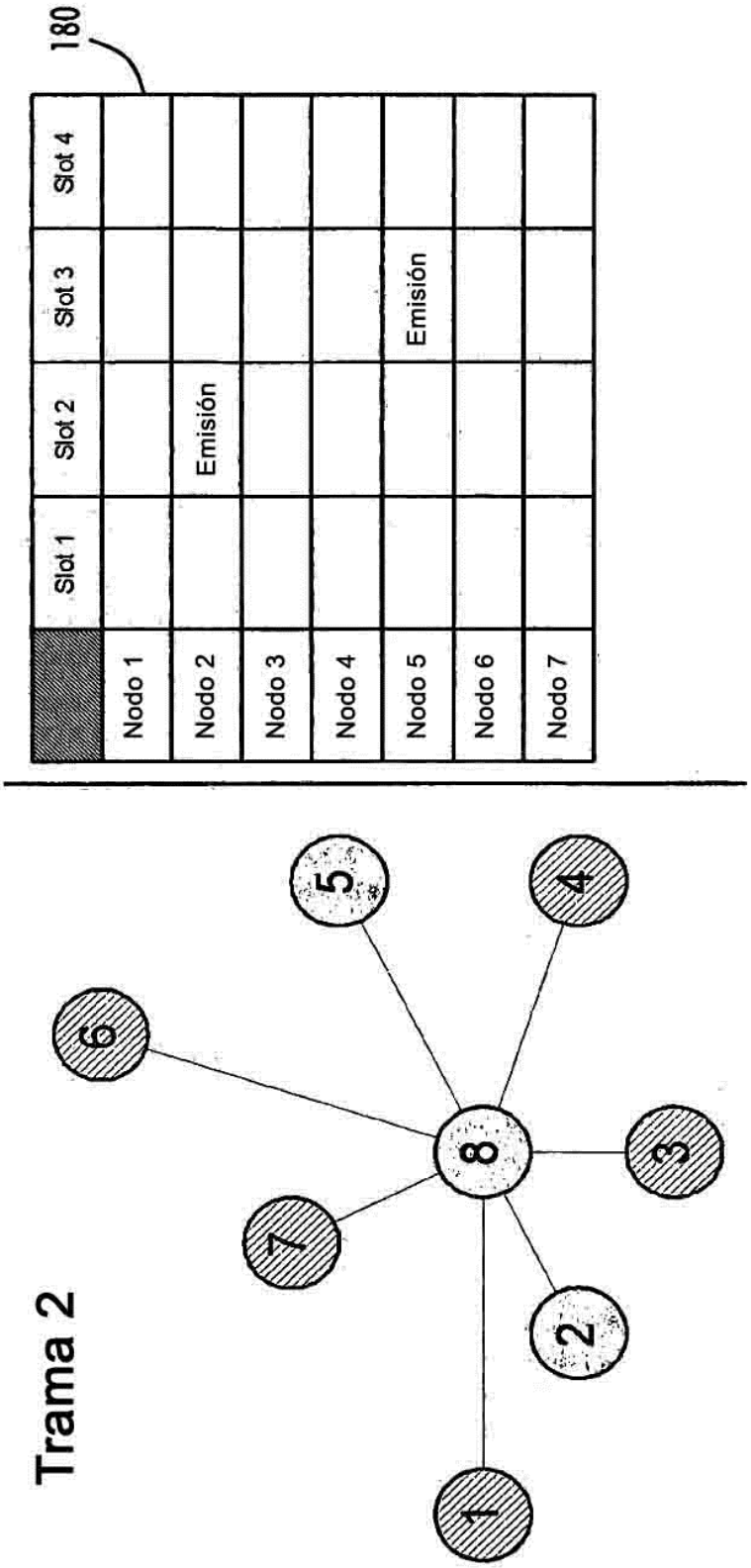

180
Slot 1
Slot 2
Slot 3
Slot 4
Nodo 1
Nodo 2
Nodo 3
Nodo 4
Nodo 5
Nodo 6
Nodo 7
Emisión
Emisión
Trama 2
1
2
3
4
5
6
7
8

FIG.9

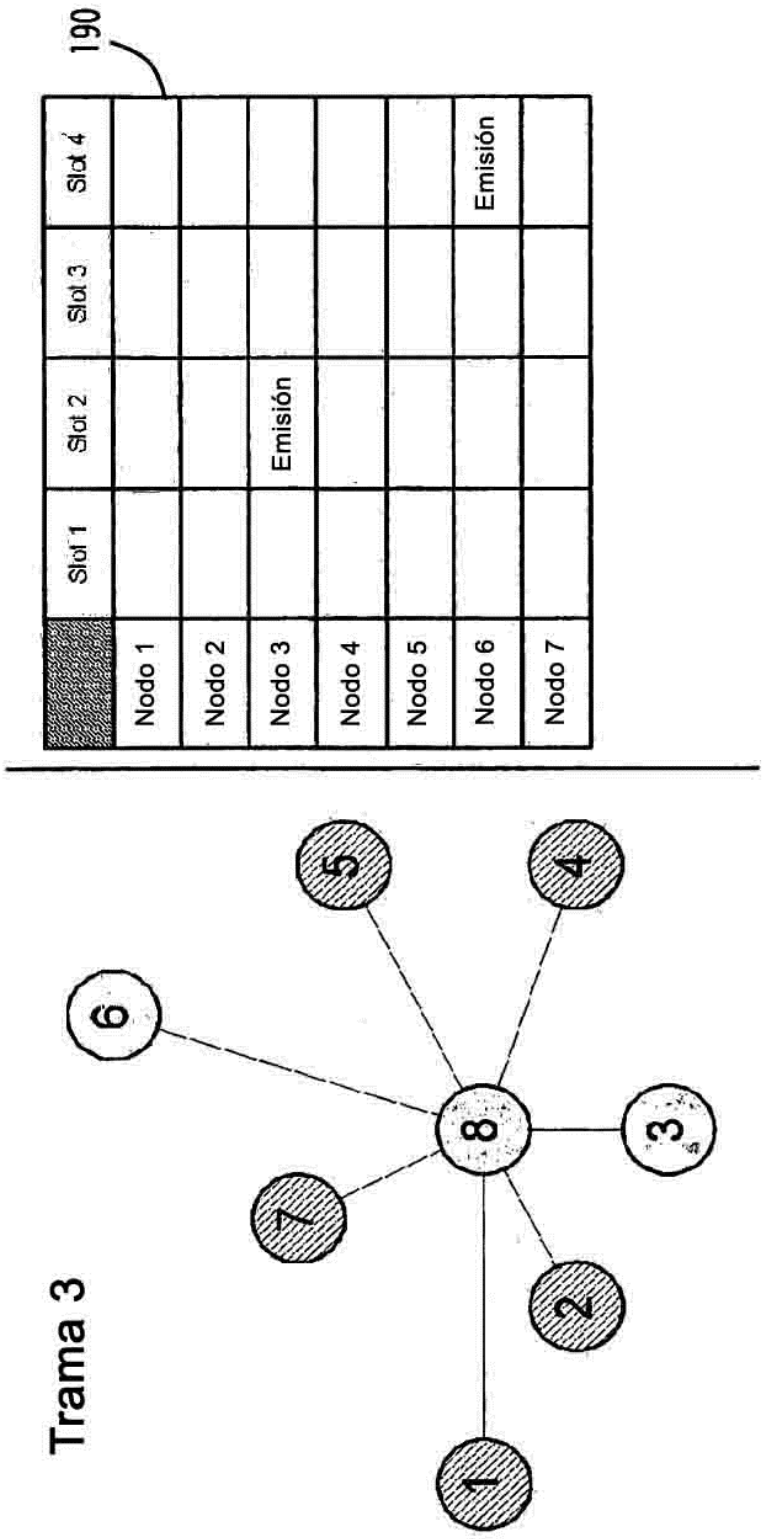

190
Slot 1
Slot 2
Slot 3
Slot 4
Nodo 1
Nodo 2
Nodo 3
Nodo 4
Nodo 5
Nodo 6
Nodo 7
Emisión
Emisión
Trama 3
1
2
3
4
5
6
7
8

FIG.10

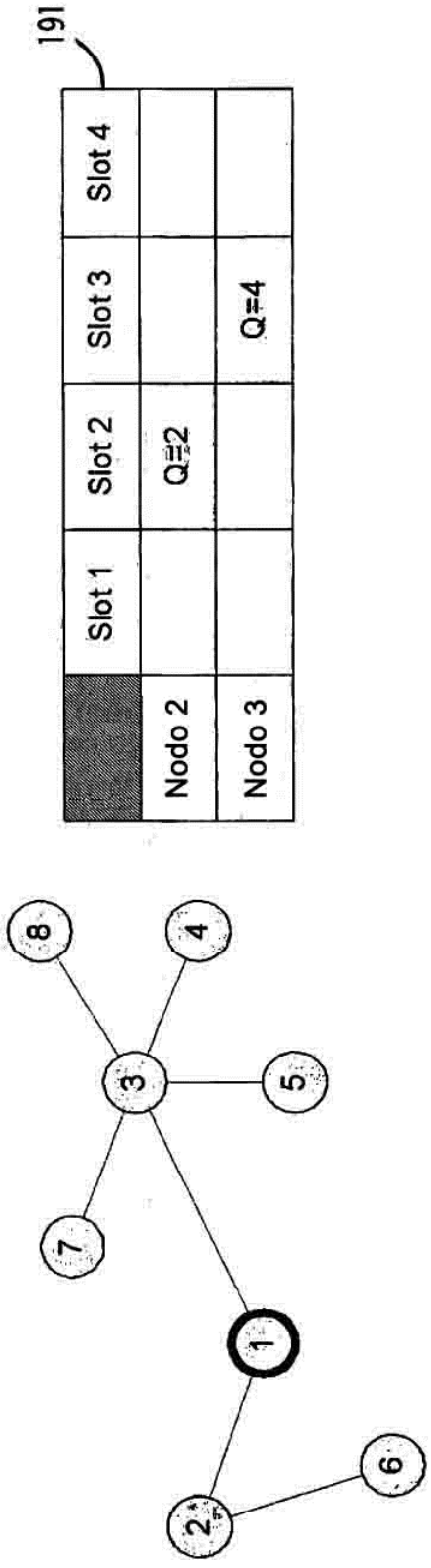


FIG.11

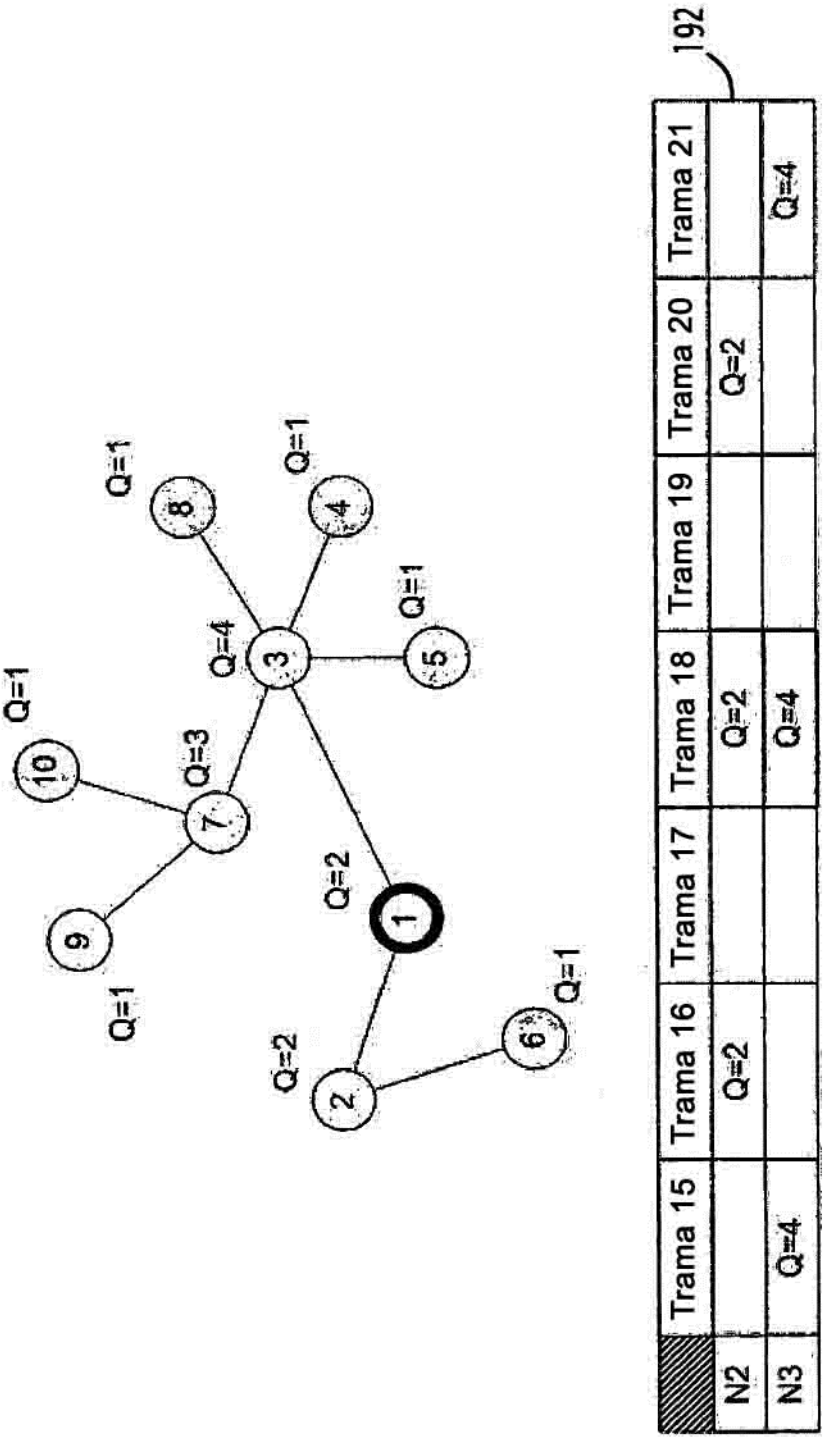

Q=1
8
Q=1
4
Q=4
3
Q=1
5
Q=1
10
Q=3
7
9
Q=1
Q=2
1
Q=2
2
6
Q=1
192
Trama 15
Trama 16
Trama 17
Trama 18
Trama 19
Trama 20
Trama 21
N2
Q=2
Q=2
Q=2
N3
Q=4
Q=4
Q=4

FIG.12

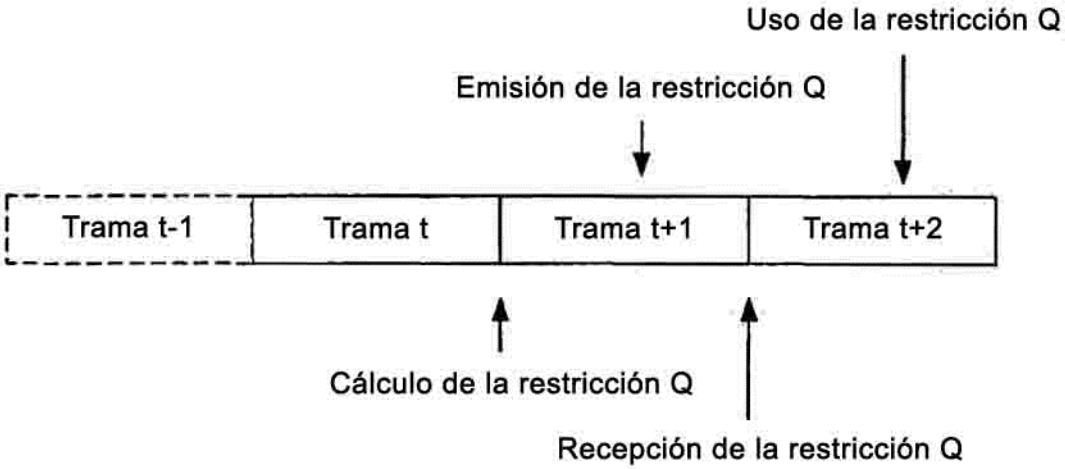

Uso de la restricción Q
Emisión de la restricción Q
Trama t-1
Trama t
Trama t+1
Trama t+2
Cálculo de la restricción Q
Recepción de la restricción Q

FIG.13

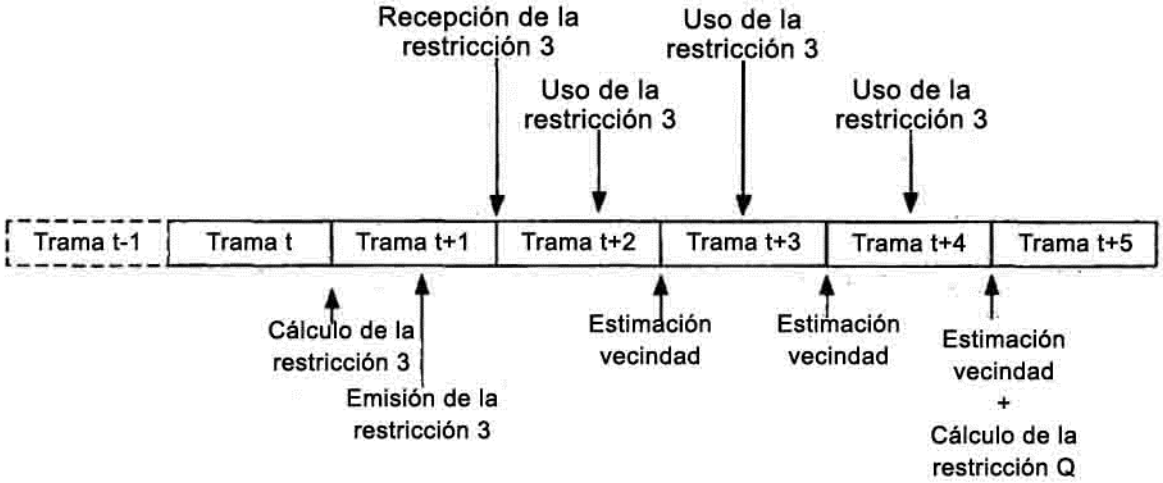

Recepción de la restricción 3
Uso de la restricción 3
Uso de la restricción 3
Uso de la restricción 3
Trama t-1
Trama t
Trama t+1
Trama t+2
Trama t+3
Trama t+4
Trama t+5
Cálculo de la restricción 3
Emisión de la restricción 3
Estimación vecindad
Estimación vecindad
Estimación vecindad + Cálculo de la restricción Q

FIG.14

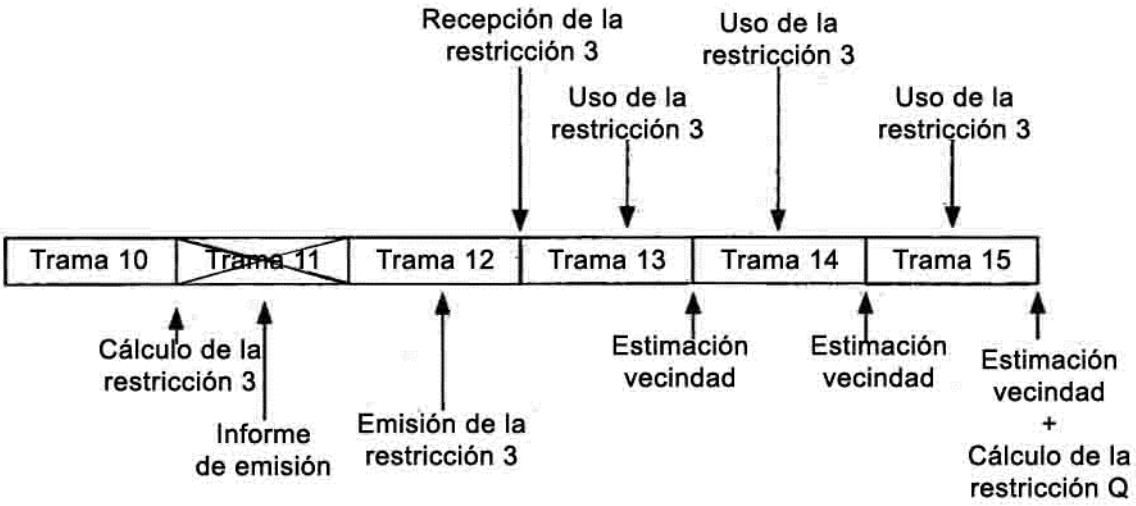


FIG.15

200

	Trama 14	Trama 15	Trama 16	Trama 17	Trama 18	Trama 19	Trama 20	Trama 21	Trama 22
N1 Q=2 q=4	E	E V(1)	E V'(1)	Tx	E	E V(1)	E V'(1)	Tx	E
N3 Q=4 q=3	V'(3)	Tx	E	E	E	E V(3)	E V'(3)	Tx	E
N7 Q=3 q=4	E V(7)	Tx	E	E	E V(7)	Tx	E	E	E V(7)

E — Estimación

E V(x) — Estimación + Cálculo de la vecindad del nodo x

Tx — Transmisión de una nueva restricción (después de actualización)

FIG.16

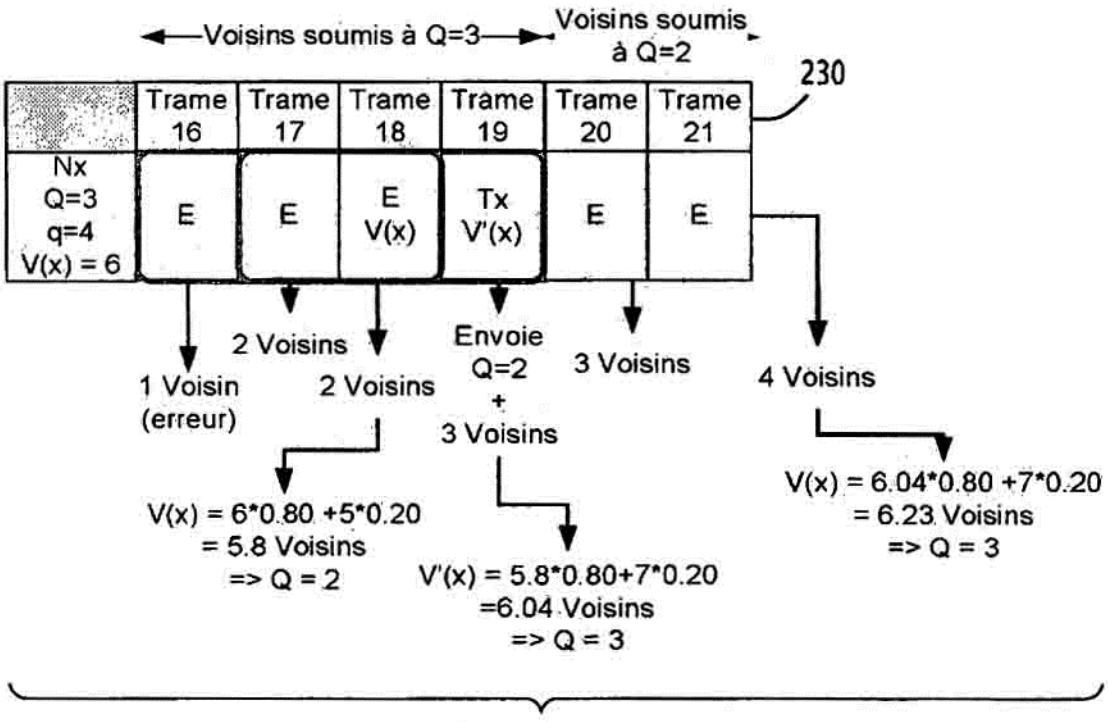
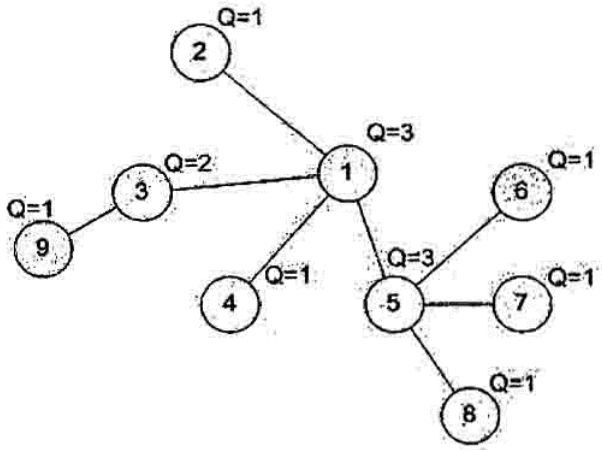

Voisins soumis à Q=3
Voisins soumis à Q=2
230
Trame 16
Trame 17
Trame 18
Trame 19
Trame 20
Trame 21
Nx
Q=3
q=4
V(x) = 6
E
E
E
V(x)
Tx
V'(x)
E
E
1 Voisin (erreur)
2 Voisins
2 Voisins
Envoie Q=2 + 3 Voisins
3 Voisins
4 Voisins
V(x) = 6*0.80 +5*0.20
= 5.8 Voisins
=> Q = 2
V'(x) = 5.8*0.80+7*0.20
=6.04 Voisins
=> Q = 3
V(x) = 6.04*0.80 +7*0.20
= 6.23 Voisins
=> Q = 3

FIG.17



194

Nodo 1: q=2

Trama 12

	Slot 1	Slot 2	Slot 3	Slot 4
Nodo 2	Tx Q=1 q=1			
Nodo 3				Tx Q=2 q=1
Nodo 4		Tx Q=1 q=1		
Nodo 5		Tx Q=3 q=1		

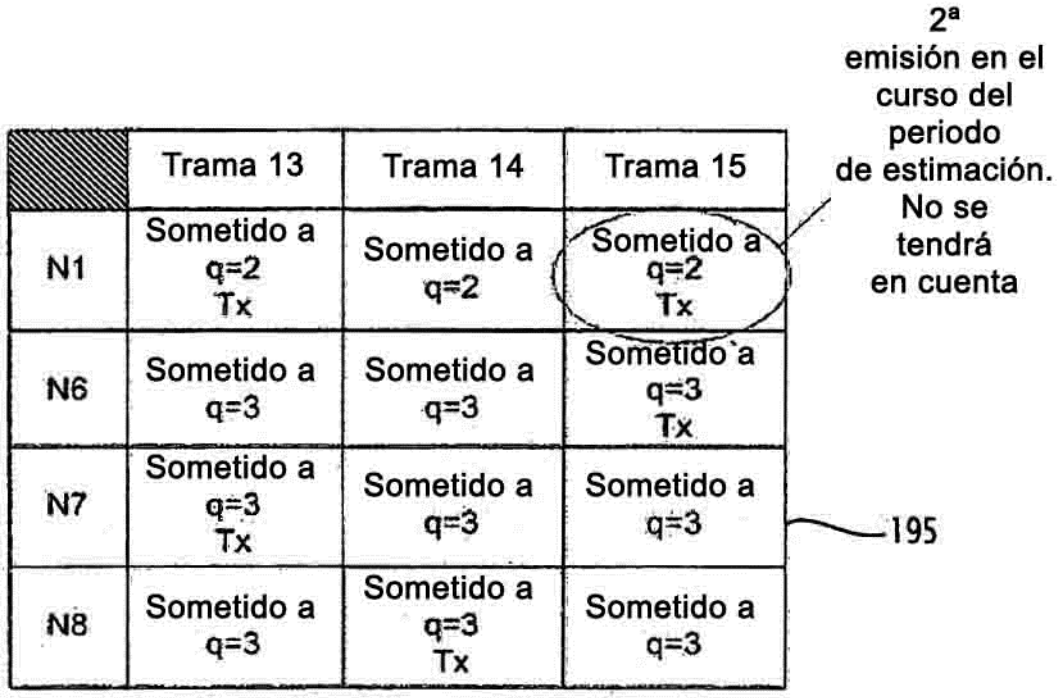
193

Nodo 5 : Detección de error:
Q(3) > q(2) ➔ "Nodo 1 no ha recibido mi restricción»

Trama 13

	Slot 1	Slot 2	Slot 3	Slot 4
Nodo 1	Tx Q=3 q=2			
Nodo 6				
Nodo 7			Tx Q=1 q=3	
Nodo 8				

FIG.18



	Trama 13	Trama 14	Trama 15
N1	Sometido a q=2 Tx	Sometido a q=2	Sometido a q=2 Tx
N6	Sometido a q=3	Sometido a q=3	Sometido a q=3 Tx
N7	Sometido a q=3 Tx	Sometido a q=3	Sometido a q=3
N8	Sometido a q=3	Sometido a q=3 Tx	Sometido a q=3

FIG.19