(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 395 955**

(21) Número de solicitud: 201200715

(51) Int. CI.:

H04L 12/715 (2013.01)

(12) SOLICITUD DE PATENTE A1

(22) Fecha de presentación:

05.07.2012

(43) Fecha de publicación de la solicitud:

18.02.2013

(71) Solicitantes:

UNIVERSIDAD DE CANTABRIA (50.0%)
Pabellón de Gobierno, Avda de los Castros s/n
39005 Santander (Cantabria) ES y
BARCELONA SUPERCOMPUTING CENTER
CENTRO NACIONAL DE SUPERCOMPUTACION
(50.0%)

(72) Inventor/es:

VALLEJO GUTIÈRREZ, Enrique;
ODRIOZOLA OLAVARRÍA, Miguel;
GARCÍA GONZÁLEZ, Marina;
BEIVIDE PALACIO, Ramón;
VALERO CORTÉS, Mateo y
LABARTA MANCHO, Jesús

(54) Título: **Método de encaminamiento adaptativo en redes jerárquicas**

(57) Resumen:

Método de encaminamiento de paquetes en una red directa jerárquica formada por una pluralidad de encaminadores, cada uno con puertos de tipo local y puertos de tipo global; cada puerto comprende una pluralidad de canales virtuales; dichos encaminadores forman grupos, donde los diferentes encaminadores de un mismo grupo están interconectados mediante una topología conexa empleando enlaces de tipo local uniendo parejas de puertos de tipo local, y los diferentes grupos están interconectados mediante una topología conexa empleando enlaces de tipo global uniendo parejas de puertos de tipo global. El método está configurado para emplear saltos por dichos enlaces de acuerdo a rutas mínimas y no mínimas; los saltos que implican rutas no mínimas pueden realizarse tanto a través de enlaces globales como locales. El número de canales virtuales necesarios en cada puerto local y global viene determinado solamente por la longitud de una ruta máxima permitida que no emplea misrouting de tipo local, empleando para ello un orden total en el recorrido de los canales virtuales, que se viola cuando se realiza un misrouting local.

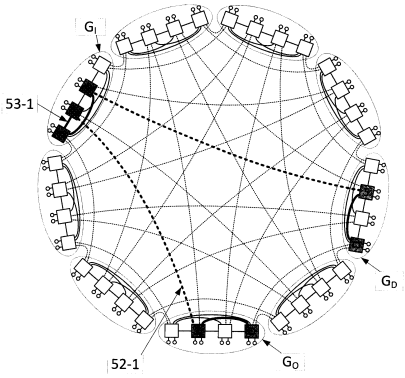

FIGURA 5

ES 2 395 955 A1

DESCRIPCIÓN

MÉTODO DE ENCAMINAMIENTO ADAPTATIVO EN REDES JERÁRQUICAS

CAMPO DE LA INVENCIÓN

La presente invención pertenece al campo de las redes para comunicaciones; más concretamente, es especialmente aplicable al campo de las redes de interconexión para computadores paralelos (multiprocesadores o multicomputadores).

ANTECEDENTES DE LA INVENCIÓN

En una red de comunicaciones basada en conmutación de paquetes, una serie de clientes (o nodos de cómputo) se comunican entre sí intercambiándose mensajes; cada uno de estos mensajes se divide en uno o más paquetes, que constituyen la unidad básica de conmutación en la red. Cada paquete tiene un cliente origen y uno o múltiples clientes destino (esto último, en el caso de paquetes *multicast*). A grandes rasgos, la red está compuesta por una serie de encaminadores (también conocidos como conmutadores, o *routers* o *switches* según los términos en inglés) que son los elementos activos de la red. Estos encaminadores están unidos mediante enlaces de comunicaciones, es decir, cables por los que se envían señales eléctricas u ópticas que transportan los paquetes de la red. Cada cliente se conecta mediante su interfaz de red a uno o más encaminadores utilizando el o los enlaces correspondientes, y a su vez los encaminadores se conectan entre sí mediante otros enlaces. Un encaminador dispone de múltiples puertos, a los que se conectan los enlaces correspondientes a otros encaminadores o clientes. Los clientes envían paquetes a los encaminadores, que se encargan de transportarlos de un encaminador a otro hasta llegar al cliente destino. La topología de la red es una descripción matemática de la forma en la que se conectan los diferentes encaminadores y clientes de la red.

Para que la comunicación sea posible, un encaminador debe ser capaz de recibir cada paquete que llegue por un cierto puerto de entrada, almacenarlo temporalmente,

procesarlo para determinar la ruta a seguir, y reenviarlo por el puerto de salida correspondiente. Para todo ello, los encaminadores 10 suelen tener una estructura interna similar al esquema presentado en la figura 1. Cada puerto de entrada p_{in} tiene asociada una unidad de entrada 11 con una o más memorias (también denominadas *bufferes* o colas) 12 en las que se almacenan datos correspondientes a los paquetes que se reciben por ese puerto p_{in}. Estos múltiples bufferes 12 se suelen utilizar para separar diferentes paquetes según su prioridad, tipo, o según una política de evitación de bloqueos (como se explica más adelante). Los paquetes almacenados en los bufferes 12 comparten el mismo enlace físico y puerto de entrada al switch; por ello, cuando hay varios de estos bufferes 12 se suelen denominar canales virtuales o "clases de buffer". Existe una lógica de encaminamiento que se encarga de determinar por qué puerto de salida p_{out} es apropiado reenviar cada paquete de los canales virtuales de entrada 12, y si acaso, en cuál de los canales virtuales del puerto de entrada del siguiente encaminador hay que introducir el paquete. A su vez, cada puerto de salida p_{out} puede tener o no una cierta memoria para almacenar los paquetes que tienen que salir por dicho puerto. La conexión entre los bufferes 12 de los puertos de entrada p_{in} y los puertos de salida p_{out} se realiza típicamente mediante un *crossbar* 13(en ocasiones traducido como *matriz de cruces*) que puede unir en cada ciclo de conmutación cualesquiera parejas de buffer de entradas y puerto de salida, una a una. Cada pareja de puertos de entrada y salida se conecta con un único enlace bidireccional. Un asignador ("*allocator*") regula el uso de recursos compartidos. Múltiples paquetes pueden solicitar un mismo puerto de salida, pero sólo se concede a uno de ellos cada vez. Un asignador ("*allocator*") puede ser de tipo dividido (es decir, separable) o no dividido (es decir, unificado). En caso de que el arbitraje sobre los recursos compartidos esté implementado mediante un asignador no separable, de acuerdo con William Dally y Brian Towles en *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003, el asignador (*allocator*) asigna los puertos de salida p_{out} en función de todas las rutas posibles que pueda seguir cada paquete en un puerto de entrada p_{in}. Para ello, se calcula para cada paquete el conjunto de rutas (puertos de salida p_{out} y canal virtual) por el que puede salir, y se pasan al asignador todos aquellas en las que hay hueco suficiente para el paquete. Después, el asignador busca la asignación de salidas a cada

paquete que maximice el throughput del router. En caso de que el arbitraje esté implementado mediante un asignador dividido, de acuerdo con William Dally y Brian Towles en *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003, cada puerto de entrada p_{in} elige uno de los canales virtuales que tengan un paquete listo para avanzar (por ejemplo, mediante una política round-robin), y el paquete decide, de entre todas las rutas posibles proporcionadas por la lógica de encaminamiento, la que más le interese. Después, se pasa la ruta seleccionada al asignador, que realiza la asignación de los puertos de salida p_{out} a los solicitantes.

Para poder enviar datos de un paquete a través de un puerto de salida sin pérdida de datos, es necesario que exista espacio de almacenamiento suficiente en el buffer, cola o canal virtual 12 del puerto de entrada p_{in} correspondiente del siguiente conmutador o encaminador. Existen dos mecanismos típicos de control de flujo para gestionar dicho espacio en el buffer de entrada del siguiente encaminador: *virtual cut-through* (*VCT*) (en ocasiones denominado *paso a través* en castellano) permite el avance de un paquete solo si existe espacio disponible para almacenarlo completo; *wormhole* (*WH*) (encaminamiento de agujero de gusano, aunque el nombre en castellano no se suele emplear) divide los paquetes en *flits (flow control digit*, en español unidad de control de flujo), y permite que avancen tantos *flits* como hueco haya en el siguiente buffer. Así, WH permite que un paquete se quede parado en la red ocupando dos o más canales virtuales o bufferes de entrada de diferentes encaminadores consecutivos. De hecho, VCT requiere que los bufferes o canales virtuales tengan una capacidad igual o superior al tamaño máximo de un paquete mientras que WH solo precisa que tengan capacidad para uno o más flits. De esta manera, WH se suele emplear en entornos en los que el área del chip o el consumo energético (generado por dichos bufferes) resulta crítico, como redes dentro del chip. Sin embargo, en redes de sistema , es habitual el uso de VCT al ser más sencilla su implementación. En cualquiera de los dos casos, si no existe espacio suficiente en el siguiente buffer o canal virtual, típicamente es necesario esperar a que los datos del dicho buffer avancen hacia su destino, y se libere espacio. En este caso, se dice que existe una dependencia entre un conmutador y el siguiente.

En una red, el interbloqueo (habitualmente denominado *deadlock* según el término inglés, o simplemente *bloqueo*) es la situación en la que ningún paquete de un conjunto de paquetes dado no puede avanzar hacia su destino porque se produce una dependencia cíclica entre los recursos solicitados para implementar dicho avance. Por ejemplo, cada paquete está ocupando un hueco en una cola de un encaminador, y para poder avanzar necesita que se libere un hueco en la cola correspondiente del siguiente encaminador, que a su vez está esperando a que se libere un hueco en un tercer encaminador, etc., formándose al final un ciclo de colas llenas en el que ninguno de los paquetes puede avanzar y nunca se libera un hueco. Esta situación es crítica para una red, ya que provoca una parada completa de su funcionamiento de la que no se puede salir. Para evitar este problema del interbloqueo (*deadlock*), se han desarrollado diferentes técnicas, que o bien detectan y solventan esta situación (por ejemplo, descartando alguno de los paquetes que conforman la dependencia cíclica y liberando su "hueco" en el buffer) o bien no permiten que se llegue a ella (por ejemplo, mediante restricciones en el encaminamiento de los paquetes en la red que no permiten que se generen dependencias cíclicas). Las primeras técnicas se denominan de resolución de bloqueos, y son más frecuentes en las redes con pérdidas, que son aquellas que se asumen poco fiables y que no garantizan la entrega de los datos en el destino (como Ethernet). En cambio, las técnicas del segundo tipo se denominan de evitación de bloqueos, y son preferibles en redes sin pérdidas ya que no precisan de la retransmisión de los datos. Los mecanismos de evitación de bloqueos están en muchos casos íntimamente relacionados con la topología de la red, el mecanismo de control de flujo (VCT o WH), y con el uso de los diferentes canales virtuales de cada puerto de entrada.

Una topología propuesta recientemente para su uso en redes de interconexión de sistemas de alta escala es la denominada *Dragonfly* , descrita en la solicitud de patente estadounidense US2010/0049942 A1 y en Technology-driven, highly-scalable dragonfly topology publicada en *ISCA '08: Proceedings of the 35th annual International Symposium on Computer Architecture*, pages 77–88, 2008, por John Kim et al., o *red directa jerárquica*, descrita por B. Arimilli et al. en The PERCS High-Performance Interconnect. In *Proceedings of 18th Symposium on High-Performance*

Interconnects (Hot Interconnects 2010). IEEE, Aug. 2010. Se muestra un caso concreto de esta topología en la figura 2. La idea general de esta topología es emplear encaminadores 20 de alto número de puertos unidos en "grupos" 21. En el ejemplo mostrado en la figura 2 cada grupo 21 está formado por un mismo número de encaminadores 20, pero no tiene por qué ser así. Globalmente, todos los grupos 21 están unidos entre sí, con un único enlace entre cada pareja de grupos, de acuerdo a lo que se denomina topología de grafo completo. La figura 2 muestra en trazo discontinuo los enlaces globales 22. Localmente, los conmutadores de cada grupo también están conectados entre sí, típicamente también con una topología de grafo completo con un único enlace entre cada pareja de conmutadores. La figura 2 muestra en trazo continuo los enlaces locales 23. Por tanto, un grupo 21 está formado por un conjunto de conmutadores o encaminadores 20 cercanos y por los clientes del sistema que están conectados a ellos (nodos 24). Todo ello se ubica típicamente en un mismo *cabinet* o varios cabinets vecinos. Los enlaces que unen conmutadores de diferentes grupos se denominan enlaces globales 22 y emplean típicamente tecnología óptica, debido a su mayor longitud. Los enlaces entre los conmutadores de un grupo se denominan enlaces locales 23, y pueden utilizar tecnología eléctrica gracias a su menor longitud.

El mecanismo de encaminamiento de la red es el que determina la ruta que siguen los paquetes desde un nodo origen (o, de manera equivalente, desde un conmutador origen) hasta un nodo (o conmutador) destino. El mecanismo de encaminamiento propuesto para estas redes en la bibliografía permite dos tipos de rutas: i) La ruta *mínima,* que utiliza la única ruta más corta entre cualquier pareja de encaminadores origen y destino. Considerando las topologías de grafo completo tanto a nivel local de los grupos como global, la ruta mínima atraviesa a lo sumo tres enlaces: local – global – local. Un ejemplo de ruta mínima está mostrado en la figura 3, en la que para enrutar un paquete desde el encaminador origen "O" hasta el encaminador destino "D" el paquete da un primer salto local 23-1 seguido de un salto global 22-1 y un segundo salto local 23-2. ii) Una ruta *no-mínima*, o ruta Valiant, tal y como se describe por L. G. Valiant en A scheme for fast parallel communication. *Journal on Computing*, 11(2):350–361, 1982, en la que primero se envía el paquete a un grupo intermedio (diferente del grupo origen

y destino) para balancear el tráfico, empleando hasta dos enlaces, local y global, y a partir de ahí hasta el destino, utilizando la ruta mínima. Por tanto, este mecanismo permite rutas de longitud máxima 5: local – global – local – global – local. Este encaminamiento no-mínimo tiene sentido cuando el enlace global de la ruta mínima correspondiente está saturado, ya que aunque se recorren más enlaces de la red, se sortea el enlace saturado. Este encaminamiento está representado en el ejemplo de la figura 4, en el que se numera la secuencia de encaminadores atravesados para enrutar un paquete desde el encaminador origen "O" hasta el encaminador destino "D": El paquete, que se encuentra en un encaminador origen "O" que pertenece a un grupo origen G_O da un primer salto local 43-1 hasta un encaminador "1" del mismo grupo origen G_O. A continuación, se produce un primer salto global 42-1 hasta un nodo "2" perteneciente a un grupo intermedio G_I. Seguidamente el paquete recorre un segundo camino local 43-2 hasta llegar a un encaminador "3" del mismo grupo intermedio G_I. Entonces el paquete sigue un segundo salto global 42-2 hasta un encaminador "4" perteneciente al grupo destino G_D. Finalmente el paquete toma un camino local 43-3 hasta el nodo destino "D".

En el caso general, es deseable que el encaminamiento se adapte a las circunstancias de ocupación de la red mediante un mecanismo adaptativo que seleccione entre una ruta mínima o no-mínima en función del estado de los enlaces de la red. El problema de estos mecanismos es que, para tomar la decisión de utilizar la ruta mínima o una no-mínima en el router o encaminador de origen, necesitan información del estado del enlace global de la ruta mínima que no necesariamente está conectado a este router de origen, como ocurre en los ejemplos de las figuras 3 y 4. Por ello, esta decisión debe tomarse mediante una estimación del estado de la red a partir de información remota. Entre estos mecanismos están, por ejemplo, UGAL, Piggybacking (PB), Credit Round-Trip Time (CRT) o Reservation (RES) propuestos por Nan Jiang et al. en Indirect adaptive routing on large scale interconnection networks en *ISCA '09: Proceedings of the 36th annual International Symposium on Computer Architecture*, pages 220–231, 2009 y por John Kim et al. en el anteriormente citado Technology-driven, highly-scalable dragonfly topology. In *ISCA '08: Proceedings of the 35th annual International*

Symposium on Computer Architecture, pages 77–88, 2008. El mecanismo Progressive Adaptive Routing (PAR), introducido por Nan Jiang et al., permite que cuando se elige una ruta mínima, tras el primer salto se modifique a una ruta no-mínima comenzando en el router en curso (el segundo de la ruta mínima) y utilizando la información local del router en curso. Este mecanismo con adaptatividad en tránsito es interesante porque permite adaptar el encaminamiento más rápido en presencia de cambios del tráfico de la red, pero a costa de rutas máximas más largas ya que permite un primer salto local adicional.

El hecho de realizar un salto por un enlace de la red que no acerca el paquete a su destino final se denomina técnicamente *misrouting*. En el caso anterior del encaminamiento no-mínimo, el primer salto global 42-1 (entre los encaminadores numerados como "1" y "2") es un *misrouting global*, que se utiliza para sortear un enlace global saturado. Esta saturación de enlaces globales puede ser frecuente en una topología Dragonfly, ya que existe un único enlace global entre cada pareja de grupos, con múltiples nodos en cada grupo. Cuando los nodos de estos grupos se comunican entre sí, el único enlace global que los une tiende a saturarse, y mediante el *misrouting* dicho enlace se puede evitar a costa de pasar por un grupo intermedio aleatorio. Nótese que, en función del grupo elegido, se puede necesitar un primer salto local previo al *misrouting* global (en la figura 4, el salto local 43-1 entre los encaminadores numerados como "0" y "1"). De manera análoga se define un *misrouting local* como un salto a través de un enlace local (por tanto, que no sale del grupo en que se encuentra el paquete) que no acerca el paquete a su destino. El *misrouting* local tiene el objetivo de sortear enlaces locales saturados dentro de un grupo. En la solicitud de patente europea EP2451127A1 se sugiere el uso de *misrouting* local en cualquier grupo de la ruta del paquete. La figura 5 muestra un ejemplo de una ruta no mínima, con un *misrouting* global 52-1 seguido de un *misrouting* local 53-1 en el grupo intermedio G_I.

El mecanismo de evitación de bloqueos propuesto para esta topología Dragonfly en Technology-driven, highly-scalable dragonfly topology en *ISCA '08: Proceedings of the 35th annual International Symposium on Computer Architecture*, pages 77–88,

2008, por John Kim et al. y en B. Arimilli et al. en The PERCS High-Performance Interconnect en *Proceedings of 18th Symposium on High-Performance Interconnects (Hot Interconnects 2010)*. IEEE, Aug. 2010, se basa en una técnica original propuesta por K. Günther en Prevention of deadlocks in packet-switched data transport systems en *Communications, IEEE Transactions on*, 29(4):512 – 524, Abril 1981. Dicho mecanismo evita la aparición de bloqueos en base al uso ordenado de tantos canales virtuales (*Virtual Channels*, VCs) en los puertos de entrada de los encaminadores como la longitud en saltos de la ruta más larga permitida en la red. La idea clave es que cada vez que un encaminador reenvía un paquete, se incrementa el índice del canal virtual utilizado para ello. Dicho de otra manera, se impone una relación de orden total en el uso de los recursos de la red, en este caso los canales virtuales en cada uno de los puertos de entrada de los diferentes encaminadores. De esta manera se evita el interbloqueo o deadlock, lo que puede demostrarse intuitivamente de manera recursiva: Los paquetes en el canal virtual con el índice más alto no se bloquean, porque van a consumirse; los paquetes en un cierto canal virtual no se bloquean, porque o bien van a consumirse, o bien dependen del inmediatamente superior que está libre de bloqueo.

El problema de esta implementación es que, en general, requiere un elevado número de canales virtuales, lo que se traduce en una elevada área de silicio y una mayor complejidad de diseño de los encaminadores. Si se permite el encaminamiento no-mínimo con una ruta *local – global – local – global – local* como la mostrada en la figura 4, entonces son necesarios 5 canales virtuales, denominados VC0 a VC4. Sin embargo, al coincidir que cada salto de la ruta recorre siempre el mismo tipo de enlace, local en el caso de los saltos impares y global en de los pares, la implementación del encaminador puede hacerse con solo 3 VCs en los puertos locales y 2 VCs en los globales. En el caso de que un paquete siga una ruta más corta, basta con omitir los canales correspondientes a los saltos que no aparecen en la ruta. En el caso de emplear el encaminamiento PAR, introducido por Nan Jiang et al., como permite rutas de longitud 6, es necesario un VC más, en concreto en los puertos locales (en total, 4 VCs locales y 2 VCs globales).Previamente se ha argumentado el interés de que el mecanismo de encaminamiento permita *misrouting* local. Sin embargo, es evidente que

este *misrouting* alarga la ruta máxima permitida en la red, y por tanto aumenta, según el mecanismo de evitación de bloqueos anterior, el número de canales virtuales necesarios. Si no se restringe el número de *misroutings* locales permitidos, hace falta un número ilimitado de canales virtuales con el mecanismo anterior, lo que claramente es no implementable. En concreto, si se limita a un máximo de un *misrouting* local por cada grupo que atraviesa el paquete, la ruta se alargará en hasta tres saltos locales (por los tres grupos que puede atravesar el paquete: origen, intermedio y destino) lo que daría lugar a 6 VCs para los enlaces locales y 2 VCs para los globales.

En resumen, no se conoce ningún mecanismo de encaminamiento en el estado del arte que permita el uso de *misroutings* locales, con adaptatividad completa en tránsito sin restringir el encaminamiento y sin emplear un elevado número de canales virtuales. De existir, dicho mecanismo sería muy deseable, ya que permitiría obtener un elevado rendimiento (por el *misrouting* local que permite sortear enlaces locales saturados), se adaptaría rápidamente a cambios en el tipo de tráfico (por la adaptatividad en tránsito), utilizaría de manera balanceada los recursos de la red por la adapatatividad completa y no emplearía recursos adicionales.

RESUMEN DE LA INVENCIÓN

La presente invención trata de resolver los inconvenientes mencionados anteriormente mediante un método de encaminamiento para redes directas jerárquicas.

Concretamente, en un primer aspecto de la presente invención, se proporciona unmétodo de encaminamiento de paquetes en una red directa jerárquica formada por una pluralidad de encaminadores, cada uno de ellos con una pluralidad de puertos de tipo local y una pluralidad de puertos de tipo global, donde cada uno de los puertos comprende una pluralidad de canales virtuales, y donde los encaminadores forman grupos, donde los diferentes encaminadores de un mismo grupo están

interconectados mediante una topología conexa empleando enlaces de tipo local uniendo parejas de puertos de tipo local, y a su vez los diferentes grupos están interconectados mediante una topología conexa empleando enlaces de tipo global uniendo parejas de puertos de tipo global. El método está configurado para emplear saltos por esos enlaces de acuerdo a rutas mínimas y no mínimas, donde los saltos que implican rutas no mínimas pueden realizarse tanto a través de enlaces globales como locales. Además, el número de canales virtuales necesarios en cada puerto local y global viene determinado solamente por la longitud de una ruta máxima permitida que no emplea *misrouting* de tipo local, empleando para ello un orden total en el recorrido de los canales virtuales, que se viola cuando se realiza un *misrouting* local.

En una realización particular, la conexión entre los diferentes encaminadores de un mismo grupo se realiza de acuerdo a un grafo completo, y la conexión entre los diferentes grupos también se realiza de acuerdo a un grafo completo, y cada puerto local comprende solo 3 canales virtuales y cada puerto global comprende solo 2 canales virtuales.

En una realización particular, para cada paquete situado en un canal virtual de un puerto de un encaminador, el método comprende:

-calcular al menos un salto de acuerdo a un encaminamiento mínimo entre el encaminador en que se encuentra dicho paquete y el encaminador al que está conectado el nodo al que dicho paquete está dirigido;

-calcular al menos un salto de acuerdo a un encaminamiento no-mínimo que comprende un *misrouting* global, a través de un grupo de encaminadores intermedio diferente del grupo al que pertenecen el encaminador de origen y el encaminador al que está conectado el nodo destino del paquete;

-calcular, si no se ha alcanzado un determinado límite de *misrouting* locales y el encaminamiento mínimo ha calculado saltos de tipo local, al menos un salto local no-mínimo diferente a dichos saltos calculados mediante el encaminamiento mínimo;

-seleccionar uno de dichos saltos en función de un determinado criterio.

En la realización anterior, el método puede emplear un asignador unificado en cada encaminador para llevar a cabo la selección de uno de dichos saltos para que avance un paquete.

Alternativamente, se selecciona una ruta para cada paquete en cada ciclo de arbitraje mediante una comparación de la ocupación del canal virtual de entrada en el siguiente encaminador correspondiente a una ruta mínima seleccionada, frente a la ocupación de los canales virtuales de entrada de los encaminadores correspondientes a otras rutas.

Alternativamente, se selecciona una ruta para cada paquete en cada ciclo de arbitraje mediante una comparación de los valores de una pluralidad de contadores de contención, existiendo tantos contadores como puertos de salida tiene el encaminador, registrando dichos contadores el número de paquetes de los puertos de entrada cuya ruta mínima avanza por el puerto de salida correspondiente.

Alternativamente, se selecciona una ruta para cada paquete en cada ciclo de arbitraje de acuerdo a una combinación tanto de la información obtenida de la ocupación de los canales virtuales de entrada de los encaminadores vecinos como de una pluralidad de contadores de contención, registrando dichos contadores el número de paquetes de los puertos de entrada cuya ruta mínima avanza por el puerto de salida correspondiente.

En otra realización particular, se emplea control de flujo *wormhole* en todos los saltos en que se respeta el orden total establecido para los canales virtuales, y control de flujo *virtual cut-through* en los saltos en que se hace un *misrouting* local que viola dicho orden total, permitiendo que todos los canales virtuales tengan un tamaño

inferior al tamaño máximo del paquete menos los correspondientes al primer canal virtual.

En otra realización particular, el orden total puede ser un orden ascendente o un orden descendente.

En otro aspecto de la invención, se proporciona una red directa jerárquica formada por una pluralidad de encaminadores, cada uno de ellos con una pluralidad de puertos de tipo local y una pluralidad de puertos de tipo global, donde cada uno de los puertos comprende una pluralidad de canales virtuales, donde los encaminadores forman grupos, donde los diferentes encaminadores de un mismo grupo están interconectados mediante una topología conexa empleando enlaces de tipo local uniendo parejas de puertos de tipo local, y a su vez los diferentes grupos están interconectados mediante una topología conexa empleando enlaces de tipo global uniendo parejas de puertos de tipo global. La red directa jerárquica comprende medios para llevar a cabo el método anterior.

Como puede apreciarse, este método de encaminamiento permite evitación de bloqueos, adaptatividad en tránsito y el uso de *misrouting* local, sin requerir para ello más de los 3 VCs locales y los 2 VCs globales necesarios en mecanismos de encaminamiento del estado del arte previo.

Otras ventajas de la invención se harán evidentes en la descripción siguiente.

BREVE DESCRIPCIÓN DE LAS FIGURAS

Con objeto de ayudar a una mejor comprensión de las características de la invención, de acuerdo con un ejemplo preferente de realización práctica del mismo, y para complementar esta descripción, se acompaña como parte integrante de la misma, un juego de dibujos, cuyo carácter es ilustrativo y no limitativo. En estos dibujos:

La figura 1 muestra un esquema de la arquitectura de un encaminador.

La figura 2 muestra un ejemplo de la topología Dragonfly o red directa jerárquica.

La figura 3 muestra un ejemplo de encaminamiento en una red directa jerárquica, siguiendo una ruta mínima entre dos grupos.

La figura 4 muestra un ejemplo de encaminamiento en una red directa jerárquica, siguiendo una ruta Valiant o ruta no-mínima entre dos grupos, en la que se numera la secuencia de encaminadores atravesados.

La figura 5 muestra un ejemplo de encaminamiento no-mínimo con *misrouting* global y local en el grupo intermedio.

La figura 6 muestra un esquema de un encaminador de la red sobre la que se implementa el método de encaminamiento de la invención.

La figura 7 muestra siete ejemplos de encaminamiento entre dos grupos y rutas resultantes del método de encaminamiento de la invención, desde un encaminador origen O hasta el destino D. En concreto, se ejemplifica: a) Ruta mínima. b) Primer y segundo salto ruta no mínima y, a continuación, ruta mínima hasta alcanzar el destino. c) Primer salto ruta no mínima y, a continuación, ruta mínima hasta alcanzar el destino. d) Primer salto ruta mínima, a continuación ruta no mínima y por último ruta mínima hasta alcanzar el destino. e) Misma ruta que en el caso c, con *misrouting* local en el grupo intermedio. f) Misma ruta que en el caso e, con *misrouting* local en el grupo destino. g) Misma ruta que en el caso b, con *misrouting* local en el grupo intermedio.

La figura 8 muestra tres ejemplos de orden en el uso de los canales virtuales, de acuerdo con una posible implementación del método de encaminamiento de la invención, considerando las rutas empleadas en las figuras 7-b. 7-g y 7-f: La primera a) tiene todos los saltos ascendentes, la segunda b) tiene un único *misrouting* local que viola dicho

orden ascendente y la tercera c) tiene 3 *misroutings* locales que violan dicho orden ascendente.

DESCRIPCIÓN DETALLADA DE LA INVENCIÓN

En este texto, el término "comprende" y sus variantes no deben entenderse en un sentido excluyente, es decir, estos términos no pretenden excluir otras características técnicas, aditivos, componentes o pasos.

Además, los términos "aproximadamente", "sustancialmente", "alrededor de", "unos", etc. deben entenderse como indicando valores próximos a los que dichos términos acompañen, ya que por errores de cálculo o de medida, resulte imposible conseguir esos valores con total exactitud.

Se define *misrouting* como el acto de realizar un salto por un enlace de la red que no acerca el paquete a su destino final. En español podría llamarse "desvío", aunque suele utilizarse el término inglés.

Cuando ese *misrouting* se realiza entre encaminadores de diferente grupo, se trata de un *misrouting* global, que se utiliza para sortear un enlace global saturado.

Cuando ese *misrouting* se realiza entre encaminadores del mismo grupo, se trata de un *misrouting* local, que se utiliza para sortear un enlace local saturado dentro de un mismo grupo.

Las siguientes realizaciones preferidas se proporcionan a modo de ilustración, y no se pretende que sean limitativas de la presente invención. Además, la presente invención cubre todas las posibles combinaciones de realizaciones particulares y preferidas aquí indicadas. Para los expertos en la materia, otros objetos, ventajas y características de la invención se desprenderán en parte de la descripción y en parte de la práctica de la invención.

El método de encaminamiento de la invención es aplicable a redes directas jerárquicas, tal y como se esquematiza de forma general en la figura 2, donde cada uno de los encaminadores 20 responde de forma general a la arquitectura representada en la figura 6. La red está formada por una pluralidad de encaminadores 20, cada uno de los cuales comprende varios puertos de inyección y consumo a los que se conectan o pueden conectarse nodos de cómputo 24. Los encaminadores 20 se agrupan formando grupos 21 mediante una topología conexa, es decir, en la que existe al menos un camino para comunicar cualquier pareja de nodos. Igualmente, los diferentes grupos se interconectan mediante una topología conexa. Aunque en la figura 2 todos los grupos tienen un mismo número de encaminadores, en general grupos diferentes pueden tener un número diferente de encaminadores. La figura 2 muestra también los enlaces locales 23, es decir, entre encaminadores de un mismo grupo, y los enlaces globales 22, es decir, entre encaminadores de grupos diferentes. Preferentemente, el presente método de encaminamiento es aplicable cuando la conexión entre encaminadores de un mismo grupo se corresponde, al menos, con un grafo completo (también conocido como *flattened butterfly* de 1 dimensión), con al menos un enlace entre cada pareja de encaminadores. Además, en caso de disponer de puertos adicionales en los encaminadores pueden existir enlaces locales 23 paralelos entre una misma pareja de encaminadores del mismo grupo. También preferentemente, el presente método de encaminamiento es aplicable cuando la conexión entre grupos se corresponde, al menos, con un grafo completo.

En la arquitectura del encaminador se asume que al menos el canal virtual VC0 tiene capacidad suficiente para albergar un paquete del tamaño máximo permitido en la red.

La figura 6 muestra un esquema de un encaminador 60 de la red sobre la que se implementa el método de encaminamiento de la invención. Cada encaminador 60 necesita tres canales virtuales (VC0, VC2 y VC4, referenciados en la figura como 62-0, 62-2 y 62-4) en los puertos locales 62, y dos canales virtuales (VC1 y VC3, referenciados en la figura como 61-1 y 61-3) en los puertos globales 61. El etiquetado

concreto de los canales virtuales puede variar mientras se mantenga la cantidad de puertos y el orden relativo. El puerto que comunica cada nodo de cómputo con un encaminador de la red se denomina *puerto de inyección*, no ilustrado en la figura 6 Dicho puerto no precisa estar dividido en canales virtuales, y si se especifica un índice para el mismo, es irrelevante para la evitación de bloqueos. A efectos del orden, los paquetes en un puerto de inyección se considera que están en el canal virtual -1.

El método o mecanismo propuesto proporciona, en cada encaminador de la red, el conjunto de rutas que puede seguir un paquete. Estas rutas se corresponden con rutas mínimas (desde el encaminador en curso hasta el destino, independientemente de la ruta seguida previamente por el paquete), rutas no-mínimas que utilizan un *misrouting* global para pasar por un grupo intermedio, así como los *misrouting* locales internos a un grupo. Después, alguna lógica del encaminador en curso se encarga de seleccionar una opción entre todas las posibles para realizar el avance del paquete; es decir, se permite adaptatividad en tránsito. Esta lógica puede utilizar información del estado de la red para seleccionar la ruta más apropiada. La ocupación de los bufferes o canales virtuales de los diferentes caminos, que se deriva de la cuenta de créditos de las salidas de los encaminadores, es un ejemplo de un indicador del estado de la red.

Como se explica más adelante, el mecanismo de encaminamiento está libre de interbloqueo. Para garantizarlo, las rutas mínima y Valiant (no-mínima con *misrouting* global) siguen un orden ascendente de índices en los canales virtuales utilizados, lo que garantiza que los paquetes se pueden encaminar hasta el destino utilizando dichas rutas sin bloqueo. En un caso general, se puede seguir cualquier orden total en el uso de los canales virtuales, no necesariamente el citado orden ascendente. Por el contrario, el *misrouting* local viola dicha relación de orden total, reutilizando el mismo canal en curso, o uno inferior. Esta violación hace que no se pueda garantizar el avance de los paquetes para las rutas que emplean *misrouting* local, y únicamente se pueda realizar este avance cuando exista hueco en un enlace local apropiado; sin embargo, en la práctica esto es frecuente, y se consigue así sortear los enlaces locales saturados.

El mecanismo de encaminamiento propuesto, R, es del tipo $R: C \times N \longmapsto \mathrm{P}(C)$, es

decir: el mecanismo de encaminamiento R se implementa como una función que, dado un paquete situado en un canal virtual C de un puerto de entrada de un encaminador dado, y para un nodo destino N, devuelve el conjunto de canales virtuales de los puertos de salida $\mathrm{P}(C)$ por los que puede avanzar el paquete. Así, R no especifica únicamente el puerto por el que puede salir un paquete, sino también el canal virtual por el que debe avanzar en el encaminador vecino si sale por dicho puerto. Para calcular las posibles rutas a seguir, R emplea la identificación del encaminador en curso (su índice E_j en la red y el grupo al que pertenece G_i), el puerto y canal virtual en los que se encuentra el paquete (denominados puerto de entrada y canal virtual de entrada) así como la información de encaminamiento presente en los metadatos del paquete (el nodo de origen N_{origen}, que está conectado al encaminador E_{origen} perteneciente al grupo de origen G_{origen}; y el nodo de destino $N_{destino}$, conectado al encaminador $E_{destino}$ en el grupo $G_{destino}$). Esta información de encaminamiento puede aparecer de forma explícita en los metadatos del paquete, o bien pueden ser calculados (por ejemplo, si E y G se derivan matemáticamente a partir de N y de las propiedades de la red, como el número de nodos por encaminador, encaminadores por grupo, etc). Además, asumimos que el paquete dispone de unos *flags* o contadores para limitar la cantidad de veces que se puede hacer *misrouting* tanto local como global, y que existen unos límites al número de veces que se permite usar *misrouting*: $L_{local-misrouting}$ y $L_{global-misrouting}$. $L_{local-misrouting}$ puede referirse al número de veces que se permite el *misrouting* local bien en toda la ruta del paquete, o bien por grupo. Aunque se podría generalizar, asumimos $L_{global-misrouting} = 1$, al igual que en todos los trabajos previos.

El mecanismo de encaminamiento propuesto R se puede expresar como la unión de tres subfunciones de encaminamiento separadas: $R = R_{min} \cup R_{non-min} \cup R_{local-misr.}$ Cada uno de estos tres mecanismos devuelve un conjunto de puertos y canales virtuales de salida para un paquete en un nodo dado y un canal virtual dado, independientes de la ruta seguida previamente por el paquete. En general, para

alcanzar su destino, el paquete puede seguir cualquiera de las rutas proporcionadas por el mecanismo de encaminamiento. Sin embargo, en función del estado de la red (ocupación de las colas o algún otro indicador), la lógica de los encaminadores puede seleccionar un subconjunto de opciones para conseguir el mejor rendimiento.

Nótese que en caso de emplear una topología de grafo completo (sin enlaces paralelos) tanto entre los encaminadores de un grupo como entre grupos, existe una única ruta mínima R_{min}. Sin embargo, si se han empleado enlaces adicionales para aprovechar puertos disponibles en los encaminadores, entonces dicha ruta no será única.

El mecanismo propuesto R se define completamente si se definen cada una de sus tres componentes, lo que se hace a continuación. Se considera la siguiente terminología: E_i (encaminador en el que se encuentra el paquete), G_i (grupo al que pertenece el encaminador en el que se encuentra el paquete), N_{origen} (nodo origen), E_{origen} (encaminador al que se encuentra conectado N_{origen}), G_{origen} (grupo al que pertenece E_{origen}), $N_{destino}$ (nodo destino), $E_{destino}$ (encaminador al que se encuentra conectado $N_{destino}$), $G_{destino}$ (grupo al que pertenece $E_{destino}$), E_{out} (conjunto de encaminadores de G_i que disponen de un enlace global que conecta directamente con $G_{destino}$).

- R_{min} se corresponde con el encaminamiento mínimo de un paquete desde E_i hasta $N_{destino}$. Así, R_{min} genera un conjunto de rutas (al menos una) de acuerdo a la primera de las siguientes condiciones que sea cierta:
  a) Si $E_i = E_{destino}$. En este caso la ruta viene dada por el puerto que conecta a $N_{destino}$.
  b) Si $G_i = G_{destino}$. En este caso, las rutas (al menos una) vienen dadas por el conjunto de puertos que conectan directamente E_i con $E_{destino}$.
  c) Si E_i tiene al menos un enlace global que lo conecta directamente con $G_{destino}$. En este caso las rutas (al menos una) vienen dadas por el

conjunto de puertos que conectan directamente con $G_{destino}$.

d) Si $G_i \neq G_{origen}$. Entonces R_{min} comprende todas las rutas (al menos una) que unen con E_{out}.

e) Si $G_i = G_{origen}$. R_{min} comprende todas las rutas (al menos una) que unen con E_{out}, pero además se debe emplear explícitamente el canal virtual VC0.

El índice del canal virtual de la salida en los casos a)-d) depende del tipo de puerto por el que se recibe el paquete (local o global) y por el que se tiene que reenviar en cada encaminador por el que pase, así como del índice del canal virtual en el que está el paquete en el puerto de entrada. Si ambos puertos (entrada y salida) del encaminador en curso, son del mismo tipo (local-local o global-global) el índice se aumenta en dos unidades; en otro caso, se aumenta en una unidad.

- $R_{non-min}$ se corresponde con el encaminamiento no-mínimo de un paquete a través de un grupo intermedio, sin *misrouting* local. $R_{non-min}$ genera un conjunto de rutas (al menos una) de acuerdo a la primera de las siguientes condiciones que sea cierta:

  f) Si $G_i \neq G_{origen}$, $R_{non-min}$ devuelve el mismo resultado que R_{min}.

  g) Si $E_i \neq E_{origen}$ y E_i no dispone de un enlace global que le conecta directamente con $G_{destino}$, entonces el paquete puede salir por cualquiera de los puertos globales de E_i, utilizando el canal virtual VC1.

  h) En otro caso, o bien $E_i = E_{origen}$ o bien $E_i \neq E_{origen}$ y además dispone de un enlace global que lo conecta directamente con $G_{destino}$. En este caso el paquete puede salir por cualquiera de los puertos locales de E_i utilizando el canal virtual VC0 o por cualquiera de los puertos globales de E_i utilizando el canal virtual VC1.

Se propone una implementación alternativa a la descrita en las reglas f), g) h), que es más restrictiva en la generación de las rutas de $R_{non-min}$ con el

objetivo de balancear el tráfico saliente por los diferentes enlaces del grupo G_{origen} a la vez que se minimiza la longitud de las rutas recorridas, de acuerdo a las siguientes reglas (alternativas a f)-h), nótese que i) coincide con f)):

i) Si $G_i \neq G_{origen}$, $R_{non-min}$ devuelve el mismo conjunto de rutas que R_{min}.

j) Si $E_i = E_{origen}$ y el paquete se encuentra en el puerto de inyección, el paquete puede salir por cualquiera de los enlaces globales de E_i utilizando el canal virtual VC1.

k) Si E_i no dispone de un enlace global que le conecta directamente con $G_{destino}$, entonces el paquete puede salir por cualquiera de los puertos globales de E_i, utilizando el canal virtual VC1.

l) Si E_i sí dispone de un enlace global que le conecta directamente con $G_{destino}$, entonces el paquete puede salir por cualquiera de los puertos locales de E_i, utilizando el canal virtual VC0.

- $R_{local-misr}$ se corresponde con el *misrouting* local en el grupo destino o un grupo intermedio. En el grupo de origen, $R_{non-min}$ es el que permite realizar un *misrouting* (global, en dicho caso) cuando se detecta congestión. $R_{local-misr}$ genera sus rutas de acuerdo a la primera de las siguientes condiciones que sea cierta:

m) Si $G_i = G_{destino}$, y además $E_i \neq E_{destino}$, y la cuenta de *misroutings* locales del paquete es menor que $L_{local-misrouting}$, entonces $R_{local-misr}$ comprende todos los enlaces locales de E_i.

n) Si $G_i \neq G_{destino}$, $G_i \neq G_{origen}$, y además E_i no tiene un enlace global que conecta con $G_{destino}$ y la cuenta de *misroutings* locales del paquete es menor que $L_{local-misrouting}$, entonces $R_{local-misr}$ comprende todos los enlaces locales de E_i.

o) En otro caso, no se genera ninguna ruta.

En ambos casos, el canal virtual utilizado es el mismo que el que contiene el

paquete en el puerto de entrada (si éste es de tipo local) o una unidad inferior (si es global).

A continuación se muestran algunos ejemplos de posibles rutas generadas por el mecanismo ilustradas en la figura 7.

La figura 7-a muestra una ruta desde E_{origen} (O) hasta $E_{destino}$ (D) en la que todos los saltos vienen dados por el encaminamiento mínimo (condiciones a-e).

La figura 7-b muestra una ruta desde E_{origen} (O) hasta $E_{destino}$ (D). Estando un paquete en E_{origen}, éste sale por uno de sus puertos locales devueltos por $R_{non-min}$ utilizando el canal virtual VC0 (condición h)), 71. Seguidamente, en el encaminador 1 el paquete sale por uno de los puertos globales devueltos por $R_{non-min}$, utilizando el canal virtual VC1 (condición h)), 72. A partir de ahí, el paquete se encamina por la ruta mínima hasta $E_{destino}$ (condiciones a) a d)).

La figura 7-c muestra el encaminamiento desde E_{origen}(O) hasta $E_{destino}$ (D) cuando se realiza un *misrouting* global 73 para un paquete que se encuentra en uno de los puertos de inyección de E_{origen} (condición j)). A partir de ahí, el paquete se encamina por la ruta mínima hasta $E_{destino}$ (condiciones a) a d)).

La figura 7-d muestra una ruta desde E_{origen} (O) hasta $E_{destino}$ (D) en la que se realiza un salto local (no mínimo) previo a un *misrouting* global. Estando un paquete en E_{origen}, éste sale por uno de los puertos locales 74 de E_{origen} devueltos por R_{min} utilizando el canal virtual VC0 (condición e)). A continuación, estando el paquete en el encaminador 1, motivado por ejemplo por la congestión en el enlace global marcado por R_{min}, el paquete sale por uno de sus puertos locales devueltos por $R_{non-min}$ 75, utilizando el canal virtual VC0 (condición l)). Seguidamente, en el encaminador 2 el paquete sale por uno de los puertos globales devueltos por $R_{non-min}$, utilizando el canal virtual VC1 (condición k)), 76. A partir de ahí, el paquete se encamina por la ruta mínima hasta $E_{destino}$. Nótese que los dos saltos locales 74 y 75 son equivalentes a haber hecho un *misrouting* local en el grupo de origen (previo a un salto global no mínimo), aunque la ruta haya sido generada por

R_{min} y $R_{non-min}$.

La figura 7-e muestra una ruta desde E_{origen} (O) hasta $E_{destino}$ (D). El paquete es encaminado del mismo modo que en la figura 7.d pero con un *misrouting* local adicional 77 en el grupo intermedio (condición n)).

La figura 7-f muestra una ruta desde E_{origen} (O) hasta $E_{destino}$ (D). El paquete es encaminado del mismo modo que en la figura 7.e pero con un *misrouting* local adicional 78 en el grupo destino (condición m)).

La figura 7-g muestra el encaminamiento desde E_{origen} (O) hasta $E_{destino}$ (D). El paquete es encaminado del mismo modo que en la figura 7.b pero con un *misrouting* local adicional 79 en el grupo intermedio (condición n).

La figura 8 muestra el orden seguido en la secuencia de canales virtuales atravesados, en la que los bloques grises representan los encaminadores de la ruta. Las flechas continuas representan saltos con orden ascendente de canales virtuales, mientras que las flechas con trazo discontinuo representan aquellos saltos que violan dicho orden ascendente de canales virtuales. Se muestran tres ejemplos: a) una ruta que no emplea *misrouting* local (como la ruta de la figura 7-b) que sigue un orden estrictamente ascendente; b) una ruta que emplea un *misrouting* local (como la ruta de la figura 7-g), donde los saltos correspondientes al *misrouting* local violan dicho orden estrictamente ascendente, mientras que el resto de saltos respeta tal orden; c) una ruta que emplea varios *misroutings* locales (como la ruta de la figura 7-f), donde se emplean tres saltos locales que no incrementan el índice de canal virtual, manteniéndolo o decrementándolo respecto al salto anterior, mientras que el resto de saltos sí que incrementan el índice empleado.

El criterio de elección de una ruta concreta entre las diferentes permitidas por el mecanismo admite múltiples implementaciones. De forma no limitativa, éstas pueden basarse por ejemplo, en la asignación por parte de un *allocator* unificado, en los créditos disponibles por cada puerto de salida, o en contadores de contención sobre los puertos de salida. Algunas implementaciones se muestran en los ejemplos de aplicación 1 a 3.

El mecanismo de encaminamiento $R = R_{min} \cup R_{non-min} \cup R_{local-misr}$ permite encaminar tráfico entre cualquier pareja de nodos origen y destino en ausencia de deadlock. Para demostrarlo, es suficiente demostrar que existe una subfunción de routing que es libre de deadlock, de acuerdo con William Dally y Brian Towles en *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003. En concreto, mostraremos que $R_{non-min}$ es capaz de encaminar cualquier paquete hasta el destino sin bloqueos. Para ello, se define inicialmente un invariante, que es una propiedad que permanece constante en la red con cualquier movimiento que se realice. Se define entonces el siguiente invariante:

Invariante 1: Dado un paquete en un canal virtual VC*i* de un puerto de entrada de un encaminador o router, la suma de *i* (el índice de su canal virtual) más su distancia hasta el encaminador de destino es siempre menor o igual que 4.

El invariante 1 se prueba de manera constructiva. De acuerdo a los casos e) y h) ó l) se deduce que los enlaces locales de G_{origen} solo pueden utilizarse por el canal VC0, y de acuerdo a los casos c), y g) y h) ó j) y k), resulta que los paquetes solo pueden salir del grupo de origen a través del canal VC1 de un enlace global. De manera similar, si se atraviesa un grupo intermedio, de acuerdo con d), f) ó i) y n) resulta que solo se pueden utilizar los enlaces locales de índice VC0 para el *misrouting* local y VC2 para llegar a E_{out}, y por tanto, de acuerdo con c) y f) ó i), resulta que los paquetes pueden utilizar VC1 ó VC3 para atravesar el enlace global con el que llegan al grupo de destino. Finalmente, en el grupo de destino de acuerdo con b), f) o m) los paquetes pueden utilizar cualquier canal virtual de los enlaces locales (VC0, VC2 o VC4) para llegar al encaminador de destino, en donde se consume el paquete.

Basta entonces considerar que el diámetro de la Dragonfly es 3 y las distancias hasta el destino para comprobar que el invariante es cierto. Consideremos un paquete en un puerto de entrada de un encaminador E_i en un grupo $G_i \neq G_{destino}$. Entonces, si el encaminador está directamente conectado a $G_{destino}$, a lo sumo se habrá llegado por

el canal virtual VC2, y su distancia a $E_{destino}$ será a lo sumo 2 (por los dos saltos, $global - local$, que le separan del destino). En cualquier otro encaminador de G_i la distancia hasta $E_{destino}$ será 3, pero el paquete estará en un canal virtual de entrada VC0 (si el puerto es local) o VC1 (si es global). Por otra parte, en el grupo de destino $E_{destino}$ es el único encaminador que puede alcanzarse por el canal virtual VC4; los demás encaminadores están a distancia 1 de $E_{destino}$ y se alcanzan, a lo sumo, por el canal virtual VC3, por lo que en todos los casos se mantiene el invariante 1.

A partir del invariante 1 se puede demostrar la ausencia de deadlock. Una función de routing $R: C \times N \longmapsto \mathrm{P}(C)$ es libre de deadlock si contiene una subfunción de routing, en este caso $R_{non-min}$, que es conexa y sin ciclos en el grafo extendido de dependencias. $R_{non-min}$ es conexa ya que permite encaminar cualquier paquete hasta el destino gracias al invariante 1: siempre existe una ruta ya que un paquete nunca está en un canal virtual demasiado alto para no poder llegar al destino. Por otra parte, cuando el modelo se usa bajo control de flujo *virtual cut-through*, el grafo de dependencias extendido se reduce al grafo de dependencias; y dado que las relaciones a) - d) y g) - h) (o j) - l)) definen un recorrido estrictamente creciente de los índices de los canales virtuales, entonces no existen ciclos.

Además de esta implementación para control de flujo Virtual Cut-through, bajo ciertas condiciones que se explican a continuación el mecanismo es aplicable a redes con control de flujo *wormhole*. Los saltos que violan el orden estrictamente ascendente de canales virtuales generan dependencias extendidas entre los canales involucrados. En el caso mostrado en la figura 8 b), un paquete puede quedarse parado repartido entre los canales virtuales VC0 en los encaminadores marcados como 1 y 3, y el VC1 en el encaminador 2. Por tanto, el paquete está ocupando el canal virtual VC1 de un encaminador, por lo que puede estar parando a otro paquete que se encuentre en el VC0 de otro encaminador; por ello, aparece una dependencia circular que puede bloquear la red. Para evitar dicho bloqueo, es suficiente emplear control de flujo VCT solo en los canales virtuales locales que pueden ser el destino de un *misrouting* local (VC0 y VC2 únicamente), exigiendo que haya hueco para el paquete completo antes de permitir el *misrouting* local. De esta manera, el

mecanismo propuesto puede funcionar en *wormhole en el caso general* en que se sigue un orden ascendente en los índices de canal virtual empleado, y permitiendo el *misrouting* local solo si existe hueco para el paquete completo en el buffer de destino. Evidentemente, para emplear este mecanismo es necesario que los bufferes correspondientes al *misrouting* local (al menos, VC0) tengan capacidad para el tamaño máximo del paquete. Por tanto, dicha implementación permite reducir el tamaño total de los bufferes de la red, ya que solo se precisa hueco para el paquete completo en un único buffer.

A continuación se listan algunos ejemplos de aplicación del mecanismo propuesto en diferentes implementaciones para mejor comprensión de su funcionamiento.

Ejemplo de aplicación 1: En este ejemplo de aplicación se considera una red con una topología Dragonfly en la que el arbitraje de los encaminadores está implementado mediante un *allocator* ("asignador") no separable (es decir, unificado). Esta configuración aprovecha el *misrouting* local y el encaminamiento en tránsito, pero tiene el problema de no favorecer el aprovechamiento de rutas cortas para reducir la latencia y aumentar el throughput.

Ejemplo de aplicación 2: En este ejemplo de aplicación se considera una red con topología Dragonfly en la que cada encaminador utiliza un asignador del tipo dividido (o separable). La selección de la ruta es crítica para conseguir un buen rendimiento. En general, es recomendable que las rutas sean lo más cortas posibles para reducir la carga de tráfico en la red. Sin embargo, en situaciones de saturación de los enlaces correspondientes a las rutas mínimas, resulta más interesante tomar una ruta más larga que sortee el enlace saturado. Por ello, debe existir una lógica que seleccione entre una de las rutas mínimas proporcionadas por R_{min}, o una ruta no-mínima proporcionada por $R_{non-min}$ o $R_{local-misr}$. Una implementación posible es comparar la ocupación de los bufferes: Si la ocupación de la cola mínima seleccionada supera un cierto umbral, entonces se permite la elección de una salida no-mínima cuya ocupación sea inferior a la de la cola mínima seleccionada escalada por un factor de ajuste. Si no existe ninguna cola no-mínima disponible, o bien la

ocupación de la cola mínima sleccionada es inferior al umbral fijado, entonces se utiliza la ruta mínima seleccionada de entre aquellas rutas fijadas por R_{min}. Esta decisión se repite en cada ciclo en que el paquete siga esperando, porque el asignador no le asignó la salida elegida en el ciclo de arbitraje anterior.

Ejemplo de aplicación 3: Al igual que en el ejemplo 2, se emplea una topología Dragonfly con encaminadores con asignador separable. Para seleccionar la ruta mínima o no-mínima en cada salto, se emplea la siguiente estrategia. Existe un "contador de contención" por cada puerto de salida. Cuando se calculan las rutas para un paquete de un puerto de entrada (bien al entrar en la cola, o bien cuando alcanza la cabecera de dicha cola de entrada), se incrementa el contador de contención correspondiente a la salida mínima seleccionada, de entre aquellas indicadas por R_{min}. Dicho contador se vuelve a decrementar cuando el paquete consigue avanzar y salir completamente del nodo. La lógica de encaminamiento utiliza los contadores de contención (opcionalmente, junto con el estado de ocupación de las colas) para seleccionar en cada encaminador entre una de las rutas mínimas indicadas por R_{min} o una de las rutas indicadas por $R_{non-min}$ o $R_{local-misr}$. Un modelo concreto podría utilizar un umbral estático a partir del cual, si la ruta mínima seleccionada tiene más contención, se selecciona una ruta no-mínima al azar. Otro modelo concreto podría comparar la contención del contador de la ruta mínima seleccionada con el promedio de todos los contadores del encaminador para tomar esta decisión.

REIVINDICACIONES

1. Un método de encaminamiento de paquetes en una red directa jerárquica formada por una pluralidad de encaminadores, cada uno de ellos con una pluralidad de puertos de tipo local y una pluralidad de puertos de tipo global, donde cada uno de dichos puertos comprende una pluralidad de canales virtuales, donde dichos encaminadores forman grupos, donde los diferentes encaminadores de un mismo grupo están interconectados mediante una topología conexa empleando enlaces de tipo local uniendo parejas de puertos de tipo local, y a su vez los diferentes grupos están interconectados mediante una topología conexa empleando enlaces de tipo global uniendo parejas de puertos de tipo global, estando el método configurado para emplear saltos por dichos enlaces de acuerdo a rutas mínimas y no mínimas, donde los saltos que implican rutas no mínimas pueden realizarse tanto a través de enlaces globales como locales,

estando el método caracterizado por que el número de canales virtuales necesarios en cada puerto local y global viene determinado solamente por la longitud de una ruta máxima permitida que no emplea *misrouting* de tipo local, empleando para ello un orden total en el recorrido de los canales virtuales, que se viola cuando se realiza un *misrouting* local.

2. El método de la reivindicación 1, donde la conexión entre los diferentes encaminadores de un mismo grupo se realiza de acuerdo a un grafo completo, y la conexión entre los diferentes grupos también se realiza de acuerdo a un grafo completo, y cada puerto local comprende solo 3 canales virtuales y cada puerto global comprende solo 2 canales virtuales.

3. El método de cualquiera de las reivindicaciones anteriores, donde para cada paquete situado en un canal virtual de un puerto de un encaminador, comprende:

-calcular al menos un salto de acuerdo a un encaminamiento mínimo entre el encaminador en que se encuentra dicho paquete y el encaminador al que está conectado el nodo al que dicho paquete está dirigido;

-calcular al menos un salto de acuerdo a un encaminamiento no-mínimo que comprende un *misrouting* global, a través de un grupo de encaminadores intermedio diferente del grupo al que pertenecen el encaminador de origen y el encaminador al que está conectado el nodo destino del paquete;

-calcular, si no se ha alcanzado un determinado límite de *misrouting* locales y el encaminamiento mínimo ha calculado saltos de tipo local, al menos un salto local no-mínimo diferente a dichos saltos calculados mediante el encaminamiento mínimo;

-seleccionar uno de dichos saltos en función de un determinado criterio.

4. El método de la reivindicación 3, que emplea un asignador unificado en cada encaminador para llevar a cabo la selección de uno de dichos saltos para que avance un paquete.

5. El método de la reivindicación 3, seleccionándose una ruta para cada paquete en cada ciclo de arbitraje mediante una comparación de la ocupación del canal virtual de entrada en el siguiente encaminador correspondiente a una ruta mínima seleccionada, frente a la ocupación de los canales virtuales de entrada de los encaminadores correspondientes a otras rutas.

6. El método de la reivindicación 3, seleccionándose una ruta para cada paquete en cada ciclo de arbitraje mediante una comparación de los valores de una pluralidad de contadores de contención, existiendo tantos contadores como puertos de salida tiene el encaminador, registrando dichos contadores el número de paquetes de los puertos de entrada cuya ruta mínima avanza por el puerto de salida correspondiente.

7. El método de la reivindicación 3, seleccionándose una ruta para cada paquete en cada ciclo de arbitraje de acuerdo a una combinación tanto de la información obtenida de la ocupación de los canales virtuales de entrada de los encaminadores vecinos como de una pluralidad de contadores de contención, registrando dichos contadores el número de paquetes de los puertos de entrada cuya ruta mínima avanza por el puerto de salida correspondiente.

8.- El método de cualquiera de las reivindicaciones anteriores, en el que se emplea control de flujo de agujero de gusano o *wormhole* en todos los saltos en que se respeta el orden total establecido para los canales virtuales, y control de flujo de paso a través o *virtual cut-through* en los saltos en que se hace un *misrouting* local que viola dicho orden total, permitiendo que todos los canales virtuales tengan un tamaño inferior al tamaño máximo del paquete menos los correspondientes al primer canal virtual.

9.- El método de cualquiera de las reivindicaciones anteriores, donde dicho orden total puede ser un orden ascendente o un orden descendente.

10.- Una red directa jerárquica formada por una pluralidad de encaminadores, cada uno de ellos con una pluralidad de puertos de tipo local y una pluralidad de puertos de tipo global, donde cada uno de dichos puertos comprende una pluralidad de canales virtuales, donde dichos encaminadores forman grupos, donde los diferentes encaminadores de un mismo grupo están interconectados mediante una topología conexa empleando enlaces de tipo local uniendo parejas de puertos de tipo local, y a su vez los diferentes grupos están interconectados mediante una topología conexa empleando enlaces de tipo global uniendo parejas de puertos de tipo global, estando dicha red directa jerárquica caracterizada por que comprende medios para llevar a cabo el método según cualquiera de las reivindicaciones de la 1 a la 9.

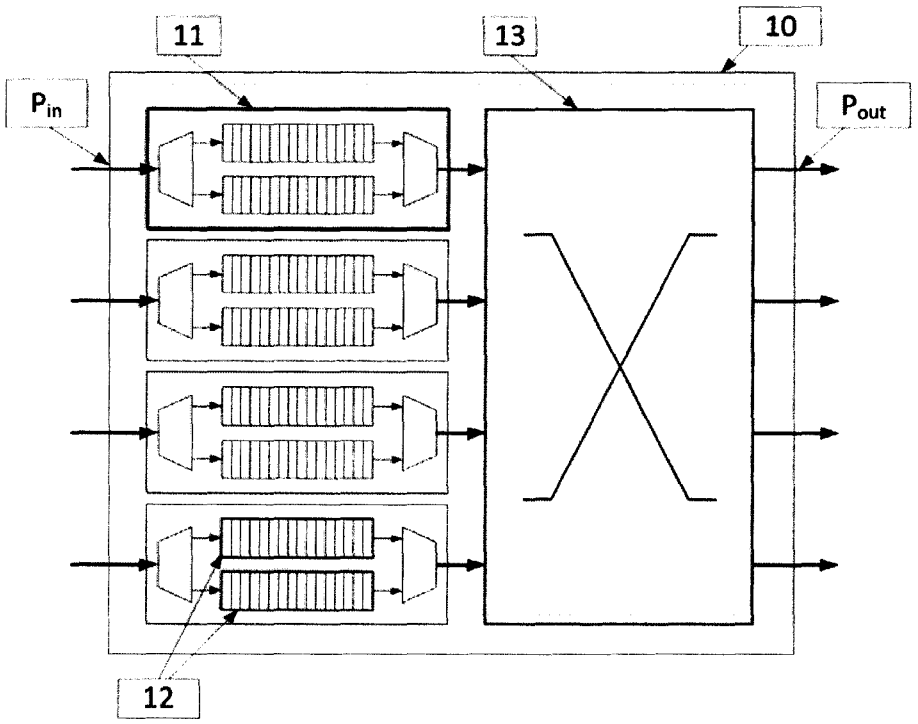

FIGURA 1

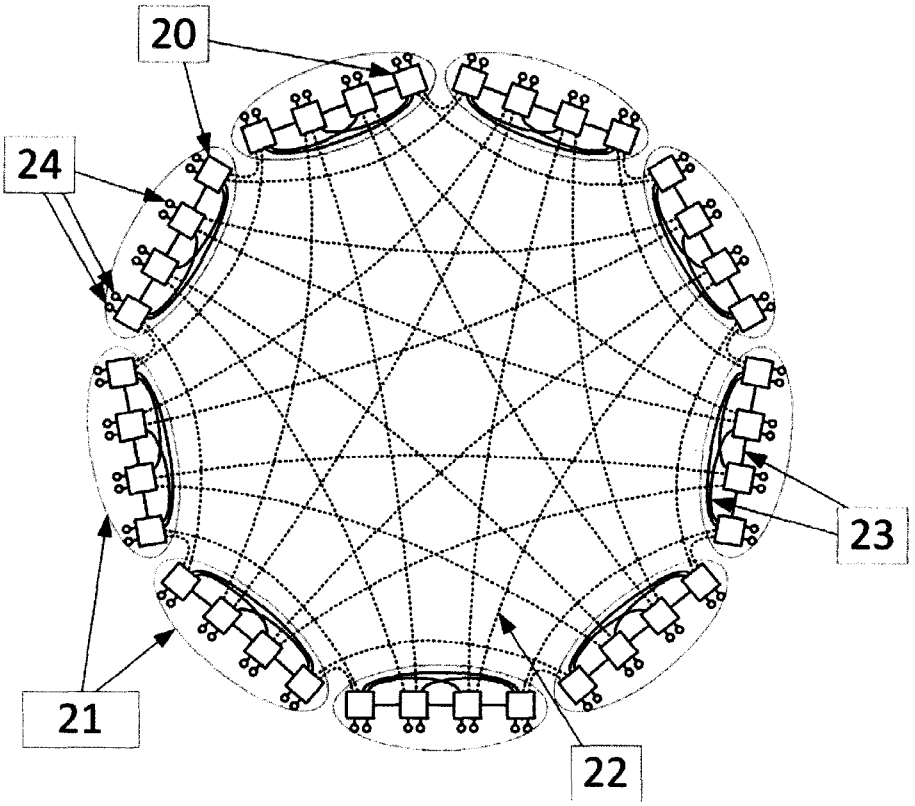

FIGURA 2

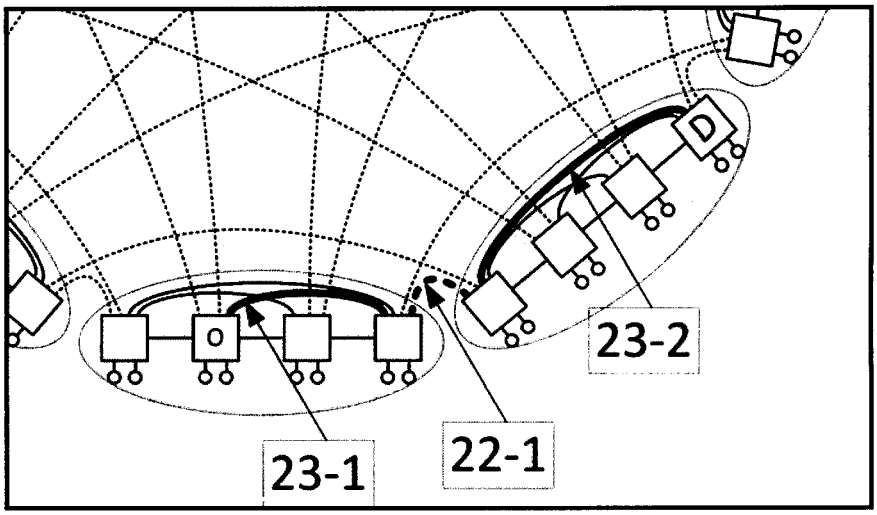

FIGURA 3

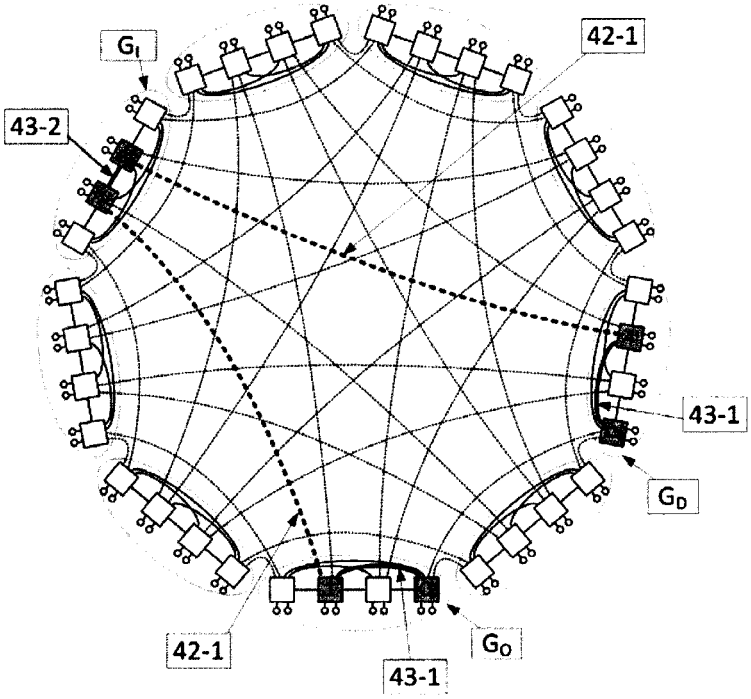

FIGURA 4

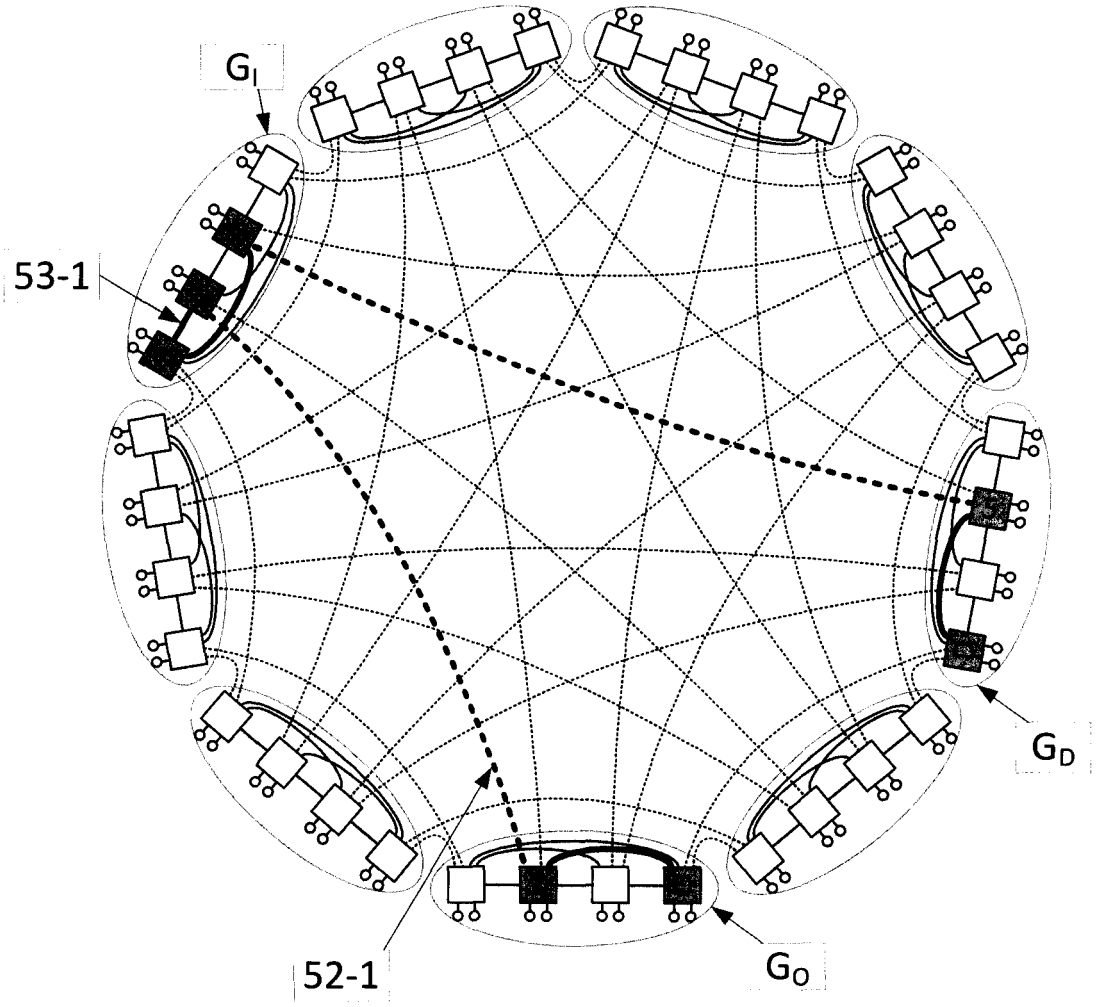

FIGURA 5

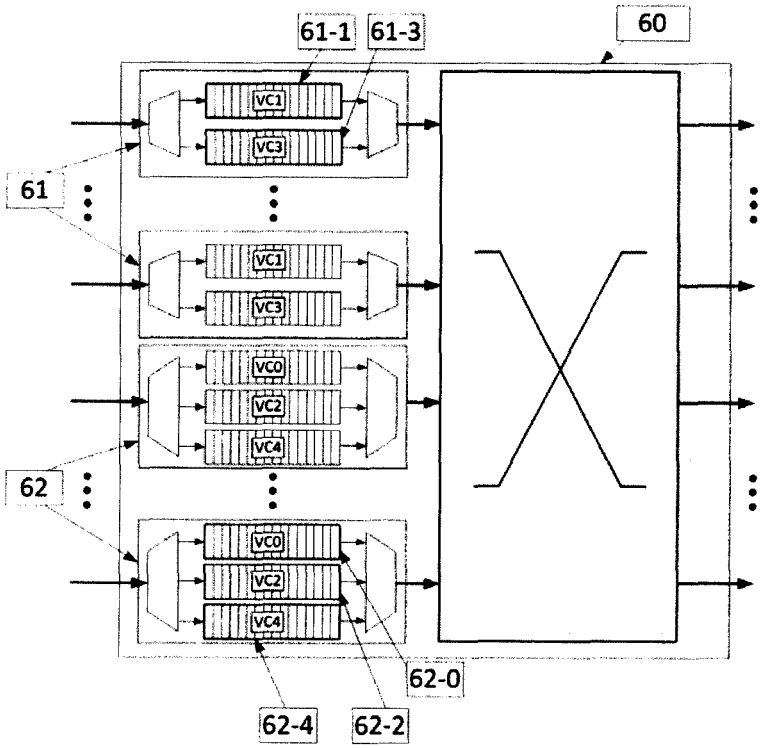
61-1
61-3
60
VC1
VC3
61
VC1
VC3
VC0
VC2
VC4
62
VC0
VC2
VC4
62-0
62-4
62-2

FIGURA 6

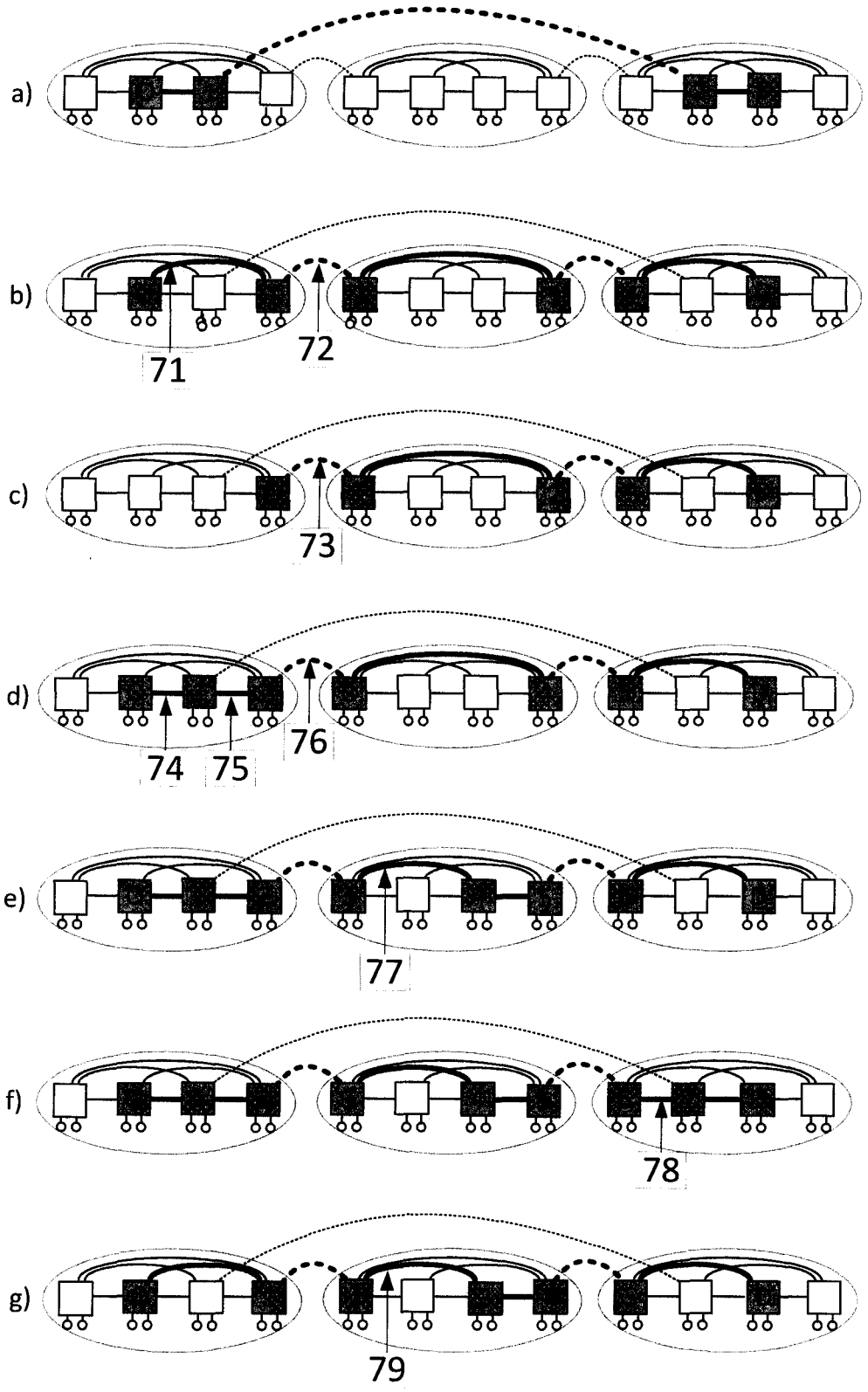

FIGURA 7

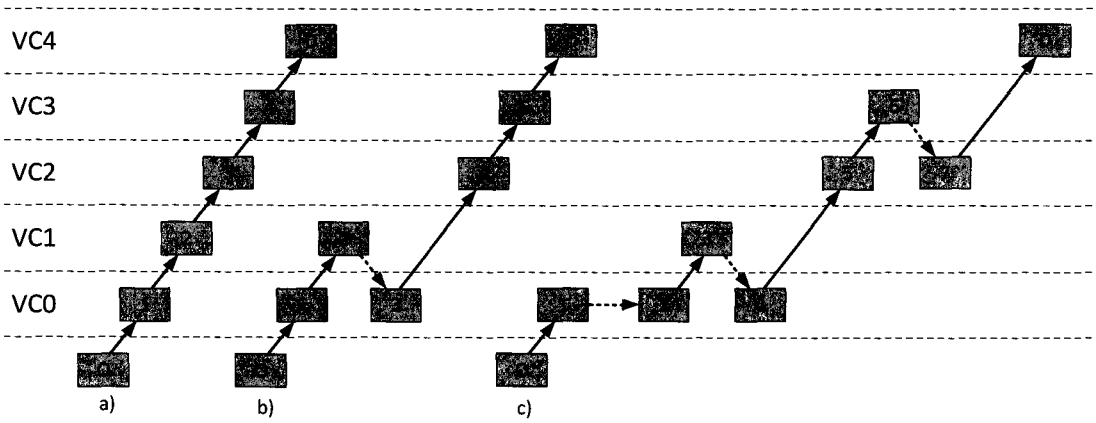

FIGURA 8

OFICINA ESPAÑOLA
DE PATENTES Y MARCAS

ESPAÑA

(21) N.º solicitud: 201200715

(22) Fecha de presentación de la solicitud: 05.07.2012

(32) Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TECNICA

(51) **Int. Cl. :** ***H04L12/715*** (2013.01)

DOCUMENTOS RELEVANTES

Categoría	(56) **Documentos citados**	**Reivindicaciones afectadas**
A	BABA ARIMILLI; RAVI ARIMILLI; VICENTE CHUNG; SCOTT CLARK; WOLFGANG DENZEL; BEN DRERUP; TORSTEN HOEFLER; JODY JOYNER; JERRY LEWIS; JIAN LI; NAN NI; RAM RAJAMONY; "The PERCS High-Performance Interconnect"; 2010 IEEE 18th Annual Symposium on High Performance Interconnects (HOTI), págs 75-82ISBN 978-1-4244-8547-5 ; ISBN 1-4244-8547-9	1
X	KIM J; DALLY W J; SCOTT S; ABTS D; Technology-Driven, "Highly-Scalable Dragonfly" ;35th International Symposium on Computer Architecture, 2008. ISCA '08; págs. 77 - 88 ISBN 978-0-7695-3174-8 ; ISBN 0-7695-3174-1	10
A	GUNTHER K. D. "Prevention of Deadlocks in Packet-Switched DataTransport Systems", IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-29, NO. 4, APRIL 1981; págs 512-524; ISSN 0090-677	1

Categoría de los documentos citados
X: de particular relevancia
Y: de particular relevancia combinado con otro/s de la misma categoría
A: refleja el estado de la técnica
O: referido a divulgación no escrita
P: publicado entre la fecha de prioridad y la de presentación de la solicitud
E: documento anterior, pero publicado después de la fecha de presentación de la solicitud

El presente informe ha sido realizado

☒ para todas las reivindicaciones ☐ para las reivindicaciones nº:

Fecha de realización del informe	**Examinador**	**Página**
17.01.2013	**M. Muñoz Sanchez**	1/4

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

H04L

Bases de datos electrónicas consultadas durante la búsqueda (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

INVENES, EPODOC, WPI, XPIEE, XPI3E, XPESP, XPESP2, XPIETF,NPL

Fecha de Realización de la Opinión Escrita: 17.01.2013

Declaración

Novedad (Art. 6.1 LP 11/1986)	Reivindicaciones	1-9	**SI**
	Reivindicaciones	10	**NO**
Actividad inventiva (Art. 8.1 LP11/1986)	Reivindicaciones	1-9	**SI**
	Reivindicaciones		**NO**

Se considera que la solicitud cumple con el requisito de aplicación industrial. Este requisito fue evaluado durante la fase de examen formal y técnico de la solicitud (Artículo 31.2 Ley 11/1986).

Base de la Opinión.-

La presente opinión se ha realizado sobre la base de la solicitud de patente tal y como se publica.

1. Documentos considerados.-

A continuación se relacionan los documentos pertenecientes al estado de la técnica tomados en consideración para la realización de esta opinión.

Documento	Número Publicación o Identificación	Fecha Publicación
D01	BABA ARIMILLI; RAVI ARIMILLI; VICENTE CHUNG; SCOTT CLARK; WOLFGANG DENZEL; BEN DRERUP; TORSTEN HOEFLER; JODY JOYNER; JERRY LEWIS; JIAN LI; NAN NI; RAM RAJAMONY; "The PERCS High-Performance Interconnect"; 2010 IEEE 18th Annual Symposium on High Performance Interconnects (HOTI), págs 75-82ISBN 978-1-4244-8547-5 ; ISBN 1-4244-8547-9	18.08.2010
D02	KIM J; DALLY W J; SCOTT S; Abts D; Technology-Driven, "Highly-Scalable Dragonfly" ;35th International Symposium on Computer Architecture, 2008. ISCA '08; págs. 77 - 88 ISBN 978-0-7695-3174-8 ; ISBN 0-7695-3174-1	21.06.2008
D03	GUNTHER K. D. "Prevention of Deadlocks in Packet-Switched DataTransport Systems", IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-29, NO. 4, APRIL 1981;págs 512-524; ISSN 0090-677	01.04.1981

2. Declaración motivada según los artículos 29.6 y 29.7 del Reglamento de ejecución de la Ley 11/1986, de 20 de marzo, de Patentes sobre la novedad y la actividad inventiva; citas y explicaciones en apoyo de esta declaración

Se considera D01 el documento más próximo del estado de la técnica al objeto de la solicitud.

Reivindicaciones independientes

Reivindicación 1: El documento D01, divulga un método para el encaminamiento de paquetes de datos por una red en la que hay conexiones (a través de puertos) entre nodos a dos niveles, unas locales dentro de un supernodo(grupo que contiene nodos interconectados a través de puertos locales) y otras globales entre supernodos. Las topologías de conexión entre nodos locales y entre supernodos son conexas. El método elige rutas mínimas y no mínimas para enrutar los paquetes de datos a través de conexiones locales y globales y dichas rutas no mínimas pueden comprender conexiones tanto locales como globales. Se sigue un orden total en el recorrido de canales virtuales basado en la ruta de máxima longitud (número de saltos) permitida evitándose así problemas de interbloqueo.

El documento D02 describe la topología general de una red dragonfly en los términos de la reivindicación 1. No se hace mención a un orden total.

El documento D03 propone un ordenamiento total de los recursos de una red que se refieren a las buffers (“colas”) de mensajes para envitar el interbloqueo. El orden propuesto es lineal y se basa en niveles. La petición en curso de un buffer para un mensaje solo puede ser para un buffer de nivel superior al del anterior utilizado en la transmisión del mensaje.
Ninguno de los documentos del estado de la técnica repara en la observación de que el interbloqueo local en redes dragonfly no tiene la importancia (por su frecuencia) necesaria como para que sea fundamental evitarlo. Esto permite usar un número mínimo de canales virtuales para las rutas no mínimas. Debido a que el documento de la solicitud define un problema técnico que no aparece en el estado de la técnica y lo resuelve se considera que el documento D01 posee actividad inventiva según el artículo 8.1 de la Ley de Patentes.

Reivindicación 10: unos medios para llevar a cabo el método de una de las reivindicaciones 1-9 podrían ser elementos de red habituales de una topología dragonfly que tienen la potencialidad de llevar a cabo dicho método si se configuran para ello. Por tanto se considera que el documento D02 afecta a la novedad de esta reivindicación según el artículo 6.1 de la Ley de Patentes.

Reivindicaciones dependientes

Reivindicaciones 2-9: las reivindicaciones presentan actividad inventiva según el artículo 8.1 de la Ley de Patentes por depender de la reivindicación 1 que también la posee.