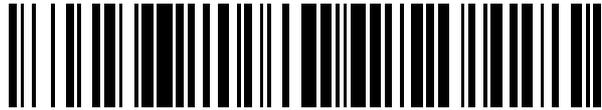


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 533 774**

51 Int. Cl.:

H04L 29/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **15.11.2005 E 05819159 (4)**

97 Fecha y número de publicación de la concesión europea: **07.01.2015 EP 1813081**

54 Título: **Procedimiento y dispositivo de ordenación de paquetes para su enrutamiento en una red con determinación implícita de los paquetes que procesar con prioridad**

30 Prioridad:

15.11.2004 FR 0412063

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

15.04.2015

73 Titular/es:

**ORANGE (100.0%)
78, rue Olivier de Serres
75015 Paris, FR**

72 Inventor/es:

**ROBERTS, JAMES;
QUESLATI, SARA y
KORTEBI, ABDESSELEM**

74 Agente/Representante:

LINAGE GONZÁLEZ, Rafael

ES 2 533 774 T3

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

DESCRIPCIÓN

Procedimiento y dispositivo de ordenación de paquetes para su enrutamiento en una red con determinación implícita de los paquetes que procesar con prioridad

5

Antecedentes de la invención

La invención se refiere al ámbito de la arquitectura de redes de telecomunicaciones.

10 La invención apunta más concretamente a un procedimiento, un dispositivo de ordenación y un enrutador de paquetes de datos de multiservicio para mejorar la calidad del servicio de enrutamiento de paquetes de datos de multiservicio incluidos en distintos flujos de esta red, de escasa complejidad.

Se aplica en particular a la red de Internet.

15

El concepto de "flujo" utilizado en este documento es por ejemplo el de la arquitectura "flow-aware" expuesta en el artículo de Bonald et al. "IP traffic and QoS control: the need for a flow-aware architecture", World Telecom Conference, París, 2002.

20 Internet tiene vocación de multiservicio y está llamado a proporcionar el soporte para una gran gama de servicios y aplicaciones. Se distinguen dos grandes clases de tráfico en esta red, el tráfico en tiempo real (o tráfico "streaming") producido en general por las aplicaciones de audio o vídeo, y el tráfico de datos (o tráfico elástico) correspondiente a la transferencia de documentos digitales. El tráfico en tiempo real posee exigencias de calidad de servicio correspondientes a la necesidad de conservación de la señal – las variaciones de velocidad que caracterizan la
 25 señal producida por la fuente deben preservarse cuando la señal cruza la red. La calidad de servicio del tráfico de datos se mide por el tiempo de transferencia de los documentos. Este tiempo o, de manera equivalente, la velocidad media alcanzada durante la transferencia, depende de toda la cadena de comunicación, desde la fuente hasta el destino. Un objetivo de calidad de servicio para una red de Internet podría ser parecer transparente al tráfico de datos sin introducir una reducción de velocidad adicional respecto de las limitaciones introducidas en otras partes
 30 (servidor, redes de acceso, equipo del usuario); en este sentido la red garantiza el mantenimiento de la velocidad de los flujos de datos.

La red de Internet pública ofrece un servicio de transporte a clientes usuarios en un contexto comercial. La cuestión de la tarificación es, por lo tanto, importante. La arquitectura de red debe permitir una rentabilidad respecto de la
 35 inversión para el operador con precios competitivos para los servicios de calidad solicitados por los usuarios.

Se conoce un procedimiento de ordenación de escasa complejidad (de orden $O(1)$, es decir independiente del número de flujos) denominado DRR (para "Deficit Round Robin"), y descrito por M. SHREEDHAR y G. VARGHESE: "Efficient fair queuing using Deficit round robin", IEEE/ACMT Transactions on Networking, Volumen 4, 3ª publicación,
 40 Junio de 1996, páginas 375-385.

El procedimiento DRR es un procedimiento de ordenación basado en el procesamiento de los flujos de manera cíclica, según el principio "round robin". Este procedimiento asocia una fila de espera a cada flujo, con cada una de estas filas de espera tratada cíclicamente, autorizada a emitir, por ciclo, paquetes hasta alcanzar una cuota
 45 (cantidad de datos, por ejemplo medida en bits).

Este procedimiento DRR permite tener un grado de equidad razonable gracias al mantenimiento de un contador de déficit que permite compensar la posible diferencia de tamaño de los paquetes de los distintos flujos.

50 El artículo TSAO et al: "Pre-order Deficit Round Robin: a new scheduling algorithm for packet-switched network" de 3 de febrero de 2001, publicado en el documento COMPUTER NETWORKS ELSEVIER SCIENCE PUBLISHERS B.V. AMSTERDAM, NL, vol. 35, nº 2-3, describe un procedimiento de ordenación de paquetes pertenecientes a distintos flujos, apuntando este procedimiento a la adaptación del algoritmo DRR para mejorar la equidad de reparto de un ancho de banda entre los distintos flujos.

55

Unas variantes del procedimiento DRR han permitido asociar un procesamiento prioritario a los paquetes de ciertos flujos.

60 Especialmente, el procedimiento DRR+, descrito en el artículo de SHREEDHAR y VARGHESE mencionado anteriormente, permite procesar con prioridad los flujos sensibles al plazo, en detrimento de los flujos "best effort". En este procedimiento, los flujos prioritarios deben cumplir un contrato, es decir, no emitir más de cierto volumen de datos durante un periodo de tiempo predeterminado.

65 El documento FR2854296 del solicitante propone un procedimiento de ordenación con diferenciación implícita de los paquetes que procesar con prioridad, que permite, en consecuencia, librarse ventajosamente de este tipo de contrato. Pero el procedimiento de ordenación descrito en este documento, del tipo "self-clock fair queuing" es del

orden $O(\log n)$, siendo n el número de flujos que tomar en cuenta, lo que puede aparecer como una limitación para su aplicación en ciertos tipos de enrutadores.

La invención pretende resolver los inconvenientes anteriores.

5

Objeto y sumario de la invención

A tal efecto, y según un primer aspecto, la invención se refiere a un procedimiento y un dispositivo de ordenación conformes al objeto de las reivindicaciones.

10

Por lo tanto, la ordenación según la invención se distingue de las del estado de la técnica conocido hasta hoy, porque realiza una distinción de la prioridad de los flujos en función de sus características intrínsecas de velocidad, no en función de un contrato o de una asignación externa de un nivel de prioridad asociado a los paquetes o a los flujos, manteniendo al mismo tiempo una complejidad independiente del número de flujos.

15

El dispositivo y el procedimiento de ordenación ordenan los paquetes en una fila de espera, según un algoritmo de reparto equitativo con prioridad.

20

Una ordenación del tipo "reparto equitativo con prioridad" permite dar prioridad a los paquetes de flujos cuya velocidad es inferior a un umbral dinámico. Este umbral corresponde a la velocidad que alcanzaría un flujo que tuviese siempre paquetes que emitir.

25

Ventajosamente, este procedimiento y este dispositivo de ordenación se basan en el procedimiento de ordenación DRR mencionado anteriormente.

30

En un modo particular de realización, el procedimiento de ordenación según la invención incluye, además, una etapa de medición de un contador de congestión que memoriza el volumen de los paquetes emitidos a partir de la fila prioritaria, con este contador destinado a ser procesado para el cálculo de estimadores de congestión utilizados para un control de admisión.

35

Estos estimadores de congestión pueden servir especialmente para un control de admisión en el núcleo de la red.

40

La invención propone asimismo un procedimiento de control de admisión que utiliza un estimador (o parámetro) de congestión constituido por un valor de carga prioritaria, correspondiente a un volumen de los paquetes prioritarios, transmitido durante un intervalo de tiempo, dividido por la duración de este intervalo, obteniéndose este volumen mediante un procedimiento de ordenación como el mencionado anteriormente.

45

En un modo particular de realización, la ordenación según la invención está asociada a un control de admisión por flujo del tipo "fair queuing".

50

De este modo, y según un segundo aspecto, la invención se refiere asimismo a un enrutador de paquetes como el reivindicado.

55

En un modo particular de realización, el módulo de admisión está, además, adaptado para enrutar directamente los paquetes pertenecientes a flujos denominados protegidos, es decir los flujos para los que al menos un paquete ha sido recibido por dicho módulo de admisión desde un intervalo de tiempo predeterminado.

60

Dicho dispositivo o procedimiento puede funcionar, sin medios de control de admisión, especialmente en el marco de una red de acceso en la que el riesgo de congestión es más controlable que en el núcleo de la red.

65

En un modo particular de realización, las etapas del procedimiento de ordenación se ejecutan en un ordenador bajo el control de instrucciones de un programa informático. Por consiguiente, la invención se refiere asimismo a un programa de ordenador almacenado en un soporte de información, incluyendo este programa instrucciones que permiten la aplicación de un procedimiento de ordenación como el mencionado anteriormente, cuando este programa se carga y ejecuta en un sistema informático.

70

La invención se refiere asimismo a un soporte de información legible por un sistema informático, eventualmente amovible en todo o parte, especialmente CD-ROM o soporte magnético, como un disco duro o un disquete, o soporte transmisible, como una señal eléctrica u óptica, caracterizado porque incluye instrucciones de un programa de ordenador que permite la aplicación de un procedimiento de ordenación como el mencionado anteriormente, cuando este programa se carga y ejecuta en un sistema informático.

75

Las ventajas particulares del dispositivo de ordenación, del enrutador, del soporte de información y del programa de ordenador son idénticas a las del procedimiento de ordenación mencionado anteriormente, por lo que no se detallarán en lo sucesivo.

80

Breve descripción de los dibujos

Otros aspectos y ventajas de la presente invención se pondrán de manifiesto mediante la lectura de la siguiente descripción de los modos particulares de realización, proporcionada únicamente a modo de ejemplo no limitativo y realizada con referencia a los dibujos adjuntos, en los que:

- 5 - la figura 1 representa, en forma de organigrama, las principales etapas de un operador de colocación en fila en un procedimiento de ordenación según la invención, en un modo particular de realización;
- 10 - la figura 2 representa, en forma de organigrama, las principales etapas de un operador de extracción de fila en un procedimiento de ordenación según la invención, en un modo particular de realización;
- las figuras 3A a 3F representan un ejemplo de funcionamiento del procedimiento de ordenación de las figuras 1 y 2;
- 15 - la figura 4 representa un dispositivo de ordenación según la invención en un modo particular de realización; y
- la figura 5 representa un dispositivo de procesamiento de paquetes según la invención en un modo particular de realización

20 Descripción detallada de un modo de realización

A continuación se va a describir un procedimiento de ordenación según la invención en un modo particular de realización. En el ejemplo descrito en el presente documento, este procedimiento de ordenación se aplica mediante un programa informático PROG_ORD.

25 Según la invención, este procedimiento de ordenación es un procedimiento del tipo DRR, modificado para permitir un procesamiento prioritario a los paquetes de los flujos cuya velocidad no supera la velocidad equitativa.

30 En el modo particular de realización descrito en el presente documento, este procedimiento de ordenación permite, además, realizar mediciones de congestión necesarias para un módulo de control de admisión descrito posteriormente con referencia a la figura 4.

Más adelante, se supondrá que el procedimiento de ordenación procesa paquetes p , pertenecientes a flujos de índice i .

35 En el modo particular de realización descrito en el presente documento, el procedimiento de ordenación según la invención utiliza filas de espera distintas $Queue_i$ para memorizar los paquetes p de cada uno de los flujos activos i .

40 Cabe señalar que el procedimiento de ordenación según la invención permite una cuota Q_i distinta por flujo i , aunque la aplicación considerada de diferenciación implícita de calidad de servicio supone que las cuotas Q_i sean idénticas e iguales al tamaño máximo de un paquete MTU (Maximum Transfert Unit).

En cualquier caso, la cuota Q_i es superior o igual al tamaño máximo MTU.

45 En el ejemplo detallado descrito en el presente documento, estas filas de espera están numeradas de 1 a n , siendo n el número máximo de flujos activos.

50 El procedimiento de ordenación descrito en el presente documento utiliza, además, una fila de espera adicional reservada para un flujo ficticio destinado a la medición de la velocidad equitativa, debiendo entenderse la velocidad equitativa como la velocidad que alcanzaría un flujo de datos que tuviese permanentemente paquetes que emitir.

Según la invención, el procedimiento de ordenación utiliza asimismo una fila de espera prioritaria PQ para entregar los paquetes determinados prioritarios en función de las características intrínsecas de la velocidad de los flujos a los que pertenecen.

55 En el resto de la descripción, se distinguirán las filas de espera "reales" (PQ, $Queue_1$, ... $Queue_n$) de la fila de espera "ficticia" para la medición de la velocidad equitativa.

60 Las filas de espera reales son todas filas de espera del tipo FIFO (first in first out) conocido por el experto en la materia.

65 El procedimiento de ordenación según la invención descrito en el presente documento utiliza asimismo una lista de los flujos activos $ActiveList$, y dos operadores, $InsertActiveList()$ y $RemoveActiveList()$, respectivamente adaptados para añadir un índice de flujo al final de la lista de los flujos activos $ActiveList$ y para quitar un índice de flujo de esta lista.

Este procedimiento de ordenación utiliza asimismo una función FreeBuffer() adaptada para liberar un espacio de memoria rechazando el paquete situado en la cabeza de la fila de espera Queue_i más larga, según el procedimiento descrito en "Stochastic Fairness Queueing", P. Mc Kenney, Internetworking: Research and Experience, Vol. 2, enero de 1991, páginas 113-131.

5 Se pueden utilizar asimismo otras estrategias de liberación de la memoria que procuran preservar la equidad entre los flujos.

10 Como en la mayoría de los procedimientos de ordenación conocidos, el procedimiento de ordenación según la invención descrito en el presente documento utiliza principalmente dos operadores estándares de manipulación de filas de espera.

- un operador InsertinQueue() de inserción de un paquete al final de la fila;

15 - un operador RemoveFromQueue() adaptado para devolver el elemento a la cabeza de la fila.

Durante una etapa preliminar de inicio, el programa de ordenación PROG_ORD inicia los contadores de déficit DC_i desde 0 para todas las filas de espera Queue_i, e inserta, en la lista de flujos activos ActiveList, el índice 0 correspondiente al flujo ficticio para la medición de la velocidad equitativa.

20 A continuación se describen, con referencia a la figura 1, las principales etapas del operador de colocación en fila Enqueue().

25 Al recibir un paquete p, el operador de colocación en fila Enqueue() aplica una primera etapa E10 durante la que prueba si la zona de memoria intermedia ("buffer zone" en inglés) en la que están memorizados los paquetes de todos los flujos activos está saturada o no.

Si la memoria intermedia está saturada, el resultado de la prueba E10 es positivo.

30 Esta prueba va entonces seguida de una etapa E20 durante la que se libera espacio de memoria rechazando el paquete situado en la cabeza de la fila de espera Queue_i más larga, recurriendo a la función FreeBuffer() mencionada anteriormente.

35 La etapa E20 de liberación de memoria va seguida de una etapa E30, durante la cual se obtiene el número i del flujo al que pertenece el paquete p.

Por el contrario, si la memoria intermedia no está saturada, el resultado de la prueba E10, anteriormente descrita, es negativo. Esta prueba va entonces seguida de la etapa E30 de obtención descrita anteriormente.

40 La etapa E30 de obtención del índice de flujo va seguida de una prueba E40, durante la cual se comprueba si el flujo i del paquete p es activo, lo que significa comprobar si el flujo i está contenido en la lista de flujos activos ActiveList.

45 Cuando no es el caso, el resultado de la prueba E40 es negativo. Esta prueba va seguida entonces de una etapa E42, durante la cual se inserta el flujo i en la lista de flujos activos ActiveList según la función InsertActiveList() mencionada anteriormente.

Esta etapa E42 de inserción va seguida de una etapa E44, durante la cual se inicia desde 0 un contador de déficit DC_i de la fila de espera Queue_i asociada al flujo activo i.

50 Este contador de déficit DC_i, es similar al del procedimiento de ordenación DRR conocido por el experto en la materia.

Según la invención, cada flujo recién recibido puede emitir, por ciclo, datos con prioridad hasta alcanzar su cuota Q_i.

55 A tal efecto, la etapa E44 de inicio va seguida de una etapa E46, durante la cual se memoriza en una variable BytesCount_i, el tamaño size(p) del paquete p, utilizándose esta variable BytesCount_i para memorizar el número de bits de los paquetes del flujo i por procesar con prioridad.

60 Esta etapa E46 de asignación va seguida de una etapa E48, durante la cual se añade el paquete p al final de la fila prioritaria PQ, utilizando la función InsertinQueue() de inserción de un paquete al final de una fila de espera.

Esta etapa E48 de colocación en fila va seguida de una etapa E50, durante la cual se inicia una variable Silence con un valor booleano FALSE para memorizar el hecho de que al menos un paquete p está listo para ser emitido.

65 Como se describe posteriormente, este indicador Silence se utiliza para medir la velocidad equitativa en caso de

ausencia de tráfico, con el estado de ausencia de tráfico representado por el hecho de que el identificador es igual al valor booleano TRUE.

Esta etapa E50 termina el procedimiento de colocación en fila del paquete p en este modo de realización.

5 Cuando el paquete p en procesamiento es el paquete de un flujo i activo, el resultado de la prueba E40 descrita anteriormente es positivo.

10 Esta prueba va entonces seguida de una etapa E52, durante la cual se añade al contenido de la variable mencionada anteriormente BytesCount_i, el tamaño del paquete p, utilizándose esta variable BytesCount_i para memorizar el número de bits de los paquetes del flujo i por procesar con prioridad.

15 Esta etapa de cúmulo va seguida de una prueba E54, durante la cual se comprueba si el número de bits del flujo i por procesar con prioridad (es decir BytesCount_i) es inferior o igual a la cuota Q_i de la fila de espera Queue_i reservada para este flujo.

20 Si tal es el caso, el resultado de la prueba E54 es positivo. Esta prueba va entonces seguida de una etapa de colocación en fila E56, similar a la etapa de colocación en fila E48, ya descrita, durante la que se añade el paquete p al final de la fila prioritaria PQ.

Por el contrario, si el número de bits BytesCount_i que procesar con prioridad para el flujo i supera la cuota Q_i para este ciclo, el resultado de la prueba E54 es negativo. Esta prueba va entonces seguida de una etapa E58 de colocación en fila del paquete p en la fila normal, es decir no prioritaria, Queue_i reservada para este flujo i.

25 Las etapas E56 y E58 de colocación en fila terminan el procedimiento de colocación en fila de este modo particular de realización.

A continuación se describen, con referencia a la figura 2, las principales etapas del operador de extracción de fila Dequeue().

30 Este operador Dequeue() de final de procesamiento de un paquete se aplica:

- al final de cada operación de envío de un paquete p; y

35 - tras la llegada de un paquete cuando el sistema está vacío, cumpliéndose esta segunda condición cuando cada fila de espera Queue_i, para n variando de 1 a n, está vacía y la fila PQ está vacía.

En el ejemplo de realización descrito en el presente documento, este operador Dequeue() se realiza en forma de un bucle sin fin constituido por las etapas F10 a F84 que se describen a continuación.

40 Durante una primera etapa F10, se comprueba si la fila prioritaria PQ está vacía. Si no es así, esta prueba F10 va seguida de una etapa F20, durante la cual se extrae el paquete p en cabeza de lista de la fila prioritaria PQ.

45 Esta operación se efectúa utilizando la función RemoveFromQueue() adaptada para devolver un elemento en cabeza de una fila de espera.

Esta etapa F20 de extracción va seguida de una etapa F22, similar a la etapa E30 ya descrita con referencia a la figura 1, de obtención del número de flujo i de este paquete p.

50 Esta etapa F22 de obtención va seguida de una etapa F24 durante la cual se emite el paquete p.

Esta etapa de envío F24 va seguida de una etapa F26, durante la cual se sustrae del contador de déficit DC_i de la fila de espera Queue_i asociada al flujo i del paquete p retirado de la fila prioritaria PQ en la etapa F20, el tamaño size(p) de este paquete p.

55 Esta etapa F26 de sustracción va seguida de una etapa F28, durante la cual se añade el tamaño del paquete p al contenido de una variable PB, utilizándose esta variable PB para memorizar el volumen del tráfico por procesar con prioridad.

60 Esta etapa de cúmulo F28 va seguida de la prueba F10 ya descrita, durante la cual se comprueba si la fila prioritaria PQ está vacía. De este modo, las etapas F10 a F28 constituyen un bucle que se prolonga mientras hay paquetes presentes en la fila prioritaria PQ.

65 Durante este bucle, los contadores de déficit DC_i asociados a los flujos i cuyos paquetes han sido emitidos con prioridad son desincrementados con el fin de impedir que cada uno de los flujos supere su cuota Q_i en el ciclo actual.

Quando la fila prioridad PQ está vacía, el resultado de la prueba F10 es positivo. Esta prueba va seguida entonces de una prueba F30, durante la cual se comprueba si al menos un flujo real está activo.

5 En el modo de realización descrito en el presente documento, esto significa comprobar que la lista de flujos activos ActiveList contiene más de un elemento, incluyendo esta lista, en cualquier caso, el flujo ficticio 0 utilizado para la medición de la velocidad equitativa.

10 Cuando no es así, el resultado de esta prueba F30 es negativo. Esta prueba va seguida entonces de una etapa F32 durante la cual se asigna a la variable Silence, descrita anteriormente en la etapa E50 de la figura 1, la constante booleana TRUE, con el fin de representar el hecho de que no hay paquete alguno que emitir, tanto en la fila prioritaria PQ como en una de sus filas de espera no prioritarias Queue_i.

15 Esta etapa de asignación F32 va seguida de la etapa F10 ya descrita, durante la cual se prueba si la fila prioritaria PQ está vacía o no.

20 Cuando al menos un flujo real está activo, el resultado de la prueba F30 es positivo. Esta prueba va seguida entonces de una etapa F34, durante la cual se obtiene el identificador del flujo i en cabeza de la lista de los flujos activos ActiveList.

Esta etapa de obtención F34 va seguida de un conjunto de etapas F40 a F44, destinado a medir el volumen de bits FB que habría podido emitir el flujo ficticio 0, sirviendo este volumen FB, como se describe posteriormente, para estimar la velocidad equitativa.

25 Más concretamente, durante la etapa F40, se prueba si el flujo obtenido en cabeza de la lista de los flujos ActiveList en la etapa anterior F34, es el flujo ficticio 0.

30 En tal caso, el resultado de la prueba F40 es positivo. Esta prueba va seguida entonces de una etapa F42, durante la cual se añade al volumen FB la cuota Q₀ asignada al flujo ficticio 0.

Esta etapa de cúmulo F42 va seguida de una etapa F44, durante la cual se retira la fila ficticia 0 de la cabeza del ciclo para añadirla inmediatamente al final del ciclo, utilizando las operaciones estándares InsertActiveList() y RemoveActiveList() mencionadas anteriormente.

35 Cuando el flujo activo i obtenido en la prueba de la lista de los flujos activos ActiveList, durante la etapa F34 es un flujo real, el resultado de la prueba F40 es negativo.

40 Esta prueba va seguida entonces de una etapa E50, durante la cual se añade al valor actual del contador de déficit DC_i para este flujo i, su cuota Q_i.

Esta etapa de cúmulo F50 va seguida de una serie de etapas F60 a F68 para emitir los paquetes del flujo i hasta alcanzar su cuota Q_i.

45 Más concretamente, durante una prueba F60, se comprueba que el contador de déficit DC_i es estrictamente positivo y que la fila de espera no prioritaria Queue_i no está vacía.

Quando esto es así, el resultado de la prueba F60 es positivo. Esta prueba va entonces seguida de una etapa F61, durante la cual se memoriza en una variable s el tamaño del paquete en cabeza de la fila de espera Queue_i.

50 Esta etapa F61 va seguida de una prueba F62 durante la cual se comprueba si dicho tamaño s es inferior o igual al contador de déficit DC_i para este flujo.

55 Cuando esto es así, el resultado de la prueba F62 es positivo. Esta prueba va seguida entonces de una etapa F64, durante la cual se envía este paquete, después de haberlo retirado de la fila de espera Queue_i, mediante el uso de la función RemoveFromQueue().

La etapa F64 de envío va seguida de una etapa F66, durante la cual se sustrae del contador de déficit DC_i el tamaño s de este paquete.

60 Esta etapa de sustracción F66 va seguida de la prueba F60 ya descrita, durante la cual se comprueba que la fila de espera Queue_i asociada al flujo i no está vacía y que el contador de déficit DC_i para esta fila es estrictamente positivo.

65 Cuando todos los paquetes del flujo han sido emitidos, hasta la cuota DC_i autorizada, el resultado de la prueba F60 o de la prueba F62 es negativo.

Se retira entonces, durante una etapa F70 el flujo i de la lista de los flujos activos `ActiveList` utilizando el operador `RemoveActiveList`, ya mencionado.

5 Esta etapa F70 de retirada va seguida de una prueba F80, durante la cual se comprueba si la fila de espera `Queuei` asociada al flujo i está vacía.

Si esto es así, se inicia el contador de déficit `DCi` asociado a este flujo i desde 0, durante una etapa F82 de inicio.

10 En caso contrario, se inserta el flujo i al final de la lista de los flujos activos `ActiveList`, utilizando la función `InsertActiveList` durante una etapa de inserción F84.

Las etapas F82 de inicio del contador de déficit `DCi` y F84 de inserción del flujo i en la lista de los flujos activos `ActiveList` van seguidas de la prueba F10 ya descrita.

15 El experto en la materia comprenderá que, en la práctica, la lista de los flujos activos `ActiveList` debe ser dimensionada para una escasa probabilidad de desbordamiento.

20 En un modo particular de realización, cuando se recibe un paquete de un nuevo flujo y que la lista de los flujos activos `ActiveList` está llena, este paquete se coloca en fila prioritaria PQ pero el flujo al que pertenece no se añade a la lista de los flujos activos `ActiveList`.

Se envía entonces este paquete (etapa F24) pero no se actualiza la información asociada al flujo al que pertenece.

25 De este modo, los paquetes a los que se aplica el procesamiento de excepción son prioritarios incluso si la velocidad de su flujo supera la velocidad equitativa actual.

Este incumplimiento no es molesto en la medida en que la probabilidad de permanecer en este estado es muy escasa para un flujo de velocidad superior a la velocidad equitativa a medida que van llegando paquetes.

30 Cabe señalar que el algoritmo de ordenación, según la invención, basado en el procedimiento DRR conocido es de complejidad $O(1)$.

35 Las figuras 3A a 3F ilustran un ejemplo de funcionamiento del procedimiento de ordenación según la invención descrito anteriormente, con referencia a las figuras 1 y 2.

Se supone en el presente documento que esta descripción empieza con la hipótesis de que tres filas de espera `Queue1`, `Queue2`, `Queue3` contienen paquetes p de los flujos respectivos flujo 1, flujo 2, flujo 3. El tamaño de estos paquetes se indica en estas figuras.

40 Se supone asimismo que, al inicio de este escenario, la fila prioritaria PQ está vacía.

Finalmente, el valor de la cuota `Qi` asignada a cada flujo i se supone igual a 1.500.

45 En estas distintas figuras, la flecha negra indica la cabeza de la lista de los flujos activos en el ciclo, siguiendo este ciclo el siguiente orden de las filas no prioritarias: `Queue1`, `Queue2`, `Queue3`, `Queue4`.

El paquete en servicio aparece en gris en cada una de las distintas etapas.

50 Inicialmente, como se ha representado en la figura 3A, el flujo que está en cabeza de lista es el flujo 1, asociado a la fila de espera `Queue1`.

Se aumenta su contador de déficit `DC1` de su cuota `Q1`, es decir 1.500.

55 El enrutador 100 puede, por lo tanto, empezar a entregar el primer paquete, en gris, de esta fila de espera `Queue1`, siendo el tamaño de este paquete en el presente caso, de 500 bits.

Se supone, con referencia a la figura 3B, que un paquete perteneciente a un nuevo flujo 4, llega durante la entrega del paquete anterior.

60 Se crea una fila `Queue4` y se asocia a este flujo 4.

Contrariamente a lo que ocurriría en el marco de una ordenación de tipo DRR clásica, este paquete queda insertado en la fila prioritaria PQ, como se ha descrito anteriormente con referencia a la etapa E48.

65 Sin embargo, se sigue con el procesamiento del flujo 1, siendo el tamaño del segundo paquete de la `Queue1`, es decir 500 bits, inferior al nuevo déficit `DC1` (resultados de las pruebas F60 y F62 positivos).

En la etapa representada en la figura 3C, el procesamiento del flujo 1 está terminado, ya que el déficit DC_1 asociado a la fila de espera Queue_1 es igual a 500, siendo este déficit inferior al tamaño 1.000 del próximo paquete (resultado de la prueba F62 negativo).

5 Según la invención, y contrariamente a lo que ocurriría en el marco de una ordenación del tipo DRR clásica, se entrega la fila prioritaria PQ (resultado de la prueba F10 negativo), conteniendo la misma el paquete perteneciente al flujo 4 cuyo déficit DC_4 es desincrementado (véase la etapa F66).

10 Durante la siguiente etapa representada en la figura 3D, habiéndose vaciado la fila prioritaria PQ (prueba F10 positiva), se procesa el primer flujo real activo (prueba F30 positiva), es decir el flujo 2, asociado a la fila de espera Queue_2.

15 Durante la siguiente etapa representada en la figura 3E, se procesa el siguiente flujo (flujo 3), ya que el tamaño del paquete que permanece en la fila de espera Queue_2 es superior a la cuota DC_2 asociada a esta fila de espera (resultado de la prueba F62 negativo).

20 La última etapa representada en la figura 3F constituye el inicio de un nuevo ciclo en el que se empieza de nuevo a entregar el flujo 1. El flujo 4 ya no está activo dado que su fila Queue_4 está vacía, por lo que ya no forma parte de ActiveList.

La figura 4 representa un dispositivo de ordenación 100 según la invención, en un modo particular de realización. Típicamente, este dispositivo de ordenación puede implementarse en un enrutador, o en un procesador o microprocesador para un enrutador.

25 De manera clásica, este dispositivo recibe paquetes p, que ordena para su emisión, o rechaza en caso de desbordamiento.

30 En esta figura 4, la referencia 34 designa la salida de los paquetes (etapas F24, F64), mientras que la referencia 32 designa el rechazo de los paquetes (etapa E20).

Este dispositivo de ordenación 100 incluye un procesador 110, una memoria muerta 120, una memoria viva 130, un reloj 140.

35 El dispositivo de ordenación 100 incluye asimismo medios materiales y de software de comunicación 150 constituidos, por ejemplo, por una tarjeta de red conectada a la red de Internet 10 y por capas de software adaptadas para aplicar el protocolo de comunicación TCP/IP.

40 Estos medios de comunicación 150 están especialmente adaptados para recibir un paquete p y para memorizarlo en una zona 131 de la memoria viva 130 y para leer un paquete p a partir de esta zona para enviarlo a la red de Internet 10.

De manera conocida, los distintos materiales del dispositivo de ordenación 100 están conectados entre sí por un sistema de bus 160.

45 Según la invención, la memoria muerta 120 incluye el programa de ordenación PROG_ORD, y la memoria viva 130 una estructura de datos 170 para la aplicación de este programa de ordenación PROG_ORD,

50 Esta estructura de datos 170 está representada en forma de una tabla en la que cada línea corresponde a una fila de espera Queue_i, asociada a un flujo i. Más concretamente, en esta tabla:

- La primera columna memoriza el índice i del flujo.

55 - La segunda columna memoriza el identificador del flujo i.

- La tercera columna memoriza la cuota Q_i de la fila Queue_i.

- La cuarta columna memoriza el contador de déficit DC_i de la fila Queue_i.

60 - La quinta columna permite gestionar el ciclo de flujos de la lista de flujos activos ActiveList, memorizando el índice de la fila de espera que sigue la fila Queue_i en esta lista. La gestión de estos punteros permite, de manera conocida, la inserción y la extracción de flujos en la lista de flujos activos ActiveList.

65 - La última columna proporciona la dirección de memorización del paquete en cabeza de la fila de espera Queue_i en la memoria viva 130. De manera conocida, los siguientes paquetes en la fila de espera Queue_i son accesibles por medio de una lista de punteros encadenados.

En un modo particular de realización, la comprobación de la actividad de un flujo i se efectúa comparando el identificador de un flujo con el contenido de la segunda columna de la estructura de datos 170, por ejemplo empleando una memoria CAM (Content Addressable Memory).

5 A continuación, se describe, con referencia a la figura 5, un enrutador 50 según la invención en un modo particular de realización.

10 Este enrutador 50 incluye un dispositivo de ordenación 100 idéntico o similar al descrito anteriormente con referencia a la figura 4.

En el modo particular de realización descrito aquí, este enrutador incluye, además, un módulo 24 de control de admisión de cada paquete p de flujos entrantes (flecha en trazo continuo referenciada 31).

15 Los paquetes p presentados en este módulo están determinados, por ejemplo, mediante funciones de enrutamiento clásicas que pueden incluir un reparto de carga, obtenido aplicando una función "hash" a un subconjunto de los campos del identificador de flujo.

20 En el ejemplo de realización descrito aquí, este módulo 24 de control de admisión efectúa dos niveles de control que pueden aplicarse de manera independiente.

25 A tal efecto, el módulo 24 de control de admisión consulta y actualiza una lista 30 de flujos denominados "protegidos", es decir los flujos admitidos por el módulo 24 de control de admisión, activos (es decir que un paquete de este flujo ha sido identificado desde un intervalo de tiempo predeterminado), y decide, sobre la base de esta lista 30 de flujos protegidos, si los paquetes de un flujo dado deben ser directamente enrutados o no, es decir sin que sea necesario comprobar las condiciones de admisión para enrutar el paquete hacia el dispositivo de ordenación.

30 En concreto, esta lista de flujos protegidos 30 es una lista de identificadores de flujos que indica, para cada uno de ellos, la hora de llegada del último paquete.

En un modo particular de realización, cada lista está asociada a una partición del espacio de los identificadores, para limitar las capacidades de cada lista y, por lo tanto, garantizar la extensibilidad.

35 En este modo de realización, se borra un flujo de la lista de identificadores 30 cuando el tiempo transcurrido desde el último paquete recibido para este flujo supera un umbral o un intervalo de temporización. La longitud de este intervalo es un parámetro del sistema, por ejemplo del orden de algunos segundos.

40 En un modo particular de realización, esta lista 30 está dimensionada de manera a limitar la probabilidad de saturación, situación en la que un flujo debería ser colocado en la lista, mientras que la misma ya está llena.

La consecuencia de dicha saturación sería que un flujo tardaría en adquirir el estado de flujo protegido. Estos paquetes serían, sin embargo, enrutados correctamente si el estado de congestión lo permite. De manera conocida, la probabilidad de saturación puede llegar a ser suficientemente escasa mediante un dimensionamiento adecuado.

45 Si el paquete pertenece a un flujo protegido, se enruta directamente (flecha 38) hacia el dispositivo de ordenación 100 correspondiente a la interfaz correcta de salida, actualizándose la hora de llegada del último paquete de este flujo protegido en la lista de identificadores 30.

50 Si el flujo no está ya protegido, se debe proceder a una toma de decisión de enrutamiento, tomándose esta decisión sobre la base de condiciones de admisión recibidas del módulo de ordenación.

Según la invención, otra función determina las condiciones de admisión a partir de medidas de congestión efectuadas por el dispositivo de ordenación 100. El suministro de estas condiciones se materializa mediante la flecha referenciada 36 en la figura 5.

55 En un modo particular de realización, se utilizan dos estimadores de congestión, la velocidad equitativa DE y la carga prioritaria CP:

60 - la velocidad equitativa es una medición de la velocidad que alcanzaría un flujo de datos que tuviera, permanentemente, paquetes por emitir; y

- la carga prioritaria es la suma de la longitud de los paquetes prioritarios transmitidos en un cierto intervalo, dividida por la duración de este intervalo.

65 Efectuando mediciones periódicas, se obtiene un recuento continuo del estado de carga del enlace controlado. El periodo de las mediciones es típicamente distinto para los dos valores de velocidad equitativa (por ejemplo, del

orden del centenar de milisegundos) y de carga prioritaria (por ejemplo, del orden de la decena de milisegundos).

En el modo particular de realización de la invención descrito aquí, estos estimadores de congestión, CP y DE se calculan mediante un proceso de medición distinto del procedimiento de ordenación descrito anteriormente a partir de los contadores de congestión PB y FB regularmente actualizados mediante el procedimiento de ordenación en las etapas F28 y F42.

Por otra parte, el proceso de medición (u otro módulo) incluye medios, no detallados aquí, adaptados para medir el tiempo total *Slicence_Time*, durante el cual la variable lógica *Silence* es igual al valor booleano *TRUE*, recordándose que esta variable *Silence* se sitúa en el valor *TRUE* en la etapa F32 y en el valor *FALSE*, en la etapa E50.

Muestreando el contador de congestión PB a intervalos regulares, el proceso de medición deduce una estimación de la carga prioritaria, como la diferencia entre los valores PB medidos al inicio y al final del intervalo de medición, dividida por la duración de este intervalo.

Siendo:

- PB(T), el valor en bits de PB en el instante T,
- (T1, T2), un intervalo de medición (en segundos), y
- C, la velocidad del enlace (en bits/segundo);

entonces, un estimador de carga prioritaria CP para este intervalo es:

$$CP(T1, T2) = (PB(T2) - PB(T1)) \times 8 / (T2 - T1) / C.$$

Según la invención, se supone, para estimar la velocidad equitativa DE, que el flujo ficticio 0 emite continuamente paquetes de tamaño fijo igual a *Q_0* (cuota asociada al flujo ficticio 0).

En un periodo en el que la fila prioritaria está constantemente ocupada, el número de bits que habría podido emitir el flujo ficticio 0 se deduce de la evolución del contador de congestión FB. Cuando la fila está vacía, el flujo ficticio 0 habría podido emitir a la velocidad C del enlace.

Conjugando la sucesión de periodos de actividad y de silencio, se deduce una estimación de la velocidad equitativa DE de la siguiente manera. Siendo:

- FB(T), el valor en bits de FB en el instante T,
- (T1, T2), un intervalo de medición, y
- *Silence_Time*, la duración total de silencio durante el intervalo (T1, T2);

se obtiene:

$$DE(T1, T2) = \max (Silence_Time \times C / (T2-T1), (FB(T2)-FB(T1)) \times 8 / (T2-T1)).$$

El experto en la materia comprenderá fácilmente que, en el cálculo de la velocidad equitativa DE, el primer término prevalece cuando la carga del enlace es escasa (ya que el flujo ficticio 0 habría utilizado toda la capacidad del enlace que queda disponible), prevaleciendo el segundo término en periodo de actividad, midiendo este término aproximadamente la velocidad alcanzada por un flujo real que posee siempre al menos un paquete en la fila de espera que tiene asociada.

El módulo de ordenación 100 determina a continuación una condición de admisión a partir de los estimadores de congestión CP (carga prioritaria) y DE (velocidad equitativa). En la práctica, esta determinación puede efectuarse utilizando una función *Admit*(CP, DE) cuyo resultado es igual a 1 (respectivamente 0), cuando los paquetes p de nuevos flujos que no tienen aún el estado de flujo protegido, deben ser enrutados (respectivamente rechazados), proporcionándose el resultado de esta función de admisión (flecha 36) mediante el módulo de ordenación 100 al módulo de enrutamiento 24.

Cuando las condiciones de admisión son favorables, el nuevo flujo cuyo paquete es enrutado, se inserta en la lista de los flujos protegidos y adquiere de este modo el estado de flujo protegido.

En esta figura, la flecha de trazo continuo 39 representa el rechazo de los primeros paquetes de los nuevos flujos por el módulo de enrutamiento 24.

En esta figura, la flecha de trazo continuo 38 representa el enrutamiento de los paquetes de los flujos protegidos hacia el módulo de ordenación 100.

- 5 Las condiciones aplicadas pueden depender de un atributo particular del paquete cuyo valor del campo "clase de tráfico" en IPv6 o el campo ToS en IPv4 o de las direcciones IP fuente y destino.

El control de admisión permite garantizar la transparencia de los flujos admitidos: conservación de la señal para los flujos en tiempo real, conservación de la velocidad para los flujos de datos.

- 10 En realidad, esta transparencia solo se ofrece a los flujos cuya velocidad de pico (determinada por las limitaciones externas) permanece inferior a cierto umbral.

- 15 Para elegir umbrales de admisión relativos a la velocidad equitativa y a la carga prioritaria que permiten realizar un control de admisión que permite garantizar la transparencia de los flujos, el experto en la materia podrá referirse al documento FR2854296 del solicitante anteriormente mencionado.

En la descripción anterior, el proceso de medición y el procedimiento de ordenación son distintos uno de otro. En una variante, el proceso de medición puede integrarse en el procedimiento de ordenación.

REIVINDICACIONES

- 1.- Procedimiento de ordenación de paquetes pertenecientes a distintos flujos (1), una fila de espera no prioritaria denominada Queue_i y un contador de déficit denominado DC_i asociados a cada flujo i, así como una fila prioritaria denominada PQ, incluyendo este procedimiento, tras la recepción de un paquete:
- la determinación del número de flujo i al que pertenece el paquete;
 - una etapa (E40, E54) de determinación de la prioridad de dicho paquete, denominado p, siendo un paquete prioritario por una parte:
 - el primer paquete de un flujo no presente en una lista denominada ActiveList de flujos activos, insertándose el índice de este flujo al final de dicha ActiveList; y por otra parte
 - los siguientes paquetes de este flujo recibidos durante un ciclo, mientras el número de bits denominado ByteCount_i de este flujo recibidos en dicho ciclo es inferior a una cuota denominada Q_i asociada al flujo para dicho ciclo; y
 - si dicho paquete se determina no prioritario, una etapa (E58) de colocación en fila de dicho paquete en la fila de espera no prioritaria denominada Queue_i asociada al flujo de dicho p;
 - si dicho paquete se determina prioritario, una etapa (E48, E56) de colocación en fila de dicho p al final de dicha fila PQ;
- con dichas filas Queue_i respectivamente asociadas a los flujos de dicha lista ActiveList procesadas cíclicamente según su orden en dicha ActiveList, con la fila PQ procesada con prioridad respecto de cada una de las filas Queue_i en dicho ciclo;
- consistiendo el procesamiento de dicha fila PQ en emitir (F24) todos los paquetes contenidos en dicha PQ y en disminuir (F26), para cada uno de dichos paquetes emitidos, dicho DC_i asociado al flujo i de dicho paquete del tamaño de dicho paquete;
- consistiendo el procesamiento de una fila Queue_i en:
- aumentar (F50) dicho DC_i asociado a dicha fila Queue_i de dicha cuota; y
 - emitir (F64) paquetes contenidos en dicha fila Queue_i, mientras dicho DC_i asociado a dicha fila Queue_i es estrictamente positivo (F60), emitiéndose un paquete si su tamaño es inferior a dicho DC_i (F62), disminuyéndose dicho DC_i (F66) del tamaño de dicho paquete emitido;
 - retirar el índice de dicho flujo asociado a esta fila Queue_i de dicha lista ActiveList y reiniciar dicho DC_i si dicha fila Queue_i está vacía tras dicho procesamiento (F82); y
 - desplazar el índice de dicho flujo al final de dicha lista ActiveList en el caso contrario.
- 2.- Procedimiento de ordenación según la reivindicación 1, caracterizado porque incluye además, una etapa (F28, F42) de medición de un contador (PB) de congestión que memoriza el volumen de los paquetes emitidos a partir de dicha fila PQ, destinado a ser procesado para el cálculo de un estimador (CP) de congestión utilizado para un control de admisión.
- 3.- Procedimiento de control de admisión que utiliza un estimador de congestión constituido por:
- un valor (CP) de carga prioritaria, correspondiente a un volumen (PB) de paquetes prioritarios, transmitidos durante un intervalo de tiempo, dividido por la duración de este intervalo, obteniéndose dicho volumen (PB) mediante un procedimiento de ordenación según la reivindicación 2.
- 4.- Dispositivo de ordenación de paquetes p pertenecientes a distintos flujos, una fila de espera no prioritaria denominada Queue_i y un contador de déficit denominado DC_i asociados a cada flujo i, así como una fila prioritaria denominada PQ, incluyendo este dispositivo, al recibir un paquete:
- medios para determinar el número de flujo i al que pertenece el paquete;
 - medios de determinación de la prioridad de dicho paquete, denominado p, siendo un paquete prioritario por una parte:

- el primer paquete de un flujo no presente en una lista denominada ActiveList de flujos activos, con el índice de este flujo insertado al final de dicha ActiveList; y por otra parte
- 5 - los siguientes paquetes de este flujo recibidos durante un ciclo mientras el número de bits denominado BytesCount_i de este flujo recibidos en dicho ciclo es inferior a una cuota denominada Q_i asociada al flujo para dicho ciclo; y
 - si dicho paquete se determina no prioritario, medios de colocación en fila de dicho paquete en la fila denominada Queue_i asociada al flujo de dicho p; y
 - 10 • si dicho paquete se determina prioritario, medios de colocación en fila de dicho p al final de dicha fila PQ;
 - medios para tratar cíclicamente dichas filas Queue_i respectivamente asociadas a los flujos de dicha ActiveList según su orden en dicha ActiveList, siendo la fila PQ procesada con prioridad respecto de cada una de las filas Queue_i en dicho ciclo, incluyendo dichos medios de procesamiento:
 - 15 - medios de emisión de todos los paquetes en dicha fila PQ, con dicho DC_i asociado al flujo i de cada uno de dichos paquetes emitidos disminuido del tamaño de dicho paquete;
 - 20 - medios de emisión de los paquetes contenidos en dicha fila Queue_i, mientras dicho DC_i asociado a dicha fila Queue_i es estrictamente positivo, con dicho contador DC_i aumentado de la cuota Q_i asociada a dicha fila Queue_i, emitiéndose un paquete si su tamaño es inferior a dicho DC_i, disminuyéndose dicho contador de déficit del tamaño de dicho paquete emitido, siendo dichos medios:
 - 25 - capaces de retirar el índice de dicho flujo asociado a esta fila Queue_i de dicha ActiveList y de reiniciar dicho DC_i si dicha fila Queue_i está vacía tras dicho procesamiento; o
 - medios para desplazar el índice de dicho flujo al final de dicha ActiveList en el caso contrario.
- 30 5.- Dispositivo de ordenación según la reivindicación 4, caracterizado porque incluye, además, medios de medición de un contador (PB) de congestión que memoriza el volumen de los paquetes emitidos a partir de dicha fila PQ, destinado a ser procesado para el cálculo de un estimador de congestión utilizado para un control de admisión.
- 35 6.- Enrutador de paquetes caracterizado porque incluye un dispositivo de ordenación (100) según la reivindicación 5, y un módulo (24) de control de admisión que utiliza un estimador de congestión constituido por un valor (CP) de carga prioritaria, correspondiente a un volumen (PB) de paquetes prioritarios, transmitidos durante un intervalo de tiempo, dividido por la duración de este intervalo, obteniéndose este volumen (PB) con la ayuda de los medios de medición de un contador de congestión del dispositivo de ordenación.
- 40 7.- Enrutador según la reivindicación 6, caracterizado porque dicho módulo de admisión está, además, adaptado para enrutar directamente los paquetes pertenecientes a flujos denominados protegidos, es decir los flujos para los que se ha recibido al menos un paquete por parte de dicho módulo de admisión desde un intervalo de tiempo predeterminado.
- 45 8.- Programa de ordenador almacenado en un soporte de información, incluyendo dicho programa instrucciones que permiten la aplicación de un procedimiento de ordenación según una cualquiera de las reivindicaciones 1 a 2, cuando este programa se carga y ejecuta en un sistema informático.
- 50 9.- Soporte de información legible por un sistema informático, eventualmente amovible en todo o parte, especialmente CD-ROM o soporte magnético, como un disco duro o un disquete, o soporte transmisible, como una señal eléctrica u óptica, caracterizado porque incluye instrucciones de un programa de ordenador que permite la aplicación de un procedimiento de ordenación según una cualquiera de las reivindicaciones 1 a 2, cuando este programa se carga y ejecuta en un sistema informático.

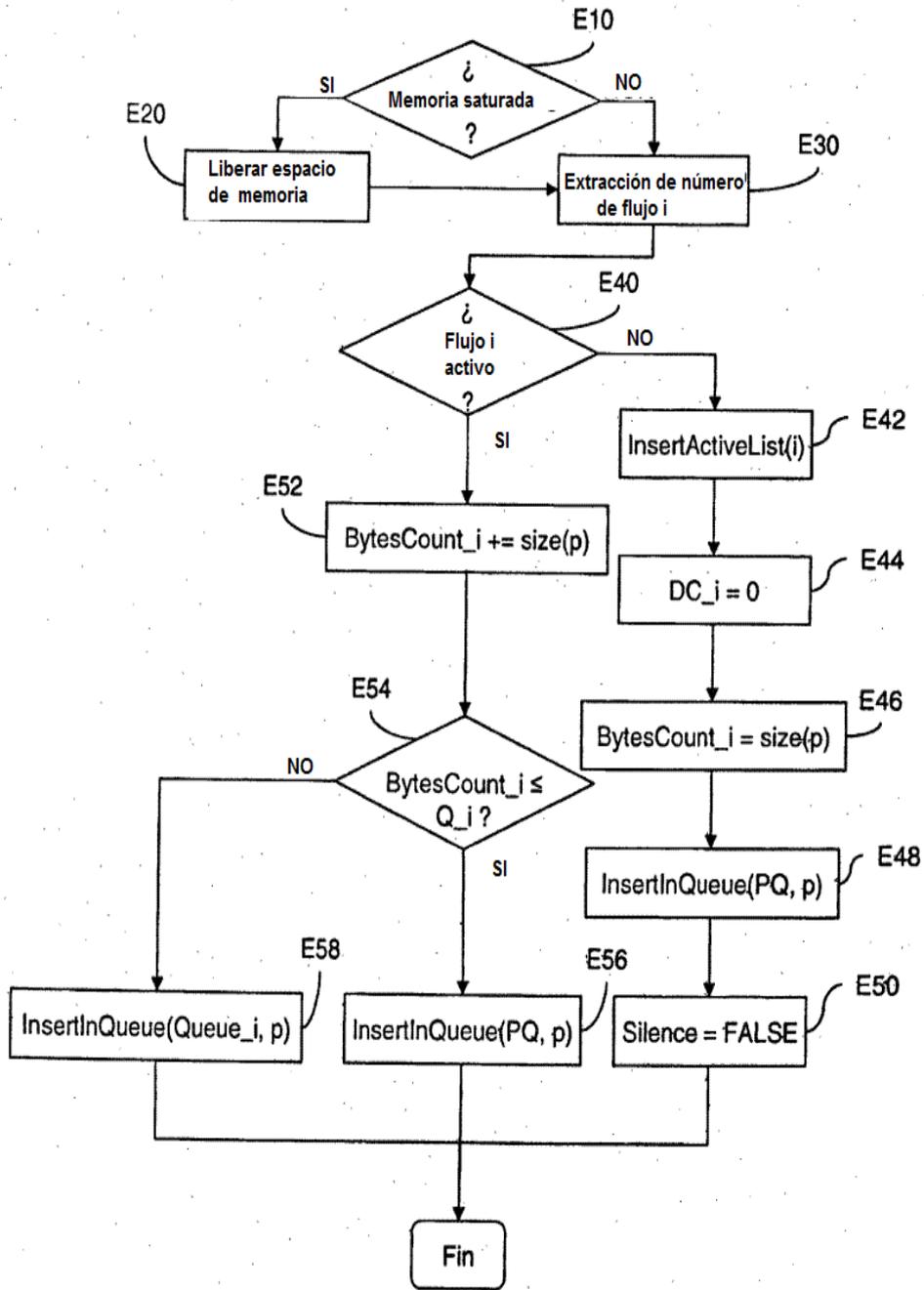


FIG. 1

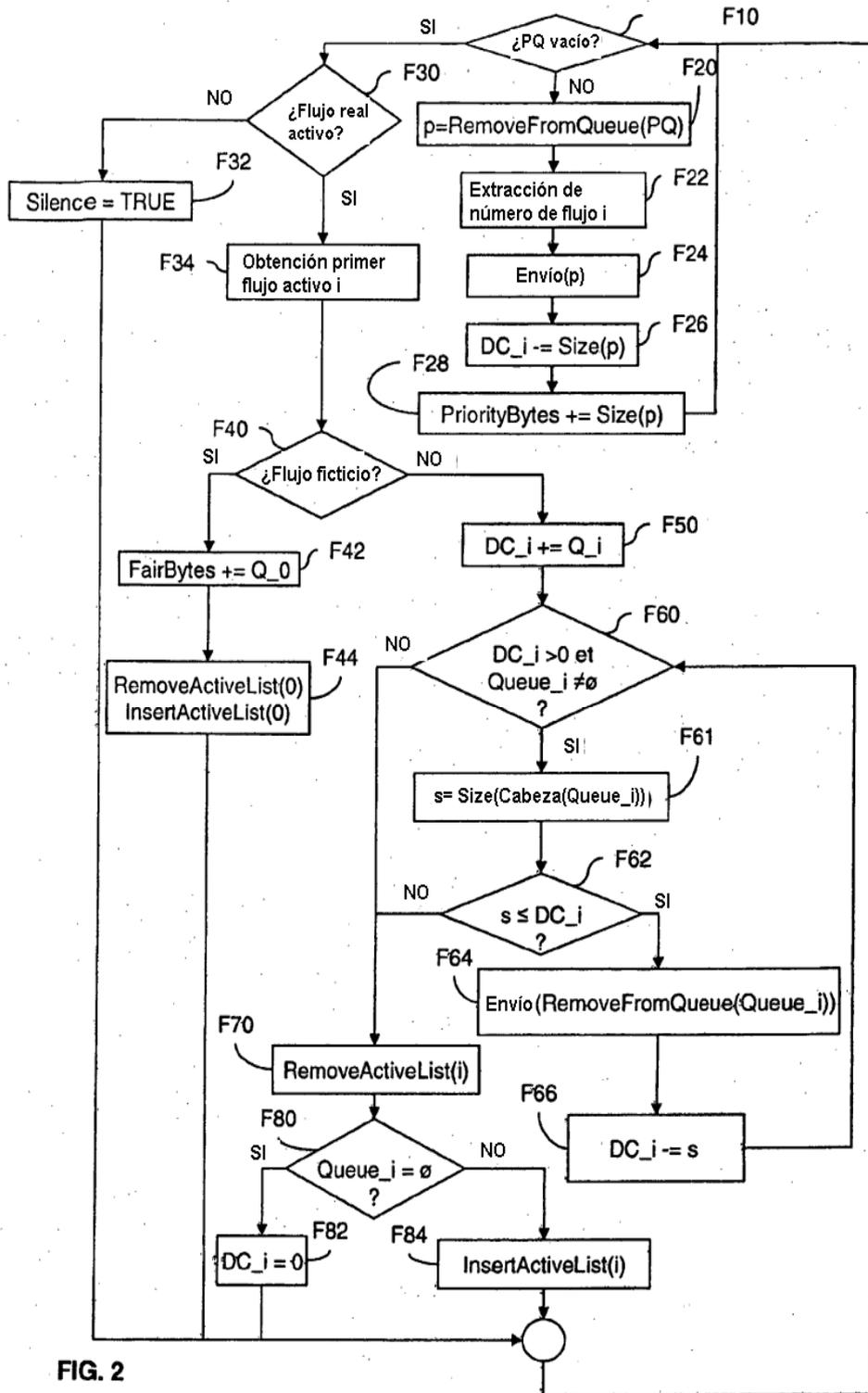


FIG. 2

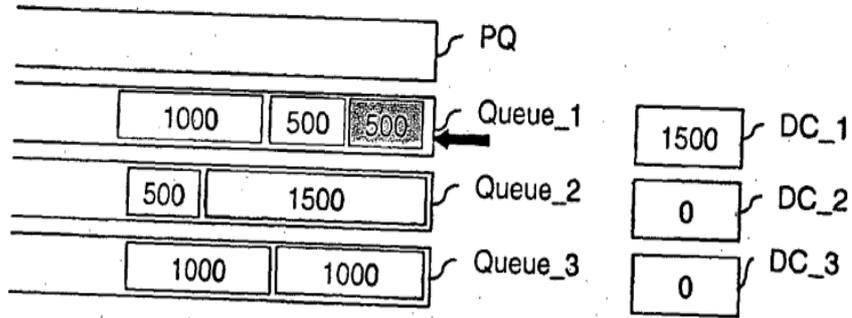


FIG. 3A

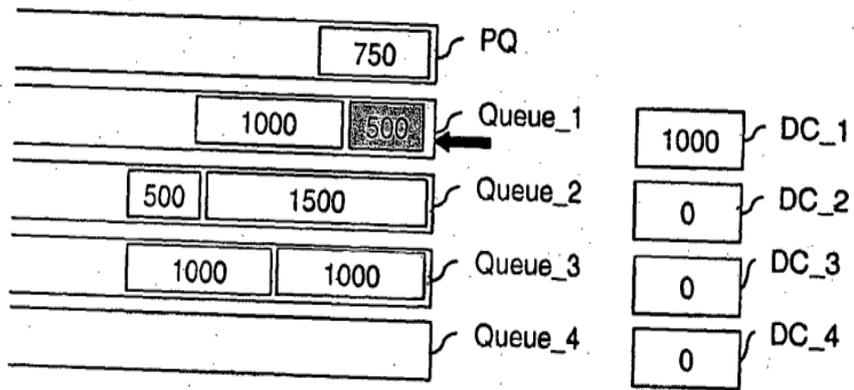


FIG. 3B

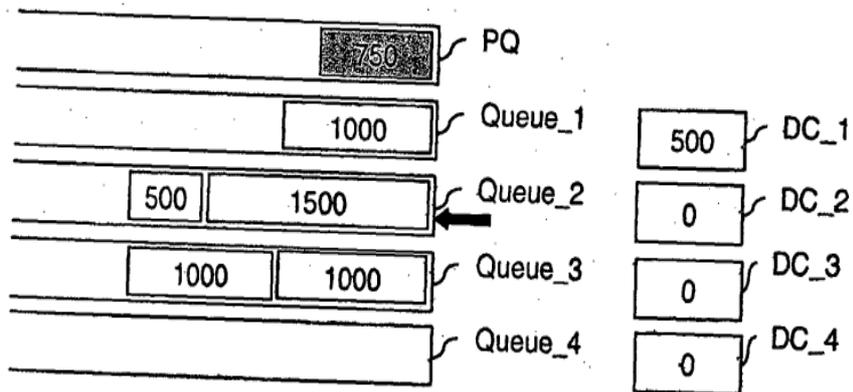


FIG. 3C

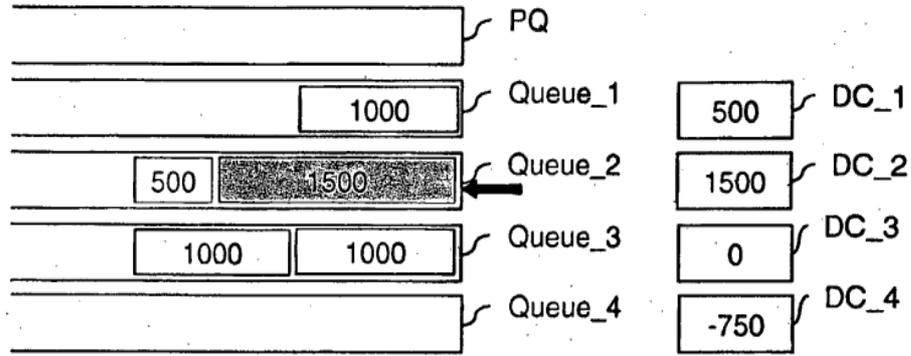


FIG. 3D

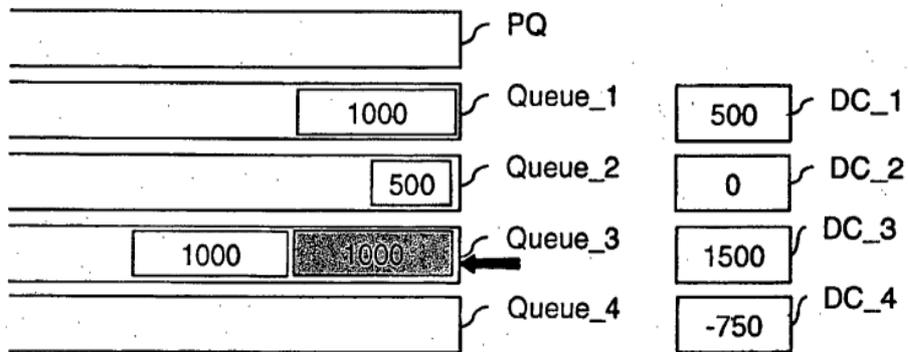


FIG. 3E

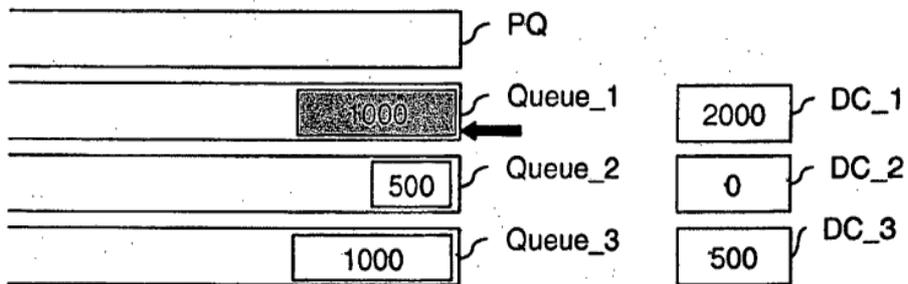
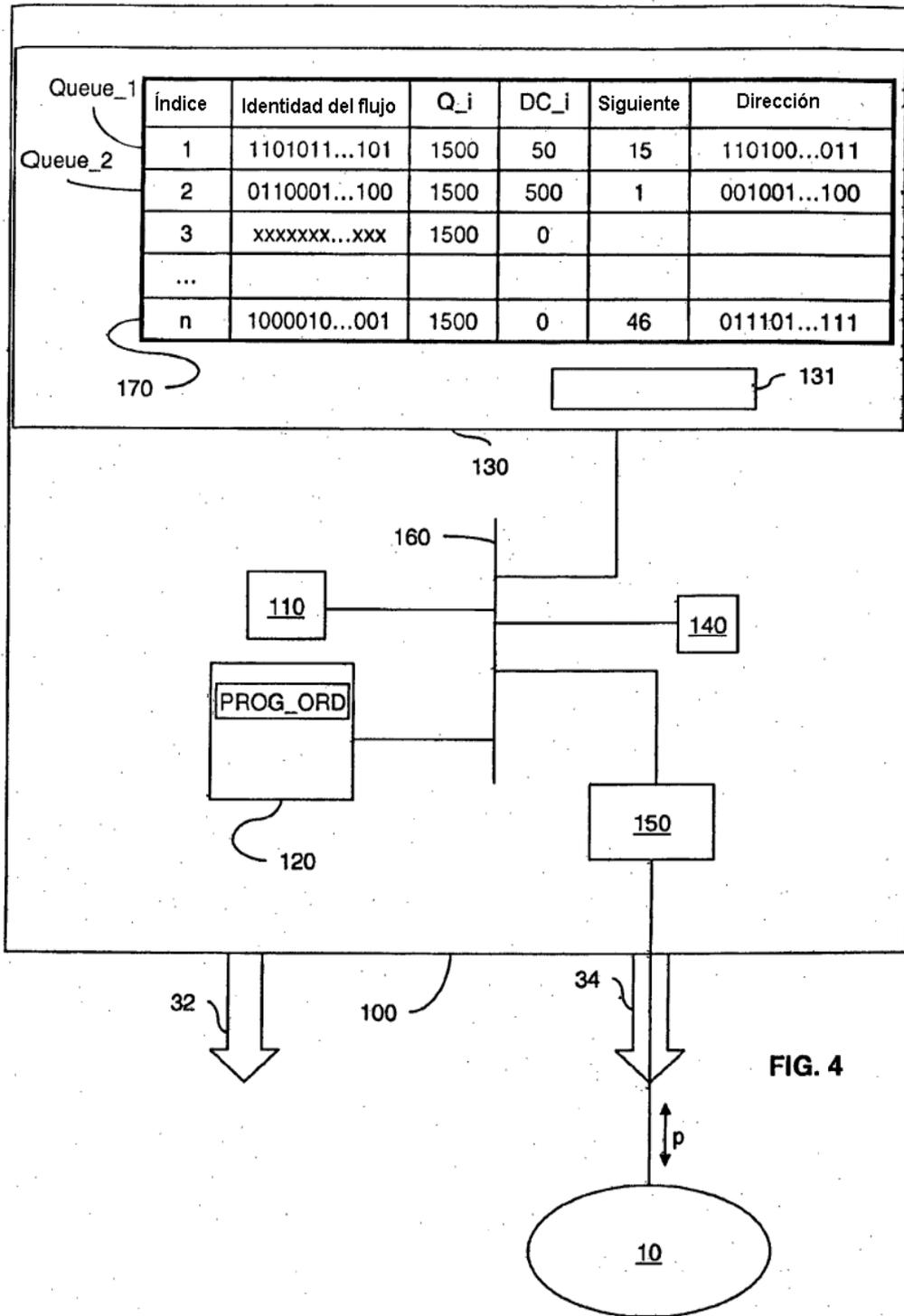


FIG. 3F



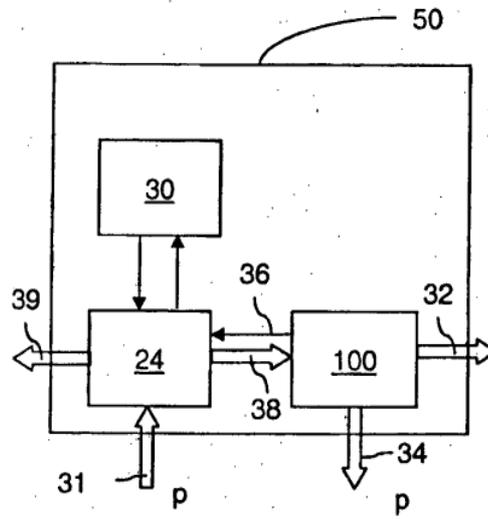


FIG. 5