

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 799 698**

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 15150092 (3)**

97 Fecha y número de publicación de la concesión europea: **01.04.2020** **EP 2940960**

54 Título: **Procedimiento y dispositivo de ordenación de paquetes para su enrutado en una red con determinación implícita de los paquetes a tratar con prioridad**

30 Prioridad:

15.11.2004 FR 0412063

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

21.12.2020

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:

ISERN JARA, Jorge

ES 2 799 698 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 enrutado en una red con determinación implícita de los paquetes a tratar con prioridad

5 Antecedentes de la invención

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

10 Se dirige más particularmente, a un procedimiento, a un dispositivo de ordenación y a un enrutador de paquetes de datos multiservicio que se dirigen a mejorar la calidad del servicio de encaminamiento de paquetes de datos multiservicio incluidos en diferentes flujos de esta red, de reducida complejidad.

15 Se aplica en particular a la red de Internet.

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

20 La Internet tiene vocación multiservicio y está llamada a suministrar 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 de "streaming") producido en general por las aplicaciones de audio o vídeo y el tráfico de datos (o tráfico elástico) que corresponde a la transferencia de documentos digitales. El tráfico en tiempo real tiene unas exigencias de calidad de servicio que corresponden a la necesidad de conservación de la señal —las variaciones de velocidad que caracterizan la señal

25 producida por la fuente deben preservarse cuando la señal atraviesa 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 realizada durante la transferencia, depende de toda la cadena de comunicación desde el origen al destino. Un objetivo de calidad de servicio para una red de Internet podría ser parecer transparente al tráfico de datos no introduciendo reducción de velocidad suplementaria con relación a las limitaciones introducidas en otros sitios

30 (servidor, redes de acceso, equipo del usuario); en este sentido la red asegura la conservación de la velocidad de los flujos de datos.

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

35 Se conoce un método de ordenación de reducida complejidad (del orden $O(1)$, es decir independiente del número de flujos) bautizado DRR (por "Deficit Round Robin") y descrito por M. SHREEDHAR y G. VARGHESE: "Efficient fair queuing using Deficit round robin", IEEE/ACM Transactions on Networking, Volumen: 4, Edición: 3, junio de 1996, páginas 375-385.

40 El método DRR es un método de ordenación basado en el tratamiento de los flujos de manera cíclica, según el principio "round robin". Este método asocia una cola de espera a cada flujo, tratados cíclicamente cada uno de estas colas de espera, estando autorizado para emitir, por ciclo, unos paquetes hasta el nivel de una cuota (cantidad de datos, por ejemplo medida en octetos).

Este método DRR permite tener un grado de equidad razonable gracias al mantenimiento de un contador de déficit que permite compensar la diferencia eventual de tamaño de los paquetes de los diferentes flujos.

50 Unas variantes del método DRR han permitido asociar un tratamiento prioritario a los paquetes de ciertos flujos.

En particular, el método DRR+, descrito en el artículo antes citado, permite tratar con prioridad los flujos sensibles al retardo, en detrimento de los flujos de "best effort". En este método, los flujos prioritarios deben respetar un contrato, a saber el de no emitir más que un cierto volumen de datos durante un periodo de tiempo predeterminado.

55 El artículo "Pre-Order Decifit Round Robin: a new scheduling algorithm for packet-switched networks", COMPUTER NETWORKS, ELSEVIER SCIENCE PUBLISHERS B.V., AMSTERDAM NL, vol. 35, nº 2-3, febrero de 2001, propone igualmente un procedimiento de ordenación de paquetes que pertenecen a diferentes flujos, que se basa en este mismo método de DRR.

60 El documento FR2854296 del presente solicitante propone un método de ordenación con diferenciación implícita de los paquetes a tratar con prioridad y que permite en consecuencia superar ventajosamente este tipo de contrato. Pero el método de ordenación descrito en este documento, de tipo "self-clock fair queuing" es de orden $O(\log n)$, siendo n el número de flujos a tener en cuenta, lo que puede aparecer como una limitación para su implementación en ciertos

65 tipos de enrutadores.

La invención se dirige a resolver los inconvenientes anteriores.

Objeto y sumario de la invención

5 Con este fin, y según un primer aspecto, se refiere a un procedimiento de ordenación de paquetes que pertenecen a diferentes flujos según la reivindicación 1. Correlativamente, la invención se refiere también a un dispositivo de ordenación de paquetes que pertenecen a diferentes flujos según la reivindicación 4. Las reivindicaciones dependientes describen modos de realización preferidos.

10 De este modo, la ordenación según la invención se distingue de las del estado de la técnica conocida hasta el momento, en que realiza una distinción de la prioridad de los flujos en función de sus características intrínsecas de velocidad y no en función de un contrato o de una atribución externa de un nivel de prioridad asociado a los paquetes o a los flujos, todo mientras se mantiene una complejidad independiente del número de flujos.

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

Una ordenación de 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 produciría un flujo si tuviera siempre paquetes a emitir.

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

25 En un modo particular de realización, el procedimiento de ordenación según la invención incluye además una etapa de medida de contadores de congestión destinados a ser tratados para el cálculo de estimadores de congestión utilizados para un control de admisión. Estos estimadores de congestión pueden servir principalmente para un control de admisión en el núcleo de la red.

30 En un modo particular de realización, estos estimadores (o parámetros) de congestión están constituidos por:

- un valor de velocidad equitativa, que es una medida de la velocidad que produciría un flujo de datos que tuviera siempre paquetes a emitir; y

35 - un valor de carga prioritaria, que corresponde al volumen de dichos paquetes prioritarios, transmitidos durante un intervalo de tiempo, dividido por la duración de este intervalo.

Según un segundo aspecto, la invención se dirige también a un enrutador de paquetes que incluye un dispositivo de ordenación tal como el mencionado anteriormente y a un módulo de control de admisión de los paquetes en función de los parámetros de congestión medidos por el dispositivo de ordenación.

40 En un modo particular de realización, el módulo de admisión está adaptado además para encaminar directamente los paquetes que pertenecen a unos flujos llamados protegidos, a saber los flujos para los que se ha recibido al menos un paquete por dicho módulo de admisión después de un intervalo de tiempo predeterminado.

45 Dicho dispositivo o procedimiento puede funcionar, sin medios de control de admisión, principalmente 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.

50 En un modo particular de realización, las etapas del procedimiento de ordenación se ejecutan por un ordenador bajo el control de instrucciones de un programa informático. Por consiguiente, la invención se refiere igualmente a un programa informático almacenado en un soporte de informaciones, incluyendo este programa instrucciones que permiten la implementación de un procedimiento de ordenación tal como se ha mencionado anteriormente, cuando este programa es cargado y ejecutado por un sistema informático.

55 La invención se dirige también a un soporte de informaciones legible por un sistema informático, eventualmente total o parcialmente extraíble, principalmente un CD-ROM o soporte magnético, tal como un disco duro o disquete, o un soporte transmisible, tal como una señal eléctrica u óptica, caracterizado por que incluye instrucciones de un programa informático que permiten la implementación de un procedimiento de ordenación tal como se ha mencionado anteriormente, cuando este programa es cargado y ejecutado por un sistema informático.

60 Las ventajas particulares del dispositivo de ordenación, del enrutador, del soporte de informaciones y del programa informático, al ser idénticas a las del procedimiento de ordenación mencionado anteriormente, no se recordarán aquí.

Breve descripción de los dibujos

65 surgirán más claramente otros aspectos y ventajas de la presente invención con la lectura de la descripción de los

modos particulares de realización que sigue, haciéndose esta descripción únicamente a título 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 puesta en cola en un procedimiento de ordenación de acuerdo con la invención, en un modo particular de realización;
- la figura 2 representa, en forma de organigrama, las principales etapas de un operador de extracción de cola en un procedimiento de ordenación de acuerdo con la invención, en un modo particular de realización;
- 10 - las figuras 3A a 3F representan un ejemplo de funcionamiento del procedimiento de ordenación de las figuras 1 y 2;
- la figura 4 representa un dispositivo de ordenación de acuerdo con la invención en un modo particular de realización; y
- 15 - la figura 5 representa un dispositivo de tratamiento de paquetes de acuerdo con la invención en un modo particular de realización.

Descripción detallada de un modo de realización

20 Se va a describir ahora un procedimiento de ordenación de acuerdo con la invención en un modo particular de realización. En el ejemplo descrito en el presente documento, este procedimiento de ordenación se implementa mediante un programa informático PROG_ORD.

25 De acuerdo con la invención, este procedimiento de ordenación es un procedimiento de tipo DRR, modificado con el fin de permitir un tratamiento prioritario a los paquetes de flujos cuya velocidad no sobrepasa la velocidad equitativa.

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

Se supondrá en lo que sigue que el procedimiento de ordenación trata unos paquetes p , que pertenecen a unos flujos de índice i .

35 En el modo particular de realización descrito en este caso, el procedimiento de ordenación según la invención utiliza unas colas de espera diferentes $Queue_i$ para memorizar los paquetes p de cada uno de los flujos activos i .

40 Se observará que el procedimiento de ordenación según la invención permite una cuota Q_i diferente por flujo i , aunque la aplicación propuesta 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 Transfer Unit).

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

45 En el ejemplo detallado descrito en el presente documento, se enumeran estas colas de espera de 1 a n , siendo n el número máximo de flujos activos.

El procedimiento de ordenación descrito en el presente documento utiliza además una cola de espera suplementaria reservada a un flujo activo destinado a la medida de la velocidad equitativa, debiendo entenderse la velocidad equitativa como la velocidad que produciría un flujo de datos que tuviera permanentemente unos paquetes a emitir.

50 De acuerdo con la invención, el procedimiento de ordenación utiliza igualmente una cola de espera prioritaria PQ para servir a los paquetes determinados como prioritarios en función de las características intrínsecas de la velocidad de los flujos a los que pertenecen.

55 En lo que sigue de la descripción se distinguirán las colas de espera "reales" (PQ, $Queue_1$, ..., $Queue_n$) de la cola de espera "ficticia" para la medida de la velocidad equitativa.

Las colas de espera reales son todas unas colas de espera de tipo FIFO (first in first out) conocidas para el experto en la materia.

60 El procedimiento de ordenación según la invención descrito en el presente documento utiliza igualmente una lista de 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 a extraer un índice de flujo de esta lista.

65 El procedimiento de ordenación utiliza igualmente una función $FreeBuffer()$ adaptada para liberar un lugar de memoria rechazando el paquete situado en la cabeza de la cola de espera $Queue_i$ más largo, según el método descrito en "Stochastic Fairness Queueing", P. McKenney, *Internetworking: Research and Experience*, Vol. 2, enero 1991, páginas

113-131.

Pueden utilizarse también otras estrategias de liberación de la memoria que se ocupan de preservar la equidad entre los flujos.

5 Como la mayor parte de los procedimientos de ordenación conocidos, el procedimiento de ordenación según la invención descrito en el presente documento utiliza principalmente dos operadores estándar de manipulación de colas de espera:

- 10 - un operador InsertInQueue() de inserción de un paquete al final de cola;
- un operador RemoveFromQueue() adaptado para devolver el elemento en cabeza de la cola.

15 En el transcurso de una etapa preliminar de inicialización, el programa de ordenación PROG_ORD inicializa los contadores de déficit DC_i a 0 para todas las colas de espera Queue_i e inserta, en la lista de los flujos activos ActiveList, el índice 0 que corresponde al flujo ficticio para la medida de la velocidad equitativa.

Se van a describir ahora, con referencia a la figura 1, las principales etapas del operador de puesta en cola Enqueue().

20 Tras la recepción de un paquete p, el operador de puesta en cola Enqueue() implementa una primera etapa E10 en el transcurso de la que comprueba si la zona de memoria tampón ("buffer zone" en inglés) en la que se memorizan los paquetes de todos los flujos activos está o no saturada;

25 si la memoria tampón está saturada, el resultado de la prueba E10 es positivo.

Esta prueba es seguida entonces por una etapa E20 en el transcurso de la que se libera del sitio memoria rechazando el paquete situado en la cabeza de la cola de espera Queue_i más largo, y esto por llamada a la función FreeBuffer() antes citada.

30 La etapa E20 de liberación de memoria es seguida por una etapa E30, en el curso de la que se obtiene el número i del flujo al que pertenece el paquete p.

35 Por el contrario, si la lista de flujos activos ActiveList no está llena, el resultado de la prueba E10, anteriormente descrita, es negativo. Esta prueba es seguida entonces por la etapa E30 de obtención descrita anteriormente.

La etapa E30 de obtención del índice de flujo es seguida por una prueba E40, en el curso de la que se verifica si el flujo i del paquete p está activo, lo que se convierte en verificar si el flujo i está contenido en la lista de flujos activos ActiveList.

40 Cuando no es ese el caso, el resultado de la prueba E40 es negativo. Esta prueba es seguida entonces por una etapa E42, en el curso de la que se inserta el flujo i en la lista de los flujos activos ActiveList por llamada a la función InsertActiveList() antes citada.

45 Esta etapa E42 de inserción es seguida por una etapa E44, en el curso de la que se inicializa a 0 un contador de déficit DC_i de la cola de espera Queue_i asociada al flujo activo i.

Este contador de déficit DC_i, es similar al del método de ordenación DRR conocido para el experto en la materia.

50 De acuerdo con la invención, cada flujo nuevamente recibido puede emitir, por ciclo, unos datos con prioridad a la altura de su cuota Q_i.

55 Para esto, la etapa E44 de inicialización es seguida por una etapa E46, en el curso de la que se memoriza en una variable BytesCount_i, el tamaño size(p) del paquete p, utilizándose esta variable BytesCount_i para memorizar el número de octetos de los paquetes del flujo i a tratar con prioridad.

Esta etapa E46 de asignación es seguida por una etapa E48, en el curso de la que se añade el paquete p al final de la cola prioritaria PQ, utilizando la función InsertInQueue() de inserción de un paquete al final de una cola de espera.

60 Esta etapa E48 de puesta en cola es seguida por una etapa E50, en el curso de la que se inicializa una variable Silence con un valor booleano FALSO con el fin de 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, siendo representado el estado de ausencia de tráfico por el hecho de que el identificador es igual al valor booleano VERDADERO.

Esta etapa E50 finaliza el procedimiento de puesta en cola del paquete p en este modo de realización.

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

Esta prueba es seguida entonces por una etapa E52, en el curso de la que se añade, al contenido de la variable antes citada BytesCount_i, el tamaño del paquete p, utilizándose esta variable BytesCount_i para memorizar el número de octetos de los paquetes del flujo i a tratar con prioridad.

10 Esta etapa de acumulación es seguida por una prueba E54, en el curso de la que se verifica si el número de octetos del flujo i del paquete p a tratar con prioridad (a saber BytesCount_i) es inferior o igual a la cuota Q_i de la cola de espera Queue_i reservado a este flujo.

15 Si ese es el caso, el resultado de la prueba E54 es positivo. Esta prueba es seguida entonces por una etapa de puesta en cola E56, similar a la etapa de puesta en cola E48, ya descrita, en el curso de la que se añade el paquete p al final de la cola prioritaria PQ.

20 Por el contrario, si el número de octetos BytesCount_i a tratar con prioridad para el flujo i excede la cuota Q_i para este ciclo, el resultado de la prueba E54 es negativo. Esta prueba es seguida entonces por una etapa E58 de puesta en cola del paquete p en la cola normal, es decir no prioritaria, Queue_i reservada a este flujo i.

Las etapas E56 y E58 de puesta en cola finalizan el procedimiento de puesta en cola de este modo particular de realización.

25 Se van a describir ahora, con referencia a la figura 2, las principales etapas del operador de extracción de cola Dequeue().

Este operador Dequeue() de fin del tratamiento de un paquete se implementa:

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

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

35 En el ejemplo de realización descrito aquí, este operador Dequeue() se realiza en la forma de un bucle sin fin, constituido por las etapas F10 a F84 que se van a describir ahora.

40 En el curso de una primera etapa F10, se verifica si la cola prioritaria PQ está vacía. Si no es ese el caso, esta prueba F10 es seguida por una etapa F20, en el curso de la que se extrae el paquete p en cabeza de la lista de la cola prioritaria PQ.

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

45 Esta etapa F20 de extracción es seguida por 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.

Esta etapa F22 de obtención es seguida por una etapa F24 en el curso de la que se emite el paquete p.

50 Esta etapa de envío F24 es seguida por una etapa F26, en el curso de la que se sustrae, al contador de déficit DC_i de la cola de espera Queue_i asociada al flujo i del paquete p retirado de la cola prioritaria PQ en la etapa F20, el tamaño size(p) de este paquete p.

55 Esta etapa F26 de sustracción es seguida por una etapa F28, en el curso de la que se añade el tamaño del paquete p al contenido de una variable PB, utilizándose esta variable PB para memorizar el volumen del tráfico a tratar con prioridad.

60 Esta etapa de acumulación F28 es seguida por la prueba F10 ya descrita, en el curso de la que se verifica si la cola prioritaria PQ está vacía. De este modo, las etapas F10 a F28 constituyen un bucle que se prolonga en tanto que haya paquetes presentes en la cola prioritaria PQ.

En el curso de este bucle, los contadores de déficit DC_i asociados a los flujos i cuyos paquetes se han emitido con prioridad se incrementa con el fin de impedir que cada uno de los flujos sobrepase su cuota Q_i en el ciclo actual.

65 Cuando la cola prioritaria PQ está vacía, el resultado de la prueba F10 es positivo. Esta prueba es seguida entonces por una prueba F30, en el curso de la que se verifica si al menos un flujo real está activo.

En el modo de realización descrito en este caso, esto se convierte en verificar que la lista de flujos activos `ActiveList` contiene más de un elemento, conteniendo esta lista de cualquier manera el flujo ficticio 0 utilizado para la medida de la velocidad equitativa.

5 Cuando no es ese el caso, el resultado de esta prueba F30 es negativo. Esta prueba es seguida entonces por una etapa F32 en el curso de la que se asigna a la variable `Silence`, descrita anteriormente en la etapa E50 de la figura 1, la constante booleana `VERDADERO`, con el fin de representar el hecho de que no hay ya ningún paquete a emitir, tanto si está en la cola prioritaria PQ como en una de sus colas de espera no prioritarias `Queue_i`.

10 Esta etapa de asignación F32 es seguida por la etapa F10 ya descrita, en el curso de la que se comprueba si la cola prioritaria PQ está vacía o no.

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

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

De manera más precisa, en el curso de la etapa F40, se comprueba si el flujo obtenido en cabeza de la lista de los flujos `ActiveList` en la etapa precedente F34, es el flujo ficticio 0.

25 Si ese es el caso, el resultado de la prueba F40 es positivo. Esta prueba es seguida entonces por una etapa F42, en el curso de la que se añade al volumen FB la cuota `Q_0` asignada al flujo ficticio 0.

30 Esta etapa de acumulación F42 es seguida por una etapa F44, en el curso de la que se retira la cola ficticia 0 de la cabeza del ciclo para volver a añadirla inmediatamente al final del ciclo utilizando las operaciones estándar `InsertActiveList()` y `RemoveActiveList()` antes citadas.

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

35 Esta prueba es seguida entonces por una etapa F50, en el curso de la que, se añade, al valor actual del contador de déficit `DC_i` para este flujo `i`, su cuota `Q_i`.

40 Esta etapa de acumulación F50 es seguida por una serie de etapas F60 a F68 para emitir los paquetes del flujo `i` hasta una altura de su cuota `Q_i`.

De manera más precisa, en el curso de una prueba F60, se verifica que el contador de déficit `DC_i` es estrictamente positivo y que la cola de espera no prioritaria `Queue_i` no está vacía.

45 Cuando ese es el caso, el resultado de la prueba F60 es positivo. Esta prueba es seguida entonces por una etapa F61, en el curso de la que se memoriza en una variable `s` el tamaño del paquete en la cabeza de la cola de espera `Queue_i`.

50 Esta etapa F61 es seguida por una prueba F62 en el curso de la que se verifica si este tamaño `s` es inferior o igual al contador de déficit `DC_i` para este flujo.

Cuando ese es el caso, el resultado de la prueba F62 es positivo. Esta prueba es seguida entonces por una etapa F64, en el curso de la que se envía este paquete, después de haberla retirado de la cola de espera `Queue_i` mediante la utilización de la función `RemoveFromQueue()`.

55 La etapa F64 de envío es seguida por una etapa F66, en el curso de la que se sustrae, del contador de déficit `DC_i`, el tamaño `s` de este paquete.

60 Esta etapa de sustracción F66 es seguida por la prueba F60 ya descrita, en el curso de la que se verifica que la cola de espera `Queue_i` asociada al flujo `i` no está vacía y que el contador de déficit `DC_i` para esta cola es estrictamente positivo.

Cuando se han emitido todos los paquetes del flujo `i`, hasta la cuota `DC_i` autorizada, el resultado de la prueba F60 o de la prueba F62 es negativo.

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

ES 2 799 698 T3

Esta etapa F70 de retirada es seguida por una prueba F80, en el curso de la que se verifica si la cola de espera Queue_i asociada al flujo i está vacía.

5 Si ese es el caso, se inicializa a 0 el contador del déficit DC_i asociado a este flujo i, en el curso de una etapa F82 de inicialización.

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

10 Las etapas F82 de inicialización del contador de déficit DC_i y F84 de inserción del flujo i en la lista de los flujos activos ActiveList son seguidas por la prueba F10 ya descrita.

15 El experto en la materia comprenderá que, en la práctica, la lista de los flujos activos ActiveList debe redimensionarse para una reducida probabilidad de desbordamiento.

En un modo particular de realización, cuando se recibe un paquete de un nuevo flujo y la lista de los flujos activos ActiveList está llena, este paquete se pone en la cola prioritaria PQ pero el flujo al que pertenece no se vuelve añadir a la lista de los flujos activos ActiveList.

20 Este paquete es enviado entonces (etapa F24) pero las informaciones asociadas al flujo al que pertenece no se actualizan.

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

Esta violación no es molesta en la medida en la que la probabilidad de permanecer en este estado es muy reducida para un flujo de velocidad superior a la velocidad equitativa a medida que vayan llegando paquetes.

30 Se ha de observar que el algoritmo de ordenación, según la invención, basado en el método DRR conocido es de complejidad en $O(1)$.

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

35 Se supondrá en este caso que esta descripción comienza con la hipótesis de tres colas de espera Queue_1, Queue_2, Queue_3 que contienen unos paquetes p de los flujos respectivos flujo 1, flujo 2, flujo 3. El tamaño de sus paquetes se notifica en estas figuras.

40 Se supondrá también que al comienzo de este escenario, la cola prioritaria PQ está vacía.

Finalmente, el valor de la cuota Q_i añadida a cada flujo i se supone igual a 1.500.

45 En estas diferentes figuras, la flecha negra indica la cabeza de la lista de los flujos activos en el ciclo, siguiendo este ciclo el orden de las colas no prioritarias siguiente: Queue_1, Queue_2, Queue_3, Queue_4.

El paquete en curso de servicio se sombrea en cada una de las diferentes etapas. Inicialmente, como se representa en la figura 3A, el flujo que está en cabeza de la lista es el flujo 1 asociado a la cola de espera Queue_1.

50 Su contador de déficit DC_1 se aumenta por su cuota Q_1, a saber 1.500.

El enrutador 100 debe comenzar por tanto a servir el primer paquete, sombreado, de esta cola de espera Queue_1, siendo en este caso el tamaño de este paquete de 500 octetos.

55 Se supone, con referencia a la figura 3B, que llega un paquete que pertenece a un nuevo flujo 4, durante el servicio del paquete anterior.

Se crea una cola Queue_4 y se asocia a este flujo 4.

60 Contrariamente a lo que pasaría en el marco de una ordenación de tipo DRR clásica, este paquete se inserta en este caso en la cola prioritaria PQ, como se ha descrito anteriormente, con referencia a la etapa E48.

Sin embargo, se prosigue el tratamiento del flujo 1, siendo el tamaño del segundo paquete de la Queue_1, a saber 500 objetos, inferior al nuevo déficit DC_1 (resultados de las pruebas F60 y F62 positivas).

65 En la etapa representada en la figura 3C, el tratamiento del flujo 1 se alcanza puesto que el déficit DC_1 asociado a

la cola 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 De acuerdo con la invención y, contrariamente a lo que pasaría en el marco de una ordenación de tipo DRR clásica, se sirve la cola prioritaria PQ (resultado negativo de la prueba F10), conteniendo este el paquete que pertenece al flujo 4 del que se resta el déficit DC_4 (véase la etapa F66).

10 En el curso de la etapa siguiente representada en la figura 3D, habiéndose convertido la cola prioritaria PQ en vacía (prueba F10 positiva), se trata el primer flujo real activo (prueba F30 positiva), a saber en este caso el flujo 2, asociado a la cola de espera Queue_2.

15 En el curso de la etapa siguiente representada en la figura 3E, se trata el flujo siguiente (flujo 3) porque el tamaño del paquete que permanece en la cola de espera Queue_2 es superior a la cuota DC_2 asociada a esta cola de espera (resultado negativo de la prueba F62).

La última etapa representada en la figura 3F, constituye el comienzo de un nuevo ciclo en el que se vuelve a comenzar a servir el flujo 1. El flujo 4 ya no está activo porque su cola Queue_4 está vacía, no forma parte por tanto del ActiveList.

20 La figura 4 representa un dispositivo de ordenación 100 de acuerdo con 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 los paquetes p, que ordena con el fin de su emisión, o que rechaza en caso de desbordamiento.

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

30 Este dispositivo de ordenación 100, incluye un procesador 110, una memoria inactiva 120, una memoria activa 130, un reloj 140.

35 El dispositivo de ordenación 100 incluye igualmente unos medios de equipos y de software de comunicación 150 constituidos por ejemplo por una tarjeta de red conectada a la red de Internet 10 y por unas capas de software adaptadas para implementar el protocolo de comunicación TCP/IP.

Estos medios de comunicación 150 están principalmente adaptados para recibir un paquete p y para memorizarlo en una zona 131 de la memoria activa 130 y para leer un paquete p desde esta zona para enviarlo en la red de internet 10.

40 De manera conocida los diferentes equipos del dispositivo de ordenación 100 se conectan entre sí mediante un sistema de bus 160.

45 De acuerdo con la invención, la memoria inactiva 120 incluye el programa de ordenación PROG_ORD y la memoria activa 130 en una estructura de datos 170 para la implementación de este programa de ordenación PROG_ORD.

Esta estructura de datos 170 se representa en la forma de una tabla en la que cada línea corresponde a una cola de espera Queue_i, asociada a un flujo i. De manera más precisa, en esta tabla:

- 50 - la primera columna memoriza el índice i del flujo;
- la segunda columna memoriza el identificador del flujo i;
- la tercera columna memoriza la cuota Q_i de la cola Queue_i;
- 55 - la cuarta columna memoriza el contador de déficit DC_i de la cola Queue_i;
- la quinta columna permite gestionar el ciclo de los flujos de la lista de los flujos activos ActiveList memorizando el índice de la cola de espera que sigue a la cola 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 los flujos activos ActiveList; y
- 60 - la última columna da la dirección de memorización del paquete en cabeza de la cola de espera Queue_i en la memoria activa 130. De manera conocida, los paquetes siguientes en la cola de espera Queue_i son accesibles por medio de una lista de punteros encadenados.

65 En un modo particular de realización, la verificación 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).

Se van a describir ahora, con referencia a la figura 5, un enrutador 50 de acuerdo con la invención en un modo particular de realización.

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

10 En el modo particular de realización descrito en este caso, este enrutador incluye además un módulo 24 de control de admisión de cada paquete p de los flujos entrantes (flecha continua de referencia 31).

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

En el ejemplo de realización descrito aquí, este módulo 24 de control de admisión efectúa dos niveles de control que pueden implementarse independientemente.

20 Para esto, el módulo 24 de control de admisión consulta y actualiza una lista 30 de los flujos llamados "protegidos", a saber los flujos, admitidos por el módulo 24 de control de admisión, activos (es decir un paquete de este flujo se ha identificado desde un intervalo de tiempo predeterminado) y decide, sobre la base de esta lista 30 de los flujos protegidos, si los paquetes de un flujo dado deben encaminarse directamente o no, es decir sin que sea necesario verificar las condiciones de admisión para encaminar el paquete hacia el dispositivo de ordenación.

25 Concretamente, esta lista de los flujos protegidos 30 es una lista de identificadores de flujos que indican para cada uno de entre ellos, la hora de llegada del último paquete.

30 En un modo particular de realización, cada lista se asocia a una partición del espacio de los identificadores, y esto con el fin de limitar las capacidades de cada lista y por tanto de garantizar la extensibilidad.

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

35 En un modo particular de realización, esta lista 30 se dimensiona de manera que se limita la probabilidad de saturación, situación en la que un flujo debería ser puesto en la lista, mientras que esta ya está llena.

40 La consecuencia de una saturación de ese tipo sería que un flujo tardaría en adquirir el estatuto de flujo protegido. Estos paquetes serían encaminados sin embargo correctamente si el estado de congestión lo permite. De manera conocida, la probabilidad de saturación puede hacerse suficientemente reducida mediante un dimensionamiento adecuado.

45 Si el paquete pertenece a un flujo protegido, se encamina directamente (flecha 38) hacia el dispositivo de ordenación 100 que corresponde a la buena interfaz de salida, siendo actualizada la hora de llegada del último paquete de este flujo protegido en la lista de identificadores 30.

Si el flujo ya no está protegido, es necesario proceder a una toma de decisión de encaminamiento, siendo tomada esta decisión sobre la base de las condiciones de admisión recibidas del módulo de ordenación.

50 De acuerdo con 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 esas condiciones se materializa por la flecha de referencia 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:

- la velocidad equitativa es una medida de la velocidad que produciría un flujo de datos que tuviera, permanentemente, unos paquetes a emitir; y

60 - 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 unas medidas periódicas, se obtiene un contaje continuo del estado de carga del enlace controlado. El período de las medidas es típicamente diferente para los dos valores de velocidad equitativa (por ejemplo, del orden de la centena de milisegundos) y de carga prioritaria (por ejemplo, del orden de la decena de milisegundos).

ES 2 799 698 T3

En el modo particular de realización de la invención descrito en este caso, estos estimadores de congestión, CP y DE, se calculan mediante un proceso de medida 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.

5 Por otra parte, el proceso de medida (u otro módulo) incluye unos medios, no detallados aquí, adaptados para medir el tiempo total Silence_Time, durante el que la variable lógica Silence es igual al valor booleano VERDADERO, recordándose que esta variable Silence se actualiza al valor VERDADERO en la etapa F32 y al valor FALSO en la etapa E50.

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

15 Sea:

- PB(T) el valor, en octetos, de PB en el instante T;

- (T1, T2) un intervalo de medida (en segundos); y

20 - C la velocidad del enlace (en bits/segundo), entonces, un estimador de carga prioritaria CP para este intervalo es:

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

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

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

Conjugando la sucesión de períodos de actividad y de silencio, se deduce una estimación de la velocidad equitativa DE que la manera siguiente, sea

35 - FB (T) el valor, en octetos, de FB en el instante T;

- (T1, T2) un intervalo de medida; y

- Silence_Time la duración total del silencio durante el intervalo (T1, T2), se obtiene:

40 $DE (T1, T2) = \max (Silence_Time \times C / (T2-T1), (FB(T2)-FB(T1)) \times 8 / (T2-T1)).$

45 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 reducida (porque el flujo ficticio 0 habría utilizado toda la capacidad restante disponible del enlace), prevaleciendo el segundo término en período de actividad, midiendo aproximadamente este término la velocidad producida mediante un flujo real que tiene siempre al menos un paquete en la cola de espera que se le asocia.

50 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 resultados es igual a 1 (respectivamente 0), cuando los paquetes p de nuevos flujos, que no tienen aún el estatus de flujo protegido, deben encaminarse (respectivamente rechazarse), suministrándose el resultado de esta función de admisión (fecha 36) por el módulo de ordenación 100 al módulo de encaminamiento 24.

55 Cuando las condiciones de admisión son favorables, el nuevo flujo cuyo paquete se encamina, se inserta en la lista de los flujos protegidos y adquiere de ese modo el estatus de flujo protegido.

60 En esta figura, la flecha continua 39 representa el rechazo de los primeros paquetes de los nuevos flujos por el módulo de encaminamiento 24.

En esta figura, la flecha continua 38 representa el encaminamiento de los paquetes de flujos protegidos hacia el módulo de ordenación 100.

65 Las condiciones aplicadas pueden depender del atributo particular del paquete incluyendo el valor del campo "clase de tráfico" en IPv6 o el campo ToS en IPv4 o las direcciones IP de origen y destino.

El control de admisión permite asegurar 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.

- 5 De hecho, esta transparencia no se ofrece más que a los flujos cuya velocidad de pico (determinada por las limitaciones externas) permanece inferior a un cierto umbral.

- 10 Para elegir unos umbrales de admisión relacionados con la velocidad equitativa y con la carga prioritaria que permitan realizar un control de admisión permitiendo asegurar la transparencia de los flujos, el experto en la materia podrá referirse al documento FR2854296 del presente Solicitante anteriormente citado.

En la descripción que antecede, el proceso de medida y el procedimiento de ordenación son distintos entre sí. Como variante, el proceso de medida puede estar integrado en el procedimiento de ordenación.

REIVINDICACIONES

1. Procedimiento de ordenación de paquetes que pertenecen a diferentes flujos (i), asociándose una cola de espera (Queue_i) a cada uno de dichos flujos, incluyendo este procedimiento, tras la recepción de un paquete (p):

- la determinación del flujo (i) al que pertenece el paquete (p);

- una etapa (E40, E54) de determinación de la prioridad de dicho paquete (p), siendo considerado un paquete como prioritario cuando:

- dicho paquete es el primer paquete de un flujo no activo, siendo considerado un flujo como no activo cuando no se haya recibido ningún paquete de este flujo en un ciclo actual o

- dicho paquete es uno de los paquetes de un flujo activo (i), siendo considerado un flujo como activo después de la recepción de un primer paquete que pertenece a este flujo y hasta que la cola de espera (Queue_i) asociada a este flujo esté vacía, en tanto que el número de octetos de este flujo (i) recibidos en el transcurso del ciclo actual sea inferior a una cuota (Q_i) asociada a un flujo (i) para este ciclo actual; y

- si dicho paquete (p) se determina como que es no prioritario, una etapa (E58) de puesta en cola de este paquete (p) en la cola de espera (Queue_i);

comprendiendo dicho procedimiento además la emisión de paquetes en el transcurso de dicho ciclo en el siguiente orden:

- las colas de espera se tratan cíclicamente, consistiendo el tratamiento de una cola de espera (Queue_i) en emitir unos paquetes contenidos en esta cola hasta el nivel de dicha cuota (Q_i) y

- los paquetes prioritarios se emiten con prioridad con relación a los paquetes contenidos en las colas de espera.

2. Procedimiento de ordenación según la reivindicación 1, caracterizado por que incluye, además, una etapa (F28, F42) de medida de contadores (FB, PB) de congestión destinados a ser tratados para el cálculo de estimadores (CP, DE) de gestión utilizados para un control de admisión.

3. Procedimiento de ordenación según la reivindicación 2, caracterizado por que dichos parámetros de congestión están constituidos por:

- un valor (DE) de velocidad equitativa, que es una medida de la velocidad que produciría un flujo de datos que tuviera siempre paquetes a emitir; y

- un valor (CP) de carga prioritaria, que corresponde al volumen de dichos paquetes prioritarios, transmitidos durante un intervalo de tiempo, dividido por la duración de este intervalo.

4. Dispositivo de ordenación de paquetes que pertenecen a diferentes flujos (i), asociándose una cola de espera (Queue_i) a cada uno de dichos flujos, incluyendo este dispositivo, tras la recepción de un paquete (p):

- unos medios para determinar el flujo (i) al que pertenece el paquete (P);

- unos medios de determinación de la prioridad de dicho paquete (p), siendo considerado un paquete como prioritario cuando:

- dicho paquete es el primer paquete de un flujo no activo, siendo considerado un flujo como no activo cuando no se haya recibido ningún paquete de este flujo en un ciclo actual o

- dicho paquete es uno de los paquetes de un flujo activo (i), siendo considerado un flujo como activo después de la recepción de un primer paquete que pertenece a este flujo y hasta que dicha cola de espera (Queue_i) asociada a este flujo esté vacía, en tanto que el número de octetos de este flujo (i) recibidos en el transcurso del ciclo actual sea inferior a una cuota (Q_i) asociada a un flujo (i) para este ciclo actual;

- unos medios de puesta en cola de este paquete (p) en la cola de espera (Queue_i) cuando dicho paquete p se determina no prioritario; y

- unos medios para emitir unos paquetes en el transcurso de dicho ciclo en el siguiente orden:

- las colas de espera se tratan cíclicamente, consistiendo el tratamiento de una cola de espera (Queue_i) en emitir unos paquetes contenidos en esta cola hasta el nivel de dicha cuota (Q_i) y

ES 2 799 698 T3

- los paquetes prioritarios se emiten con prioridad con relación a los paquetes contenidos en las colas de espera.

- 5 5. Dispositivo de ordenación según la reivindicación 4, caracterizado por que incluye además unos medios de medida de contadores (FB, PB) de congestión destinados a ser tratados para el cálculo de estimadores (CP, DE) de congestión utilizados para un control de admisión.
6. Dispositivo de ordenación según la reivindicación 5, caracterizado por que dichos parámetros de congestión están constituidos por:
- 10 - un valor (DE) de velocidad equitativa, que es una medida de la velocidad que produciría un flujo de datos que tuviera siempre paquetes a emitir; y
- 15 - un valor (CP) de carga prioritaria, que corresponde al volumen de dichos paquetes prioritarios, transmitidos durante un intervalo de tiempo, dividido por la duración de este intervalo.
7. Enrutador de paquetes caracterizado por que incluye un dispositivo de ordenación (100) según la reivindicación 5 o 6 y un módulo (24) de control de admisión de dichos paquetes en función de los parámetros de congestión medidos por dicho dispositivo de ordenación.
- 20 8. Enrutador según la reivindicación 7, caracterizado por que dicho módulo de admisión está adaptado además para encaminar directamente los paquetes que pertenecen a unos flujos llamados protegidos, a saber los flujos para los que se ha recibido al menos un paquete por dicho módulo de admisión después de un intervalo de tiempo predeterminado.
- 25 9. Programa informático almacenado en un soporte de informaciones, incluyendo dicho programa instrucciones que permiten la implementación de un procedimiento de ordenación según una cualquiera de las reivindicaciones 1 a 3, cuando este programa es cargado y ejecutado por un sistema informático.
- 30 10. Soporte de informaciones legible por un sistema informático, eventualmente total o parcialmente extraíble, principalmente un CD-ROM o soporte magnético, tal como un disco duro o disquete, o un soporte transmisible, tal como una señal eléctrica u óptica, caracterizado por que incluye instrucciones de un programa informático que permiten la implementación de un procedimiento de ordenación según una cualquiera de las reivindicaciones 1 a 3, cuando este programa es cargado y ejecutado por un sistema informático.

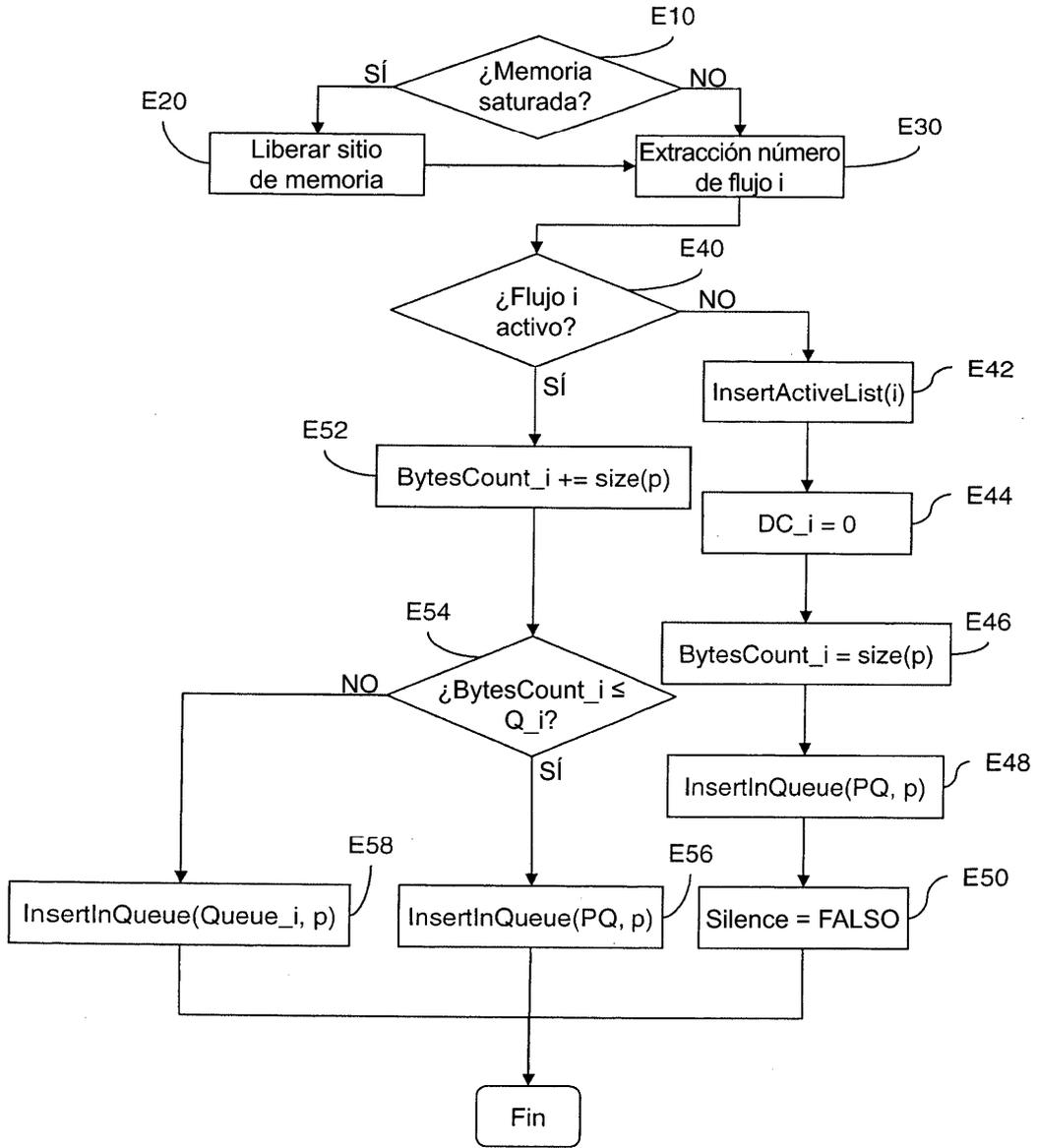


FIG. 1

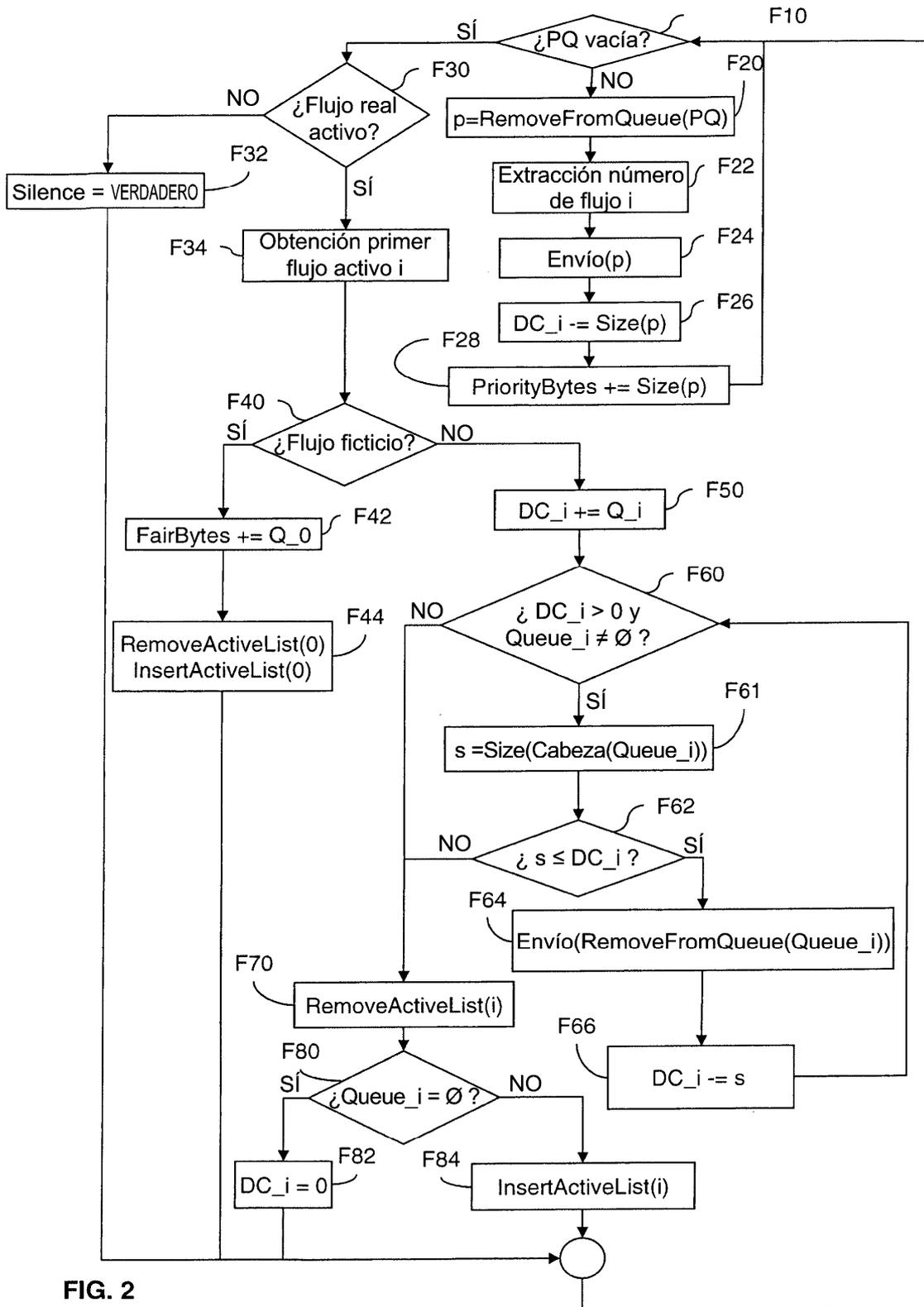


FIG. 2

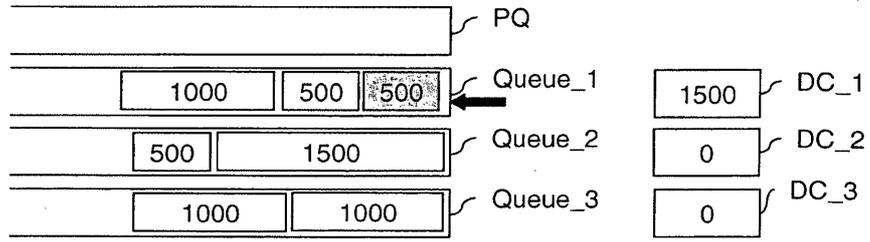


FIG. 3A

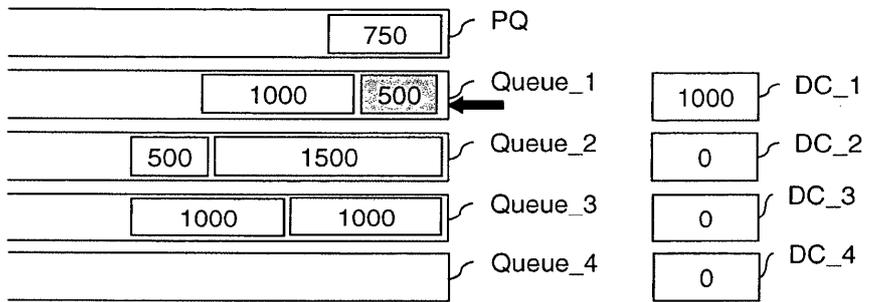


FIG. 3B

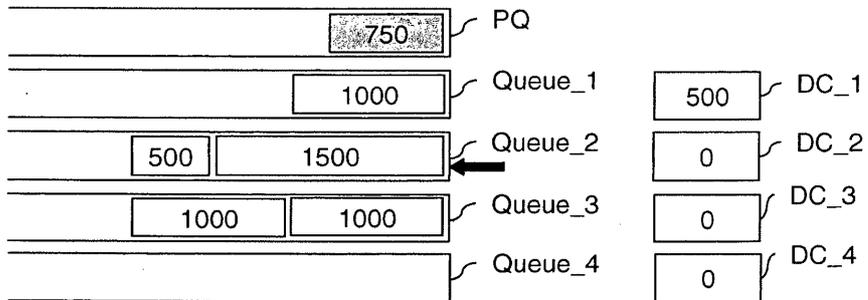


FIG. 3C

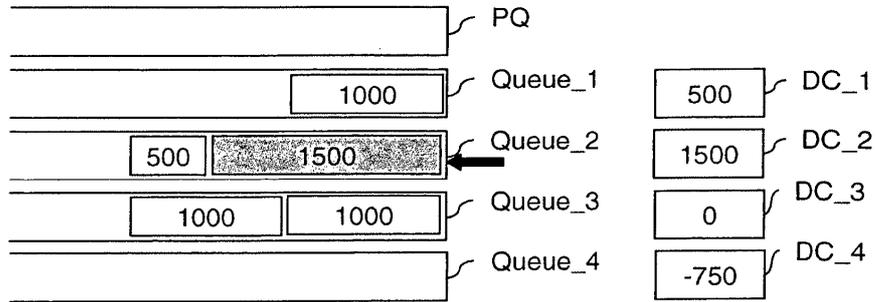


FIG. 3D

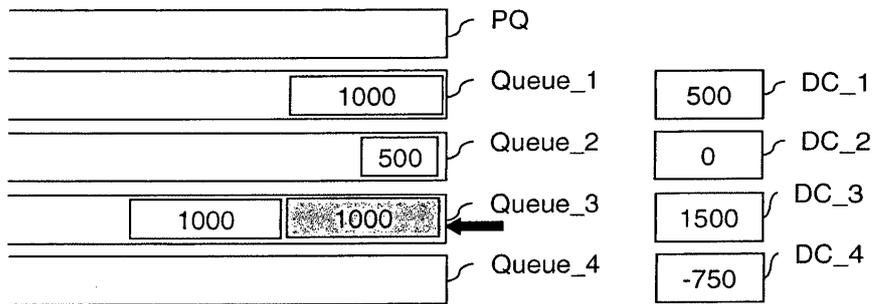


FIG. 3E

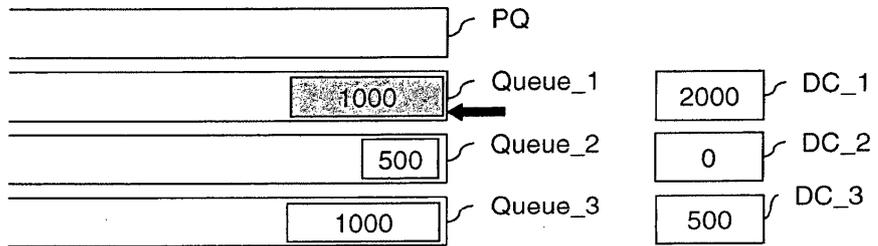
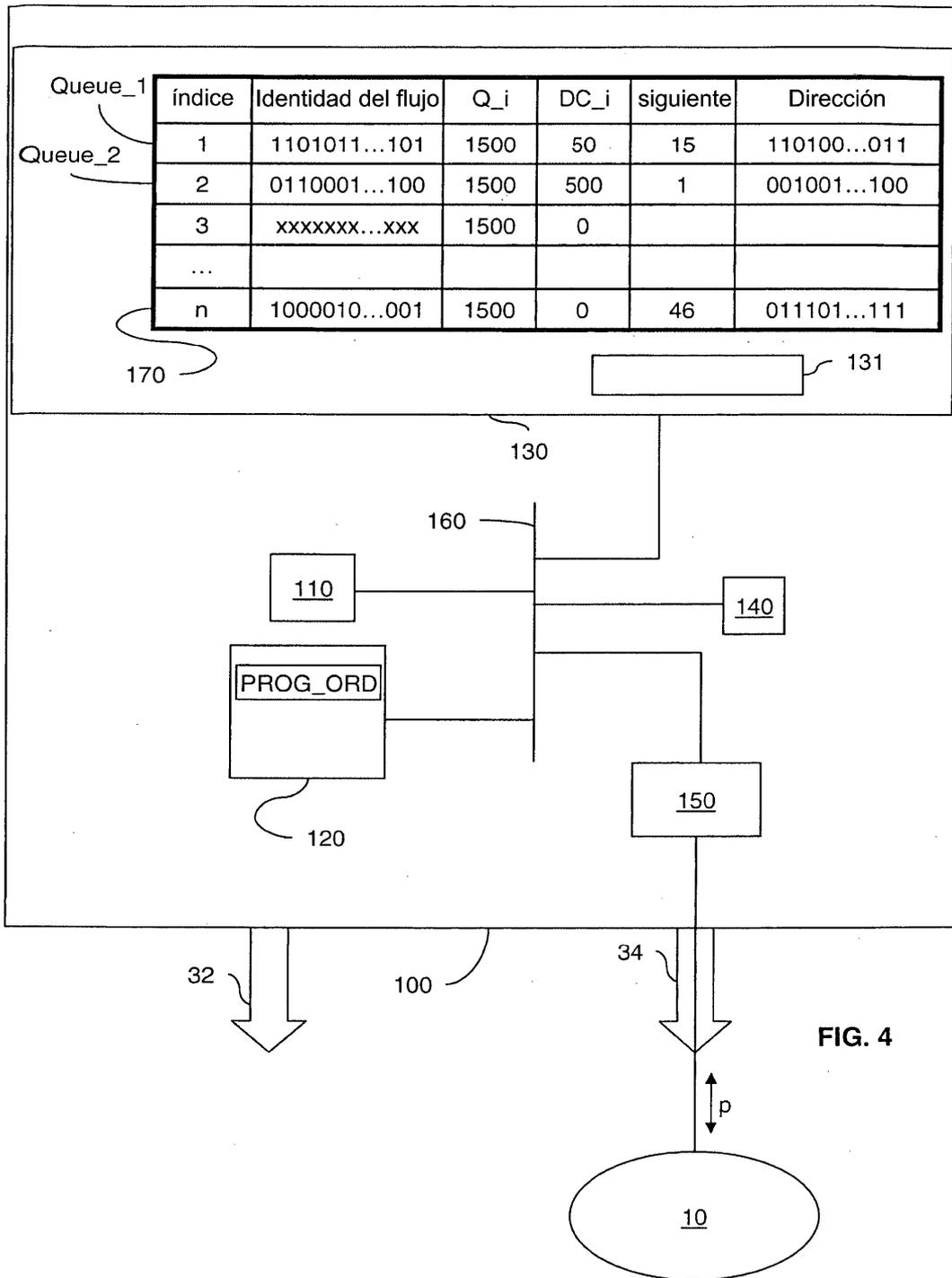


FIG. 3F



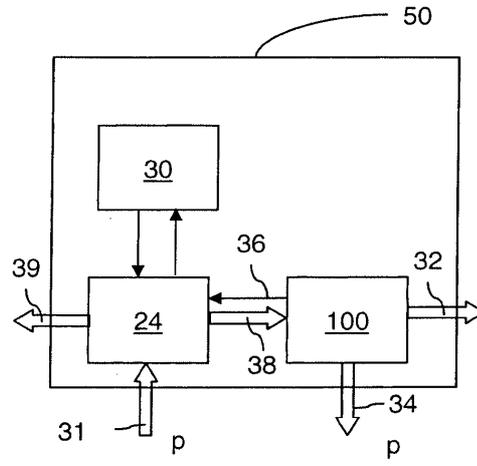


FIG. 5