

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 401 374**

51 Int. Cl.:

H04L 12/66 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **21.05.2010 E 10721155 (9)**

97 Fecha y número de publicación de la concesión europea: **13.02.2013 EP 2433400**

54 Título: **Procedimiento para poner en cola paquetes entre diferentes capas de un protocolo de comunicación**

30 Prioridad:

22.05.2009 ES 200930197

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

19.04.2013

73 Titular/es:

**TELEFÓNICA, S.A. (100.0%)
Gran Vía, 28
28013 Madrid, ES**

72 Inventor/es:

**KIRNECH NEBOT, XAVIER;
RODRÍGUEZ RODRÍGUEZ, PABLO;
GONZÁLEZ CALABOZO, JOSÉ MARÍA;
ABILLA NAVARRO, XAVIER IGNASI;
CUMPLIDO DOMÍNGUEZ, RAÚL;
GONZÁLEZ BLANCO, RUBÉN;
CARPIO ESQUINA, MARÍA DOLORES;
PERALES FERRERA, DAVID y
FERRAN BENNSTROM, CHRISTIAN**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 401 374 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 para poner en cola paquetes entre diferentes capas de un protocolo de comunicación

Campo de la invención

5 La presente invención se aplica al campo de las redes digitales de paquetes y, más concretamente, a la multiplexación inversa de canales heterogéneos en dichas redes.

Antecedentes de la invención

10 Las redes actuales tienen una arquitectura basada en capas: cada capa usa los servicios de la capa inmediatamente inferior y ofrece servicios a las capas superiores. El funcionamiento de capa es transparente para las capas superiores. En los sistemas operativos actuales, en la implementación de esta arquitectura de capas, cada interfaz de red físico tiene asociado en la capa de enlace una cola de transmisión, en la que se almacenan los paquetes de las capas superiores para su envío a través del canal de comunicaciones. Esta cola de transmisión se vacía a la velocidad a la que el interfaz físico puede enviar información por el canal de comunicaciones. En las implementaciones actuales de la arquitectura de capas de red, cada capa genera tráfico para las capas inferiores y se desentiende de lo que ocurre en la capa inferior. En el caso de que las capas superiores generen tráfico más rápidamente del que el dispositivo físico puede transmitir, la cola de transmisión asociada al dispositivo se llena y el sistema operativo expulsa los paquetes, produciéndose pérdidas de paquetes locales al sistema que realiza la transmisión, tal como se muestra en la figura 1. Es decir, en la situación actual de la arquitectura de capas de red, la independencia entre capas y el exceso de velocidad en las capas superiores provocan pérdidas de paquetes locales al sistema que realiza la transmisión (sin salir al medio físico). En estos casos, tal como está diseñada la arquitectura de capas de red actual, los protocolos de las capas superiores son los responsables de tratar extremo a extremo de la comunicación la pérdida de paquetes, bien sea local al sistema que transmite o en cualquier otro punto intermedio de la comunicación entre capas.

25 Por otra parte, agregar un ancho de banda de varios canales de comunicaciones de bajo coste para crear un único canal con un ancho de banda superior al de un único canal es una técnica común en el mundo de las redes que se conoce como multiplexación inversa. Si se realiza a nivel físico o de enlace se conoce como "bonding". Cuando los diferentes medios que se desea agregar son homogéneos, la agregación de ancho de banda es simple y efectiva, ya que se puede hacer a partir de reglas predefinidas (en función del ancho de banda nominal de cada canal).

30 Sin embargo, cuando los diferentes medios que se desea agregar son de naturaleza heterogénea y con características cambiantes dinámicamente, la agregación del ancho de banda se vuelve un problema complejo. En estos casos, la distribución del flujo de información generado por las capas superiores (en el extremo de la aplicación) por los diferentes medios, se ha de realizar teniendo en cuenta las características de latencia y ancho de banda de cada medio en el momento de enviar tráfico por el mismo.

35 Esto es especialmente complejo en el caso de las redes de comunicaciones móviles para aplicaciones que requieran alto ancho de banda y comunicaciones en tiempo real en el enlace de subida, ya que el ancho de banda del canal de comunicaciones móvil varía dinámicamente y la adaptación a las condiciones del canal para aplicaciones en tiempo real ha de ser quasi-instantánea.

40 La agregación de canales o conexiones paralelas para la transferencia de datos estáticos (páginas web, ficheros), donde el orden de llegada de la información en el destino no es importante, es ampliamente conocida en Internet y existen diversos algoritmos que a día de hoy lo implementan: uso de conexiones TCP paralelas para la transferencia de ficheros mediante FTP y descarga de páginas web, etc. Además, la mayoría de estas implementaciones usan los enlaces descendentes, donde el ancho de banda es normalmente superior a los enlaces ascendentes.

45 La patente europea EP 1653706 B1 trata de resolver este problema mediante mecanismos de multiplexación inversa del flujo de datos (en concreto, vídeo) de alta velocidad de transmisión, para dividir ese flujo en una pluralidad de flujos de menor velocidad de transmisión. De esta forma, se consigue doblar el ancho de banda en sentido ascendente. En concreto, la distribución del flujo en el sistema descrito en EP 1653706 B1 se basa en la paridad de cada paquete entrante, distribuyéndose cada paquete de forma alternativa en cada una de las diversas conexiones destinadas a tal efecto. El algoritmo clasifica los paquetes entrantes por su número de secuencia. Además, con el fin de regular el flujo de paquetes que llegan de manera aleatoria al servidor de multiplexación, se utiliza un algoritmo de cubo goteante (*leaky bucket*).

50 Sin embargo, el mecanismo de multiplexación inversa de EP 1653706 B1 carece de adaptación en tiempo real, debido a que se basa en algoritmos que distribuyen la carga de una forma predefinida (en este caso, por la paridad). Por esta razón, la división del flujo es estática. Es decir, en esta solución, el algoritmo de multiplexación inversa se configura y se pone en marcha, pero luego no se adapta (pues está pensado para situaciones en las que se conoce el ancho de banda de antemano).

55 Además, en otros algoritmos de multiplexación inversa, la distribución del tráfico se realiza en un solo proceso algorítmico y a partir de reglas predeterminadas, en las que a partir del flujo de entrada se decide cómo se distribuye

el flujo entre los interfaces de salida.

La solicitud de patente de los Estados Unidos US 2006/2567930 A1 divulga un procedimiento para distribuir tráfico en una red, en la que la velocidad de transmisión de los paquetes es superior a la capacidad de transmisión del canal de transporte. Los paquetes que se van a transmitir deben ser puestos en cola, y el documento US 2006/2567930 describe cómo entran y salen los paquetes de forma secuencial a/desde una memoria en la que los paquetes son organizados como una cola. La cola se lleva a cabo por separado para cada capa del protocolo de comunicación. Más particularmente, dos capas vecinas del protocolo de comunicación son conectadas por un buffer de cola, de tal manera que cada paquete de la capa superior puede ser conectado a cada paquete de la capa inferior usando el respectivo buffer de cola. No obstante, esta técnica de control de flujo no asocia las colas de paquetes en la capa superior de una forma biunívoca con las colas de paquetes en la capa inferior.

Resumen de la invención

La presente invención trata de resolver los problemas mencionados anteriormente mediante un procedimiento de transmisión de paquetes entre una aplicación de un equipo emisor y una aplicación de un equipo receptor a través de una red de paquetes.

Concretamente, en un primer aspecto de la presente invención, se proporciona un procedimiento de transmisión de paquetes entre una aplicación de un equipo emisor y una aplicación de un equipo receptor a través de una red de paquetes, donde la velocidad de transmisión de dichos paquetes es superior a la capacidad de transmisión del canal de transporte de una conexión de dicha red de paquetes. El procedimiento comprende la etapa de, en un equipo emisor, almacenar en una cola de entrada de una determinada capa de protocolo, una pluralidad de paquetes procedentes de una capa de protocolo de orden superior. El procedimiento comprende además las etapas de: en dicho equipo emisor, controlar, mediante una pluralidad de procesos implementados en dicha determinada capa de protocolo, la inserción de cada uno de dichos paquetes en una cola de salida de una pluralidad de colas de salida de una capa de protocolo de orden inferior, donde cada una de dichas colas de salida está asociada de forma biunívoca a uno de dichos procesos y cada una de dichas colas de salida está a su vez asociada a un determinado interfaz físico, estando dichos procesos configurados para controlar la inserción de paquetes en dichas colas de salida en función del flujo de salida de paquetes de dichos interfaces físicos para evitar la pérdida de paquetes en dichas colas de salida; y transmitir los paquetes insertados en dichas colas de salida asociadas a dichos interfaces físicos a través de sendos canales de transporte de dicha red de paquetes.

En una realización particular, al menos dos de dichos interfaces físicos son diferentes entre sí.

En una realización particular, al menos dos de dichos interfaces físicos tienen velocidades máximas diferentes, transmitiéndose los paquetes a través de cada interfaz físico a la velocidad del canal de transporte de dicho interfaz físico.

Preferentemente, dicha determinada capa de protocolo es una capa de demultiplexado o multiplexación inversa.

Preferentemente, dicha etapa de controlar, mediante dicha pluralidad de procesos, la inserción de paquetes en dichas colas de salida, comprende las etapas de: cada uno de dichos procesos monitoriza el estado de su correspondiente cola de salida; y dichos procesos compiten entre sí para enviar paquetes desde dicha cola de entrada hacia su cola de salida correspondiente.

Preferentemente, dicha etapa de controlar, mediante dicha pluralidad de procesos, la inserción de paquetes en dichas colas de salida, comprende las etapas de: analizar, por parte de cada proceso, la carga de paquetes acumulados en su cola de salida asociada en la capa inferior; y si un proceso tiene su cola de salida llena por encima de un cierto umbral, dicho proceso no trata de desencolar un determinado paquete de la cola de entrada, mientras que si el número de paquetes de una cola de salida es inferior o igual a un cierto umbral, el proceso que controla dicha cola de salida trata de desencolar un determinado paquete de la cola de entrada para insertarlo en su cola de salida asociada, consiguiéndolo el proceso que invoque a la cola de entrada antes.

Preferentemente, cada cola de salida se vacía según la velocidad del interfaz físico asociado a la misma.

En una realización particular, antes de dicha etapa de almacenar en dicha cola de entrada dicha pluralidad de paquetes procedentes de una capa de protocolo de orden superior, se produce la etapa de: re-encapsular dichos paquetes y asignarles un número de secuencia respectivo para facilitar su posterior ordenación en el extremo receptor.

En una realización particular, dicha determinada capa de protocolo se implementa en la capa de aplicación y, más concretamente, a nivel del Protocolo de Transporte en Tiempo real.

En otro aspecto de la presente invención, se proporciona un procedimiento de recepción de paquetes por parte de una aplicación de un equipo receptor, donde dichos paquetes han sido enviados a través de una red de paquetes, y donde la velocidad de transmisión de dichos paquetes es superior a la capacidad de transmisión del canal de transporte de una conexión de dicha red de paquetes. Dichos paquetes han sido enviados según el procedimiento de transmisión de

paquetes anteriormente descrito.

Preferentemente, dicho procedimiento comprende la etapa de: reordenar dichos paquetes recibidos en una capa de protocolo) antes de entregarlos a una capa de protocolo de orden superior.

5 En una realización particular, cualquiera de los procedimientos anteriormente descritos comprende la etapa de ordenación y recuperación de paquetes mediante la implementación de un mecanismo basado en uno de los siguientes aspectos: en la duplicación de paquetes para minimizar la probabilidad de pérdida, en una ventana de tiempo, en una ventana de secuencia, en una diferencia de marca de tiempo, en la dispersión dentro de una ventana de ordenación.

10 Además, cualquiera de dichos procedimientos puede comprender la etapa de recuperación de paquetes por reconocimiento negativo.

En una realización particular, los paquetes se transmiten a través de una red de telefonía móvil que se conecta a dicha red de paquetes y dichos interfaces físicos se eligen del siguiente grupo: GPRS, EDGE, 3G, HSPA y Ethernet.

15 En otro aspecto de la presente invención, se proporciona la utilización de cualquiera de los procedimientos descritos para la transmisión de vídeo-conferencia de alta calidad y bajo retardo o para la transmisión de imágenes de vídeo-vigilancia.

20 Por último, en otro aspecto de la presente invención se recoge un sistema de transmisión de paquetes que comprende: un equipo emisor que comprende una determinada aplicación; una red de paquetes; un equipo receptor que comprende esa misma determinada aplicación, donde la velocidad de transmisión de dichos paquetes es superior a la capacidad de transmisión del canal de transporte de una conexión de dicha red de paquetes. El equipo emisor comprende medios para llevar a cabo el procedimiento de transmisión descrito anteriormente.

Preferentemente, el equipo receptor comprende medios para llevar a cabo el procedimiento de recepción anteriormente descrito.

25 Como ha podido apreciarse, la invención proporciona un mecanismo que permite aplicar la multiplexación inversa sobre canales heterogéneos y de ancho de banda variable en el enlace de subida, tales como canales de comunicaciones móviles, para aplicaciones que necesiten transmitir información en tiempo real con bajo retardo (videoconferencia, streaming,...), consiguiendo distribuir la carga adaptándose constantemente a la velocidad del canal. Esta adaptación dinámica a la velocidad del canal se consigue a partir del estado de las colas de transmisión de las capas inferiores, de forma que cuando una cola de salida en la capa inferior se satura, esto se detecta y el proceso de control asociado a la cola deja de consumir de la cola de entrada hasta que la cola de salida vuelva a no estar saturada (el número de paquetes baje de un determinado umbral), con lo que se adapta quasi instantáneamente.

30 El procedimiento de la invención implementa un mecanismo de distribución de tráfico basado en múltiples colas y un algoritmo que podemos denominar de control de flujo inverso: desde las capas inferiores de la arquitectura de red se controla el flujo de paquetes de las capas superiores. Es decir, en lugar de controlar el flujo teniendo en cuenta la velocidad de la entrada de una capa, la velocidad a la que se transmite hacia las capas inferiores se controla monitorizando la salida de la capa física (capa inferior) para modular la velocidad de escritura de las capas superiores que generan el tráfico (es decir, una capa inferior actúa sobre el funcionamiento de una capa superior). Este procedimiento se describe en detalle más adelante.

35 Frente a la arquitectura de capas de las redes actuales, el procedimiento de la presente invención se basa en degenerar ligeramente la arquitectura de capas actual para poder monitorizar la velocidad de una determinada capa inferior desde una capa superior, donde se implementa el procedimiento de distribución de tráfico descrito en esta invención. Esta monitorización se hace indirectamente a través de la carga de la cola de salida asociada al interfaz físico, midiendo la ocupación de dicha cola. El procedimiento evita así esa pérdida de paquetes local (dropping), manteniendo la cola de salida de cada dispositivo completamente ocupada hasta un límite, pero sin que se produzcan pérdidas locales de paquetes.

40 Las ventajas de la invención propuesta se harán evidentes en la descripción siguiente.

Breve descripción de los dibujos

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

La figura 1 muestra la expulsión de paquetes (dropping) en capas inferiores por exceso de velocidad en las capas superiores, situación que ocurre en las implementaciones de la arquitectura de capas actual.

La figura 2 muestra un esquema del control de flujo inverso según una realización de la presente invención.

La figura 3 muestra el control de flujo inverso para una realización de la presente invención.

La figura 4 muestra una posible implementación del procedimiento de la presente invención, que incluye la etapa de demultiplexación y la de multiplexación.

5 La figura 5 describe un ejemplo concreto de implementación del procedimiento de la invención a nivel de capa de red IP.

La figura 6 muestra una posible aplicación de la presente invención a una solución de captura electrónica de noticias usando varios canales móviles multiplexados inversamente.

La figura 7 muestra otra posible aplicación de la presente invención a una solución de video vigilancia usando varios canales móviles multiplexados inversamente

10 **Descripción detallada de la invención**

A lo largo de esta memoria descriptiva, el término "comprende" y sus derivados no debe interpretarse en un sentido excluyente o limitativo, es decir, no debe interpretarse en el sentido de excluir la posibilidad de que el elemento o concepto al que se refiere incluya elementos o etapas adicionales.

15 Además, a lo largo de la presente descripción aparece el término inglés *socket*, que se ha incorporado comúnmente al español para designar, en términos informáticos, un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada. Un *socket* queda definido por una dirección, un protocolo y un número de puerto.

20 La figura 2 muestra un esquema del algoritmo de control de flujo inverso del procedimiento de la invención. En este algoritmo, unos procesos (denominados SenderQ_1 SenderQ_2 ... SenderQ_n en la figura 2), monitorizan el estado de la cola de salida y compiten entre sí para enviar tráfico desde la cola de entrada InputQueue hacia el interfaz con la capa inferior. De esta forma, el flujo se controla en sentido inverso: desde la capa de salida al mundo físico se decide cómo cargar el tráfico en el interfaz físico, en función de la velocidad del mismo adaptando el flujo de entrada.

25 Como muestra la figura 2, se implementa una capa intermedia de demultiplexado (en la figura 2, Capa Demux) entre dos capas convencionales. En la implementación concreta de la figura 2, la capa intermedia de demultiplexado se implementa entre la capa de red (preferentemente IP) y la de enlace (a su vez seguida de la capa física). No obstante, el procedimiento de la invención es independiente de la capa en la que se implemente, pudiendo implementarse entre otras capas.

30 El tráfico entrante en la capa de demultiplexado a través del punto de acceso al servicio (SAP, *service access point*) experimenta un proceso de re-encapsulado y de asignación de número de secuencia, como muestra la etapa (1) de la figura 2. Esta etapa (1) es necesaria cuando en el destino, al multiplexar, hay que reordenar los paquetes. De esta forma, la multiplexación inversa es transparente para los protocolos de las capas superiores. En caso contrario, esta etapa (1) no es necesaria. A continuación, se acumula en una cola o buffer de entrada InputQueue, como se indica en la etapa (2). Este tráfico que se acumula en la cola o buffer de entrada InputQueue es el tráfico procedente de la
35 de salida de la capa de red (IP).

40 Asociado a cada dispositivo de comunicaciones (y por tanto, a cada interfaz con el medio físico), el algoritmo proporciona un proceso (que en la figura 2 se ha llamado SenderQ_1 SenderQ_2 ... SenderQ_n) responsable de recoger tráfico de la cola o buffer de entrada InputQueue (2) y cargarlo en una cola de salida, que se representa en la etapa (3). Los procesos SenderQ_1 SenderQ_2 ... SenderQ_n consumen (3) el tráfico de entrada acumulado en la cola o buffer InputQueue (2) de acuerdo a los niveles de ocupación de las colas de salida Output_1 Output_2 ... Output_n asociados a los interfaces físicos. Estos procesos se encargan del control de flujo inverso para evitar tirar paquetes (*dropping*). Estos procesos SenderQ_1 SenderQ_2 ... SenderQ_n asociados a cada dispositivo físico de transmisión NIC1 NIC2 ... NICn o interfaz físico compiten entre sí por extraer paquetes de la cola de entrada
45 InputQueue (2) y escribirlos o insertarlos en la cola de salida correspondiente Output_1 Output_2 ... Output_n (3).

50 Esta competición entre procesos SenderQ_1 SenderQ_2 ... SenderQ_n depende de la implementación concreta del algoritmo asociado a los procesos. En una realización particular, se implementa mediante procesos paralelos o *threads* en un mismo proceso. Estos procesos paralelos o *threads* están constantemente en un bucle que puede desempeñar la siguiente tarea, que se muestra a modo de ejemplo en la figura 2: Si un proceso o *thread* (por ejemplo, SenderQ_n) no ha llenado la cola de salida que controla Output_n hasta un cierto umbral considerado OutQMax, sino que la cola sólo está parcialmente ocupada OutQ, el proceso o *thread* va a la cola de entrada InputQueue de la etapa (2) a desencolar un determinado paquete p_i. Todos los procesos SenderQ_1,... SenderQ_n tratan de hacer esto a la vez, compiten entre sí, y el primero que invoca o llama a la cola de entrada InputQueue es el que lo consigue. Si, por el contrario, un proceso o *thread* (SenderQ_2) tiene su cola Output_2 llena (es decir, ha
55 llegado al umbral considerado OutQMax), el proceso o *thread* SenderQ_2 no trata de desencolar un determinado paquete p_i de la cola de entrada InputQueue hasta que el interfaz físico consiga desencolar paquetes a la salida. Es

decir, cada cola de salida Output_1... Output_n, que dirige paquetes a un respectivo interfaz físico como se ve en la etapa (4), puede aumentar su tamaño hasta un valor máximo OutQMax y los procesos SenderQ_i tratan de llenar la cola de salida hasta el límite de la pérdida de paquetes que el interfaz físico es capaz de soportar.

Otras implementaciones alternativas de la competición entre procesos SenderQ_1,... SenderQ_n son posibles.

- 5 Como se ha dicho anteriormente, la capa de demultiplexado puede estar a nivel de red (IP) o a un nivel superior (por ejemplo, de aplicación, como se muestra más adelante). En una realización de la invención, la capa de demultiplexado se ha implementado en la capa de aplicación y, más concretamente, a nivel del protocolo RTP (Protocolo de Transporte en Tiempo real o *Real-time Transport Protocol*).

10 Cada cola de salida Output_1... Output_n se vacía según la velocidad del interfaz físico correspondiente. El proceso correspondiente SenderQ_1... SenderQ_n es el responsable de mantener su cola de salida al límite máximo OutQMax pero sin que se produzca pérdida por exceso de paquetes.

15 Cada proceso SenderQ_1... SenderQ_n está asociado a un interfaz físico correspondiente, que a su vez tiene una dirección IP. Los procesos están continuamente monitorizando su cola de salida respectiva, y en el momento en que el que dispositivo físico genera un hueco en la cola de salida Output_1... Output_n porque ha llevado a cabo la transmisión de un paquete, el proceso correspondiente SenderQ_1... SenderQ_n lo detecta y vuelve a rellenar dicho hueco desde la cola de entrada InputQueue de la etapa (2).

20 La figura 3 muestra un esquema del algoritmo explicado anteriormente. En concreto, puede observarse cómo, en función de la velocidad del dispositivo físico NIC1 NIC2, la cola de salida controlada por un proceso acepta más paquetes procedentes de la cola de entrada y también esa cola de salida se va vaciando más deprisa (gracias a la velocidad que proporciona el dispositivo físico). Es decir, este dibujo muestra la idea del algoritmo: acumular los paquetes en una cola de entrada y desde los procesos asociados a las colas de salida permitir, si no están en el límite OutQMax, el paso de paquetes hacia la salida. Por ejemplo, con respecto al proceso SenderQ_2, si el tamaño OutQ de la cola de salida Output_2 es menor que el umbral OutQMax, entonces se abre el "grifo". Si no, se mantiene cerrado.

25 Se trata de un control de flujo inverso porque son los propios procesos SenderQ_1... SenderQ_n los que inversamente, desde la salida al mundo físico, condicionan y conforman el flujo de entrada de paquetes.

30 Es de destacar que el procedimiento permite la multiplexación inversa sobre canales heterogéneos, es decir, sobre interfaces físicas NIC1 NIC2... NICn de distinta naturaleza. Ejemplos no limitativos de estos interfaces son: GPRS, EDGE, 3G, HSPA, Ethernet, etc. Cada uno de estos interfaces puede tener diferentes velocidades de transmisión y condiciones cambiantes en la velocidad de transmisión.

35 El hecho de multiplexar inversamente sobre canales heterogéneos implica que se desordenen los paquetes en recepción. Opcionalmente, si los protocolos y/o aplicaciones de las capas superiores a la multiplexación inversa de la invención lo requieren, se deben aplicar diferentes políticas y algoritmos de ordenamiento de paquetes que se han reordenado por la multiplexación inversa. Por ejemplo, si se implementa el procedimiento de la invención a nivel de IP, es decir, debajo del punto de acceso al servicio de la capa IP, no hace falta aplicar dichos algoritmos de ordenamiento de paquetes, ya que TCP ordena los paquetes. Eso sí, si se mantiene la secuencia, se facilita a las capas o aplicaciones superiores su trabajo y mejora la efectividad.

40 En caso de que sean necesarios, para volver a recomponer el flujo original después de atravesar los canales multiplexados inversamente, el procedimiento de la invención proporciona diferentes mecanismos de ordenación y recuperación de paquetes, específicos para aplicaciones en tiempo real. Estos mecanismos son configurables y pueden ser: en función de un número de secuencia, de una diferencia de marca de tiempo (*timestamp*), de la dispersión dentro de una ventana de ordenación (porcentaje de paquetes ordenados en una ventana temporal y/o de tamaño). En caso de pérdidas de paquetes en alguno de los enlaces, opcionalmente se puede activar un mecanismo de recuperación por reconocimiento negativo (*Negative ACK*).

45 Con estos algoritmos de ordenación y recuperación se consigue adaptar la multiplexación inversa a diferentes aplicaciones que requieren alto ancho de banda y tiempo real sobre canales de naturaleza heterogénea y variable como los canales de comunicaciones móviles.

A continuación se describe un ejemplo de aplicación del procedimiento de multiplexado inverso de la invención.

50 La figura 4 muestra la arquitectura de un sistema que implementa tanto el procedimiento de demultiplexación inversa como el de multiplexación inversa con sus capas intermedias correspondientes.

55 En la figura 4 se ha representado una realización genérica (es decir, sin asociarla a una capa específica) extremo a extremo. Esta realización cubre posibles realizaciones específicas de la invención, que pueden tener lugar o bien por debajo de la capa IP, o bien a nivel de capa aplicación, como Proxy específico del protocolo que controla el envío de paquetes sobre las capas inferiores. Otras realizaciones en otros niveles de la arquitectura de capas también son posibles, siempre que se tenga acceso a la cola de salida de cada dispositivo.

La parte superior de la figura 4 muestra en forma de bloques la comunicación extremo a extremo: a la izquierda se muestran las capas del extremo que envía paquetes, desde la capa de envío L seguida de la capa de multiplexación inversa L+1 hasta la capa física. Una vez que la transmisión se lleva a cabo a través del medio físico, representado por una nube o red genérica 40, los paquetes acceden al extremo receptor: primero a través de la capa física y, una vez atravesadas las posibles capas intermedias, se llega a la capa de multiplexación inversa L+1 y finalmente a la capa de recepción L.

La parte inferior de la figura 4 representa las etapas del procedimiento de la invención en relación con las capas de la parte superior. Así, la capa de envío L representa la fuente de paquetes (*Packet source*). Esta capa de envío L proporciona paquetes a la capa de multiplexación inversa L+1. Normalmente, entre estas dos capas L L+1 el ancho de banda (BW) disponible es elevado, por tratarse de una distancia muy corta.

El bloque siguiente de la figura 4 representa la implementación de la capa de demultiplexado o demultiplexor en el punto de transmisión. Esta capa es responsable de recibir el flujo de paquetes de entrada (por ejemplo, a través de un socket de entrada `inputSocket`) procedentes de la capa L y transmitirlos aplicando el control de flujo inverso sobre varios interfaces de red. Este control de flujo inverso ha sido descrito en detalle anteriormente en relación a la figura 2. Para las implementaciones en las que el orden de los paquetes es importante, este módulo lleva a cabo las secuenciación y encapsulado de los paquetes para su posterior ordenación en recepción en la capa de multiplexado del otro extremo (extremo receptor). La capa de demultiplexado comprende: un punto de acceso SAP al servicio de multiplexación inversa de paquetes a la entrada de la capa L+1 (implementado mediante un `inputSocket`); una cola o buffer de entrada `InputQueue` donde se acumulan los paquetes; y varios procesos `SenderQ_1 SenderQ_2 ... SenderQ_n`, uno por cada cola de salida `Output_1 ... Output_n`, que implementan el control de flujo inverso. El procedimiento proporciona a los interfaces físicos respectivos un flujo de paquetes acorde con la velocidad que cada dispositivo físico asociado NIC1 NIC2 NIC3 pueda procesar.

A continuación, se representa de forma esquemática el envío de paquetes desde cada dispositivo físico NIC1 NIC2 NIC3 hacia el extremo receptor, a través de la red de paquetes correspondiente 40. En general, el ancho de banda (BW) disponible en esta red 40 es un recurso limitado. Gracias a la multiplexación inversa aplicada en la capa L+1, se consigue que el ancho de banda total sea la suma de los anchos de banda de cada uno de los dispositivos físicos NIC1 NIC2 NIC3: $BW = BW_1 + BW_2 + BW_3$. En la figura 4, las referencias f1, f2, f3 indican los flujos de paquetes, asociados a cada NIC (NIC1 NIC2 NIC3), en los que se demultiplexa el flujo principal a la entrada del demultiplexor o capa de demultiplexado.

Una vez que los paquetes llegan al extremo receptor, el bloque siguiente de la figura 4 representa la implementación de la capa de multiplexado o multiplexor en el punto de recepción. Se reciben los paquetes a través de varios canales y, si lo requiere la aplicación o capa superior, se reconstruye el flujo aplicando las siguientes políticas de ordenación y recuperación del flujo (configurables según requiera la capa superior):

- Ordenación por diferencia de tiempo: Usa las marcas de los paquetes *timestamp*: espera a que se acumulen los paquetes dentro de un determinado intervalo de tiempo Δt y a partir de ese momento los envía hacia la capa superior;
- Ordenación por diferencia de tiempo y espera a una diferencia máxima: espera a que exista una diferencia mínima de tiempo entre el último paquete y el primero dentro de una ventana de ordenación;
- Por número de secuencia y dispersión dentro de una ventana de ordenación: Cuando hay un % de paquetes ordenados dentro de una ventana se permite el vaciado de la cola hacia las capas superiores;
- Por reconocimiento negativo si se detecta la pérdida de un paquete, en cuyo caso se señala al demultiplexor para que lo reenvíe. Si se activa esta opción, el demultiplexor guarda los paquetes enviados durante un margen de tiempo (configurable) para poder atender las peticiones de pérdida de paquetes.

Como se explicó anteriormente, estas políticas de ordenación y recuperación se implementan dentro de la cola de reordenación, la cual puede estar compuesta de varias colas según el mecanismo de ordenación seleccionado. En función de las necesidades de retardo, fiabilidad y ordenación de los paquetes, se puede configurar el demultiplexor y el multiplexor con cada una de las políticas anteriores.

Finalmente, el extremo inferior derecho de la figura 4 representa la entrega de paquetes desde la capa de multiplexación inversa L+1 hacia la capa de recepción L. Al igual que en el extremo emisor, normalmente entre estas dos capas L L+1 el ancho de banda (BW) disponible es elevado, por tratarse de una distancia muy corta.

La figura 5 muestra una implementación específica del procedimiento de la invención, en la que la capa de multiplexación inversa se implementa bajo la capa IP, que a su vez tiene por encima la capa TCP/UDP, que reciben paquetes de la capa de aplicación. En este ejemplo, la capa de multiplexación inversa realiza el control de flujo de paquetes que se deben entregar al nivel MAC, que a su vez entrega los paquetes a la capa física. Cada capa concreta actúa de acuerdo con la explicación de la figura 4. Además, en este ejemplo particular, se incluye la opción de recuperación de pérdidas de paquetes reconocimiento negativo en la etapa (6). En el extremo receptor se activa un mecanismo de solicitud de reconocimiento negativo (NAKRequester) y en el extremo emisor se activa el

mecanismo equivalente (NAKHandling).

5 A continuación se describen dos posibles aplicaciones del procedimiento de la invención. Estas aplicaciones llevan embebida internamente la implementación de las capas de multiplexación inversa y demultiplexación inversa como sendos componentes a modo de Proxy para paquetes RTP (Protocolo de Transporte en Tiempo Real). Ambas aplicaciones industriales están pensadas para el envío de vídeo de alta calidad, en tiempo real y bajo retardo, mediante la multiplexación inversa de la invención de canales de comunicaciones móviles 2.5G y 3G.

10 La figura 6 muestra de forma esquemática una de las aplicaciones industriales de la invención que consiste en una unidad móvil de reportero de TV ligera y transportable. La unidad móvil usa una implementación del procedimiento de multiplexación inversa para la transmisión de vídeo de alta calidad en tiempo real (vídeo-conferencia) a través de múltiples interfaces de red móvil HSDPA (3.5G).

La figura 7 muestra de forma esquemática un sistema de vídeo-vigilancia de tráfico que usa el procedimiento de multiplexación inversa de la invención para la transmisión de vídeo de alta calidad en tiempo real bajo demanda a través de múltiples interfaces de red móvil 3G y 2.5G (GPRS).

15 En ambas implementaciones (figuras 6 y 7), el demultiplexor es capaz de priorizar paquetes en la cola de entrada en función del tipo de paquete para codificaciones de vídeo tales como H.264 AVC, MPEG-4 y MPEG-TS y, en caso de producirse saturación en la cola de entrada (etapa (2) representada por el InputQueue de la figura 2), aplicar un descarte inteligente solo de los paquetes no prioritarios. A modo de ejemplo, que no debe considerarse limitativo, pueden descartarse paquetes que no son I frames, y/o en un porcentaje configurable para no degradar la calidad de vídeo. Alternativamente pueden aplicarse otras políticas de descarte inteligente.

20 El procedimiento de la invención permite la multiplexación inversa de canales heterogéneos y de ancho de banda variable en tiempo real y con bajo retardo (menor de 2 segundos entre la aplicación emisora y la receptora).

25 La implementación es configurable y extensible a otros tipos de paquetes y sistemas de codificación. El multiplexador es configurable con los algoritmos de ordenación descritos anteriormente. Ambas implementaciones están concebidas a nivel de aplicación como sendos proxy: SIP y RTSP respectivamente. Internamente, dichos proxy trabajan con paquetes RTP y aplican el control de flujo inverso objetivo de la invención descrita anteriormente.

A la vista de esta descripción y juego de dibujos, el experto en la materia podrá entender que la invención ha sido descrita según algunas realizaciones preferentes de la misma, pero pueden introducirse múltiples variaciones sin apartarse del objeto de la invención tal y como ha sido reivindicada.

REIVINDICACIONES

1. Un procedimiento de transmisión de paquetes entre una aplicación de un equipo emisor y una aplicación de un equipo receptor a través de una red de paquetes (40, 50), en el que la velocidad de transmisión de dichos paquetes es superior a la capacidad de transmisión del canal de transporte de una conexión de dicha red de paquetes (40, 50), que comprende la etapa de:
 - en un equipo emisor, almacenar en una cola de entrada (InputQueue) de una determinada capa de protocolo (Capa Demux, L+1), una pluralidad de paquetes (p_i) procedentes de una capa de protocolo de orden superior (IP, L);
 estando el procedimiento **caracterizado por** las etapas de:
 - en dicho equipo emisor, controlar, mediante una pluralidad de procesos (SenderQ₁, SenderQ₂... SenderQ_n) implementados en dicha determinada capa de protocolo (Capa Demux, L+1), la inserción de cada uno de dichos paquetes (p_i) en una cola de salida de una pluralidad de colas de salida (Output₁, Output₂... Output_n) de una capa de protocolo de orden inferior (Enlace, L-1), en el que cada una de dichas colas de salida (Output₁, Output₂... Output_n) está asociada de forma biunívoca a uno de dichos procesos (SenderQ₁, SenderQ₂... SenderQ_n) y cada una de dichas colas de salida (Output₁, Output₂... Output_n) está a su vez asociada a un determinado interfaz físico (NIC1, NIC2, ... NICn), estando dichos procesos (SenderQ₁, SenderQ₂... SenderQ_n) configurados para controlar la inserción de paquetes en dichas colas de salida (Output₁, Output₂... Output_n) en función del flujo de salida de paquetes de dichos interfaces físicos (NIC1, NIC2, ... NICn);
 - transmitir los paquetes insertados en dichas colas de salida (Output₁, Output₂... Output_n) asociadas a dichos interfaces físicos (NIC1, NIC2, ... NICn) a través de sendos canales de transporte de dicha red de paquetes (40, 50).
2. El procedimiento de la reivindicación 1, en el que al menos dos de dichos interfaces físicos (NIC1, NIC2, ... NICn) son diferentes entre sí.
3. El procedimiento de la reivindicación 1 ó 2, en el que al menos dos de dichos interfaces físicos (NIC1, NIC2, ... NICn) tienen velocidades máximas diferentes, transmitiéndose los paquetes a través de cada interfaz físico (NIC1, NIC2, ... NICn) a la velocidad del canal de transporte de dicho interfaz físico.
4. El procedimiento de cualquiera de las reivindicaciones anteriores, en el que dicha determinada capa de protocolo (Capa Demux, L+1) es una capa de demultiplexado o multiplexación inversa.
5. El procedimiento de cualquiera de las reivindicaciones anteriores, en el que dicha etapa de controlar, mediante dicha pluralidad de procesos (SenderQ₁, SenderQ₂... SenderQ_n), la inserción de paquetes (p_i) en dichas colas de salida (Output₁, Output₂... Output_n) comprende las etapas de:
 - cada uno de dichos procesos (SenderQ₁ SenderQ₂ ... SenderQ_n) monitoriza el estado de su correspondiente cola de salida (Output₁, Output₂... Output_n); y
 - dichos procesos (SenderQ₁ SenderQ₂ ... SenderQ_n) compiten entre sí para enviar paquetes (p_i) desde dicha cola de entrada (InputQueue) hacia su cola de salida correspondiente (Output₁, Output₂... Output_n).
6. El procedimiento de cualquiera de las reivindicaciones anteriores, en el que dicha etapa de controlar, mediante dicha pluralidad de procesos (SenderQ₁, SenderQ₂... SenderQ_n), la inserción de paquetes (p_i) en dichas colas de salida (Output₁, Output₂... Output_n) comprende las etapas de:
 - analizar, por parte de cada proceso (SenderQ₁, SenderQ₂... SenderQ_n), la carga de de paquetes acumulados en su cola de salida asociada (Output₁, Output₂... Output_n); y
 - si un proceso (SenderQ₁) tiene su cola de salida (Output₁) llena por encima de un cierto umbral (OutQMax), dicho proceso (SenderQ₁) no trata de desencolar un determinado paquete (p_i) de la cola de entrada (InputQueue), mientras que
 - si el número de paquetes de una cola de salida (Output_n) es inferior (OutQ) o igual a un cierto umbral (OutQMax), el proceso (SenderQ_n) que controla dicha cola de salida (Output_n) trata de sacar de la cola un determinado paquete (p_i) de la cola de entrada (InputQueue) para insertarlo en su cola de salida asociada (Output_n), consiguiéndolo el proceso que invoque a la cola de entrada (InputQueue) antes.
7. El procedimiento de cualquiera de las reivindicaciones anteriores, en el que cada cola de salida (Output₁... Output_n) se vacía según la velocidad del interfaz físico asociado a la misma (NIC1, NIC2, ... NICn).

8. El procedimiento de cualquiera de las reivindicaciones anteriores, en el que antes de dicha etapa de almacenar en dicha cola de entrada (InputQueue) dicha pluralidad de paquetes (p_i) procedentes de una capa de protocolo de orden superior (IP, L), se produce la etapa de:
- 5 - re-encapsular dichos paquetes (p_i) y asignarles un número de secuencia respectivo para facilitar su posterior ordenación en el extremo receptor.
9. El procedimiento de cualquiera de las reivindicaciones anteriores, en el que dicha determinada capa de protocolo (Capa Demux, L+1) se implementa en la capa de aplicación y, más concretamente, a nivel del Protocolo de Transporte en Tiempo Real (RTP).
10. Un procedimiento de recepción de paquetes por una aplicación de un equipo receptor, en el que dichos paquetes han sido enviados a través de una red de paquetes (40, 50), y en el que la velocidad de transmisión de dichos paquetes es superior a la capacidad de transmisión del canal de transporte de una conexión de dicha red de paquetes (40, 50), **caracterizado porque** dichos paquetes han sido enviados según un procedimiento de transmisión de paquetes de acuerdo con cualquiera de las reivindicaciones 1 a 9.
15. El procedimiento de cualquiera de las reivindicaciones anteriores, que comprende la etapa de ordenación y recuperación de paquetes mediante la implementación de un mecanismo basado en uno de los siguientes aspectos: en la duplicación de paquetes para minimizar la probabilidad de pérdida, en una ventana de tiempo, en una ventana de secuencia, en una diferencia de marca de tiempo, en la dispersión dentro de una ventana de ordenación.
20. El procedimiento de cualquiera de las reivindicaciones anteriores, que comprende la etapa de recuperación de paquetes por reconocimiento negativo.
13. El procedimiento de cualquiera de las reivindicaciones anteriores, en el que dichos paquetes se transmiten a través de una red de telefonía móvil que se conecta a dicha red de paquetes (40, 50) y dichos interfaces físicos (NIC1, NIC2, ... NICn) se eligen del siguiente grupo: GPRS, EDGE, 3G, HSUPA y Ethernet.
14. Un sistema de transmisión de paquetes que comprende:
- 25 - un equipo emisor que comprende una determinada aplicación;
- una red de paquetes (40, 50);
- un equipo receptor que comprende esa misma determinada aplicación,
- en el que la velocidad de transmisión de dichos paquetes es superior a la capacidad de transmisión del canal de transporte de una conexión de dicha red de paquetes (40, 50);
- 30 **caracterizado porque** dicho equipo emisor comprende medios para llevar a cabo las etapas del procedimiento de cualquiera de las reivindicaciones 1-9.
15. El sistema de la reivindicación 14, en el que dicho equipo receptor comprende medios para llevar a cabo el procedimiento de la reivindicación 10.

35

40

45

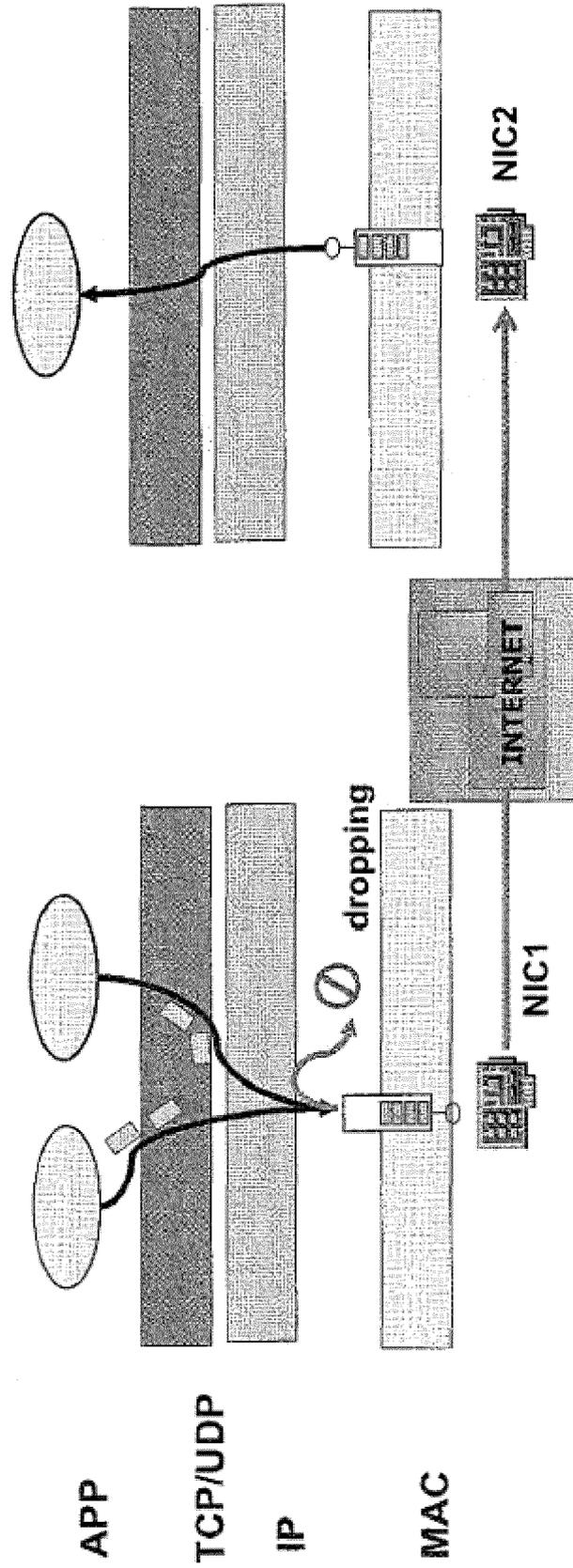


FIG. 1

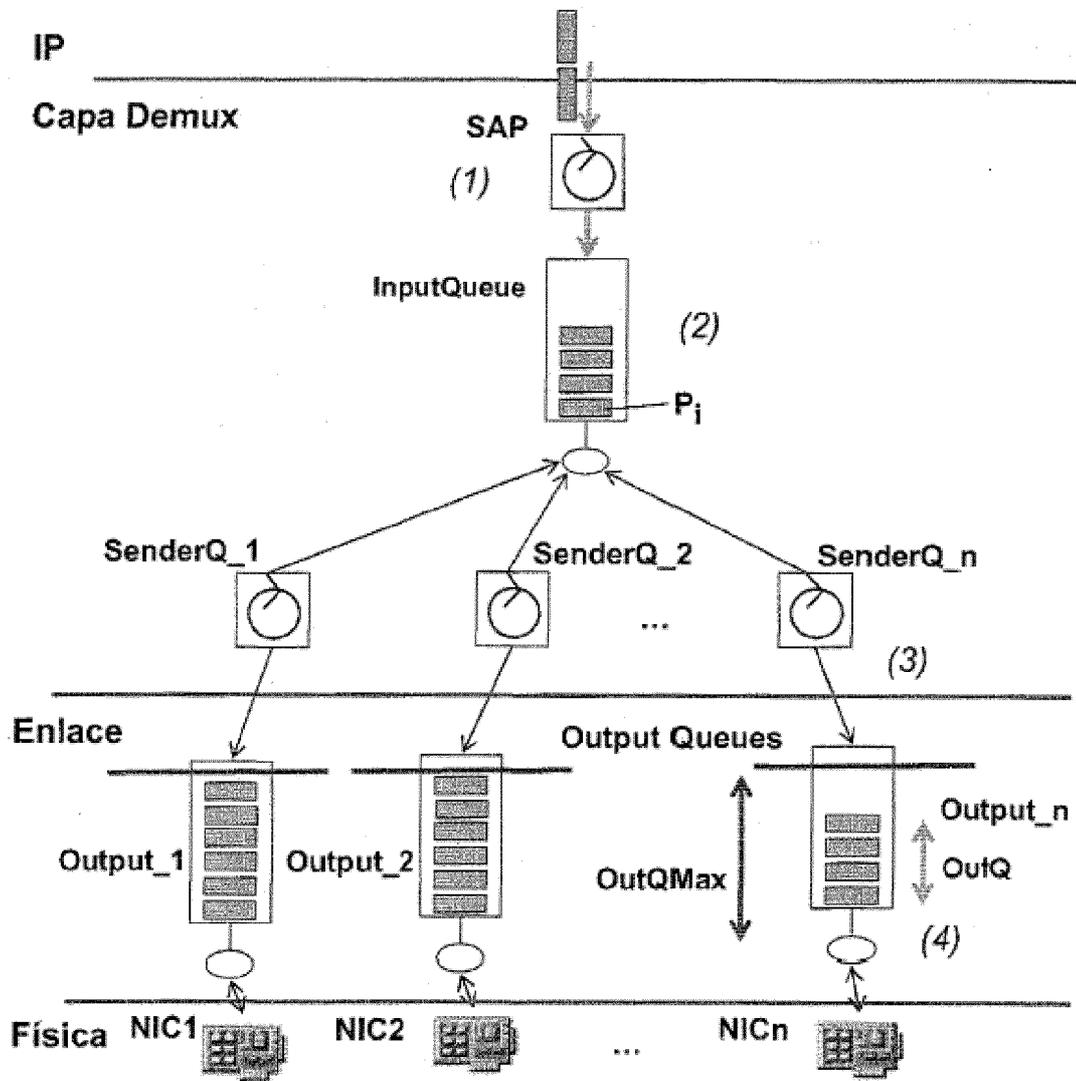


FIG. 2

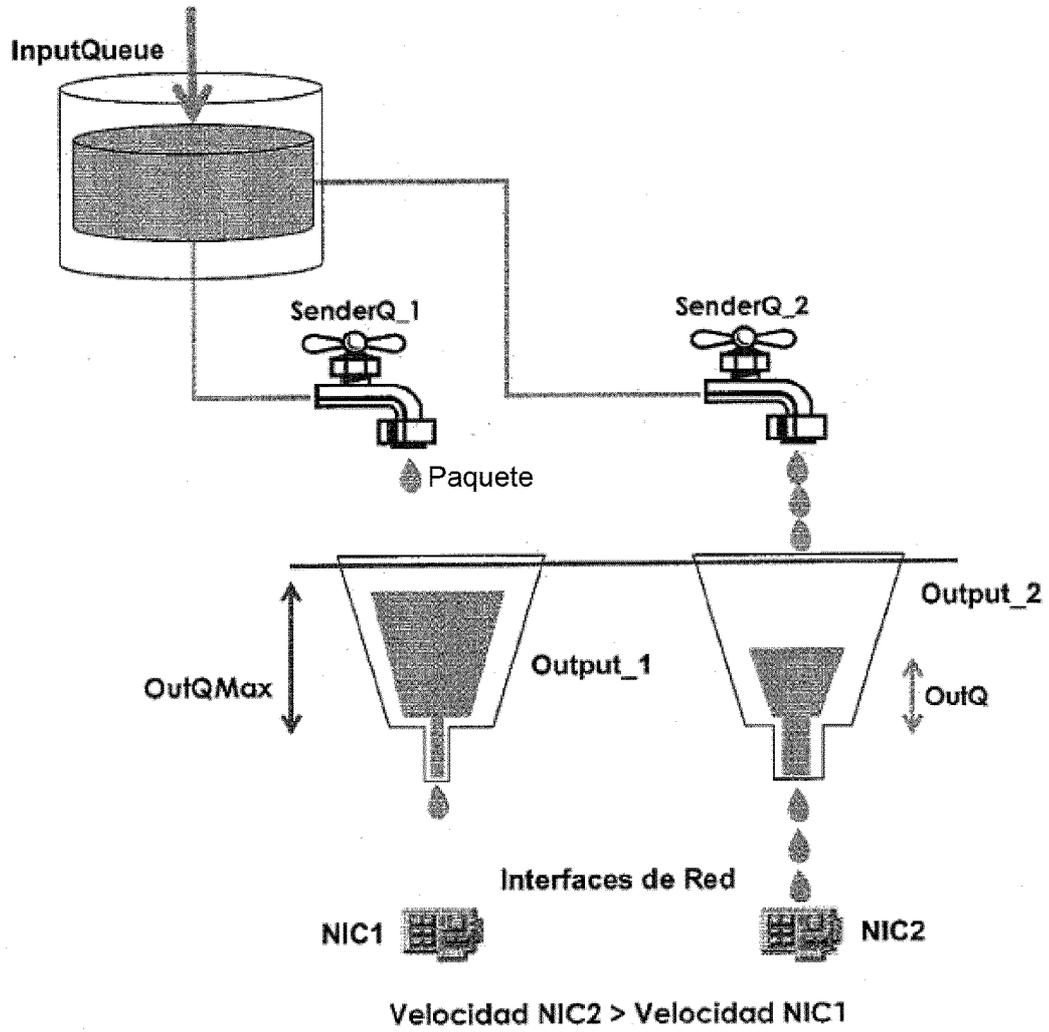


FIG. 3

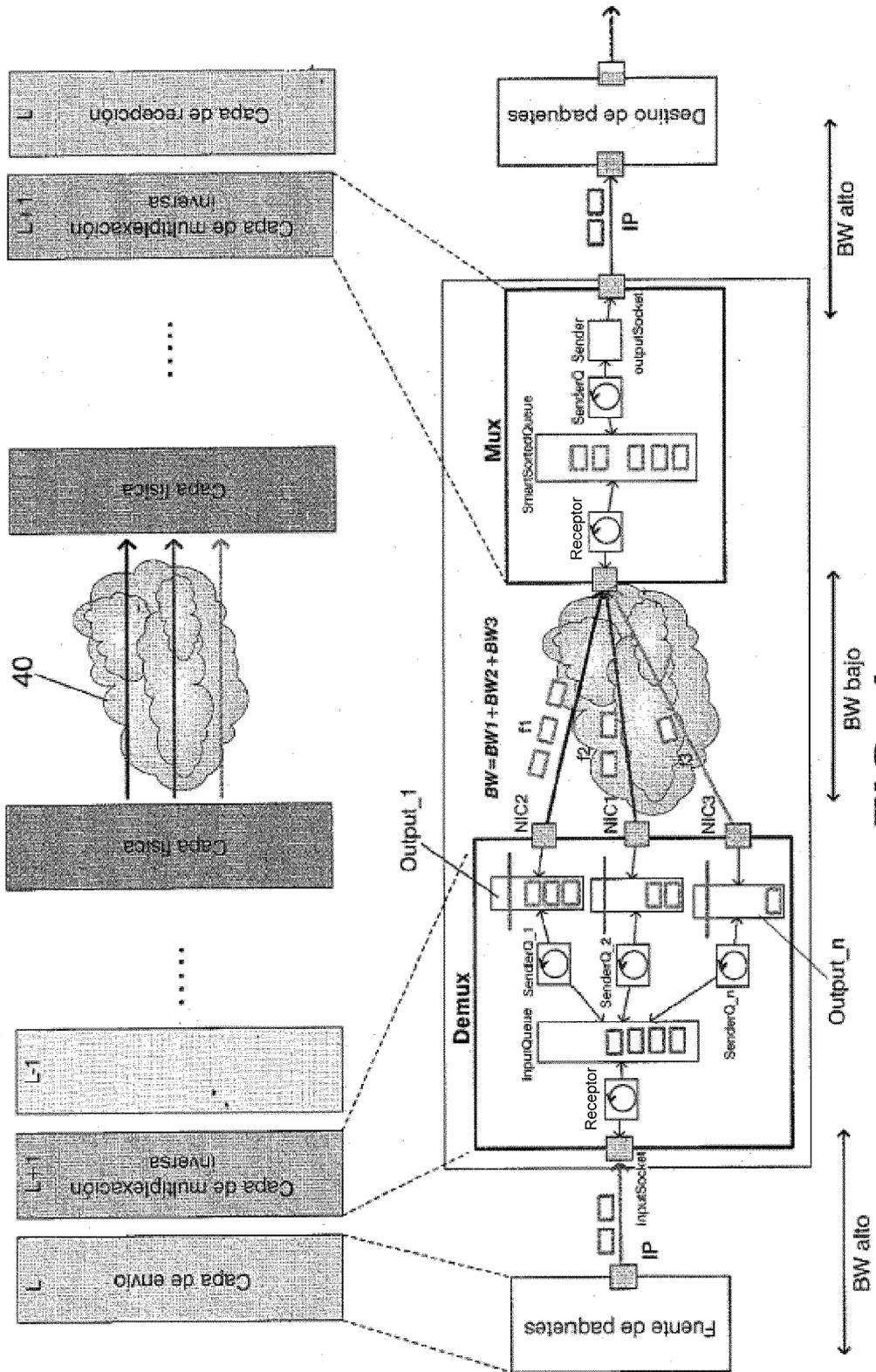


FIG. 4

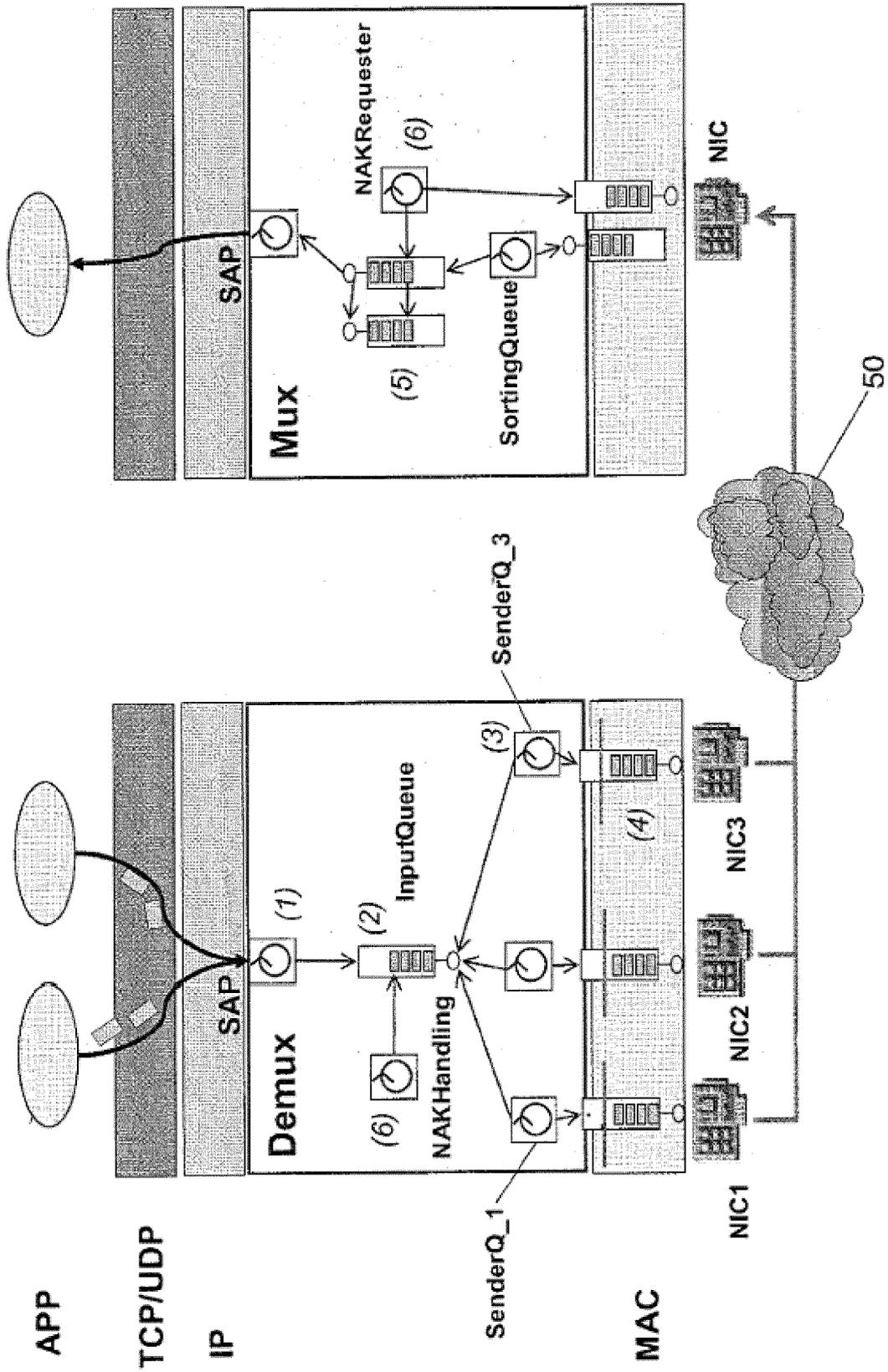


FIG. 5

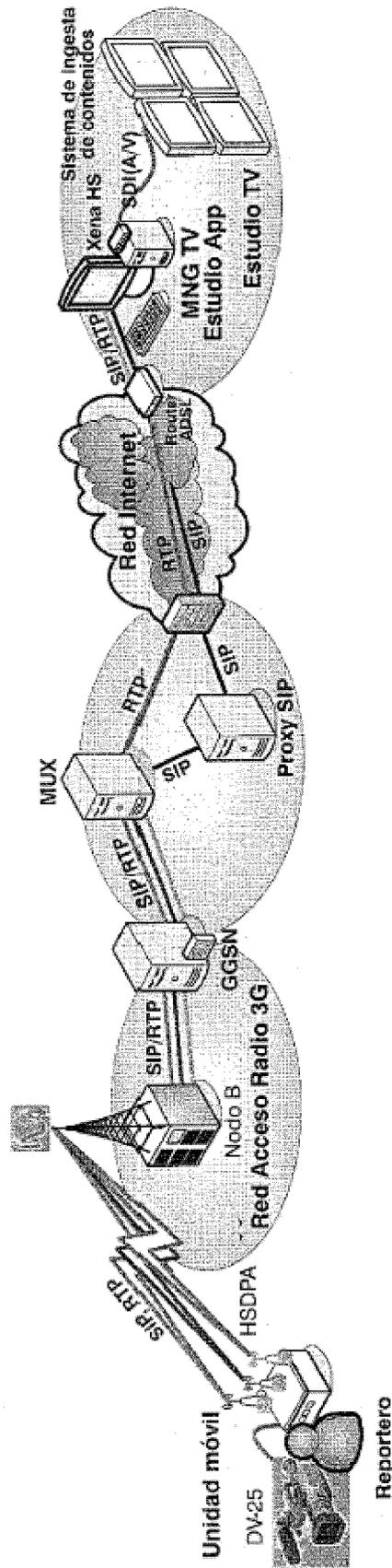


FIG. 6

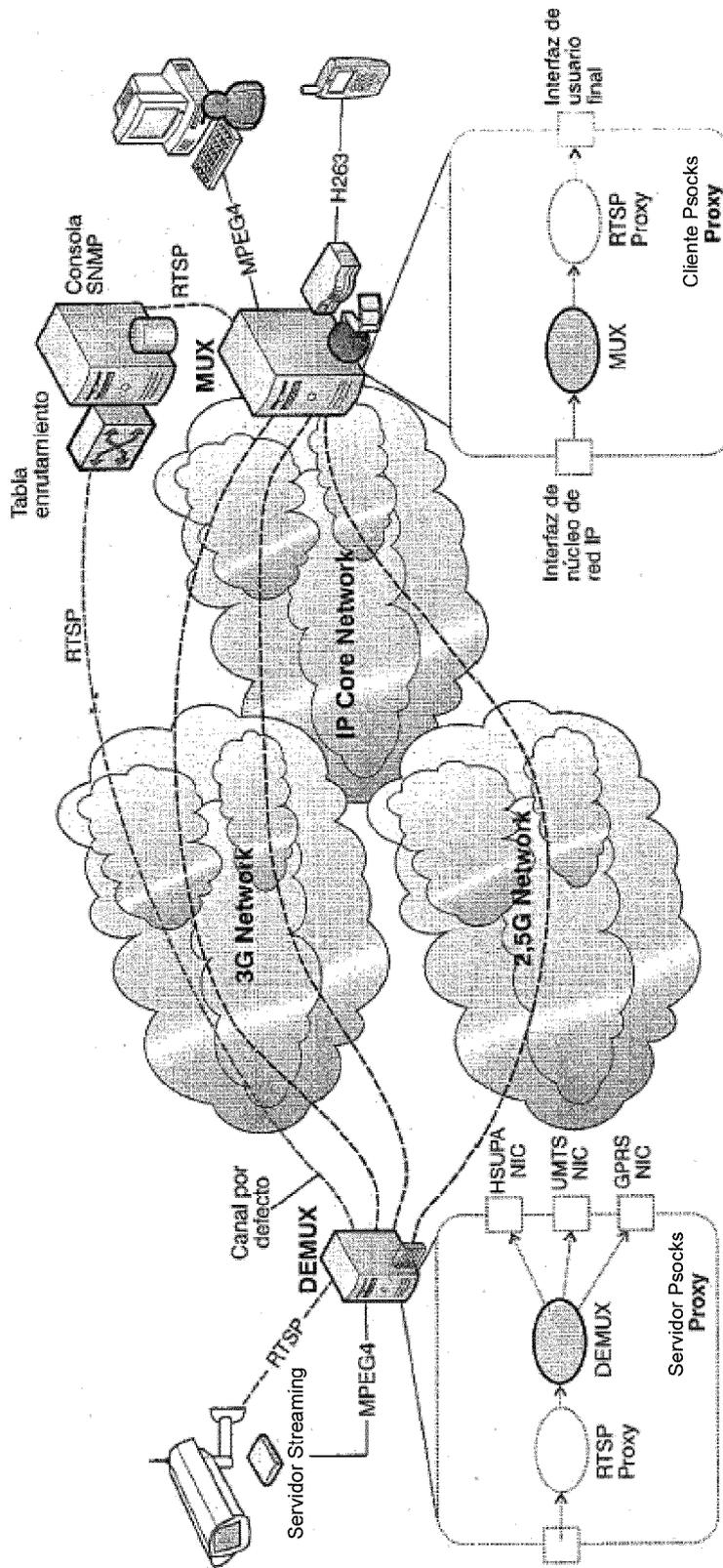


FIG. 7