



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 335 880**

51 Int. Cl.:
H04L 12/56 (2006.01)
H04Q 11/04 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **03818596 .3**
96 Fecha de presentación : **11.09.2003**
97 Número de publicación de la solicitud: **1665674**
97 Fecha de publicación de la solicitud: **07.06.2006**

54 Título: **Método para eliminar todos los segmentos correspondientes al mismo paquete en una memoria inter-media.**

45 Fecha de publicación de la mención BOPI:
06.04.2010

45 Fecha de la publicación del folleto de la patente:
06.04.2010

73 Titular/es: **Telefonaktiebolaget LM Ericsson (publ)**
164 83 Stockholm, SE

72 Inventor/es: **Meirick, Ingo;**
Nordström, Hakan;
Westerberg, Erik;
Schliwa-Bertling, Paul y
Ekström, Hannes

74 Agente: **Elzaburu Márquez, Alberto**

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

Método para eliminar todos los segmentos correspondientes al mismo paquete en una memoria intermedia.

5 Campo técnico

La presente invención se refiere generalmente a la gestión de paquetes de datos, y en particular a la gestión de memorias intermedias (buffer) de paquetes de datos en sistemas de comunicaciones de telefonía móvil.

10 Antecedentes

En un sistema de comunicaciones de telefonía móvil de hoy en día hay una tendencia cada vez mayor a usar equipos de usuario de telefonía móvil, unidades y teléfonos para otros servicios distintos a los servicios de llamada tradicional y de voz. Por ejemplo, un usuario puede por medio de su teléfono móvil o equipo terminal asociado, por ejemplo ordenador portátil o Personal Digital Assistance (PDA - Asistencia Digital Personal), acceder a Internet y descargar páginas Web al equipo de usuario de telefonía móvil. Además, fotografías, audio, vídeo y otros ficheros de datos pueden ser transmitidos entre diferentes unidades de telefonía móvil, pedidos a proveedores de contenido o servicio y/o descargados de páginas Web. Este aumento en la transmisión de paquetes de datos y, en particular, de paquetes de Internet Protocol (IP - Protocolo de Internet) en los sistemas de comunicaciones de telefonía móvil plantea elevadas demandas en el tratamiento de paquetes de datos a través del sistema de comunicaciones para garantizar el rendimiento satisfactorio en términos de tiempos de descarga, bajos tiempos de retardo, etc.

Una manera de aumentar el rendimiento de la transmisión de paquetes de datos y de tratar y, así, proporcionar tiempos de descarga satisfactorios para paquetes de datos en servicios de telefonía móvil relativos a IP/Transmission Control Protocol (TCP - Protocolo de Control de Transmisión), tal como General Packet Radio service (GPRS - Servicio de Radio en Paquetes General), Enhanced GPRS (EGPRS - GPRS Mejorado) y sistemas de Enhanced Data rates for Global Evolution (EDGE - Velocidades de Datos Mejoradas para Evolución Global)/GPRS, es proporcionar una gestión eficiente de las memorias intermedias (buffer) de datos que, temporalmente, almacenan los paquetes de datos durante su transmisión a través del sistema. En particular, la manera en la que son eliminados los paquetes de datos de las memorias intermedias (buffer) de datos que empiezan a llenarse impacta significativamente en el rendimiento de extremo a extremo del servicio.

En los servicios de GPRS, EGPRS y EDGE/GPRS de la técnica anterior, los paquetes de IP son segmentados en un Serving GPRS Support Node (SGSN - Nodo de Soporte de GPRS de Servicio) en un número de segmentos de paquetes de datos o Logical Link Control (LCC) Packet Data Units (PDUs) (Unidades de Datos en Paquetes con Control de Enlace Lógico) antes de su transmisión al Base Station System (BSS - Sistema de Estación de Base) y, eventualmente, al equipo de usuario de telefonía móvil relevante. Debido a la segmentación de los paquetes de IP, el BSS no tiene manera de identificar los segmentos (LLC PDUs) de su memoria intermedia (buffer) de datos asociada que constituyen un paquete de IP completo y, así, no puede llevar a cabo una gestión eficiente de la memoria intermedia (buffer). Tal solución tiene como resultado una gestión de la memoria intermedia (buffer) de datos lejos de ser satisfactoria, lo que lleva a tiempos de descarga innecesariamente largos de páginas Web, ficheros, fotografías, audio, vídeo y cualquier otra recepción de datos que utilice TCP/IP. Además, la solución existente hace al servicio de comunicaciones de telefonía móvil sensible a ajustes de parámetros relativos a TCP/IP, incluyendo el tamaño de ventana de TCP empleado.

El documento US 6.567.378 describe un conmutador de relé de celda que tiene una memoria intermedia (buffer) en colas para recibir una pluralidad de celdas de un paquete procedentes de una fuente. Las celdas son temporalmente almacenadas en la memoria intermedia (buffer) antes de su transmisión por un procesador de salida. Un controlador controla el almacenamiento de las celdas recibidas en la memoria intermedia (buffer) determinando un número total de celdas en el paquete y un número de celdas para ser recibidas de diferentes paquetes y decidiendo si la memoria intermedia (buffer) contiene suficiente espacio para almacenar las celdas recibidas basándose en el número total de celdas del paquete, en el número de celdas para ser recibidas de los diferentes paquetes y en las velocidades a las cuales las celdas son recibidas y eliminadas de la memoria intermedia (buffer).

El documento EP 1 100 285 describe el control de la congestión determinando cuándo un paquete correspondiente al inicio de una trama es presentado a una memoria intermedia (buffer), si el nivel de llenado de la memoria intermedia (buffer) excede un primer umbral de llenado predeterminado correspondiente a un estado de congestión. Si se excede el primer umbral, el paquete y todos los paquetes que pertenecen a la misma trama son sistemáticamente eliminados antes de llegar a la memoria intermedia (buffer).

El documento WO 01/93598 describe la transmisión de paquetes sobre un nodo de ATM. La transmisión se basa en la gestión del umbral de memoria intermedia (buffer) en lugar de en seguir canales virtuales individuales y es útil para casos en los cuales celdas de datos son encaminadas de acuerdo con su ruta virtual. En consecuencia, mientras que acepta datos de entrada, la memoria intermedia (buffer) tiene suficiente capacidad disponible para almacenar tramas completas de tantos canales virtuales como sea posible y para que siempre que la capacidad disponible de memoria intermedia (buffer) caiga por debajo de esa condición, todos los datos entrantes sean eliminados.

Sumario

La presente invención supera estos y otros inconvenientes de las disposiciones de la técnica anterior.

- 5 Un objeto general de la presente invención es proporcionar una gestión eficiente de los paquetes de datos en una memoria intermedia (buffer) de datos que comprende segmentos de paquetes de datos.

Otro objeto de la invención es permitir la identificación de un paquete de datos completo en una memoria intermedia (buffer) de datos que comprende segmentos de paquetes de datos.

- 10 Un objeto particular de la invención es proporcionar una gestión eficiente de las memorias intermedias (buffer) de datos que comprenden segmentos de paquetes de datos en un sistema de estación de base de un sistema de comunicaciones de telefonía móvil.

- 15 Estos y otros objetos son cumplidos por la invención tal como definen las reivindicaciones de patentes que se acompañan.

Brevemente, la presente invención implica la gestión de paquetes de datos en una memoria intermedia (buffer) de datos que comprende segmentos de paquetes de datos y, en particular, la identificación de paquetes de datos completos en tal memoria intermedia (buffer). La invención se refiere también a gestionar memorias intermedias (buffer) de datos que comprenden segmentos de paquetes de datos en un sistema de comunicaciones de telefonía móvil y, en particular, a eliminar paquetes de datos completos identificados de tales paquetes de datos.

- 25 De acuerdo con la presente invención, un base station system (BSS - Sistema de Estación de Base) recibe segmentos de paquetes de datos de un nodo de red que segmenta paquetes de datos en segmentos más pequeños antes de su transmisión al BSS. En el BSS, los segmentos de paquetes de datos entran en una cola de memoria intermedia (buffer) de datos, donde son, temporalmente, almacenados antes de enviarlos a un equipo de usuario de telefonía móvil. Cuando la memoria intermedia (buffer) empieza a llenarse, los segmentos tienen que ser eliminados de la misma. La presente invención proporciona una manera eficiente de eliminar segmentos permitiendo la identificación de esos segmentos que, juntos, constituyen un paquete de datos completo y a continuación elimina el paquete de datos completo identificado.

La identificación de un paquete de datos es llevada a cabo por el BSS analizando la información asociada con los segmentos de paquetes de datos. En una primera realización de la invención, la información comprende el tamaño de los segmentos individuales. Comparando por parejas el tamaño de un segmento de paquete de datos con el tamaño de un segmento consecutivo siguiente en una cola de memoria intermedia (buffer), el paquete de IP completo puede ser identificado. La comparación del tamaño incluye proporcionar y asociar un contador de segmentos k con un segmento de paquete de datos $P(k)$ actual. Si el tamaño $S(k)$ de este segmento $P(k)$ actualmente asociado con el contador k es menor que el tamaño $S(k+1)$ de un segmento consecutivo siguiente $P(k+1)$, este segmento siguiente $P(k+1)$ es identificado como el primer segmento de un paquete de datos completo y es asociado con un identificador PRIMERO. No obstante, si el tamaño $S(k)$ no es menor que el tamaño $S(k+1)$, el contador resulta ahora asociado con el siguiente segmento. La comparación de tamaño y la asociación de contador por etapas (segmento a segmento) continúa hasta que se encuentran dos segmentos vecinos en los que el tamaño $S(k)$ del segmento previo $P(k)$ es menor que el tamaño $S(k+1)$ del segmento consecutivo $P(k+1)$, o hasta que se alcanza el final de la cola de la memoria intermedia (buffer).

Una vez que el primer segmento del paquete de datos completo es encontrado en la memoria intermedia (buffer), se inicia la búsqueda del último segmento de paquete de datos completo. El contador de segmentos k se asocia ahora con el primer segmento identificado del paquete de datos completo. El tamaño $S(k)$ de este segmento $P(k)$ actual se compara entonces con el tamaño $S(k+1)$ de un segmento consecutivo $P(k+1)$ siguiente en la memoria intermedia (buffer). Siempre que dos tamaños $S(k)$, $S(k+1)$ son iguales el contador avanza una etapa, desplazándose por consiguiente segmento a segmento a través de la memoria intermedia (buffer) y comparando el tamaño de pares de segmentos hasta que los tamaños difieren, o hasta que se alcanza el extremo de la cola de la memoria intermedia (buffer). Si los tamaños $S(k)$, $S(k+1)$ difieren, el siguiente segmento $P(k+1)$ es identificado como el último segmento del paquete de datos completo y es asociado con un identificador ÚLTIMO. Así, se identifica que el paquete de datos completo comprende el segmento del paquete de datos asociado con el identificador PRIMERO, el segmento asociado con el identificador ÚLTIMO y cualquiera de los segmentos intermedios en la memoria intermedia (buffer). Una vez que este paquete de datos completo es identificado en la memoria intermedia (buffer), el BSS puede eliminarlo de ella.

En una realización preferida de la invención, el paquete de datos es un paquete de Internet Protocol (IP - Protocolo de Internet) proporcionado desde un servidor de Internet, proveedor de contenido o de servicios o equipo de usuario de telefonía móvil en un sistema de comunicaciones de General Packet Radio Service (GPRS - Servicio de Radio en Paquetes General; Enhanced GPRS (EGPRS - GPRS Mejorado), Enhanced Data rates for Global Evolution (EDGE - velocidades de Datos Mejoradas para Evolución Global)/GPRS. Típicamente, el paquete de IP es transmitido a un Gateway GPRS Support Node (GGSN - Nodo de Soporte con GPRS de Puerta de Enlace), que lo envía a un Serving GPRS Support Node (SGSN - Nodo de Soporte con GPRS de Servicio). Si el tamaño del paquete de IP recibido es mayor que un valor de umbral de tamaño, el SGSN segmenta el paquete de IP en un número de Logical Link Control Packet Data Units (LLC - PDUs - Unidades de Datos en Paquetes con Control de Enlace Lógico). Las LLC PDUs pueden entonces opcionalmente ser cifradas antes de ser enviadas en orden secuencial al BSS.

La información incluida en la cabecera de la LLC PDU (segmento de paquete de datos) por el SGSN puede incluir información de formato o de tipo, que identifica el formato de la carga útil de la LLC PDU asociada. Esta información de formato es extraída a continuación por el BSS y permite la discriminación entre LLC PDUs que contienen la carga útil del paquete de datos a partir de diferentes mensajes de control y paquetes transmitidos desde el SGSN al BSS.

Podría llevarse a cabo, alternativamente, una verificación de que los segmentos de paquete de datos (LLC PDUs) identificados como constituyentes de un paquete de datos (de IP) completo comprende realmente carga útil de IP, comparando el tamaño total de los segmentos identificados con los valores de umbral de tamaño. Puesto que el tamaño de los paquetes de IP típicamente difiere de manera significativa del tamaño de otros paquetes y mensajes, tales como los mensajes de control Transmission Control Protocol (TCP - Protocolo de Control de Transmisión), el BSC puede verificar que los segmentos identificados realmente contienen carga útil de paquetes de IP.

La invención ofrece las siguientes ventajas:

- Permite una eficiente gestión de la memoria intermedia (buffer) de datos en un sistema de comunicaciones, lo que aumenta el rendimiento del tratamiento de los paquetes de datos y la memoria intermedia (buffer);
- permite menores tiempos de descarga de páginas Web, ficheros, imágenes, audio, vídeo y otras recepciones de datos que utilizan TCP/IP;
- estabiliza el sistema de comunicaciones de telefonía móvil hacia diferentes ajustes de parámetros relacionados con el TCP/IP, incluyendo el tamaño de la ventana de TCP;
- reduce la complejidad y el coste de optimizar el sistema de comunicaciones de telefonía móvil con respecto a la transferencia de TCP/IP; y
- proporciona una reutilización de las técnicas de gestión de memoria intermedia (buffer) de las aplicaciones de Internet de línea-fija en el sistema de comunicaciones de telefonía móvil.

Otras ventajas ofrecidas por la presente invención resultarán evidentes con la lectura de la siguiente descripción de las realizaciones de la invención.

Breve descripción de los dibujos

La invención junto con otros objetos y ventajas de la misma, puede comprenderse mejor haciendo referencia a la siguiente descripción tomada junto con los dibujos que se acompañan, en los cuales:

la Fig. 1 es una vista global de un sistema de comunicaciones al que pueden aplicarse las enseñanzas de la presente invención;

la Fig. 2 ilustra esquemáticamente el tratamiento de un paquete de datos en el sistema de comunicaciones de la Fig. 1;

la Fig. 3 es un diagrama de bloques de una memoria intermedia (buffer) de datos de acuerdo con la presente invención que comprende segmentos de paquetes de datos;

la Fig. 4 es un diagrama de flujo que ilustra una realización de un método de gestionar una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

la Fig. 5 es un diagrama de flujo que ilustra una realización de la etapa de análisis de la Fig. 4 con más detalle;

la Fig. 6 es un diagrama de flujo que ilustra una realización de un método de identificar un primer segmento de paquete de datos de un paquete de datos completo en una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

la Fig. 7 es un diagrama de flujo que ilustra una realización de un método de identificar un último segmento de paquete de datos de un paquete de datos completo en una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

la Fig. 8 es un diagrama de flujo que ilustra una realización de un método de identificar un paquete de datos en una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

la Fig. 9 es un diagrama de flujo que ilustra otra realización de un método de identificar un paquete de datos completo en una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

la Fig. 10 es un diagrama de flujo que ilustra otra realización más de un método de identificar un paquete de datos completo en una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

las Figs. 11A y B son diagramas de flujo que ilustran otra realización de un método de gestionar una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

las Figs. 12A y B son diagramas de flujo que ilustran otra realización de un método de gestionar una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

las Figs. 13A y B son diagramas de flujo que ilustran otra realización más de un método de gestionar una memoria intermedia (buffer) de datos de acuerdo con la presente invención;

la Fig. 14 es un diagrama de bloques que ilustra esquemáticamente una realización de un sistema de estación de base de acuerdo con la presente invención;

la Fig. 15 es un diagrama de bloques que ilustra esquemáticamente una realización de un gestor de memoria intermedia (buffer) de datos de acuerdo con la presente invención;

la Fig. 16 es un diagrama de bloques que ilustra esquemáticamente una realización de un identificador de paquete de datos de acuerdo con la presente invención; y

la Fig. 17 es un diagrama de bloques que ilustra esquemáticamente una realización de un nodo de red de segmentación de un paquete de datos de acuerdo con la presente invención.

Descripción detallada

En los dibujos, se utilizarán los mismos números de referencia para elementos similares o correspondientes.

La presente invención se refiere generalmente a la gestión de paquetes de datos y en particular a la gestión de memorias intermedias (buffer) de datos que comprenden segmentos de paquetes de datos en un sistema de comunicaciones de telefonía móvil. La invención proporciona también métodos y sistemas para permitir la identificación de un paquete de datos completo en tales memorias intermedias (buffer) de datos que comprenden segmentos.

Con el fin de facilitar la comprensión de la presente invención se explica primero con referencia a la Fig. 1 un ejemplo de un sistema de comunicaciones 1, al cual pueden aplicarse las enseñanzas de la presente invención.

El sistema de comunicaciones 1 como se ilustra en la Fig. 1 es instanciado por un sistema de comunicaciones General Packet Radio Service (GPRS - Servicio de Radio en Paquetes General). No obstante, como el experto entiende, la invención no está limitada a esto sino que puede ser aplicada a otro sistema de comunicaciones de telefonía móvil que soporta la transmisión de paquetes de datos, en particular sistemas que soportan Transmission Control Protocol/Internet Protocol (TCP/IP - Protocolo de control de Transmisión/Protocolo de Internet) que incluyen, pero no están limitados a, sistemas de GPRS, Enhanced GPRS (EGPRS - EGPRS Mejorado) y Enhanced Data rates for Global Evolution (EDGE - Velocidades de datos Mejoradas para Evolución Global)/GPRS. A continuación los procesos de estructura y segmentación son descritos en términos del sistema de GPRS 1. Los procesos de estructura y segmentación para los sistemas de EGPRS y EDGE/GPRS son muy similares y cualquier experto debería ser capaz de entender cómo la descripción y principios siguientes son generalizados para tales sistemas.

El sistema de GPRS 1 incluye un número de diferentes sistemas o nodos de red, que se describen también en [1]. Los nodos de relevancia para la presente invención están representados en la Fig. 1. Siguiendo la notación de [1], los nodos relevantes son el Gateway GPRS Support Node (GGSN - Nodo de Soporte de GPRS de Puerta de Enlace) 300, el Serving GPRS Support Node (SGSN - Nodo de Soporte de GPRS de Servicio) 200 y el Base Station System (BSS - Sistema de Estación de Base) 100. El sistema 1 proporciona servicios al equipo de usuario 500 conectado o asociado al mismo. El equipo de usuario 500 es típicamente un sistema o teléfono de telefonía móvil independiente 520, un Personal Digital Assistant (PDA - Asistente Digital Personal), etc. En otra realización, el equipo de usuario 500 comprende una estación de telefonía móvil 520 conectada o asociada con equipo terminal 540. Este equipo terminal 540 es típicamente un ordenador portátil, un PDA o una entidad lógica separada de la propia estación de telefonía móvil 520. Un propósito del sistema de GPRS es encaminar paquetes de de datos (paquetes de IP) que entran en el sistema de GPRS, por ejemplo, desde un servidor de Internet 400, al equipo de usuario 500 correcto.

El paquete de datos podría incluir datos de una página Web, fichero, imagen, audio, vídeo o cualesquiera otros datos que un usuario quiera recibir en su estación de telefonía móvil 520 o equipo terminal 540. Estos datos son proporcionados desde un proveedor de contenido o servicio, ejemplificado por un servidor de Internet 400 en la Fig. 1, o desde otro equipo de usuario al equipo de usuario de telefonía móvil 500 en forma de paquete de datos y segmentos de paquetes de datos, lo que se describe con más detalle a continuación. En particular, la invención se aplica bien a datos que utilizan TCP/IP para recepción y transmisión de datos.

En lo que sigue un paquete de datos será ejemplificado por un paquete de IP y un segmento de tal paquete de IP se ejemplificará por una Logical Link Control Packet Data Unit (LLC-PDU - Unidad de Datos en Paquetes con Control de Enlace Lógico). No obstante la presente invención no está limitada a este ejemplo de paquete de datos y segmento de paquete de datos.

Un paquete de IP entra en el sistema de GPRS, típicamente, en la interfaz Gi entre un servidor de Internet 400 u otro proveedor de servicio o paquete, y el GGSN 300. El tamaño del paquete de IP está regulado por las especificaciones del Internet Engineering Task Force (IETF - Equipo de Trabajo de Ingeniería de Internet) y depende, entre otros, del tipo de aplicación, cliente (equipo de usuario) y preferencias del servidor. Basándose en la información interna del GGSN combinada con información del propio paquete de IP, el GGSN 300 encamina el paquete de IP al SGSN 200. El transporte de paquetes de IP desde el SGSN 200 al equipo de usuario 500 utiliza el protocolo de LLC, definido por ejemplo en [2]. Una LLC PDU consiste en una porción de cabecera y una porción de carga útil. Los paquetes de IP son transportados en la porción de carga útil. En cualquier transferencia específica desde el SGSN 200 al equipo de usuario 500 existe un tamaño máximo de las LLC PDUs. Este tamaño es, típicamente, negociado entre el equipo de usuario 500 y un SGSN 200 y puede ser diferente de una combinación de equipo de usuario - SGSN a otra. No obstante, dentro de un flujo de equipo de usuario el tamaño máximo negociado es preferiblemente fijo. El mayor valor posible actual para el tamaño máximo de la LLC PDU es 1500 bytes, pero muchas estaciones de telefonía móvil y otros equipos de usuario de telefonía móvil usan generalmente un valor máximo significativamente menor. El tamaño máximo negociado de las LLC PDUs permitido no es conocido en el BSS 100. Como se ha mencionado anteriormente, además del campo de carga útil la LLC PDU comprende 11 bytes de cabeceras. Esto significa que el máximo número de octetos de IP que una LLC PDU puede transferir es 11 bytes menor que el tamaño máximo negociado de las LLC PDUs.

Las especificaciones de 3GPP [1], [2] y [3] especifican que si el tamaño del paquete de IP más 11 bytes es menor o igual que el tamaño permitido máximo negociado de la LLC PDU entonces ese paquete de IP será transportado en una LLC PDU. Además, el campo de carga útil de esa LLC PDU contendrá preferiblemente ni más ni menos que ese paquete de IP. En este caso hay una relación de uno a uno entre la LLC PDU y el correspondiente paquete de IP.

En contraste, si el paquete de IP es mayor que el tamaño máximo de la LLC PDU menos 11 bytes entonces el paquete de IP será preferiblemente segmentado y transportado en más de una LLC PDU. De acuerdo con las especificaciones de 3GPP, la segmentación será tal que las LLC PDUs que contienen segmentos de un paquete de IP particular son enviadas en secuencia con numeración secuencial, P, P+1 ... P+N. La segmentación es además tal que todas excepto la LLC PDU P+N son del tamaño máximo permitido. Esto se ilustra también en la Fig. 2.

La Fig. 2 ilustra esquemáticamente el tratamiento de un paquete de IP en el sistema de GPRS de la Fig. 1. En este ejemplo el tamaño del paquete de IP es 1490 bytes. Con referencia tanto a la Fig. 1 como a la Fig. 2, primeramente el paquete de IP relevante es transmitido desde el servidor de Internet 400 u otro proveedor de datos por medio del GGSN 300 al SGSN 200. En este ejemplo el tamaño máximo de la LLC PDU (segmento de paquete de datos) está negociado a 500 bytes. Puesto que el tamaño total del paquete de IP excede el tamaño de segmento máximo en 500 bytes el SGSN 200 tiene que segmentar el paquete de IP recibido en un número de LLC PDUs. La primera LLC PDU comprende los bytes 1 - 489 del paquete de IP además de la cabecera de 11 bytes. La segunda LLC PDU comprende los bytes 490 - 978 y la tercera LLC PDU incluye los bytes 979 - 1467. La cuarta y última LLC PDU incluye los restantes bytes 1468 - 1490 del paquete de IP. Así, con los parámetros de tamaño anteriores, el paquete de IP recibido es segmentado en cuatro LLC PDUs consecutivas, en las que las primeras tres son del tamaño de segmento máximo.

En la siguiente etapa hay una opción de que el SGSN 200 cifre o encripte la o las LLC PDU o LLC PDUs generadas. Tras el cifrado la única información disponible para cualquiera que no esté en posesión de las clave o claves de cifrado o encriptado es el tamaño de la LLC PDU.

La o las LLC PDU o LLC PDUs (posiblemente cifradas) que contienen cada una bien paquetes de IP completos o porciones de ellos son a continuación transmitidas al BSS 100 para otro envío a través de la interfaz de radio al equipo de usuario 500. En el BSS 100, las LLC PDUs, típicamente, son introducidas en una memoria intermedia (buffer) de datos o en otra unidad de almacenamiento de paquetes de datos, donde pueden ser puestas en cola, priorizadas, retrasadas e incluso eliminadas. Las LLC PDUs no eliminadas son eventualmente transmitidas a través de la interfaz de radio al equipo de usuario apropiado. En el equipo de usuario los posiblemente segmentados paquetes de IP son retirados de las LLC PDUs y posiblemente descifrados. Los paquetes de IP que han sido segmentados son ensamblados de nuevo.

La Fig. 3 es un diagrama de bloques de una memoria intermedia (buffer) de datos 120 que comprende segmentos (LLC PDUs) de paquetes de datos completos (paquete de IP) en el BSS del sistema de comunicaciones de la Fig. 1. La memoria intermedia (buffer) 120 comprende una cola de LLC PDUs consecutivas. Cada LLC PDU tiene un cierto tamaño S(1) a S(N) para las LLC PDU s P(1) a P(N). En la memoria intermedia (buffer) 120 de la Fig. 3, las LLC PDUs P(3) a P(6), juntas constituyen un paquete de IP completo y el número total de LLC PDUs en la memoria intermedia (buffer) es N. La memoria intermedia (buffer) de datos 120 comprende preferiblemente LLC PDUs destinadas a un solo usuario. Este usuario típicamente tiene un acuerdo de servicio, por ejemplo suscripción, con un operador de red que gestiona el sistema de comunicaciones que alberga al BSS y a la memoria intermedia (buffer) 120. Tal suscripción es típicamente manifestada en un Subscriber Identify Module (SIM) que está dispuesto en o asociado con el equipo de usuario.

La Fig. 4 ilustra un diagrama de flujo de una realización de gestionar una memoria intermedia (buffer) de datos, tal como la memoria intermedia (buffer) 120 de la Fig. 3, en un BSS de un sistema de comunicaciones de acuerdo con la presente invención. El método empieza con el BSS analizando la información del data packet (DP) en su memoria intermedia (buffer) asociada en la etapa S1. La información del DP incluye el tamaño de las LLC PDUs. En la etapa

S2, el BSS identifica a continuación un paquete de IP completo en la memoria intermedia (buffer) basándose en la información del DP analizada. En otras palabras, el BSS identifica la LLC PDU o las LLC PDUs que incluyen juntas todos los datos (bytes) de un paquete de IP completo antes de que fuese segmentado. Una vez que se ha identificado un paquete de IP completo, el BSS elimina este paquete de IP completo de la memoria intermedia (buffer) en la etapa S3.

5 En una realización alternativa, en lugar de eliminar un paquete de IP identificado de la memoria intermedia (buffer), el BSS podría retardar o priorizar la transmisión del mismo al equipo de usuario, poner en cola el paquete de IP completo, o llevar a cabo algún otro tratamiento del mismo.

Por consiguiente, la presente invención proporciona un modo inteligente y ventajoso de gestionar memorias intermedias (buffer) de datos y eliminar paquetes de datos. Se ha probado tanto por la teoría como por la experiencia que la mejor práctica de gestionar una memoria intermedia (buffer) de segmentos de paquetes de datos (LLC PDUs) que empiezan a llenarse es eliminar exactamente un paquete de datos (paquete de IP), es decir, no una fracción (4) de un paquete de datos ni dos o más paquetes de datos consecutivos. En particular, el nivel de señalización de TCP en el sistema de comunicaciones que alberga al BSS y a su memoria intermedia (buffer) asociada mejora significativamente cuando se eliminan paquetes de IP completos. Además, eliminando paquetes de IP completos de la memoria intermedia (buffer) del BSS, los paquetes de IP son eliminados cerca del cuello de botella del sistema de comunicaciones. La presente invención puede emplear diferentes versiones de Active Queue Management (AQM) conocidas en el sector para determinar cuándo y, posiblemente qué paquete de datos completo, si existen varios en la memoria intermedia (buffer), debe ser eliminado.

La Fig. 5 es un diagrama de flujo que ilustra una realización de la etapa de análisis de la Fig. 4 con más detalle. En la etapa S10, el BSS recibe la información del DP del nodo de red que llevó a cabo la segmentación de los paquetes de IP en las LLC PDUs, es decir el SGSN de la Fig. 1. La información del DP es asociada con la LLC PDU y es preferiblemente incluida en un campo de fin de la cabecera de la LLC PDU, tal como en el campo de información de una PDU con BSS GPRS Protocol (BSSGP - Protocolo de GPRS del BSS) que incluye la LLC PDU asociada. La información del DP podría comprender información que identifique a la LLC PDU asociada como el último segmento de un paquete de IP completo, es decir, que comprende a los últimos bytes restantes del paquete de IP. En otra realización la información del DP podría comprender información que identifique a la LLC PDU asociada como el primer segmento de un paquete de datos completo, es decir, comprendiendo los primeros (máximo tamaño de segmento - tamaño de la cabecera) bytes del paquete de IP. También, la información que identifica a una LLC PDU asociada como una LLC PDU intermedia podría comprender información del DP.

En la etapa S11 opcional, el BSS recibe la información de tipo o de formato asociada con la LLC PDU. Esta información permite al BSS identificar el tipo de datos incluidos en la LLC PDU, y por consiguiente permite la discriminación entre datos de carga útil de IP y otros datos, por ejemplo Voice over IP (VoIP - Voz sobre IP) y diferentes paquetes de señal de control y mensajes, tales como mensajes de TCP. El BSS compara a continuación la información del DP recibida (y posiblemente la información de formato) con los identificadores del DP (identificadores de formato) para determinar si la LLC PDU asociada es el primero, el último o un segmento intermedio del paquete de IP de la etapa S12. Generalmente, esta comparación se lleva a cabo comparando la información del DP recibida con la secuencia de bits que previamente ha sido determinada como identificadores para las LLC PDUs primera, última o intermedia. El método continúa entonces a la etapa S2, en la que el BSS identifica un paquete de IP completo basándose en la información del DP asociada con las LLC PDUs. Si la información del DP incluye tanto un primer identificador del DP como un último identificador del DP, el BSS identifica que el paquete de IP completo comprende la carga útil de la LLC PDU que está asociada con el primer identificador del DP, estando la carga útil de la LLC PDU asociada con el último identificador del DP y la carga útil de cualesquiera LLC PDUs situadas entre estas dos LLC PDUs en la memoria intermedia (buffer) de datos. No obstante, si la información del DP sólo incluye un último identificador del DP, podría identificarse que un paquete de IP completo incluye la carga útil de las LLC PDU que van desde una LLC PDU consecutiva siguiente después de una primera LLC PDU que está asociada con el último identificador del DP a una segunda LLC PDU que está asociada con el último identificador del DP. De manera similar, si la información del DP sólo incluye un primer identificador del DP, podría identificarse un paquete de IP completo que incluye la carga útil de las LLC PDUs de una primera LLC PDU que está asociada con el primer identificador del DP a una LLC PDU precedente vecina de una segunda LLC PDU que está asociada con el primer identificador del DP.

Alternativamente a recibir la información del DP que permite al BSS identificar un paquete de IP completo en su memoria intermedia (buffer) de LLC PDUs consecutivas, el propio BSS puede generar esta información del DP.

La Fig. 6 ilustra una realización de un método de identificar una primera LLC PDU de un paquete de datos completo en una memoria intermedia (buffer) de datos que comprende LLC PDUs. Este método podría ser llevado a cabo por el BSS para permitir la gestión eficiente de su memoria intermedia (buffer) y para eliminar paquetes de IP completos de la misma.

El método empieza en la etapa S20, en la que se proporciona y asocia con una de las LLC PDUs de la memoria intermedia (buffer) un contador de segmentos o puntero k. En la figura, el contador k es asociado con la LLC PDU número X (P(X)). X puede ser un número arbitrario de 1 a N, donde N es el número total de LLC PDUs en la cola de la memoria intermedia (buffer) de datos. No obstante, en la mayoría de las aplicaciones X es 1, es decir, el método empieza con la primera LLC PDU de la memoria intermedia (buffer), o al menos una de las pocas primeras LLC PDUs de la memoria intermedia (buffer). Dónde en la cola de la memoria intermedia (buffer) se inicia la búsqueda de paquetes de datos completos puede ser determinado mediante un protocolo de AQM. Además, se proporciona y

asocia con la misma LLC PDU P(X) un identificador o puntero PRIMERO de la primera LLC PDU de un paquete de IP completo como el contador de segmentos k.

En la siguiente etapa S21, el tamaño S(k) de la LLC PDU P(k) actualmente asociada con el contador k es comparado con el correspondiente tamaño S(k+1) de la siguiente LLC PDU P(k+1) consecutiva en la cola de la memoria intermedia (buffer). Si el tamaño S(k+1) de esta LLC PDU P(k+1) siguiente excede el tamaño S(k) de la LLC PDU P(k) actual, el método continúa a la etapa 23 en la que el identificador PRIMERO se asocia con este siguiente P(k+1) de LLC PDU. Así, esta LLC PDU P(k+1) de LLC PDU siguiente es, mediante la asociación con el identificador PRIMERO, identificado como la primera LLC PDU de un paquete de datos completo. El método termina entonces.

No obstante, si el tamaño S(k+1) de la siguiente LLC PDU P(k+1) no excede el tamaño S(k) de la LLC PDU S(k) actual, el contador de segmentos k es incrementado en una unidad en la etapa S22, permitiendo así comprobar la siguiente LLC PDU. El método continúa ahora a la etapa S21, donde la comparación del tamaño se repite hasta que el tamaño S(k+1) de la siguiente LLC PDU P(k+1) excede el tamaño S(k) de la LLC PDU P(k) que está actualmente asociada con el contador k. En otras palabras, empleando el bucle que comprende la etapa S21 y S22 el método en función de la etapa (segmento por segmento) examina las LLC PDUs de la memoria intermedia (buffer) empezando desde la LLC PDU número X y comparando el tamaño de dos LLC PDUs vecinas hasta que la primera LLC PDU de un paquete de IP completo es identificada (o se alcanza el final de la memoria intermedia (buffer)).

Como alternativa a la condición de tamaño en la etapa S21 de comparación ($S(k+1) > S(k)$), la condición y comparación de tamaño $S(k+1) = S(k)$ podría ser llevada a cabo. A continuación, si los dos tamaños S(k), S(k+1) son iguales el contador k de segmentos es incrementado en la etapa S22. No obstante, si S(k+1) difiere de S(k), el método continúa a la etapa S23 en la que el identificador PRIMERO es asociado con el P(k+2) de LLC PDU, es decir con el segmento P(k+2) del paquete de datos encontrado dos posiciones después del segmento P(k) actual en la cola de la memoria intermedia (buffer). Además, en otra realización de la etapa de comparación de tamaño S21, se usa la condición de tamaño $S(k+1) < S(k)$. Si el tamaño S(k+1) de la siguiente LLC PDU P(k+1) es igual o mayor que el tamaño S(k) de la LLC PDU P(k) actual, el contador k de segmentos es incrementado en la etapa S22. No obstante, si S(k+1) es menor que S(k), el identificador PRIMERO es asociado con la LLC PDU P(k+2) encontrada dos posiciones después de la LLC PDU P(k) actual en la memoria intermedia (buffer) en la Fig. S23.

El método de la Fig. 6 se aplicará ahora, como ejemplo, a la memoria intermedia (buffer) de la Fig. 3. En este ejemplo X se elige que sea 1, es decir, la investigación se inicia desde la primera LLC PDU P(1) en la memoria intermedia (buffer).

En la etapa S20 el contador k de segmentos es puesto a 1 y es, de este modo, asociado con el P(1) de LLC PDU. El identificador PRIMERO es asociado también con este segmento P(1). El tamaño S(1) de la LLC PDU P(1) es comparado a continuación con el tamaño S(2) de la siguiente (segunda) LLC PDU P(2). Puesto que el tamaño S(2) de la segunda LLC PDU P(2) es menor que el tamaño S(1), el método continúa a la etapa S22, en la que el contador k es incrementado de 1 a 2. En otras palabras, el contador k es ahora asociado con la segunda LLC PDU P(2) en la memoria intermedia (buffer). El tamaño S(2) es comparado con el correspondiente tamaño S(3) de la tercera LLC PDU P(3) en la memoria intermedia (buffer) en la etapa S21. En este caso, el tamaño S(3) es mayor que el tamaño S(2) de manera que la condición de la etapa S21 se cumple y el método avanza hacia la etapa S23. En esta etapa S23, el identificador PRIMERO es puesto igual a k+1, en este caso 3. De este modo, el identificador PRIMERO es asociado con la tercera LLC PDU P(3) de la memoria intermedia (buffer), que ahora se identifica como la primera LLC PDU de un paquete de datos completo.

La Fig. 7 ilustra una realización de un método de identificar una última LLC PDU de un paquete de IP completo en una memoria intermedia (buffer) de datos que comprende LLC PDUs. Este método podría ser llevado a cabo por el BSS para permitir una gestión eficiente de su memoria intermedia (buffer) y para eliminar de la misma paquetes de datos.

El método empieza en la etapa S30, en la que se proporciona un contador de segmentos o puntero k y se asocia con una de las LLC PDUs de la memoria intermedia (buffer). En la Figura, el contador k es asociado con la LLC PDU número X (P(X)). X puede ser un número arbitrario de 1 a N, en el que N es el número total de LLC PDUs en la cola de la memoria intermedia (buffer). No obstante, en la mayoría de las aplicaciones el método de identificar el último segmento de la Fig. 7 se usa preferiblemente en combinación con el método de la Fig. 6 que identifica la primera LLC PDU de un paquete de IP completo. En tal caso, X es preferiblemente igual a PRIMERO, es decir el método se inicia con la primera LLC PDU de un paquete de IP completo de la memoria intermedia (buffer) identificada usando el método de la Fig. 6 ó identificada de alguna otra manera, por ejemplo por medio de la información del DP recibida desde el SGSN. Además, se proporciona un identificador o puntero ÚLTIMO de la última LLC PDU de un paquete de IP completo y se asocia con el mismo P(X) de LLC PDU como el contador k de segmentos.

De manera correspondiente a la etapa S21 de la Fig. 6, en la etapa S31 se lleva a cabo una comparación de tamaño usando el tamaño S(k) de la LLC PDU P(k) actualmente asociada con el contador k y el tamaño S(k+1) de una LLC PDU P(k+1) consecutiva siguiente en la cola de la memoria intermedia (buffer). Si los dos tamaños S(k), S(k+1) difieren el método continúa a la etapa S33, en la que el identificador ÚLTIMO es puesto a k+1, es decir es asociado con la siguiente P(k+1) de LLC PDU. Así, esta LLC PDU P(k+1) es identificada como la última LLC PDU de un paquete de IP completo y el método termina.

No obstante, si los dos tamaños $S(k)$, $S(k+1)$ son iguales, como se determina en la etapa S31, el método continúa hacia la etapa S32 en la que el contador k de segmentos es incrementado en una unidad. El contador k se asocia ahora con la siguiente LLC PDU de la memoria intermedia (buffer). Las etapas S31 y S32 se repiten hasta que el tamaño de dos LLC PDUs vecinas difiere y, así, la última LLC PDU de un paquete de IP es identificada (o se alcanza el final de la memoria intermedia (buffer)).

Como alternativa a la condición de tamaño en la etapa de comparación S31 ($S(k+1)=S(k)$), puede llevarse a cabo la condición y comparación del tamaño $S(k+1)<S(k)$. A continuación, si el tamaño $S(k+1)$ de la siguiente LLC PDU $P(k+1)$ consecutiva siguiente es menor que el tamaño $S(k)$ de la LLC PDU $P(k)$ actual, el método continúa hacia la etapa S33, o bien se lleva a cabo la etapa S32. En otra realización de la etapa de comparación de tamaño S31, se usa la condición de tamaño $S(k+1)>S(k)$. En tal caso, si el tamaño $S(k+1)$ de la siguiente LLC PDU $P(k+1)$ es igual o menor que el tamaño $S(k)$ de la LLC PDU $P(k)$ actual el contador k de segmentos es incrementado en la etapa S32, o bien el identificador ÚLTIMO se asocia con la LLC PDU $P(k)$ actual en la etapa S33.

El método de la Fig. 7 se aplicará ahora, a modo de ejemplo, a la memoria intermedia (buffer) de la Fig. 3. En este ejemplo se elige que X sea 3 (PRIMERO), es decir la investigación se inicia desde la primera LLC PDU (3) del paquete de IP completo de la memoria intermedia (buffer).

En la etapa S30 el contador k de segmentos es puesto a 3 y es, de este modo, asociado con el $P(3)$ de LLC PDU. El identificador ÚLTIMO es asociado también con este segmento $P(3)$. El tamaño $S(3)$ es comparado a continuación con el tamaño $S(4)$ del siguiente (cuarto) $P(4)$ de LLC PDU. Puesto que dos tamaños $S(3)$ y $S(4)$ son iguales al método continúa hacia la etapa S32, en la que el contador k es incrementado en una unidad y es ahora asociado con el cuarto $P(4)$ de LLC PDU. La comparación de tamaño de la etapa S31 se repite pero ahora se comparan los tamaños de segmentos $S(4)$ y $S(5)$. Los tamaños de segmentos son iguales y la etapa S32 es llevada a cabo de nuevo, incrementando el contador k a 5. En la etapa S31, el tamaño $S(5)$ es comparado con el tamaño $S(6)$ del sexto $P(6)$ de LLC PDU. En este caso, $S(6)$ es menor que $S(5)$ y la condición de tamaño de la etapa S31 no se cumple de manera que el método avanza hacia la etapa S33. El identificador ÚLTIMO es ahora puesto a $k+1$, es decir 6, y, de este modo, es asociado con el sexto $P(6)$ de LLC PDU. Esta LLC PDU $P(6)$ es identificada como la última LLC PDU del paquete de IP completo y el método finaliza.

Con el fin de identificar un paquete de IP completo en una memoria intermedia (buffer), tanto la primera como la última LLC PDU tienen que ser identificadas (o la primera (última) LLC PDU de un primer paquete de IP completo en combinación con la primera (última) LLC PDU de un segundo paquete de IP completo). La Fig. 8 es una ilustración de una realización de un método de identificar un paquete de IP completo en una memoria intermedia (buffer) de datos que comprende segmentos de LLC PDU. Esta realización es básicamente una combinación del método de identificación del primer segmento de la Fig. 6 y el método de identificación del último segmento de la Fig. 7.

El método se inicia en la etapa S40, en la que un contador de segmentos o puntero k es proporcionado y asociado con un $P(X)$ de LLC PDU en la memoria intermedia (buffer) de datos. Como se ha explicado anteriormente, esta LLC PDU $P(X)$ puede ser una LLC PDU arbitraria, por ejemplo seleccionada por una unidad AQM asociada, pero es seleccionada preferiblemente desde las primeras pocas LLC PDUs, tales que la primera LLC PDU de la memoria intermedia (buffer) ($X=1$). En la siguiente etapa S41, se investiga si ésta es la primera LLC PDU de un paquete de IP completo. Esta investigación se lleva a cabo comparando el tamaño $S(k)$ de la LLC PDU $P(k)$ actualmente asociada con el contador k con el tamaño $S(k+1)$ de una siguiente LLC PDU consecutiva $P(k+1)$. Si el tamaño $S(k+1)$ es mayor que el tamaño $S(k)$ el método continúa a la etapa S43, o bien el contador de segmentos es en función de las etapas (segmento a segmento) incrementado en la etapa S42 hasta que la condición de tamaño de la etapa S41 se cumple. En la etapa S43, un identificador PRIMERO de la primera LLC PDU de un paquete de IP completo es puesto a $k+1$, es decir es asociado con el $P(k+1)$ de LLC PDU. El contador k es entonces incrementado en una unidad. La etapa S44 investiga a continuación si la LLC PDU es una LLC PDU intermedia o una LLC PDU última del paquete de IP completo llevando a cabo una comparación de tamaño. En la etapa S44, si el tamaño $S(k)$ de la LLC PDU $P(k+1)$ consecutiva difiere del tamaño $S(k+1)$ de una siguiente LLC PDU $P(k+1)$ consecutiva, el método continúa a la etapa S46. De otro modo, el contador k es "en función de los segmentos" incrementado en una unidad en la etapa S45, hasta que se identifica la última LLC PDU de un paquete de IP completo. Un identificador ÚLTIMO de la última LLC PDU de un paquete de IP completo es a continuación puesto a $k+1$, es decir es asociado con el $P(k+1)$ de LLC PDU, en la etapa S46. A partir de entonces, en la etapa S47, se identifica que el paquete de IP completo en la memoria intermedia (buffer) de datos comprende la LLC PDU $P(\text{PRIMERA})$ asociada con el identificador PRIMERO, la LLC PDU $P(\text{ÚLTIMA})$ con el identificador ÚLTIMO y cualesquiera LLC PDUs intermedias situadas entre estas dos LLC PDUs ($P(\text{PRIMERA})$ y $P(\text{ÚLTIMA})$) en la cola de la memoria intermedia (buffer).

El método de la Fig. 8 identificado anteriormente podría llevarse a cabo como un sub método del método de gestión de la memoria intermedia (buffer) de datos de la Fig. 4. En tal caso, la información del DP podría comprender el identificador o el puntero PRIMERO y ÚLTIMO permitiendo la identificación de un paquete de IP completo. El método continúa entonces desde la etapa S47 a la etapa S3 de la Fig. 4, en el que se elimina el paquete de IP completo ahora identificado.

ES 2 335 880 T3

El método de la Fig. 8 se aplicará ahora, como ejemplo, a la memoria intermedia (buffer) de la Fig. 3. En este ejemplo se elige que X sea 1, es decir la investigación se inicia desde la primera LLC PDU P(1) de la memoria intermedia (buffer).

En la etapa S40 el contador k de segmentos es puesto a 1 y es, así, asociado con la LLC PDU P(1). El tamaño S (1) de la LLC PDU P(1) es comparado a continuación con el tamaño $S(2)$ de la siguiente (segunda) LLC PDU P(2). Puesto que el tamaño $S(2)$ es menor que el tamaño $S(1)$, el método continúa a la etapa S42, en la que el contador k es incrementado de 1 a 2. El tamaño $S(2)$ es comparado a continuación con el correspondiente tamaño $S(3)$ de la tercera LLC PDU P(3) en la memoria intermedia (buffer) de la etapa S41. En este caso, el tamaño $S(3)$ es mayor que el tamaño $S(2)$ de manera que la condición de la etapa S41 se cumple y el método continúa a la etapa S43. En esta etapa S43, el identificador PRIMERO es puesto igual a $k+1$, en este caso 3. Así, la tercera LLC PDU P(3) de la memoria intermedia (buffer) es identificada como la primera LLC PDU de un paquete de IP completo. Además, el contador es incrementado de 2 a 3. Los tamaños $S(4)$ y $S(3)$ son comparados en la etapa S44 y desde que la condición se cumple el contador k es incrementado en una unidad llegando a 4 en la etapa S45. De manera correspondiente, los tamaños $S(5)$ y $S(4)$ y los tamaños $S(6)$ y $S(5)$ son comparados en la etapa S44, respectivamente. $S(6)$ es menor que $S(5)$ y se sale del bucle de la etapa S44 y S45 y se entra en la etapa S46, en la que el identificador ÚLTIMO es puesto a 6. Finalmente, en la etapa S47 se identifica que el paquete de IP completo comprende (la carga útil de) la LLC PDU P (3), P(4), P(5) y P(6).

La Fig. 9 es una ilustración de otra realización de un método de identificar un paquete de IP completo en una memoria intermedia (buffer) de datos que comprende segmentos de LLC PDU. Esta realización es básicamente una combinación de los dos métodos de identificar dos segmentos primeros de la Fig. 6.

Las etapas S50 a S53 corresponden a S40 a S43 de la Fig. 8 y no se explican más. En la etapa S54 se lleva a cabo una comparación de tamaño por medio de los tamaños de segmento $S(k)$ y $S(k+1)$ de dos LLC PDUs vecinas P(k), P($k+1$). El contador k de segmentos es incrementado en una unidad en la etapa S55, "saltando" así en función de la etapa de una LLC PDU a la siguiente LLC PDU a través de la memoria intermedia (buffer), hasta que el tamaño $S(k)$ de la LLC PDU P(k) actualmente asociada con el contador k es menor que el tamaño $S(k+1)$ de una LLC PDU P($k+1$) consecutiva siguiente. En la etapa S56, la última LLC PDU de un paquete de IP completo es identificada como la LLC PDU P(k) y el identificador ÚLTIMO es puesto al valor de k . La siguiente etapa S57 corresponde a la etapa S47 de la Fig. 8.

La Fig. 10 es una ilustración de otra realización de un método de identificar un paquete de IP completo en una memoria intermedia (buffer) de datos que comprende segmentos de LLC PDU. Esta realización es básicamente una combinación de los dos métodos consecutivos de identificar el último segmento de la Fig. 7.

La etapa S60 corresponde a la etapa S40 de la Fig. 8 y no se explica más. La comparación de tamaño de la etapa S61 y el aumento en función de la etapa (segmento a segmento) del contador l de segmentos de la etapa S62 se repite hasta que el tamaño $S(k)$ de la LLC PDU P(k) actualmente asociado con el contador k difiere del tamaño $S(k+1)$ correspondiente de una LLC PDU consecutiva P($k+1$) siguiente. Si los tamaños difieren, el identificador PRIMERO es puesto a $k+2$, es decir la LLC PDU P($k+2$) es identificada como la primera LLC PDU de un paquete de IP completo. A partir de entonces el contador k de segmentos es incrementado en 2. Las etapas restantes S64 a S67 del método corresponden a las etapas S44 a S47 de la Fig. 8.

El método de identificar el primer segmento de la Fig. 6 ó el método de identificar el último segmento de la Fig. 7 podrían, alternativamente, combinarse con la recepción de la información del DP (Fig. 5) que identifica la última LLC PDU del paquete de IP completo o de la primera LLC PDU del paquete de IP, respectivamente. La identificación del paquete de IP se lleva a cabo a continuación basándose tanto en la información del DP recibida desde el SGSN y en la información obtenida a través de las comparaciones de los tamaños de segmentos.

Las Figs. 11A y B son diagramas de flujo de una realización de un método de gestionar una memoria intermedia (buffer) de datos que comprende segmentos (LLC PDUs) de paquetes de datos (paquetes de IP) de acuerdo con la presente invención. En las Figs. 11A y B, un parámetro N corresponde con el número total de LLC PDUs en la cola de la memoria intermedia (buffer). Un parámetro MINTAMAÑO es el mínimo tamaño de un paquete de IP y un parámetro MAXTAMAÑO es el máximo tamaño de un paquete de IP que va a ser eliminado de una memoria intermedia (buffer). Estos umbrales de tamaño son empleados para discriminar entre paquetes de IP y otros paquetes que no deben ser eliminados. Tales otros paquetes y mensajes tienen generalmente un tamaño que difiere significativamente de un tamaño típico de un paquete de IP. Un ejemplo de un paquete de datos que no debe ser eliminado es la memoria intermedia (buffer) es mensajes de control de TCP o paquetes, tales como mensajes de control de synchronize (SYN - Sincronizar) y SYN acknowledgement (ACK - Reconocimiento). Tales mensajes son de manera típica significativamente más pequeños que los paquetes de IP y generalmente tienen un tamaño menor de 100 bytes. Haciendo a continuación MINTAMAÑO igual a 100, tales mensajes de control de TCP/paquetes son protegidos de ser eliminados. Como se ha mencionado en lo anterior, el máximo tamaño de un paquete de IP es regulado por las especificaciones de IETF y depende del tipo de aplicación, cliente, preferencias de servidor, etc.

El método se inicia en la etapa S70, en la que el contador k de segmentos se proporciona y asocia con una LLC PDU P(X) en la ($k+1$). De manera correspondiente, los identificadores PRIMERO y ÚLTIMO son asociados con esta

LLC PDU P(X). Un parámetro de tamaño TAMAÑO es puesto al valor del tamaño de segmento S(X) de esta LLC PDU P(X). En la etapa S71, se investiga, comparando el contador k con el número total N de LLC PDUs en la memoria intermedia (buffer), si se está actualmente mirando en la última LLC PDU P(N) de la cola de la memoria intermedia (buffer). Si la LLC PDU actual P(k) es la última LLC PDU P(N) de la memoria intermedia (buffer) el método continúa a la etapa S72 opcional, en la que el identificador ÚLTIMO se hace igual al identificador PRIMERO, es decir son asociados con una y la misma LLC PDU. El método avanza a continuación hacia la etapa S73 opcional, en la que la LLC PDU asociada con los identificadores PRIMERO y ÚLTIMO es eliminada. El método termina entonces.

No obstante, si la LLC PDU actual P(k) no es la última, su tamaño S(k) se compara con el tamaño S(k+1) de una LLC PDU consecutiva P(k+1) siguiente en la etapa S74. Si el tamaño S(k+1) es mayor que el tamaño S(k) el método continúa hacia la etapa S76, o bien el contador k se incrementa en una unidad en la etapa S75. El bucle de la etapa S71, S74 y S75 se repite entonces hasta que se identifica que la primera LLC PDU del paquete de IP completo se determina cuando S(k+1) es mayor que S(k), como se determinó en la etapa S74, o se alcanzó el final de la cola de la memoria intermedia (buffer), como se determinó en la etapa S71. En la etapa S76, el identificador PRIMERO es entonces asociado con la primera LLC PDU P(k+1) identificada del paquete de IP completo, es decir, es puesto a (k+1). El contador k es entonces incrementado en una unidad y el parámetro TAMAÑO es puesto al valor del tamaño S(PRIMERO) de esta primera LLC PDU.

A partir de entonces, la etapa S77 investiga si el contador k de segmentos es igual a N, es decir si la LLC PDU P(k) actual es la última LLC PDU P(N) de la memoria intermedia (buffer). Si k es igual a N, el identificador ÚLTIMO es puesto al valor del contador k, es decir, se asocia con la LLC PDU P(k) actual, en la etapa S78. El método continúa entonces hacia la etapa S82. Si la LLC PDU P(k) actual no es la última LLC PDU P(N) de la memoria intermedia (buffer), las etapas S77, S79 y S80 se repiten hasta que se identifica la última LLC PDU del paquete de IP completo o se alcanza el final de la memoria intermedia (buffer). La etapa S79 determina si el tamaño S(k) de la LLC PDU P(k) actual es igual al tamaño S(k+1) de una LLC PDU P(k+1) consecutiva siguiente en la cola de la memoria intermedia (buffer). Si los tamaños S(k), S(k+1) son iguales el contador k se incrementa en una unidad en la etapa S80. A partir de entonces, el tamaño S(k) (que corresponde a S(k+1) en la etapa S79 previa) es añadido al parámetro TAMAÑO. Una vez que S(k+1) difiere de S(k) el método continúa desde la etapa S79 a la etapa S81.

En la etapa S81, el identificador ÚLTIMO es puesto a k+1 y, así, es asociado con la última LLC PDU identificada ahora del paquete de IP completo. Además, el tamaño S(ÚLTIMO) de esta LLC PDU P(ÚLTIMA) identificada es añadida al parámetro TAMAÑO. El parámetro TAMAÑO es comparado con el valor de umbral MINTAMAÑO en la etapa S83. Si TAMAÑO es menor que MINTAMAÑO o mayor que MAXTAMAÑO el método se repite básicamente haciendo X igual a ÚLTIMO en la etapa S85 y retrocediendo a la etapa S70. De este modo, el método se repite para tratar de identificar un (segundo) paquete de IP completo en la memoria intermedia (buffer) que cumple los límites de tamaño. No obstante, si el tamaño está dentro de los límites, el paquete de IP completo identificado (LLC PDU P(PRIMERA) a LLC PDU P(ÚLTIMA)) es eliminado en la etapa S84. El método termina entonces.

Debe observarse que la primera LLC PDU (P(1) si X=1) en la cola de la memoria intermedia (buffer) no es eliminada si es posible identificar un paquete de IP completo en la cola restante. La razón para esto es que una primera parte del paquete de IP, al cual pertenece la LLC PDU, podría haberse transmitido ya desde la memoria intermedia (buffer). Sólo si no es posible identificar un paquete de IP en la memoria intermedia (buffer), la primera LLC PDU puede ser eliminada.

Las Figs. 12A y B son diagramas de flujo de otra realización de un método de gestionar una memoria intermedia (buffer) de datos que comprende segmentos (LLC PDU) de paquete de datos (paquetes de IP) de acuerdo con la presente invención. Las etapas S90 a S100 corresponden a las etapas S70 a S80 de la Fig. 11A y no se explicarán más.

En la etapa S99, se determina que el tamaño S(k+1) de una LLC PDU P(k+1) consecutiva siguiente difiere del tamaño S(k) de la LLC PDU P(k) actual, continuando el método hacia la comparación del tamaño de la etapa S101. En esta etapa S101, se determina si el tamaño S(k) es menor que el tamaño S(k+1). Si S(k) es el menor de los dos tamaños S(k), S(k+1), el identificador ÚLTIMO es asociado con la LLC PDU P(k) actual, es decir, se hace igual al valor (o posición en la cola de la memoria intermedia (buffer)) del contador k de segmentos en la etapa S103. El método continúa entonces hacia la etapa S104. No obstante, si S(k+1) es mayor o igual que S(k), el identificador ÚLTIMO es puesto a k+1 (asociado con la siguiente LLC PDU P(k+1)) en la etapa S102. Además, el tamaño S(ÚLTIMO) de esta LLC PDU P(ÚLTIMA) asociada ÚLTIMA es añadido al parámetro de tamaño TAMAÑO. El método avanza entonces hacia la etapa S104. Las etapas S104 a S107 corresponden a las etapas S82 a S85 de la Fig. 11B y no se explican más.

Las Figs. 13A y B son diagramas de flujo de otra realización de un método de gestionar una memoria intermedia (buffer) de datos que comprende segmentos (LLC PDUs) de paquetes de datos (paquetes 25) de acuerdo con la presente invención. De manera similar al método representado en las Figs. 11A y B, un parámetro N corresponde al número total de LLC PDUs de la memoria intermedia (buffer). Un parámetro MIN es el mínimo tamaño de una LLC PDU y un parámetro MAX es el máximo tamaño de un paquete de IP que va a ser eliminado de la memoria intermedia (buffer). Estos umbrales de tamaño se emplean para discriminar entre paquetes de IP y otros paquetes que no deberían ser eliminados.

El método se inicia en la etapa S110, en la que se proporciona un contador k de segmentos y se asocia con una LLC PDU P(X) en la memoria intermedia (buffer). De manera correspondiente, los identificadores PRIMERA y ÚLTIMA son asociados con esta LLC PDU P(X). Un parámetro de tamaño TAMAÑO se pone al valor del tamaño S(X) de esta LLC PDU P(X). La etapa S111 determina si se está mirando a la última LLC PDU de la memoria intermedia (buffer), o si el tamaño acumulado es mayor que el valor de umbral MAX. Esta última comprobación es para asegurarse de que no se intenta buscar la primera LLC PDU de un paquete de IP durante más tiempo del necesario. Si la LLC PDU P(k) actual es el último segmento de la memoria intermedia (buffer), es decir k es igual a N, o TAMAÑO > MAX el método continúa hacia la etapa S122, en la que desde la LLC PDU P(PRIMERA) a la LLC PDU P(ÚLTIMA) son eliminadas. En este caso, puesto que PRIMERA=ÚLTIMA=X la LLC PDU P(X) es eliminada. El método termina entonces. No obstante, si las condiciones de la etapa S111 no se cumplen el método continúa hacia la etapa S112, que investiga si se está mirando a la primera LLC PDU de un paquete de IP completo. En la etapa S112, el tamaño S(k) de la LLC PDU P(k) actual se compara con el tamaño S(k+1) de un P(k+1) de LLC PDU consecutivo siguiente en la cola de la memoria intermedia (buffer). Además, se comprueba si el campo del formato de esta siguiente LLC PDU P(k+1) está puesto a UI, indicando carga útil. El tamaño S(k+1) se compara también con el valor de umbral MIN. Las condiciones de la etapa S112 aseguran que la primera LLC PDU del paquete de IP completo es seleccionada sólo entre las LLC PDUs de tamaño correcto (mayor de MIN) y que comprenden la carga útil del paquete de IP (UI). Si no (todas) las condiciones de la etapa S112 se cumplen el contador k es incrementado en una unidad y el tamaño de la LLC PDU P(k) actualizada que está actualmente asociada con el contador k es añadido al parámetro TAMAÑO. El bucle de las etapas S111, S112 y S113 se repite entonces hasta que k es igual a N o TAMAÑO es mayor que MAX, desplazándose hacia la etapa S122, o el S(k+1) es mayor que S(k) y el formato FORMATO(k+1) de la siguiente LLC PDU P(k+1) es la carga útil UI y el tamaño S(k+1) es mayor que el parámetro MIN, desplazándose hacia la etapa S114.

En la etapa S114, el identificador PRIMERO es asociado con la primera LLC PDU P(k+1) de un paquete de IP completo, es decir es puesto a k+1. El contador k es entonces incrementado en una unidad y el parámetro TAMAÑO es puesto al valor del tamaño S(PRIMERO) de esta primera LLC PDU.

A partir de entonces, la etapa S115 investiga si el contador k de segmentos es igual a N, es decir si la LLC PDU P(k) actual es la última LLC PDU de la memoria intermedia (buffer) o si el parámetro TAMAÑO es mayor que el umbral MAX. Si k es igual a N o TAMAÑO es mayor que MAX, la etapa S120 pone el identificador ÚLTIMA al valor del contador k, es decir, que está asociado con la LLC PDU P(k) actual. El método continúa entonces hacia la etapa S121 en la que se identifica que el paquete de IP completo comprende desde la LLC PDU P(PRIMERA) hasta la LLC PDU P(ÚLTIMA) en la memoria intermedia (buffer). Si la LLC PDU P(k) actual no es la última LLC PDU P(N) de la memoria intermedia (buffer) y TAMAÑO es menor o igual que MAX las etapas S115, S116 y S117 se repiten hasta que se identifica la última LLC PDU del paquete de IP completo, o k=N o TAMAÑO > MAX. La etapa S116 determina si el tamaño S(k) de la LLC PDU P(k) actual es igual al tamaño S(k+1) de una LLC PDU P(k+1) consecutiva siguiente en la cola de la memoria intermedia (buffer) y si el campo de formato de esta LLC PDU P(k+1) siguiente está puesto a la carga útil UI. Si los tamaños S(k), S(k+1) son iguales y FORMATO(k+1)=UI, el contador k es incrementado en una unidad en la etapa S117. A partir de entonces, el tamaño S(k) (que corresponde a S(k+1) en la etapa S116 previa) es añadido al parámetro TAMAÑO. Una vez que S(k+1) difiere de S(k) o bien FORMATO(k+1) no es la carga útil UI el método continúa desde la etapa S116 hacia la etapa S118. Esta etapa S118 determina si el campo del formato de la LLC PDU P(k+1) es carga útil. Si es positivo, el identificador ÚLTIMA es asociado con esta LLC PDU P(k+1) siguiente en la etapa S119, considerando que si el formato no es carga útil el identificador ÚLTIMA es asociado con la LLC PDU P(k) actual en la etapa S120. El método continúa entonces hacia la etapa S121 donde el paquete de IP completo es identificado como LLC PDU P(PRIMERA), LLC PDU (ÚLTIMA) y cualesquiera LLC PDUs intermedias. El paquete de IP identificado es a continuación eliminado en la etapa S122, que a continuación finaliza el método.

La Fig. 14 es un diagrama de bloques esquemático de un base station system (BSS - Sistema de Estación de Base) 100 de acuerdo con la presente invención. El BSS 100 comprende una unidad de entrada y salida (I/O) 110 adaptada para conducir la comunicación con diferentes nodos de red en el sistema de comunicaciones que alberga al BSS 100 y con diferentes equipos de usuario de telefonía móvil. La unidad de I/O 110 está en particular adaptada para recibir LLC PDUs (segmentos de paquetes de datos) desde un nodo SGSN y para transmitir LLC PDUs al equipo de usuario relevante. El BSS 100 incluye también una memoria intermedia (buffer) de datos 120 u otra unidad de almacenamiento para, al menos temporalmente, almacenar las LLC PDUs recibidas desde el SGSN hasta que sean, posiblemente, enviadas al equipo de usuario. La memoria intermedia (buffer) de datos 120 está preferiblemente configurada para almacenar las LLC PDUs en una cola de segmentos, donde las LLC PDUs se sitúan consecutivamente en el orden en el que son recibidas desde el SGSN a través de la unidad de I/O 110. En una realización preferida todas las LLC PDUs en la cola de datos son asociadas y destinadas para una única aplicación de de telefonía móvil de usuario. En tal caso, la memoria intermedia (buffer) de datos 120 podría ser configurada para comprender varias colas de LLC PDUs, estando cada una de tales colas asociada con un único equipo de usuario o de usuarios. La Fig. 3 es una ilustración más detallada de una memoria intermedia (buffer) de datos 120 con una cola de LLC PDUs consecutivas de acuerdo con la presente invención. El BSS 100 incluye también un gestor de memoria intermedia (buffer) o medios de gestión 130, que gestiona la memoria intermedia (buffer) 120, por ejemplo introducir, eliminar y analizar LLC PDUs.

Las unidades 110 y 130 del BSS 100 pueden ser proporcionadas como software, hardware o una combinación de los mismos. La unidad de I/O 110, la memoria intermedia (buffer) 120 y el gestor de memoria intermedia (buffer) 130 pueden ser interpretados por ejemplo juntos en un único nodo de red de estación de base del BSS 100. Alternativa-

mente, también es posible una implementación distribuida con algunas otras unidades proporcionadas en diferentes nodos de red del sistema de estación de base.

La Fig. 15 es un diagrama de bloques esquemático que ilustra una realización del gestor de memoria intermedia (buffer) 130 de la Fig. 14 con más detalle. El gestor 130 incluye un analizador 132 de info, adaptado para analizar información asociada con los segmentos de paquetes de datos (LLC PDUs). En una realización el analizador 140 está adaptado para llevar a cabo una comparación de tamaño de LLC PDUs consecutivas en la memoria intermedia (buffer) de datos para proporcionar los identificadores PRIMERA y SEGUNDA y asociarlos con las LLC PDUs identificadas correctas. En cada caso, la información del DP (que comprende los identificadores PRIMERA y ÚLTIMA) es proporcionada a un paquete de IP en la memoria intermedia (buffer) asociada basándose en la información del DP. El gestor de memoria intermedia (buffer) 130 comprende también un segmento introductor o medios de inserción 134 adaptados para insertar LLC PDUs recibidas del nodo SGSN del sistema de comunicaciones en la memoria intermedia (buffer) de datos asociada. Un eliminador 136 de paquetes de datos es proporcionado en el gestor 130 para eliminar paquetes de IP completos identificados por el identificador 140 de paquetes de datos de la memoria intermedia (buffer) de datos. El eliminador 136 podría ser configurado también para eliminar únicas LLC PDUs, o grupos de LLC PDUs, por ejemplo si no se ha identificado ningún paquete de IP completo. El eliminador 136 de paquetes de datos incluye preferiblemente o tiene acceso a un protocolo o unidad de Active Queue Management (AQM - Gestión de Cola Activa) para determinar por ejemplo cuándo deben eliminarse paquetes de datos/segmento. Aunque no está ilustrado en la figura, el gestor 130 de memoria intermedia (buffer) puede incluir otros medios y unidades adaptados para tratar paquetes de datos completos identificados en la memoria intermedia (buffer). Tales unidades adicionales podrían incluir medios de retardo y priorización de paquetes de IP que operan en los paquetes de IP identificados por el identificador 140 de paquetes de datos.

Las unidades 132, 134, 136 y 140 del gestor 130 de memoria intermedia (buffer) pueden ser provistas como software, hardware o una combinación de los mismos. Las unidades pueden implementarse juntas por ejemplo en un solo nodo de red de la estación de base, estando algunas de las unidades provistas en diferentes nodos de red del sistema de la estación de base.

La Fig. 16 es un diagrama de bloques esquemático que ilustra con más detalle una realización del identificador 140 de paquetes de datos de la Fig. 15. El identificador 140 de paquetes de datos incluye típicamente medios 142 para almacenar y actualizar un contador o puntero k de segmentos. Este contador k se usa para conocer qué LLC PDU está siendo actualmente investigada durante el proceso de identificación de la primera u la última LLC PDU de un paquete de IP completo en la cola de la memoria intermedia (buffer). De manera similar los medios 142 están adaptados para almacenar y actualizar los identificadores o punteros PRIMERA y ÚLTIMA, que son asociados con y usados para identificar la primera y la última LLC PDU de un paquete de IP completo, respectivamente. Los medios 142 almacenan y actualizan también el MINTAMAÑO de tamaño TAMAÑO, que proporciona el tamaño de un paquete de IP identificado. Los medios de umbral 144 están adaptados para almacenar y determinar el número total N de LLC PDUs en la cola de la memoria intermedia (buffer) y los diferentes parámetros de umbral, incluyendo MAXTAMAÑO y MINTAMAÑO, y MAX y MIN. El valor de los parámetros de umbral de tamaño puede ser recibido de otras unidades en el BSS o en el sistema de comunicaciones de telefonía móvil. El medio o unidad de comparación de tamaño 146 está adaptado para comparar el tamaño de LLC PDUs en la memoria intermedia (buffer) de datos asociada, comparando un tamaño de una LLC PDU actualmente asociada con el contador k de segmentos en el medio 142 con un tamaño de una LLC PDU consecutiva siguiente en la cola de la memoria intermedia (buffer). El medio de determinación del formato 148 está provisto en el identificador 140 de paquetes de datos para determinar el formato o el tipo de una LLC PDU, es decir para discriminar entre los segmentos de paquete de datos que transportan carga útil de paquetes de IP y otros paquetes de datos o segmentos, incluyendo mensajes de control y paquetes. En una primera realización, el medio de determinación 148 extrae información de formato del campo de información en la cabecera de las LLC PDUs recibidas. En una segunda realización, el medio 148 comprueba el campo de formato en la cabecera de las LLC PDUs para determinar si contienen carga útil de paquetes de IP. Alternativamente, o además, el medio de determinación 148 podría basar la determinación del formato en el tamaño del paquete de IP o LLC PDU identificados, en particular comparando el tamaño del paquete de IP o de la LLC PDU con los parámetros de tamaño manejados por el medio 144. En tal caso, la unidad 148 determina un paquete de datos completo identificado como un paquete de IP si su tamaño es menor que MAXTAMAÑO y mayor que MINTAMAÑO. De manera similar, un segmento de paquete de datos de un tamaño menor que MIN podría ser determinado como que es un paquete o mensaje de señal de control, y así no contener carga útil de paquetes de IP.

Las unidades 142, 144, 146 y 148 del etapa 140 de paquetes de datos pueden ser provistas como software, hardware o una combinación de ambos. Las unidades pueden ser implementadas juntas por ejemplo en un solo nodo de red de estación de base del BSS. Alternativamente, también es posible una implementación distribuida, estando algunas de las unidades provistas en diferentes nodos de red del sistema de la estación de base.

La Fig. 17 es un diagrama de bloques esquemático que ilustra una realización de un nodo de red 200 que segmenta paquetes de datos (25) en segmentos de paquetes de datos (LLC PDUs) de acuerdo con la presente invención. En un sistema de comunicaciones de GPRS, EGPRS y EDGE/GPRS la funcionalidad de segmentación de paquetes de datos es proporcionada en un nodo SGSN. El SGSN 200 incluye una unidad de I/O 210 adaptada para dirigir la comunicación con otros nodos de red en el sistema de comunicaciones. La unidad de I/O 210 está en particular configurada para recibir paquetes de IP del nodo GGSN y para transmitir LLC PDUs al BSS. Un segmentador de paquetes de datos o medio de segmentación 220 está provisto en el SGSN 200 para segmentar los paquetes de IP

ES 2 335 880 T3

recibidos que son mayores que el máximo tamaño permitido para una LLC PDU (segmento de paquete de datos). El SGSN 200 comprende un asociador de información del DP o medio de asociación 230 que asocia la información del DP con los paquetes de datos segmentados del segmentador 220. El asociador está preferiblemente adaptado para incluir la información del DP, por ejemplo una notificación de si la LLC PDU asociada es una LLC PDU primera, última o intermedia de un paquete de IP completo, en el campo de información de la cabecera de la LLC PDU. El asociador 230 de información de DP puede también estar configurado para introducir información de formato en la cabecera del campo de información de las LLC PDUs desde el segmentador 220. Un medio de cifrado de segmentos o medio de encriptación 240 puede ser proporcionado para cifrar las LLC PDUs antes de su transmisión al BSS a través de la unidad de I/O 210.

Las unidades 210, 220, 230 y 240 del SGSN 200 pueden ser provistas como software, hardware o una combinación de los mismos.

Un experto comprenderá que pueden realizarse varias modificaciones y cambios a la presente invención sin separarse del ámbito de la misma, que está definido por las reivindicaciones adjuntas.

Referencias

[1] 3GPP TS 23.060 V6.1.0; 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS); Service Description; Stage 2 (Junio, 2003).

[2] 3GPP TS 04.64 V8.7.0: 3rd Generation Partnership Project; Technical Specification Group Core Network; Digital cellular telecommunications system (Phase 2+); General Radio Service (GPRS); Mobile Station - Serving GPRS Support Node (MS-SGSN) Logical Link Control (LLC) layer specification (Diciembre, 2001).

[3] 3GPP TS 04.65 V8.2.0; 3rd Generation Partnership Project; Technical Specification Group Core Network; Digital Cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Mobile Station - Serving GPRS Support Node (SGSN); Subnetwork Dependent Convergence Protocol (SND CP) (Septiembre, 2001).

REIVINDICACIONES

1. Un método para permitir la identificación de un paquete de datos completo en una memoria intermedia (buffer) de datos (120) que comprende una cola de segmentos de paquetes de datos consecutivos, **caracterizado** por:

- comparar un tamaño $S(k)$ de un segmento de paquete de datos ($P(k)$) con un tamaño ($S(k+1)$) de un segmento de paquete de datos ($P(k)$) en la citada memoria intermedia (buffer) de datos (120); e
- identificar el citado paquete de datos completo basándose en la citada comparación.

2. El método de acuerdo con la reivindicación 1, **caracterizado** por proporcionar un contador (k) de segmentos asociado con un segmento de paquete de datos ($P(k)$) en la citada memoria intermedia (buffer) de datos (120).

3. El método de acuerdo con la reivindicación 2, **caracterizado** por:

- comparar un tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) con un tamaño ($S(k+1)$) de un segmento de paquete de datos ($P(k+1)$) consecutivo en la citada memoria intermedia (buffer) de datos (120);
- identificar el citado siguiente segmento de paquete de datos ($P(k+1)$) como un primer segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120) si el citado tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) es menor que el citado tamaño ($S(k+1)$) del citado segmento de paquete de datos ($P(k+1)$) siguiente.

4. El método de acuerdo con la reivindicación 2, **caracterizado** por:

- (a) comparar un tamaño ($S(k)$) del segmento de paquete de datos ($P(k)$) actualmente asociado con el citado contador (k) con un tamaño ($S(k+1)$) de un siguiente segmento de paquete de datos consecutivo ($P(k+1)$) en la citada memoria intermedia (buffer) de datos (120); y
- (b) asociar el citado contador (k) con el citado siguiente segmento de paquete de datos ($P(k+1)$) si el citado tamaño ($S(k)$) del segmento de paquete de datos ($P(k)$) actualmente asociado con el citado contador (k) es igual o mayor que el citado tamaño ($S(k+1)$) del citado siguiente segmento de paquete de datos ($P(k+1)$); y
- repetir tanto la etapa de comparación (a) como la citada etapa de asociación (b) hasta que el citado tamaño ($S(k)$) del segmento de paquete de datos ($P(k)$) actualmente asociado con el citado contador (k) es menor que el citado tamaño ($S(k+1)$) del citado siguiente segmento de paquete de datos ($P(k+1)$), por lo que el citado siguiente segmento de paquete de datos ($P(k+1)$) es identificado como un primer segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120).

5. El método de acuerdo con la reivindicación 3 ó 4, **caracterizado** por asociar el citado contador (k) de segmentos con el citado segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo.

6. El método de acuerdo con la reivindicación 5, **caracterizado** por:

- comparar un tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) con un tamaño ($S(k+1)$) de un siguiente consecutivo segmento de paquete de datos ($P(k+1)$) en la citada memoria intermedia (buffer) de datos (120); e
- identificar el citado siguiente segmento de paquete de datos ($P(k+1)$) como un último segmento de paquete de datos ($P(\text{ÚLTIMO})$) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120) si el citado tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) difiere del citado tamaño ($S(k+1)$) del citado siguiente segmento de paquete de datos ($P(k+1)$).

7. El método de acuerdo con la reivindicación 6, **caracterizado** porque se identifica que el citado paquete de datos completo comprende el citado segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo, el citado segmento de paquete de datos último ($P(\text{ÚLTIMO})$) del citado paquete de datos completo y cualesquiera segmentos de paquetes de datos intermedios entre el citado primer ($P(\text{PRIMERO})$) y último ($P(\text{ÚLTIMO})$) segmento de paquete de datos del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120).

8. El método de acuerdo con la reivindicación 6, **caracterizado** por:

- determinar un tamaño total del citado primer segmento de paquete de datos (P(PRIMERO)) del citado paquete de datos completo, del citado último segmento de paquete de datos (P(ÚLTIMO)) del citado paquete de datos completo y cualesquiera segmentos de paquetes de datos intermedios entre el primer (P(PRIMERO)) y último (P(ÚLTIMO)) segmento de paquete de datos completo en la citada memoria intermedia (buffer) de datos (120);
- comparar el citado tamaño total con un umbral de tamaño mínimo; e
- identificar el citado paquete de datos completo comparando el citado primer segmento de paquete de datos (P(PRIMERO)) del citado paquete de datos completo, el citado último segmento de paquete de datos (P(ÚLTIMO)) del citado paquete de datos completo y cualesquiera segmentos de paquetes de datos intermedios entre el citado primer (P(PRIMERO)) y último (P(ÚLTIMO)) segmento de paquete de datos del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120) si el tamaño total es mayor que el citado umbral de tamaño mínimo.

9. El método de acuerdo con la reivindicación 2, **caracterizado** por:

- comparar un tamaño (S(k)) del citado segmento de paquete de datos (P(k)) asociado con el citado contador (k) con un tamaño (S(k+1)) de un siguiente segmento de paquete de datos consecutivo (P(k+1)) en la citada memoria intermedia (buffer) de datos (120); e
- identificar el citado siguiente segmento de paquete de datos (P(k+1)) como un último segmento de paquete de datos (P(ÚLTIMO)) del citado paquete de datos completo en la citada memoria intermedia (buffer) (120) si el citado tamaño S(k) del citado segmento de paquete de datos (P(k)) asociado con el citado contador (k) difiere del citado tamaño (S(k+1)) del citado segmento de paquete de datos (P(k+1)).

10. El método de acuerdo con la reivindicación 2, **caracterizado** por:

- (c) comparar un tamaño (S(k)) del segmento de paquete de datos (P(k)) actualmente asociado con el citado contador (k) con un tamaño (S(k+1)) de un siguiente MINTAMAÑO de paquete de datos consecutivo (P(k+1)) en la citada memoria intermedia (buffer) de datos (120);
- (d) asociar el citado contador (k) con el citado siguiente paquete de datos (P(k+1)) si el citado tamaño (S(k)) del segmento de paquete de datos (P(k)) actualmente asociado con el citado contador (k) es igual al citado tamaño (S(k+1)) del citado siguiente segmento de paquete de datos (P(k+1)); y
- repetir tanto la citada etapa de comparación (c) como la citada etapa de comparación (d) hasta que el citado tamaño (S(k)) del segmento de paquete de datos (P(k)) actualmente asociado con el citado contador (k) difiere del citado tamaño (S(k+1)) del citado siguiente segmento de paquete de datos (P(k+1)), por lo cual el citado siguiente segmento de paquete de datos (P(k+1)) es identificado como un último segmento de paquete de datos (P(ÚLTIMO)) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120).

11. Un método de gestionar una memoria intermedia (buffer) de datos (120) que comprende una cola de segmentos consecutivos de paquetes de datos en un sistema de estación de base (100) de un sistema de comunicaciones de telefonía móvil (1), **caracterizado** por:

- el citado sistema de estación de base (100) que identifica un paquete de datos completo en la citada memoria intermedia (buffer) de datos (120) de acuerdo con cualquiera de las reivindicaciones 1 a 10; y
- el citado sistema de estación de base (100) que elimina el citado paquete de datos completo de la citada memoria intermedia (buffer) de datos (120).

12. Un sistema (140) para permitir la identificación de un paquete de datos completo en una memoria intermedia (buffer) de datos (120) que comprende una cola de segmentos de paquete de datos consecutivos, **caracterizado** por:

- medios (146) para comparar un tamaño (S(k)) de un segmento de paquete de datos (P(k)) con un tamaño (S(k+1)) de un siguiente segmento de paquete de datos consecutivo (P(k+1)) en la citada memoria intermedia (buffer) de datos (120); y
- medios (140) para identificar el citado paquete de datos completo basándose en la citada comparación.

13. El sistema de acuerdo con la reivindicación 13, **caracterizado** por medios (142) para asociar un contador (k) de segmentos con un segmento de paquete de datos (P(k)) en la citada memoria intermedia (buffer) de datos (120).

14. El sistema de acuerdo con la reivindicación 13, **caracterizado** porque el citado medio de comparación (146) está adaptado para comparar un tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) con un tamaño ($S(k+1)$) de un siguiente segmento de paquete de datos consecutivo ($P(k+1)$) en la citada memoria intermedia (buffer) de datos (120), en la que el citado medio de identificación (140) está adaptado para identificar el citado segmento de paquete de datos ($P(k+1)$) como un primer segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120) si el citado tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) es menor que el citado tamaño ($S(k+1)$) del citado siguiente segmento de paquete de datos ($P(k+1)$).
15. El sistema de acuerdo con la reivindicación 13, **caracterizado** porque el citado medio de comparación (146) está adaptado para comparar un tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) actualmente asociado con el citado contador (k) con un tamaño ($S(k+1)$) de un siguiente segmento de paquete de datos consecutivo ($P(k+1)$) en la citada memoria intermedia (buffer) de datos (120), en el que el citado medio de asociación (142) está adaptado para asociar el citado contador (k) con el citado siguiente segmento de paquete de datos ($P(k+1)$) si el citado tamaño ($S(k)$) del segmento de paquete de datos ($P(k)$) actualmente asociado con el citado contador (k) es igual o mayor que el citado tamaño ($S(k+1)$) del citado segmento de paquete de datos ($P(k+1)$), el citado medio de comparación (146) está adaptado para repetir la citada comparación de tamaño y el citado medio de asociación (142) está adaptado para repetir la citada asociación de contador hasta que el citado tamaño ($S(k)$) del segmento de paquete de datos ($P(k)$) actualmente asociado con el citado contador (k) es menor que el citado tamaño ($S(k+1)$) del citado siguiente segmento de paquete de datos ($P(k+1)$), por lo que el citado medio de identificación (140) está adaptado para identificar el siguiente segmento de paquete de datos ($P(k+1)$) como un primer segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120).
16. El sistema de acuerdo con la reivindicación 14 ó 15, **caracterizado** porque el citado medio de asociación (142) está adaptado para asociar el citado contador (k) de segmentos con el citado primer segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo.
17. El sistema de acuerdo con la reivindicación 16, **caracterizado** porque el citado medio de comparación (146) está adaptado para comparar un tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) con un tamaño ($S(k+1)$) de un siguiente segmento de paquete de datos consecutivo ($P(k+1)$) en la citada memoria intermedia (buffer) de datos (120), en el que el citado medio de identificación (140) está adaptado para identificar el citado siguiente segmento de paquete de datos ($P(k+1)$) como un último segmento de paquete de datos ($P(\text{ÚLTIMO})$) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120) si el citado tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) difiere del citado tamaño ($S(k+1)$) del citado siguiente segmento de paquete de datos ($P(k+1)$).
18. El sistema de acuerdo con la reivindicación 17, **caracterizado** porque el citado medio de identificación (140) está adaptado para identificar que el citado paquete de datos completo comprende el citado primer segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo, el citado último segmento de paquete de datos ($P(\text{ÚLTIMO})$) del citado paquete de datos completo y cualesquiera segmentos de paquetes de datos intermedios entre el citado primero ($P(\text{PRIMERO})$) y último ($P(\text{ÚLTIMO})$) segmento de paquete de datos del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120).
19. El sistema de acuerdo con la reivindicación 17, **caracterizado** por medios (142) para determinar un tamaño total del citado primer segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo, el citado último segmento de paquete de datos ($P(\text{ÚLTIMO})$) del citado paquete de datos completo y cualesquiera segmentos de paquete de datos intermedios entre el citado primer ($P(\text{PRIMERO})$) y último ($P(\text{ÚLTIMO})$) segmento de paquete de datos del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120), porque el citado medio de comparación (146) está adaptado para comparar el citado tamaño total con un umbral de tamaño mínimo, y porque el citado medio de identificación (140) está adaptado para identificar que el citado paquete de datos completo comprende el citado primer segmento de paquete de datos ($P(\text{PRIMERO})$) del citado paquete de datos completo, el citado último segmento de paquete de datos ($P(\text{ÚLTIMO})$) del citado paquete de datos completo y cualesquiera segmentos de paquete de datos intermedios del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120) si el tamaño total es mayor que el citado umbral de tamaño mínimo.
20. El sistema de acuerdo con la reivindicación 13, **caracterizado** porque el citado medio de comparación (146) está adaptado para comparar un tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) con un tamaño ($S(k+1)$) de un siguiente segmento de paquete de datos consecutivo ($P(k+1)$) en la citada memoria intermedia (buffer) de datos (120), en el que el citado medio de identificación (140) está adaptado para identificar el citado siguiente segmento de paquete de datos ($P(k+1)$) como un último segmento de paquete de datos ($P(\text{ÚLTIMO})$) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120) si el citado tamaño ($S(k)$) del citado segmento de paquete de datos ($P(k)$) asociado con el citado contador (k) difiere del citado tamaño ($S(k+1)$) del citado segmento de paquete de datos ($P(k+1)$).
21. El sistema de acuerdo con la reivindicación 13, **caracterizado** porque el citado medio de comparación (146) está adaptado para comparar un tamaño ($S(k)$) del segmento de paquete de datos ($P(k)$) actualmente asociado con el citado contador (k) con un tamaño ($S(k+1)$) de un siguiente segmento de paquete de datos consecutivo ($P(k+1)$) en la citada memoria intermedia (buffer) de datos (120), en el que el citado medio de asociación (142) está adaptado para

asociar el citado contador (k) con el citado siguiente segmento de paquete de datos (P(k+1)) si el citado tamaño (S(k)) del paquete de datos (P(k)) actualmente asociado con el citado contador (k) es igual al citado tamaño (S(k+1)) del citado segmento de paquete de datos (P(k+1)), porque el citado medio de comparación (146) está adaptado para repetir la citada comparación de tamaño y porque el citado medio de asociación (142) está adaptado para repetir la citada asociación de contador hasta que el citado tamaño (S(k)) del segmento de paquete de datos (P(k)) actualmente asociado con el citado contador (k) difiere del citado tamaño (S(k+1)) del citado siguiente segmento de paquete de datos (P(k+1)), por lo cual el citado medio de identificación (140) está adaptado para identificar el citado siguiente segmento de paquete de datos (P(k+1)) como un último segmento de paquete de datos (P(ÚLTIMO)) del citado paquete de datos completo en la citada memoria intermedia (buffer) de datos (120).

22. Un sistema (130) para gestionar una memoria intermedia (buffer) de datos (120) que comprende una cola de segmentos consecutivos de paquetes de datos en un sistema de estación de base (100) de un sistema de comunicaciones de telefonía móvil (1), **caracterizado** por:

- un sistema (140) de acuerdo con cualquiera de las reivindicaciones 12 a 21 para identificar un paquete de datos completo en la citada memoria intermedia (buffer) de datos (120); y
- un medio (136) para eliminar el citado paquete de datos completo identificado de la citada memoria intermedia (buffer) de datos (120).

23. Un nodo de estación de base de un sistema de estación de base (100) en un sistema de comunicaciones de telefonía móvil (1) que comprende una memoria intermedia (buffer) de datos (120) que comprende una cola de segmentos consecutivos de paquetes de datos, **caracterizado** por un sistema (130) para gestionar la citada memoria intermedia (buffer) de datos (120) de acuerdo con la reivindicación 22.

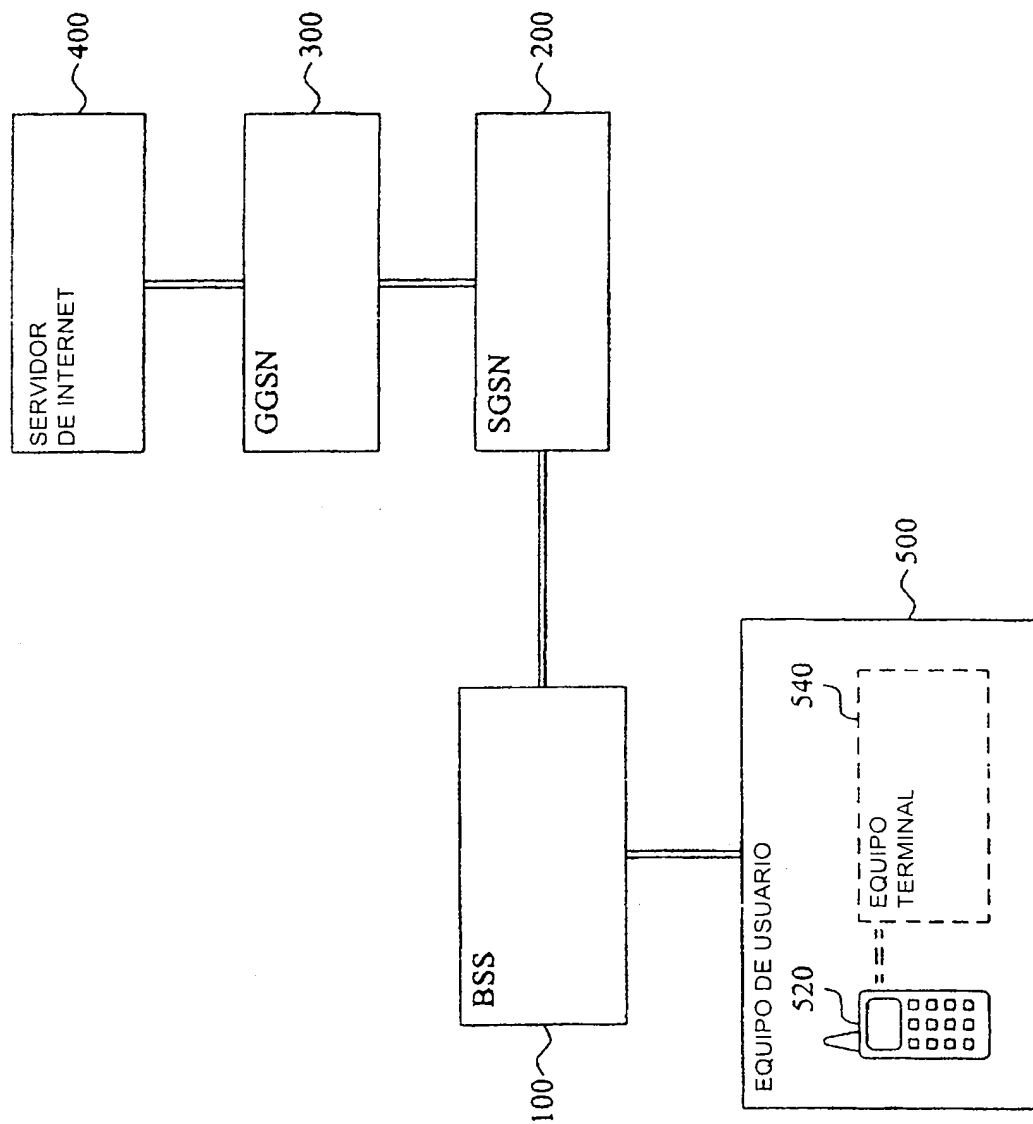


Fig. 1

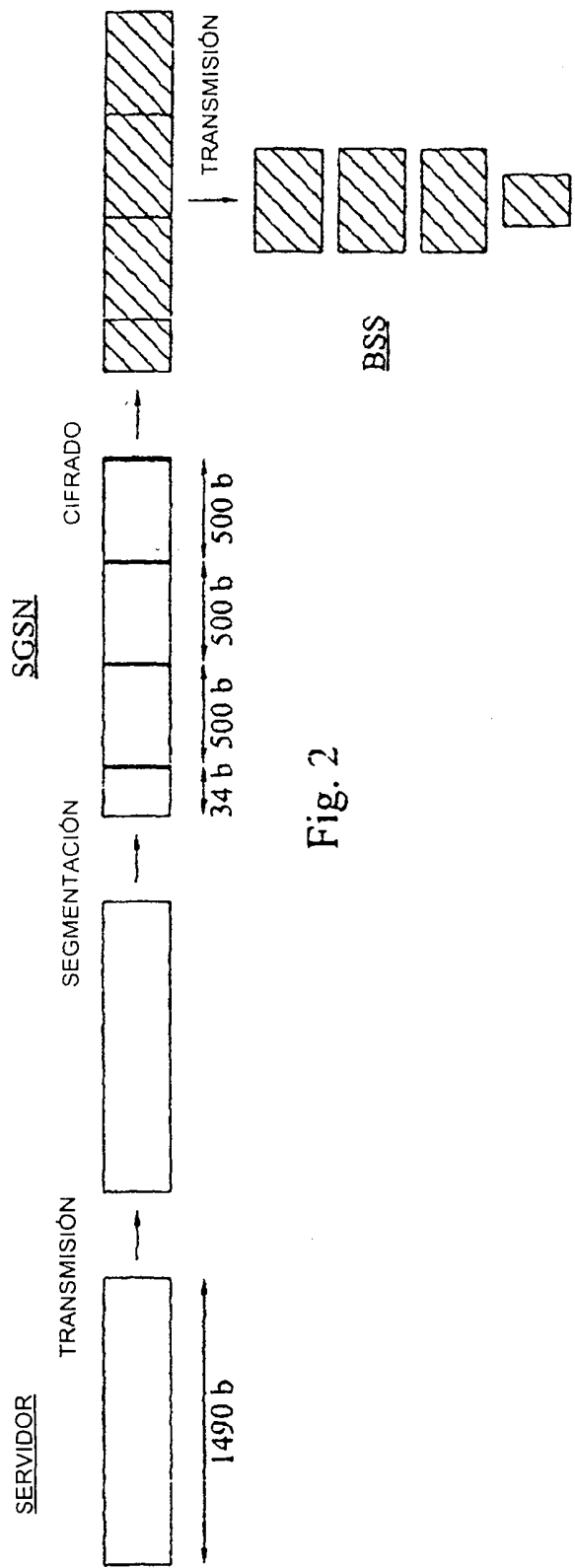


Fig. 2

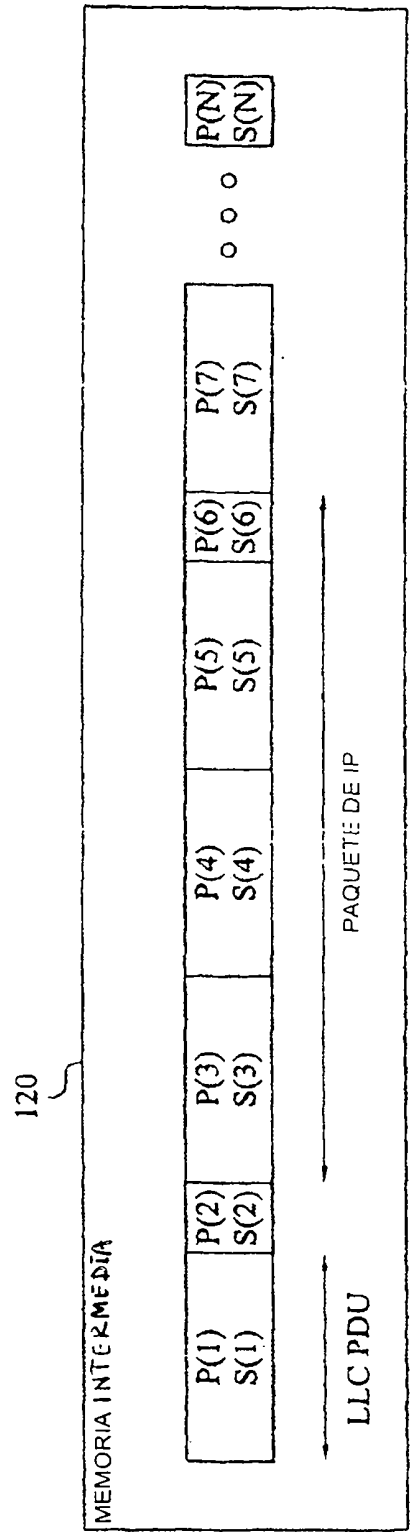


Fig. 3

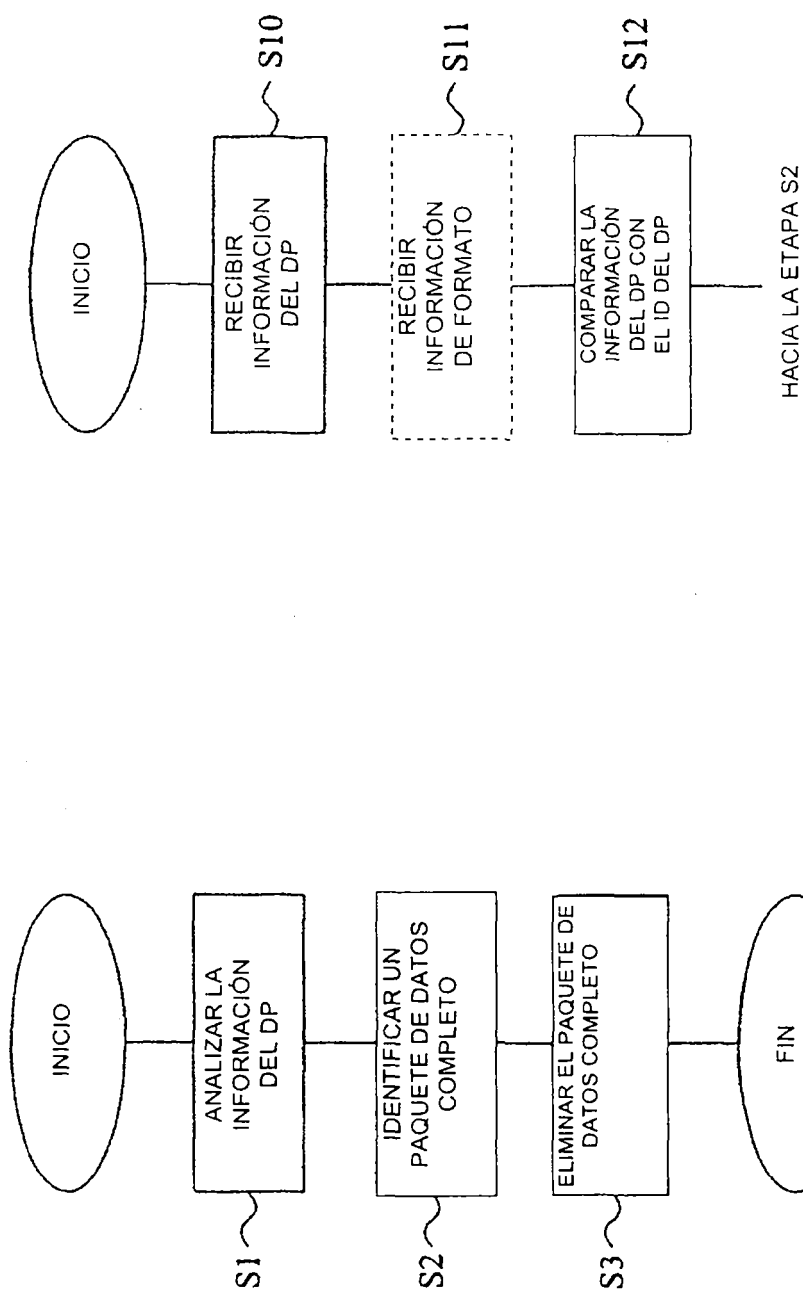


Fig. 5

Fig. 4

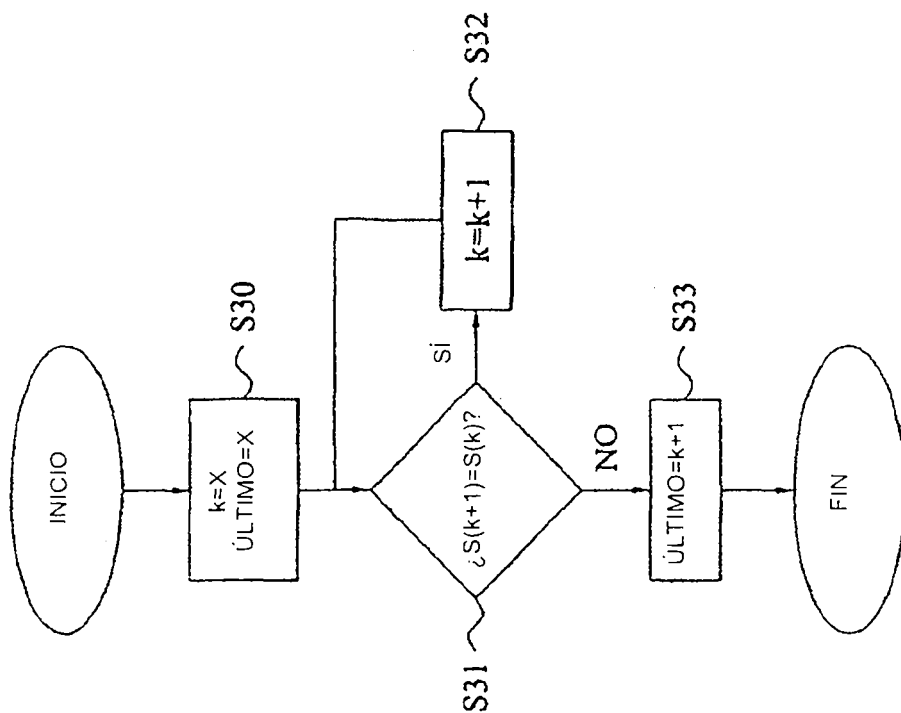


Fig. 7

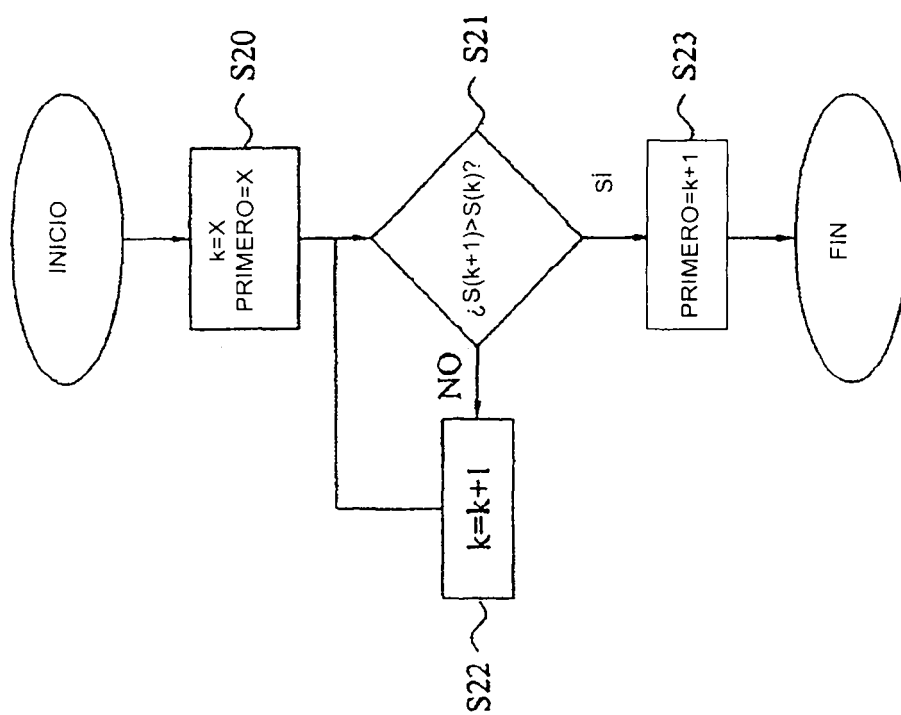


Fig. 6

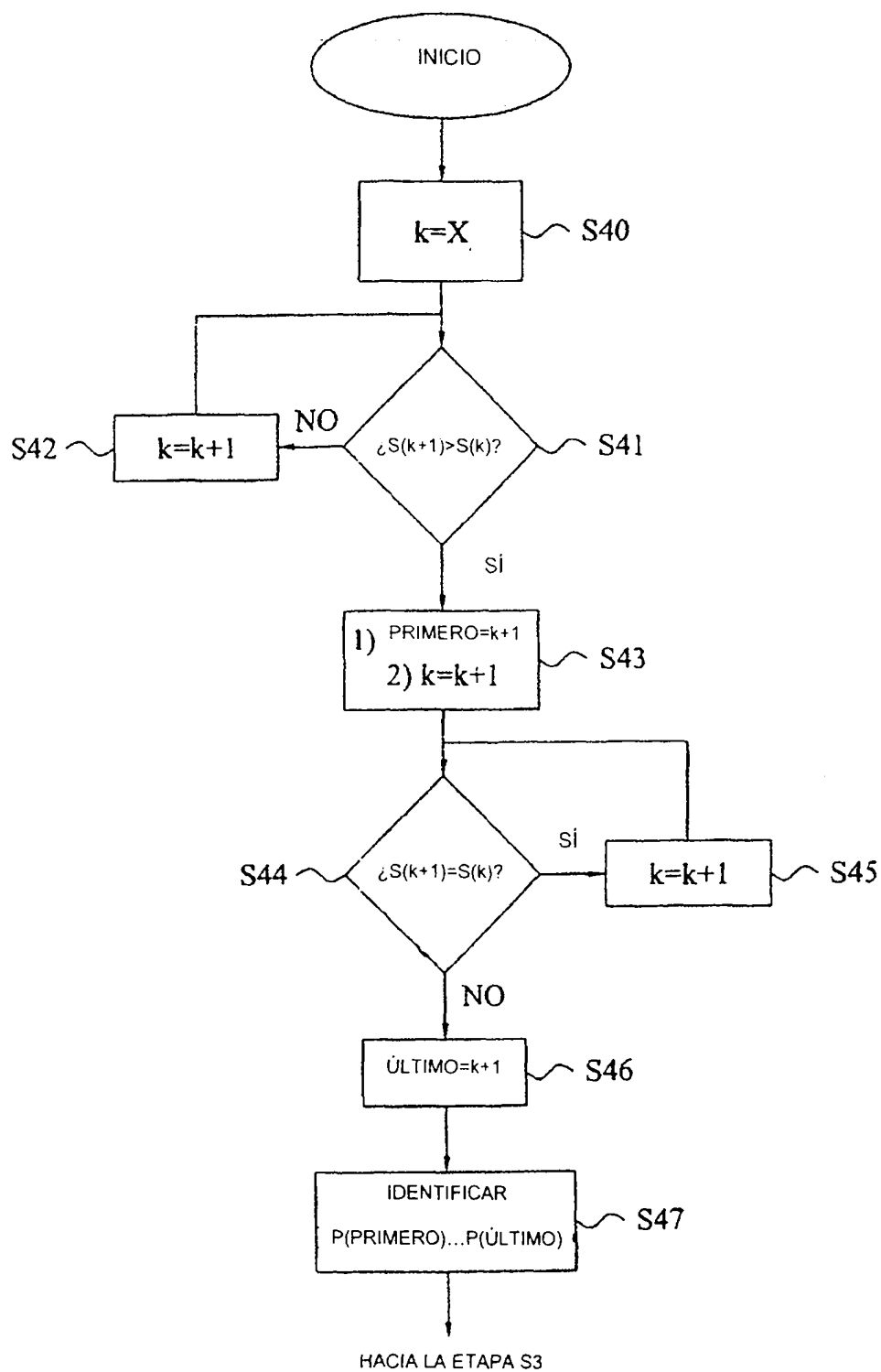
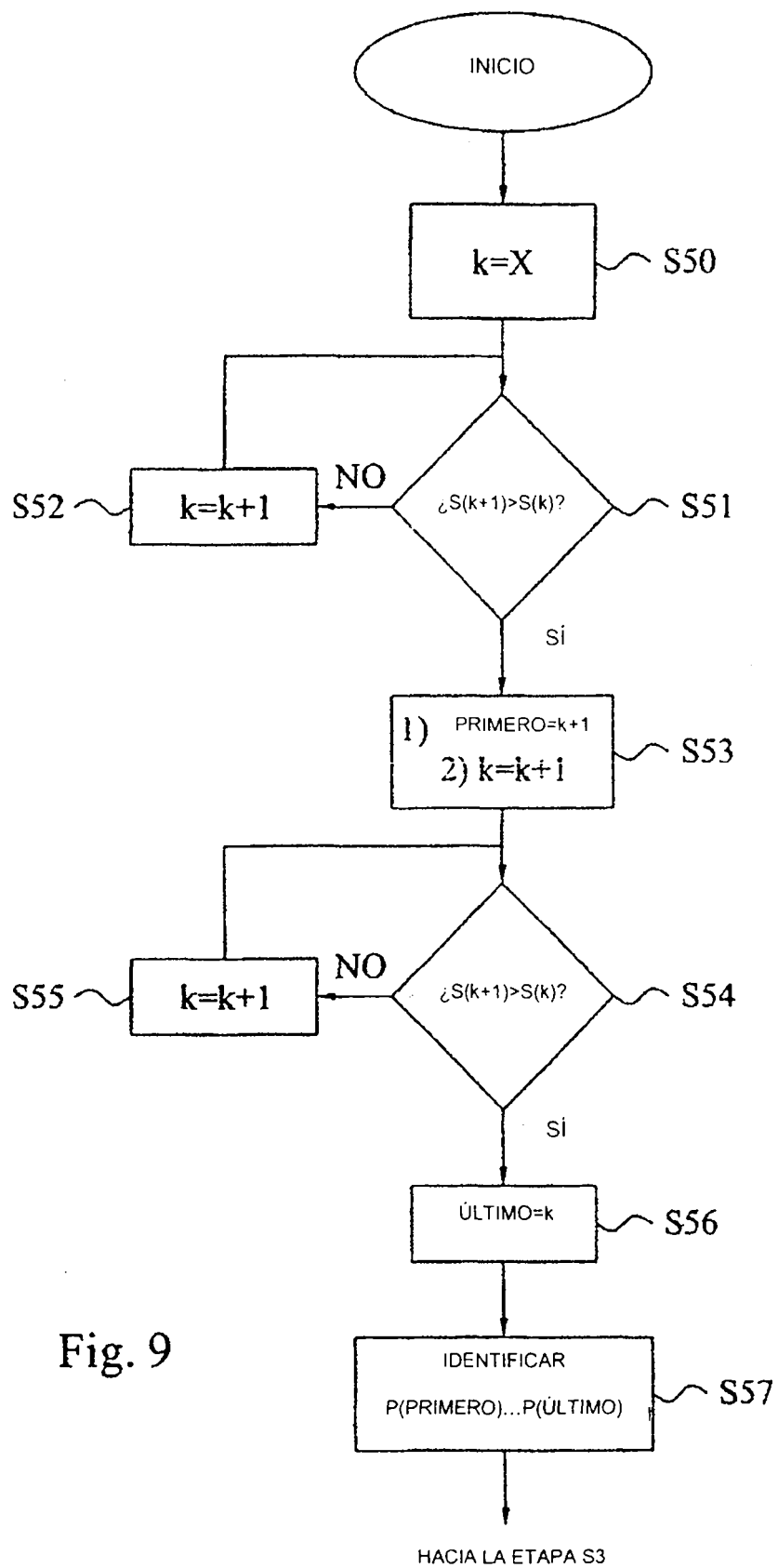


Fig. 8



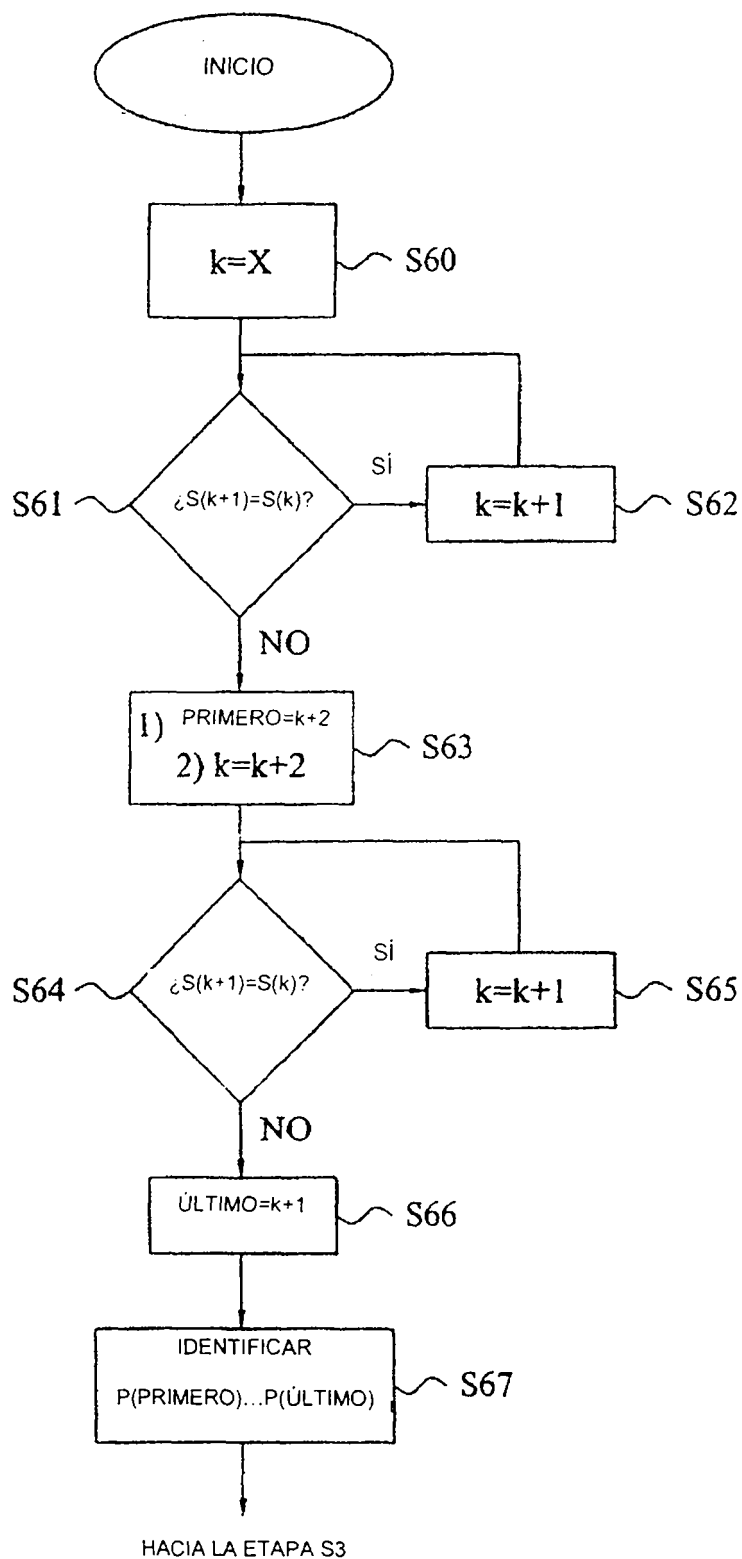
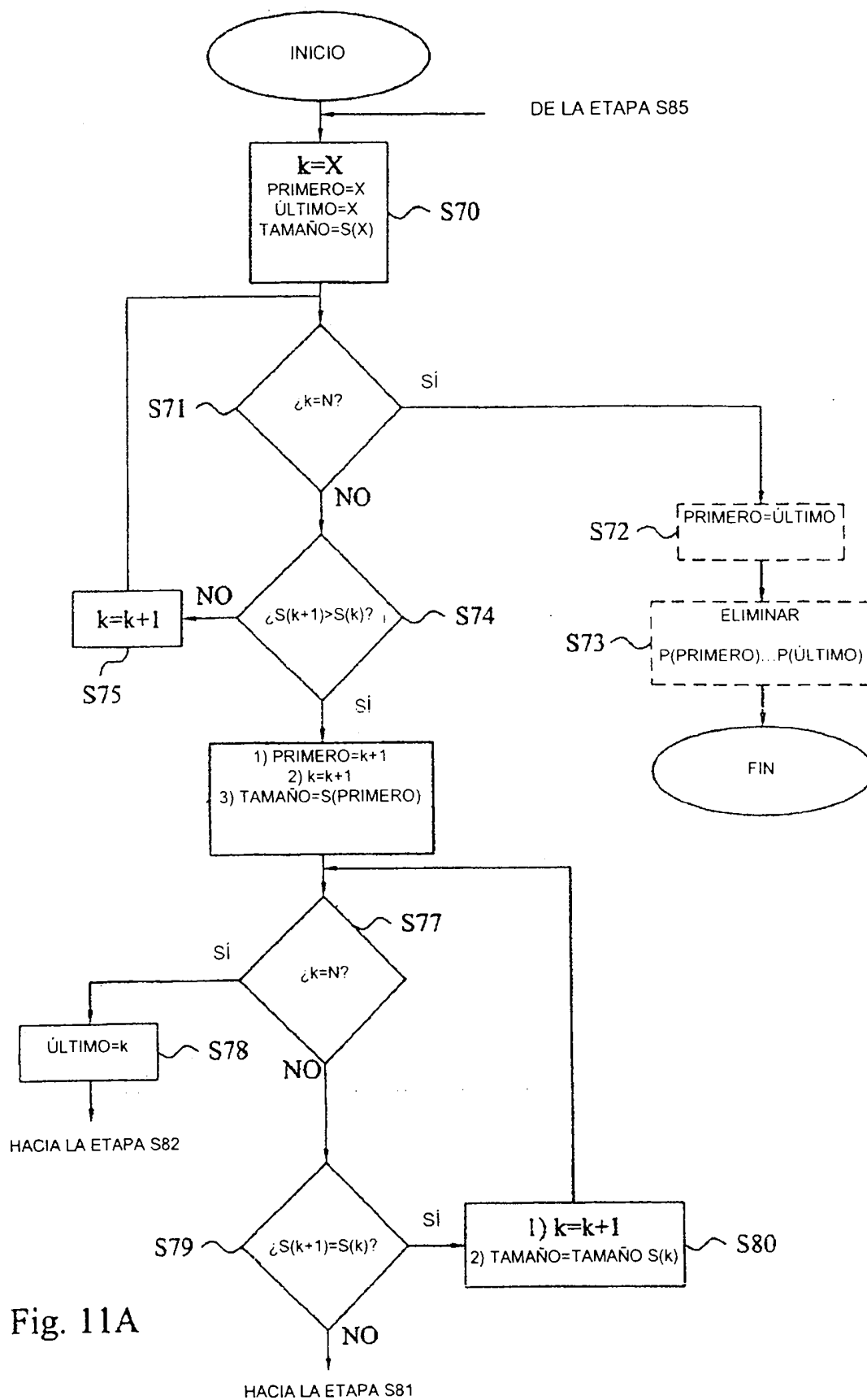


Fig. 10



DE LA ETAPA S78

DE LA ETAPA S79

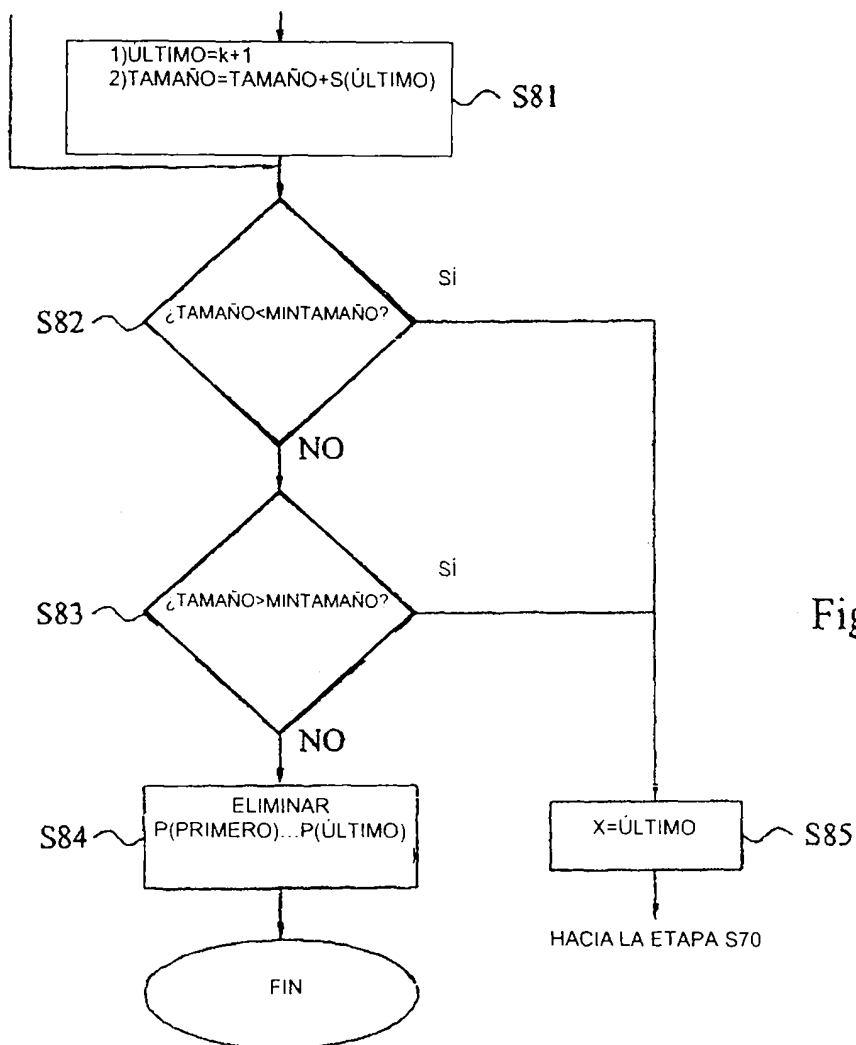


Fig. 11B

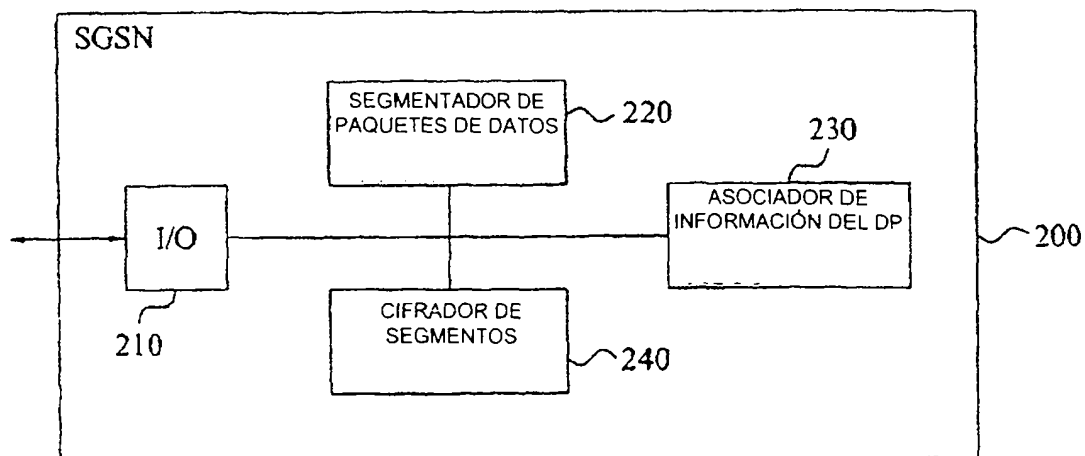
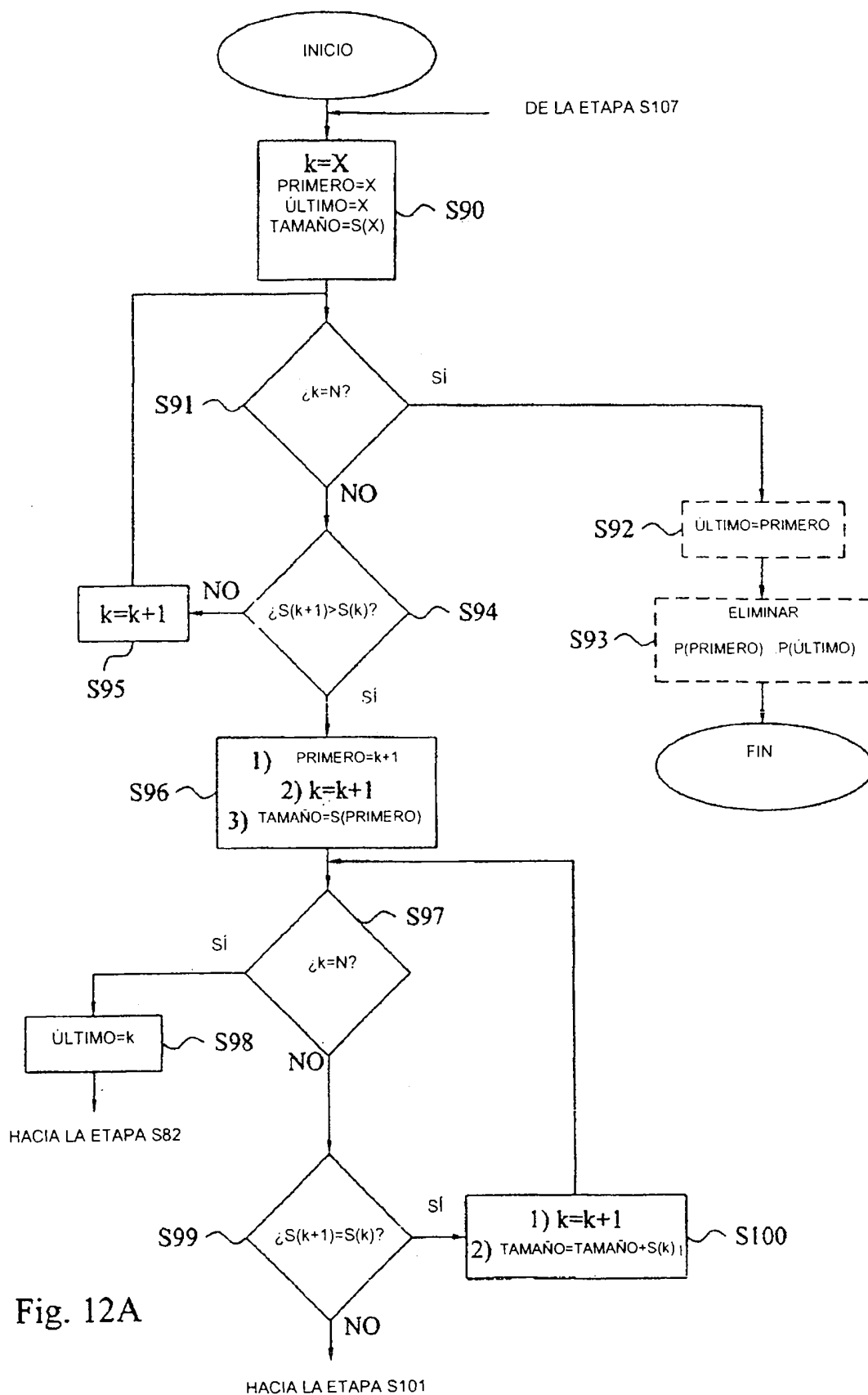


Fig. 17



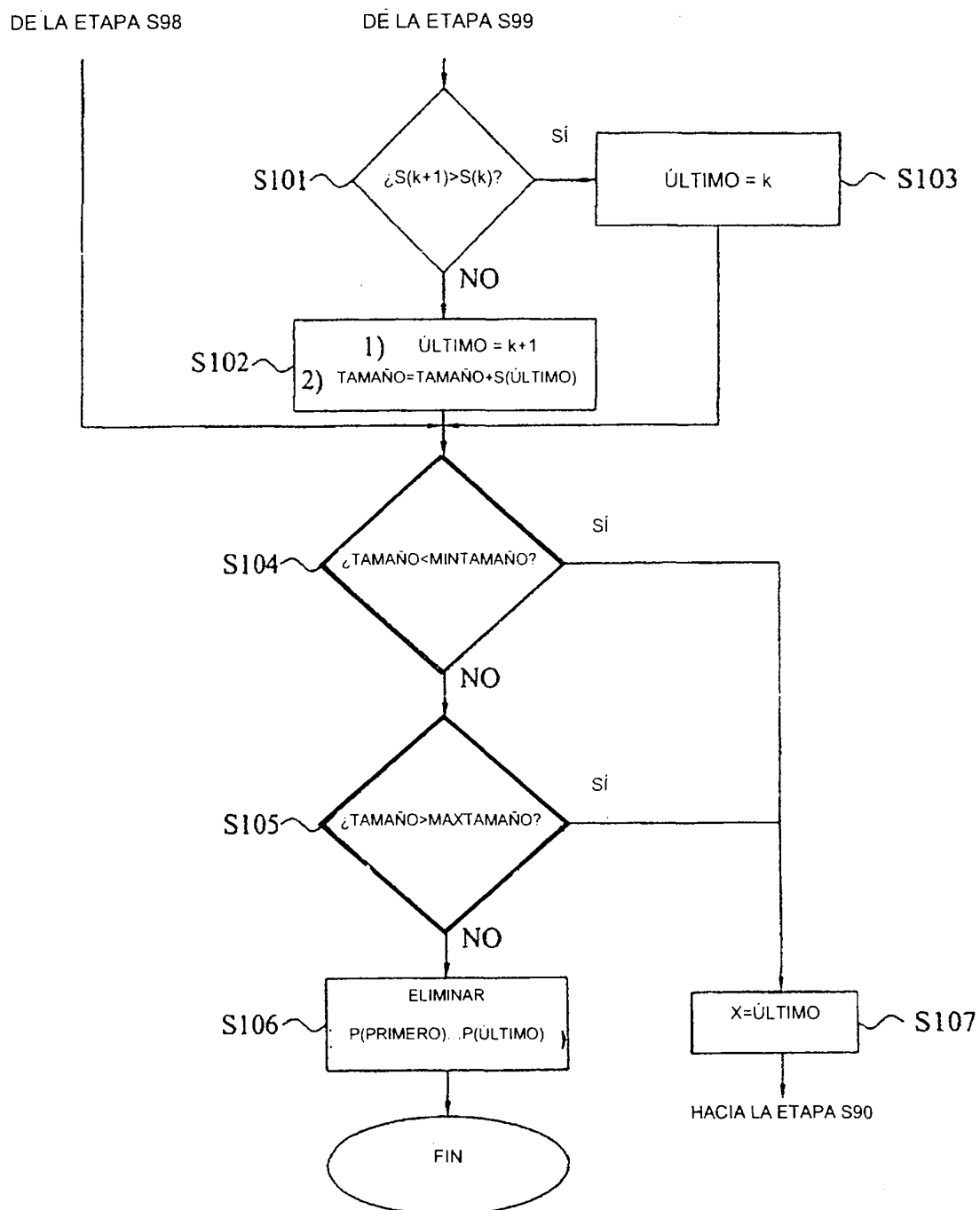


Fig. 12B

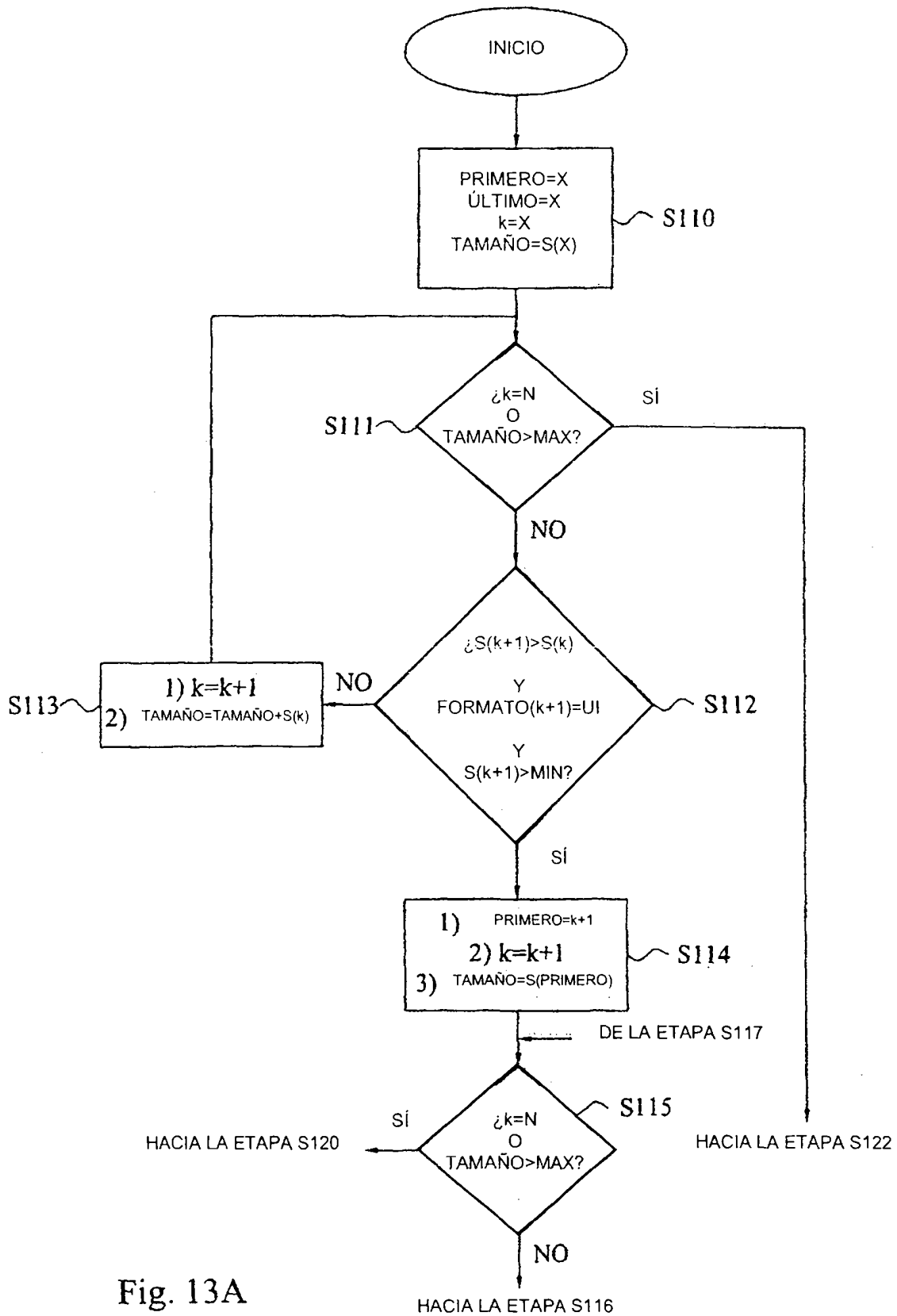


Fig. 13A

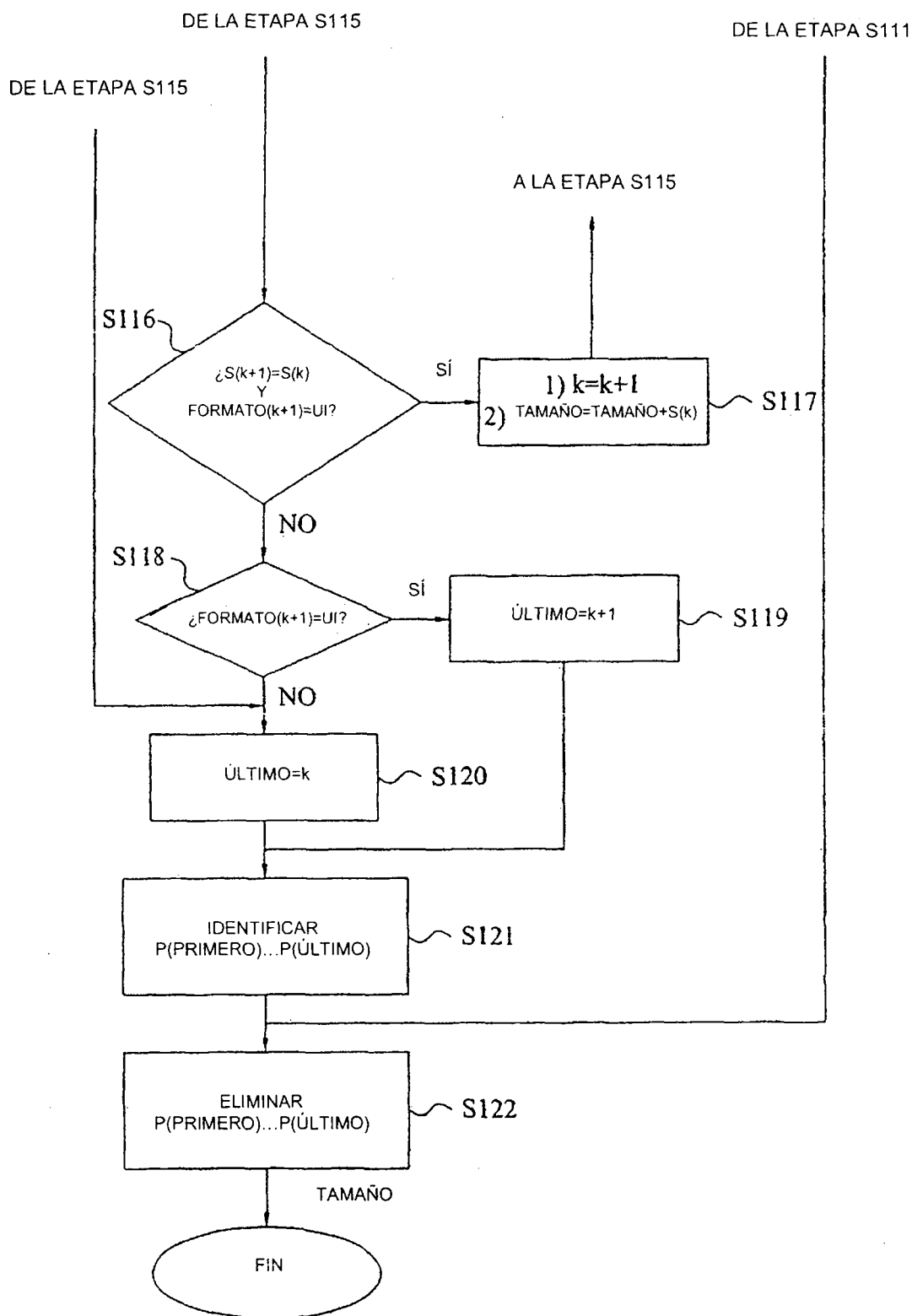


Fig. 13B

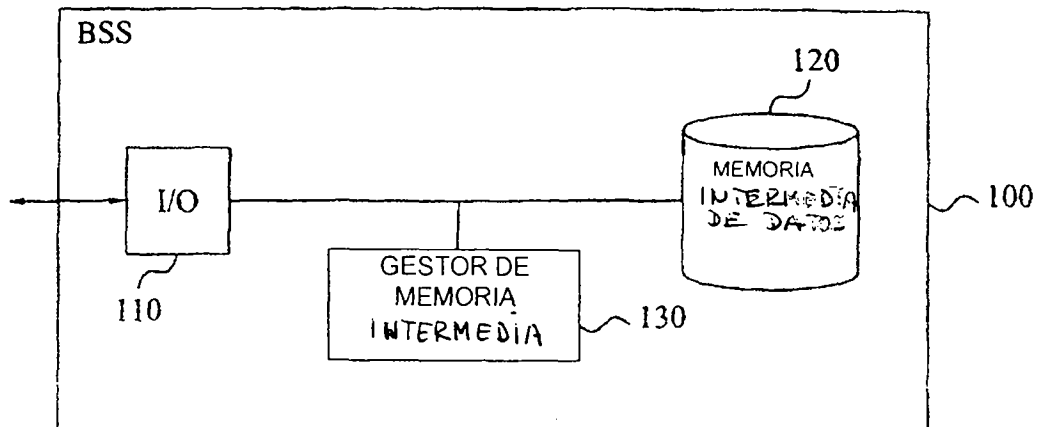


Fig. 14

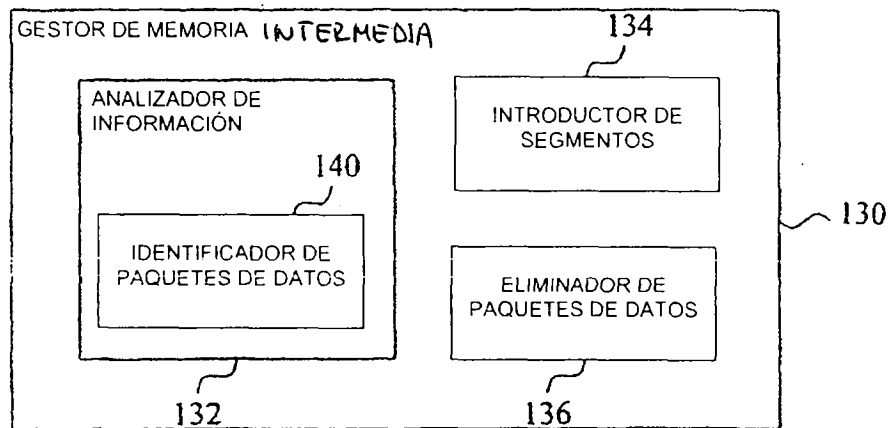


Fig. 15

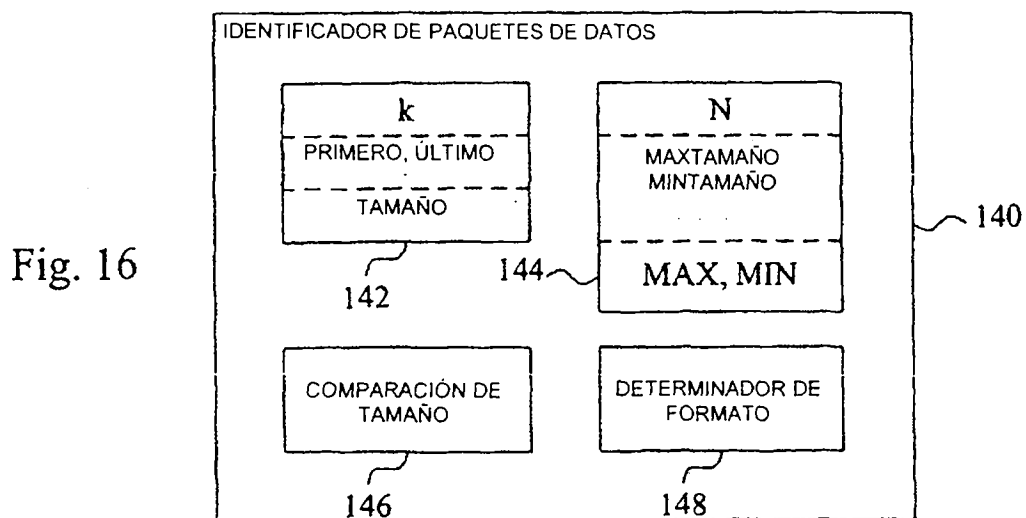


Fig. 16