

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 604 777**

51 Int. Cl.:

G06F 17/30 (2006.01)

H04L 29/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **06.12.2005** **E 05111729 (9)**

97 Fecha y número de publicación de la concesión europea: **07.09.2016** **EP 1727055**

54 Título: **Coordinación de comunicación de datos con números de secuencia**

30 Prioridad:

25.05.2005 US 685008

15.07.2005 US 182989

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

09.03.2017

73 Titular/es:

MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)

One Microsoft Way
Redmond, WA 98052, US

72 Inventor/es:

MOHAMED, AHMED;
RAMAN, BALAN SETHU y
KRUSE, DAVID

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 604 777 T3

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

DESCRIPCIÓN

Coordinación de comunicación de datos con números de secuencia

La presente solicitud reivindica prioridad a la solicitud de patente provisional de Estados Unidos con N.º de serie 60/685.008, presentada el 25 de mayo de 2005. La presente solicitud está relacionada con la solicitud de patente de Estados Unidos en trámite junto con la presente "Data Communication Protocol" número de expediente del mandatario 5612/311754, presentada de manera concurrente con la presente, asignada al cesionario de la presente invención.

Muchos protocolos de comunicación de datos que están aún en uso hoy en día, tal como el protocolo SMB (Bloque de Mensajes de Servidor), se desarrollaron en un momento cuando los recursos informáticos eran muy diferentes, por ejemplo, el ancho de banda de red estaba típicamente limitado y la memoria era muy preciosa. Como resultado, cuando se usan en redes actuales, tales protocolos pueden limitar el rendimiento global. Por ejemplo, debido a que se han diseñado cuando la memoria estaba limitada, se usan tamaños de memoria intermedia pequeños, que requieren más recorridos de ida y vuelta para comunicar grandes cantidades de datos.

Además, el protocolo SMB existente tiene otras limitaciones que se han hecho evidentes con el tiempo. Por ejemplo, el protocolo SMB existente es susceptible a ataques de denegación de servicio; el diseño del protocolo hace difícil combatir estos ataques. Análogamente, el procedimiento para asegurar la seguridad de los paquetes es complicado. Además, no existe mecanismo actual para realizar operaciones similares a calidad de servicio, en las que un cliente confiable, por ejemplo, obtiene los mismos recursos de servidor que un cliente no confiable. En resumen, aunque aún es un protocolo usado con frecuencia y valioso, las versiones de SMB existentes son menos que ideales cuando se usan con recursos de red actuales.

Tom Talpey Network Appliance y col: "NFSv4 Session Extensions" IETF Standard-Wording-Draft, Internet Engineering Task Force, IETF, CH, vol. Nfsv4, n.º 1, febrero de 2005, ISSN: 0000-0004 propone extensiones a la versión 4 de NFS, que le posibilitan soportar sesiones duraderas, gestión y operaciones de punto de extremo sobre una diversidad de transportes de RPC, incluyendo TCP y RDMA. Cada solicitud de un cliente se etiqueta con un identificador, que se denomina un id de intervalo. Este identificador debe analizarse por NSFv4 cuando se ejecuta sobre cualquier transporte. Existe un límite al número de solicitudes activas. Cuando el cliente emite una nueva solicitud, selecciona un id de intervalo en el rango 0... N-1, donde N es el límite de "solicitudes totales" actuales del servidor concedidas al cliente en la sesión a través de la que se ha de emitir la solicitud. El id de intervalo debe estar sin usar por cualquiera de las solicitudes que el cliente ya tiene activas en la sesión. "Sin usar" en este punto significa que el cliente no tiene solicitudes pendientes para ese id de intervalo. Un id de secuencia, que acompaña el id de intervalo en cada solicitud, es importante para una segunda comprobación importante en el servidor: debe poder determinarse si una solicitud que usa un cierto id de intervalo es una solicitud de retransmisión o una nueva sin precedentes. Por lo tanto, se transmite un id de secuencia junto con el id de intervalo en cada solicitud. El id de intervalo por lo tanto se encarga del papel tradicional del número de puerto en la implementación de DRC de servidor.

El documento US2005/060442 A1 desvela un procedimiento para gestionar transmisión de datos a través de una red desde un origen a un destino. El destino impone un valor de ventana en el origen que limita la cantidad de paquetes de datos que pueden enviarse desde el origen al destino sin recibir un acuse de recibo de ser recibidos por el destino.

Es por lo tanto el objeto de la presente invención proporcionar un procedimiento mejorado en un entorno informático de red que tenga un cliente y un servidor que pueda controlar el uso de recursos del cliente; así como un medio legible por ordenador correspondiente que tenga instrucciones ejecutables por ordenador para ejecutarse en un servidor y un sistema del lado de servidor correspondiente.

Este objeto se resuelve mediante la materia objeto de las reivindicaciones independientes.

Se definen realizaciones preferidas mediante las reivindicaciones dependientes.

En resumen, diversos aspectos de la presente invención están dirigidos al uso de números de secuencia para comunicación cliente-servidor, tal como incorporados en un protocolo de comunicación de datos, para controlar un uso del cliente de recursos de servidor. Diversos aspectos aplican el uso de números de secuencia a un protocolo donde la ordenación no es importante, pero proporcionan calidad de servicio, combaten la denegación de servicio, división de recursos de servidor, firma de mensajes segura y otros numerosos beneficios.

Un servidor concede los créditos de cliente, y el cliente usa un crédito para enviar cada comando al servidor. Cada crédito corresponde a un número de secuencia, formando el conjunto de números de secuencia una ventana de comando válido. El servidor hace cumplir que para cada comando recibido, el comando incluya un número de secuencia que está dentro de la ventana de comando válido y que el número de secuencia no se ha usado con otro comando. El servidor puede mantener también un tamaño de ventana máximo, de manera que incluso un cliente que tenga créditos no pueda enviar un comando con un número de secuencia por debajo de un número de secuencia máximo que corresponda al tamaño de ventana máximo.

En general, tras recibir un comando desde el cliente, el servidor verifica que el número de secuencia está dentro de la ventana y no se ha usado antes. El servidor a continuación elimina el correspondiente número de secuencia de entre aquellos que el cliente puede usar, consumiendo de esta manera un crédito. El servidor a continuación determina si conceder al cliente uno o más otros créditos.

- 5 Por lo tanto, se proporciona un mecanismo para limitar el uso de recursos de servidor controlando un número de créditos concedidos al cliente, mediante una ventana de operación válida que contiene un número único para cada crédito concedido al cliente. Un mecanismo de cumplimiento asegura que para permitir operaciones de servidor adicionales sobre un comando recibido, el comando incluya un número de secuencia que está dentro de la ventana de operación válida y que el número único no se haya usado con otro comando. Un mecanismo de localización que controla créditos concedidos al cliente y los números únicos dentro de la ventana de operación válida.

Otras ventajas serán evidentes a partir de la siguiente descripción detallada cuando se toma en conjunto con los dibujos.

Breve descripción de los dibujos

- 15 La presente invención se ilustra a modo de ejemplo y sin limitación en las figuras adjuntas en las que números de referencia similares indican elementos similares y en las que:

La Figura 1 muestra un ejemplo ilustrativo de un entorno informático de fin general en el que pueden incorporarse diversos aspectos de la presente invención.

La Figura 2 es un diagrama de bloques que representa un entorno de red de ejemplo en el que un cliente se comunica con un servidor de acuerdo con diversos aspectos de la presente invención.

- 20 Las Figuras 3-13 son representaciones de datos de cliente y servidor mantenidos para usar números de secuencia, de acuerdo con diversos aspectos de la presente invención.

Descripción detallada

Entorno de operación ejemplar

- 25 La Figura 1 ilustra un ejemplo de un entorno 100 de sistema informático adecuado en el que puede implementarse la invención. El entorno 100 de sistema informático es únicamente un ejemplo de un entorno informático adecuado y no se pretende para sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad de la invención. Ni debería interpretarse el entorno informático 100 como que tiene ninguna dependencia o requisito relacionado con uno cualquiera o combinación de los componentes ilustrados en el entorno 100 operativo ejemplar.

- 30 La invención es operacional con numerosos otros entornos o configuraciones de sistema informático de fin general o de fin especial. Ejemplos de sistemas informáticos bien conocidos, entornos y/o configuraciones que pueden ser adecuados para uso con la invención incluyen, pero sin limitación: ordenadores personales, ordenadores de servidor, dispositivos de mano o portátiles, dispositivos de tableta, sistemas multiprocesador, sistemas basados en microprocesadores, decodificadores de salón, electrónica de consumo programable, PC de red, miniordenadores, ordenadores centrales, entornos informáticos distribuidos que incluyen cualquiera de los sistemas o dispositivos anteriores y similares.

- La invención puede describirse en el contexto general de instrucciones ejecutables por ordenador, tales como módulos de programa, que se ejecutan mediante un ordenador. En general, módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, y así sucesivamente, que realizan tareas particulares o implementan tipos de datos abstractos particulares. La invención puede ponerse en práctica también en entornos informáticos distribuidos donde las tareas se realizan por dispositivos de procesamiento remotos que están enlazados a través de una red de comunicaciones. En un entorno informático distribuido, los módulos de programa pueden localizarse en medio de almacenamiento informático local y/o remoto incluyendo dispositivos de almacenamiento de memoria.

- 45 Con referencia a la Figura 1, un sistema ejemplar para implementar la invención incluye un dispositivo informático de fin general en forma de un ordenador 110. Los componentes del ordenador 110 pueden incluir, pero sin limitación, una unidad 120 de procesamiento, una memoria 130 de sistema y un bus 121 de sistema que acopla diversos componentes de sistema incluyendo la memoria de sistema a la unidad 120 de procesamiento. El bus 121 de sistema puede ser cualquiera de diversos tipos de estructuras de bus incluyendo un bus de memoria o controlador de memoria, un bus periférico y un bus local usando cualquiera de una diversidad de arquitecturas de bus. A modo de ejemplo, y sin limitación, tales arquitecturas incluyen bus de la Arquitectura Estándar de la Industria (ISA), bus de Arquitectura Micro Canal (MCA), bus de ISA mejorado (EISA), bus local de la Asociación de Normalización en la Electrónica de Vídeo (VESA), y bus de Interconexión de Componentes Periféricos (PCI) también conocido como bus Mezzanine.

El ordenador 110 típicamente incluye una diversidad de medios legibles por ordenador. Medio legible por ordenador puede ser cualquier medio disponible que pueda accederse por el ordenador 110 e incluye tanto medio volátil como no volátil, y medio extraíble y no extraíble. A modo de ejemplo, y sin limitación, medio legible por ordenador puede comprender medio de almacenamiento informático y medio de comunicación. Medio de almacenamiento informático incluye medio volátil y no volátil, extraíble y no extraíble implementado en cualquier procedimiento o tecnología para almacenamiento de información tal como instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos. Medio de almacenamiento informático incluye, pero sin limitación, RAM, ROM, EEPROM, memoria flash u otra tecnología de memoria, CD-ROM, discos versátiles digitales (DVD) u otro almacenamiento de disco óptico, casetes magnéticos, cinta magnética, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda usarse para almacenar la información deseada y pueda accederse mediante el ordenador 110. El medio de comunicación típicamente incorpora instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tal como una onda portadora u otro mecanismo de transporte e incluye cualquier medio de entrega de información. La expresión "señal de datos modulada" significa una señal que tiene una o más de sus características establecidas o cambiadas de tal manera para codificar información en la señal. A modo de ejemplo, y sin limitación, medio de comunicación incluye medio cableado tal como una red cableada o conexión cableada directa, y medio inalámbrico tal como medio acústico, RF, infrarrojo y otros. También deberían incluirse combinaciones de cualquiera de los anteriores dentro del alcance de medio legible por ordenador.

La memoria 130 de sistema incluye medio de almacenamiento informático en forma de memoria volátil y/o no volátil tal como memoria 131 de sólo lectura (ROM) y memoria 132 de acceso aleatorio (RAM). Un sistema 133 de entrada/salida básico (BIOS), que contiene las rutinas básicas que ayudan a transferir información entre elementos en el ordenador 110, tal como durante el arranque, se almacena típicamente en la ROM 131. La RAM 132 típicamente contiene datos y/o módulos de programa que son inmediatamente accesibles a y/o que se operan actualmente mediante la unidad 120 de procesamiento. A modo de ejemplo, y sin limitación, la Figura 1 ilustra el sistema 134 operativo, programas 135 de aplicación, otros módulos 136 de programa y datos 137 de programa.

El ordenador 110 puede incluir también otro medio de almacenamiento informático extraíble/no extraíble, volátil/no volátil. A modo de ejemplo únicamente, la Figura 1 ilustra una unidad 141 de disco duro que lee desde o escribe en medio magnético no extraíble no volátil, una unidad 151 de disco magnético que lee desde o escribe en un disco 152 magnético extraíble no volátil, y una unidad 155 de disco óptico que lee desde o escribe en un disco 156 óptico extraíble no volátil tal como un CD ROM u otro medio óptico. Otro medio de almacenamiento informático extraíble/no extraíble, volátil/no volátil que puede usarse en el entorno de operación ejemplar incluye, pero sin limitación, casetes de cinta magnética, tarjetas de memoria flash, discos versátiles digitales, cinta de vídeo digital, RAM de estado sólido, ROM de estado sólido y similares. La unidad 141 de disco duro está típicamente conectada al bus 121 de sistema a través de una interfaz de memoria no extraíble tal como la interfaz 140, y la unidad 150 de disco magnético y la unidad 155 de disco óptico están típicamente conectadas al bus 121 de sistema mediante una interfaz de memoria extraíble, tal como la interfaz 150.

Las unidades y su medio de almacenamiento informático asociado, anteriormente descritos e ilustrados en la Figura 1, proporcionan almacenamiento de instrucciones legibles por ordenador, estructuras de datos, módulos de programa y otros datos para el ordenador 110. En la Figura 1, por ejemplo, la unidad 141 de disco duro se ilustra como que almacena el sistema 144 operativo, programas de aplicación 145, otros módulos 146 de programa y datos 147 de programa. Obsérvese que estos componentes pueden ser los mismos que o diferentes del sistema 134 operativo, programas 135 de aplicación, otros módulos 136 de programa y datos 137 de programa. Se proporciona al sistema 144 operativo, programas de aplicación 145, otros módulos 146 de programa y datos 147 de programa diferentes números en el presente documento para ilustrar que, como mínimo, son diferentes copias. Un usuario puede introducir comandos e información en el ordenador 110 a través de dispositivos de entrada tales como una tableta o digitalizadora 164 electrónica, un micrófono 163, un teclado 162 y dispositivo 161 apuntador, denominado comúnmente como ratón, bola de mando o almohadilla táctil. Otros dispositivos de entrada no mostrados en la Figura 1 pueden incluir una palanca de mandos, control de juegos, antena parabólica, escáner, o similares. Estos y otros dispositivos de entrada a menudo están conectados a la unidad 120 de procesamiento a través de una interfaz 160 de entrada de usuario que está acoplada al bus de sistema, pero que puede estar conectada mediante otra interfaz y estructuras de bus, tal como un puerto paralelo, puerto de juegos o un bus serie universal (USB). Un monitor 191 u otro tipo de dispositivo de visualización está conectado también al bus 121 de sistema mediante una interfaz, tal como una interfaz 190 de vídeo. El monitor 191 puede integrarse también con un panel táctil o similar. Obsérvese que el monitor y/o panel de pantalla táctil pueden estar físicamente acoplados a un alojamiento en el que está incorporado el dispositivo 110 informático, tal como en un ordenador personal de tipo tableta. Además, los ordenadores tales como el dispositivo 110 informático pueden incluir también otros dispositivos de salida periféricos tales como altavoces 195 y la impresora 196, que pueden estar conectados a través de una interfaz 194 periférica de salida o similar.

El ordenador 110 puede operar en un entorno en red usando conexiones lógicas a uno o más ordenadores remotos, tales como un ordenador 180 remoto. El ordenador 180 remoto puede ser un ordenador personal, un servidor, un encaminador, un PC de red, un dispositivo de pares u otro nodo de red común, y típicamente incluye muchos o todos los elementos anteriormente descritos con relación al ordenador 110, aunque únicamente se ha ilustrado un dispositivo 181 de almacenamiento de memoria en la Figura 1. Las conexiones lógicas representadas en la Figura 1

incluyen una red 171 de área local (LAN) y una red 173 de área extensa (WAN), pero pueden incluir también otras redes. Tales entornos de red son habituales en oficinas, redes informáticas en empresas, intranets e internet.

Cuando se usa en un entorno de interconexión de red LAN, el ordenador 110 está conectado a la LAN 171 a través de una interfaz de red o adaptador 170. Cuando se usa en un entorno de interconexión en red WAN, el ordenador 110 típicamente incluye un módem 172 u otros medios para establecer comunicaciones a través de la WAN 173, tal como internet. El módem 172, que puede ser interno o externo, puede estar conectado al bus 121 de sistema mediante la interfaz 160 de entrada de usuario u otro mecanismo apropiado. En un entorno en red, los módulos de programa representados con relación al ordenador 110, o porciones del mismo, pueden almacenarse en el dispositivo de almacenamiento de memoria remoto. A modo de ejemplo, y sin limitación, la Figura 1 ilustra programas 185 de aplicación remotos como que residen en un dispositivo 181 de memoria. Se apreciará que las conexiones de red mostradas son ejemplares y que pueden usarse otros medios para establecer un enlace de comunicaciones entre los ordenadores.

Coordinación de comunicación de datos con números de secuencia

Diversos aspectos de la tecnología descrita en el presente documento están dirigidos a un mecanismo que puede emplearse en un protocolo de comunicación de datos, tal como una versión modificada (2.x o superior) del protocolo SMB. En una implementación de ejemplo descrita en general en el presente documento, el mecanismo controla el flujo de datos / comandos en este protocolo de SMB revisado, que se usa para compartición de ficheros basada en Windows®. Sin embargo, como puede apreciarse fácilmente, la presente invención no está limitada a sistemas basados en Windows®, o al protocolo SMB, sino en su lugar, la técnica de ejemplo es aplicable a otros protocolos de compartición de ficheros y protocolos de comunicación de datos en general, incluyendo aquellos que no tratan necesariamente con datos de ficheros. Por ejemplo, son factibles numerosas maneras para implementar la presente invención, incluyendo para uso en comunicación con impresoras, tuberías de datos con nombre, dispositivos genéricos y así sucesivamente. Como tal, la presente invención no está limitada a ninguno de los ejemplos basados en ficheros particulares u otros usados en el presente documento, sino que en su lugar pueden usarse numerosas maneras que proporcionan beneficios y ventajas en la informática en general.

Volviendo a la Figura 2 de los dibujos, se muestra un diagrama de bloques que representa un entorno de interconexión en red de ejemplo en el que un cliente 202 se comunica con un servidor 204 a través de uno o más canales de comunicación. Aunque las funciones y componentes de la máquina 202 de cliente y el servidor 204 se describen como que están localizadas en dos ordenadores separados, tal como el sistema 110 informático principal y el sistema 180 informático remoto de la Figura 1, los componentes de estos dos ordenadores o funciones realizadas de esta manera pueden proporcionarse en una máquina, o pueden distribuirse a través de un número de ordenadores. Por ejemplo, un sistema informático puede comprender uno de una diversidad de dispositivos de aparatos de red, tales como servidores de impresión o las propias impresoras, así como dispositivos de almacenamiento NAS y similares.

Los comandos de sistema de ficheros en red desde un programa 206 de aplicación se manejan por un componente 208 redirector de cliente, que comunica con un módulo 210 de red común homólogo (SRVNET) para llevar a cabo los comandos en el sistema 212 de ficheros. En general, el cliente 202 establece una conexión y a continuación negocia con el servidor 204 para establecer finalmente una sesión. Como parte de esto, antes de que se procesen los comandos dirigidos al sistema de ficheros, el cliente y el servidor acuerdan un protocolo de comunicación. En este ejemplo, el protocolo acordado para esta conexión / sesión es SMB 2.0, en el cual un motor 220 de SMB del lado cliente emplea un controlador de SMB 2.0 para comunicaciones hasta y desde el servidor 204. El módulo 210 de red de control (SRVNET) emplea de manera similar un proveedor 226 de SMB 2.0 para manejar comunicaciones de cliente a través de esta conexión. El proveedor 226 incluye o está asociado de otra manera con el mecanismo de cumplimiento y estructuras de datos que aseguran que un cliente usa un número de secuencia apropiado, como se describe a continuación.

Volviendo al concepto de coordinación de comunicación de datos con números de secuencia, los números de secuencia proporcionan un mecanismo mediante el cual un servidor puede acelerar la cantidad de trabajo que un cliente dado puede emitir contra el servidor. Como se entenderá, esto se consigue haciendo que el servidor proporcione una ventana de números de secuencia disponibles que el cliente está permitido a usar cuando identifica un comando dado. Para implementar números de secuencia y el comportamiento deseado, se emplea el concepto de créditos, donde un crédito concede al cliente el derecho de consumir una porción de recursos del lado de servidor, incluyendo la memoria requerida para devolver una operación y los ciclos de CPU que puede consumir. Un cliente consume un crédito en cada comando enviado, y dependiendo de la respuesta del servidor, puede concederse cero, uno o más créditos adicionales. Un cliente no está autorizado a reutilizar un número de secuencia, y por lo tanto el número de comandos que un cliente puede emitir está controlado. Obsérvese que se usan por conveniencia números de secuencia que incrementan monotónicamente, sin embargo es equivalente cualquier número único (por sesión / conexión).

Por ejemplo, si un servidor concede a un cliente cinco créditos, el servidor está proporcionando al cliente el derecho de presentar hasta cinco operaciones simultáneamente. Cuando un servidor necesita acelerar un cliente, el servidor lo hace reduciendo los créditos disponibles para ese cliente. Cuando un servidor desea proporcionar a un cliente

más recursos con los que funcionar, el servidor lo hace concediendo créditos.

Esto proporciona al servidor diversas opciones. Concediendo cero créditos a un cliente, el servidor recorta los recursos asignados a ese cliente. Como alternativa, devolviendo un crédito, el servidor mantiene el tamaño de ventana anterior. Devolviendo más de un crédito, el servidor permite al cliente más recursos para ejecutar comandos. Una restricción es que el servidor no puede permitir que el tamaño de ventana alcance cero (sin números de secuencia válidos), a menos que el servidor tenga un procedimiento para conceder un crédito fuera de banda, suponiendo que el protocolo sea estrictamente un protocolo de comando-respuesta. Obsérvese que si existe un procedimiento para conceder a un cliente créditos sin requerir que el cliente los solicite en el protocolo que está usando este procedimiento, entonces no se aplica la restricción.

- 10 Obsérvese que una solicitud / respuesta de negociación tiene un número de secuencia, (también denominado como un identificador de mensaje, o MID) de cero, y un tamaño de ventana de uno. En SMB 2.0, el siguiente encabezamiento facilita el paso de tal información, como se describe adicionalmente en la solicitud de patente de Estados Unidos en trámite junto con la presente anteriormente indicada titulada "Data Communication Protocol":

```
typedef struct _SMB2_HEADER {
    UCHAR      Protocol[4];          // Contiene 0xFE, 'S', 'M', 'B'
    USHORT     StructureSize;        // = sizeof(SMB2_HEADER).
                                        // (creación de versiones)
    USHORT     Epoch;                // incrementar cada vez que
                                        // el servidor se reinicia

    NTSTATUS    Status;              // Estado del comando

    USHORT     Command;              // El comando para esta unión de
                                        // paquete
    {
        USHORT  CreditsRequested;    // En envío de cliente, solicitar más
                                        // créditos
        USHORT  CreditsGranted;      // En respuesta de servidor, créditos
                                        // concedidos a cliente
    };

    ULONG      Flags;
    ULONG      Reserved;

    UINT64     MessageId;            // Identifica este envío/respuesta
                                        // de mensaje

    union
    {
        struct
        {
            UINT64  ProcessId;        // Procesar identificador

            UINT64  SessionId;        // Identificador de sesión
            ULONG   TreeId;           // Identificador de conexión de árbol
        };

        struct
        {
            UINT64  AsyncId;          // Usado para identificar comandos
                                        // de larga duración
        };
    };

    UCHAR      Signature[8];        // Firma para el paquete

    ULONG      NextCommand;          // Desplazamiento desde siguiente
} _SMB2_HEADER, *PSMB2_HEADER;
```

Como puede observarse a partir de la estructura de encabezamiento anterior, el cliente solicita tantos créditos como desee, sin embargo, el servidor está en control de conceder créditos al cliente. El servidor por lo tanto tiene la capacidad de reducir o aumentar la ventana basándose en la identidad del cliente, comportamiento o cualquier otro atributo o criterio. Los números de secuencia también proporcionan una manera para identificar de manera inequívoca un comando enviado desde el cliente al servidor para una conexión dada.

El cliente y servidor comienzan estableciendo una ventana de comando. La ventana de comando empieza usando cualquiera de un número de secuencia inicial (ISN) por defecto o negociado (también denominado como un identificador de mensaje inicial, o MID) y número de créditos (NoC), que representa el rango de números permisibles que el servidor aceptará para identificar un comando dado. Por lo tanto, la ventana de comando comprende inicialmente [ISN, ISN+NoC-1]. Para la mayoría de los protocolos, por defecto puede ser ISN=1, NoC=1, por lo que cuando se negocia en primer lugar la ventana de comando es simplemente [1,1], que representa que el único número de secuencia que el servidor aceptará para identificar un comando es uno (1).

A medida que la comunicación progresa, el cliente mueve hacia abajo los números en la ventana usando números hacia arriba dentro del intervalo. Una vez que se usa un número, no puede reutilizarse, ya que esto se hace cumplir por el servidor. Al mismo tiempo, el servidor puede ampliar, según se determina por el servidor, el final de la ventana concediendo más créditos al cliente. Por ejemplo, si la ventana de comando es [A, B], cuando el cliente envía el comando A, la ventana de comando válido se hace esencialmente [A+1, B]. Cuando el servidor responde al comando A, puede conceder al cliente en cualquier lugar desde cero a cualquier número práctico de créditos. Por lo tanto, si el servidor devolvió N créditos, la ventana de comando válido se hace [A+1, B+N].

El uso de los números de secuencia en el intervalo permisible no se requiere que esté en orden. El protocolo se configura para facilitar uso asíncrono de los números de secuencia, siempre que el número que se esté usando esté dentro del rango válido. Esto permite al protocolo de red enviar paquetes a medida que están disponibles, en lugar de intentar forzar envíos ordenados. Por lo tanto, si el número de secuencia A se solicita para un paquete muy grande, pero mientras que las memorias intermedias están preparando llegan los envíos para A+1 y A+2 y son muy pequeños, es legal enviar A+1 y A+2 (siempre que el final de la ventana sea $\geq A+2$) sin esperar que empiece el envío de A.

Obsérvese que si hay una ventana de comando válido de [1,5], y se envían los paquetes 2, 3, 4, el servidor puede conceder créditos de vuelta que permiten una ventana de [1,8] excepto {2, 3, 4} (que significa todos los números entre 1 y 8 excepto 2 hasta 4). Eventualmente, el servidor probablemente dejará de conceder créditos a menos que se envíe el paquete 1 y se permita a la ventana deslizarse. Si en este punto, el cliente envía 1, y el servidor responde concediendo un crédito, entonces la ventana se hace [5,9].

El cumplimiento de la ventana de comando válido se hace en el lado del servidor. Este sistema permite que una estructura del lado de cliente sea tan sencilla como tener el número de secuencia actual y el número de secuencia máximo, y usar comparados e incrementos interbloqueados como el único procedimiento de sincronización requerido.

Por lo tanto, una ventana de comando válido, también denominada como una operación válida o ventana de operación válida comprende la ventana de identificadores válidos que el servidor aceptará. El cliente envía cada comando posterior con el siguiente número de secuencia válido, (hasta sus créditos válidos), y no necesita mantener una vista de la ventana válida. Se requiere que el cliente entienda el concepto de un "Tamaño de Ventana Máximo" como se describe a continuación). Ejemplos del uso de la ventana de comando / operación válidos se exponen a continuación.

Cualquier operación que pudiera bloquear durante una cantidad de tiempo indefinida, tal como un cambio-notificación o una lectura de tubería con nombre; o crear (ya que puede depender de una pausa de bloqueo oportuno), se considera una operación de bloqueo. Para facilitar tales operaciones, el cliente puede suministrar un valor de "contexto de operación", es decir, una bandera de bloqueo, en el envío del comando. El servidor a continuación responderá cuando inicie satisfactoriamente la operación, permitiendo que se incremente el número de secuencia, incluso aunque la operación esté aún en proceso en el lado del servidor. Sin embargo, los recursos soportados por tales operaciones de ejecución largas pueden ser a menudo un subconjunto de los recursos requeridos para un comando normal. Por lo tanto, se permite a un servidor determinar el número máximo de "Créditos de operación de bloqueo" (también denominados como créditos de operación larga, o LOC) que se permite a un cliente consumir. Los números de secuencia también permiten comandos de larga ejecución y comandos con múltiples respuestas desde el servidor, equilibrados controlando cuántos recursos puede consumir el cliente.

Por lo tanto, una extensión a la ventana de comando válido es permitir que la ventana continúe deslizando de manera normal, y no soportarse por operaciones que pueden tomar una cantidad de tiempo indefinida. Para este fin, se concede al cliente un número dado de créditos de operación de bloqueo por el servidor y cualquier operación emitida con la bandera de operación de bloqueo consumirá un crédito de operación de bloqueo. Cuando el servidor recibe un comando, el servidor puede enviar una respuesta intermedia de vuelta al cliente con un conjunto de bandera de este tipo que realiza acuse de recibo de la recepción de un comando de larga duración, y devuelve un ID de comando de larga duración, también denominado como un identificador asíncrono (AsyncID). Esta respuesta

permite que la ventana de comando válido deslice como lo haría normalmente. Cuando el comando de larga duración está completado, se envía una nueva respuesta de vuelta al cliente usando el ID de comando de larga duración para indicar a qué paquete está respondiendo. Esta arquitectura de envío-respuesta-respuesta permite que la ventana continúe moviéndose, y el mecanismo de crédito permite al servidor retener el control sobre cuántos recursos puede consumir el cliente. El servidor puede incluso reducir la ventana de comando válido si hubiera un gran número de operaciones de larga duración en progreso desde el cliente.

Implementaciones alternativas incluyen un protocolo que permite al cliente sugerir al servidor que un comando de larga duración puede estar llegando. Otra implementación del concepto asíncrono puede también hacer que el cliente emita una “aceptación” intermedia o similar, tal como una operación asíncrona/de bloqueo tome la forma, “enviar-recibir-enviar-recibir” a diferencia de simplemente enviar-recibir-recibir. Obsérvese que algunos transportes subyacentes (por ejemplo, TCP) están ajustados en ocasiones para tráfico de solicitud-respuesta, y pueden introducir retardos en una situación de solicitud-respuesta-respuesta.

Debido a la naturaleza asíncrona del protocolo y los transportes, la ventana válida actual no será directamente igual al ID de secuencia mínimo actual más los créditos, puesto que algunos comandos intermedios pueden recibirse en primer lugar, o, por ejemplo, el comando con el ID de secuencia = 1 puede tomar un largo tiempo para procesar. Sin embargo, el servidor puede poner un límite en cómo de larga esta ventana válida puede aumentar antes de que se deje de aumentar. Continuando con el ejemplo anterior, el servidor puede especificar que el tamaño de ventana máximo es diez (10). Esto significa que la ventana de operación (comando) válida puede aumentar hasta [1,10] si el servidor recibe y procesa los paquetes 2, 3, 4, 5 y 6 antes de que reciba o complete el procesamiento del paquete 1. Por lo tanto, los números de secuencia que serían válidos son 1, 7, 8, 9, 10. Sin embargo, si el servidor a continuación recibe el paquete 7 y lo procesa, la ventana de operación (comando) válida no se deslizará a [1,11], sino que permanece a [1,10] con números de secuencia aceptables de 1, 8, 9, 10. La respuesta al comando 8 indicaría un crédito (-1) para decir al cliente que el cliente está alcanzado el final de sus límites aceptables, es decir, su valor de créditos es ahora 4. Esto es una manera en la que el servidor evita que el cliente nunca se salte un número dado en la secuencia que evitaría que la ventana se deslizará bien. Esto demuestra también el valor de emitir “Operaciones de Bloqueo” contra un servidor para comandos que tomarán un tiempo largo.

Rastrear la ventana de comando válido en el lado del servidor puede ser computacionalmente costoso ya que el servidor necesita rastrear conjuntos entrecruzados. Para simplificar esto, una implementación de ejemplo establecerá el tamaño de ventana máximo anteriormente descrito, como el más largo que el servidor dejara que se hiciera la ventana de comando. Una vez que este se establece, el servidor asigna una memoria intermedia que representa este tamaño, y a medida que llegan los comandos, el valor en esa localización en la memoria intermedia se cambia. Siempre que el tamaño de ventana permanezca menor o igual al tamaño de ventana máximo, entonces como el inicio de la ventana se mueve hacia delante, el servidor mueve su puntero de memoria intermedia hacia delante. Análogamente, cuando el final aumenta a medida que se conceden créditos, el servidor mueve su puntero final a lo largo de la memoria intermedia. La aritmética maneja situaciones donde la memoria intermedia “envuelve” el tamaño de ventana máximo. Es posible hacer aumentar el tamaño de ventana máximo dinámicamente, asignando una memoria intermedia mayor y copiando los valores actuales en ella. Las operaciones interbloqueadas se usan para rastrear el estado de los comandos válidos en la ventana, empezando como DISPONIBLE. Tras recibir el comando desde el cliente, pasan a EN_PROGRESO, y cuando se envía la respuesta (o respuesta intermedia para un caso de comando de ejecución larga), pasa a USADO. Si el valor que pasa a USADO es el primer valor en la ventana, la ventana se desliza hacia delante hasta que se encuentre un valor no USADO.

Otra alternativa permite que el servidor revoque créditos mediante un canal alternativo o a través de comunicación no en secuencia. Por ejemplo, considerándose que el servidor ha concedido al cliente diez créditos, pero desea recortar al cliente hasta cinco créditos, que normalmente no ocurriría hasta que el cliente usara cinco comandos. Si el cliente está en espera, el servidor puede indicar al cliente que debe usar cinco créditos en los siguientes N segundos, o estará en infracción y terminará (o pierde los créditos). Esto permite al servidor decelerar a clientes sin basarse en que el cliente mueva su propia ventana.

Volviendo a una explicación de la operación de la invención mediante el uso de diversos ejemplos, el estado actual se expondrá en el formato a continuación:

Mín:	Créditos actuales:	Créditos:	Válido: [1,5]	Máx:
1	(5,1)	(5,1)	excepto {}	[1,11]

La columna mín muestra el número de secuencia de cliente sin usar más bajo que el cliente está permitido a usar; los créditos actuales son cuántos créditos se conceden al cliente, en forma de (créditos normales, créditos de operación de bloqueo). Un cliente consume un crédito en un envío, y posiblemente se vuelve a incrementar en la recepción (dependiendo de la respuesta). La siguiente columna, simplemente “Créditos” muestra el número de créditos máximo actual que se permite al cliente.

Válido y Máx representan estructuras del lado de servidor para validación de identificador de secuencia; (obsérvese que el cliente no necesita tener ninguna idea acerca de estas). Válido muestra la ventana de operación válida junto con las excepciones de los identificadores de secuencia que ya se usaron (por ejemplo, rastreados por mapa de bits), y Máx muestra la ventana de comando máxima que el cliente puede rellenar antes de completar la primera operación, es decir, la operación que provocaría que la ventana se desplazara.

Las Figuras 3-9 representan un ejemplo de cómo la ventana 330_s de operación válida (operación válida) aumenta según se mantiene en el servidor 204; (obsérvese que en las Figuras 3-9, a medida que la ventana de operación válida cambia, el primer dígito de su etiqueta cambia, por ejemplo, la Figura 3 tiene una ventana de operación válida etiquetada 330_s, la Figura 4 tiene una ventana de operación válida etiquetada 430_s, y así sucesivamente).

En la Figura 3, se conceden 5 créditos a un cliente y un número de secuencia de inicio (o MID) de 1. Algunos criterios 320 como se han descrito en general anteriormente, por ejemplo, el tipo de cliente, se usan por un componente 322 asignador para determinar la cantidad a conceder al cliente. En este ejemplo, la ventana de operación válida actual es por lo tanto [1,5], según se representa verticalmente en la Figura 5 mediante los dígitos individuales en la ventana 330_s. El servidor 204 por lo tanto aceptará un paquete con el número de secuencia 1, 2, 3, 4 o 5 desde este cliente 202. Esto se representa de manera alternativa en el formato anteriormente descrito como sigue:

Mín:	Créditos actuales:	Créditos:	Válido: [1,5]	Máx:
1	(5,1)	(5,1)	excepto {}	[1,11]

La ventana 320_s de operación válida se usa para rechazar paquetes de manera trivial; si radican en la ventana 320_s de operación válida, el servidor 204 a continuación comprueba un mapa de excepción interno para garantizar un número de secuencia que no se ha usado aún.

En el caso típico de una recepción que incrementa monotónicamente, el cliente envía un paquete con MID = 1, y el cliente y servidor pasan al estado representado en la Figura 4 y en la tabla a continuación; (obsérvese que [1,5] se consideran válidos, pero 1 es únicamente verdaderamente válido para una respuesta desde el servidor, no para otra recepción desde el cliente, que caería frente a la comprobación Mín=2 en primer lugar).

Mín:	Créditos actuales:	Créditos:	Válido: [1,5]	Máx:
2	(4,1)	(5,1)	excepto {}	[1,11]

En la Figura 4, el servidor recibe y procesa el comando con el número de secuencia=1. Esto significa que 1 ya no es un número de secuencia válido para comandos posteriores, según se representa por el {1} entre corchetes en la columna válida / excepción en la tabla anterior y en la ventana de operación válida etiquetada 430_s en la Figura 4.

Cuando el servidor responde, concede al cliente un crédito adicional (+1) en la respuesta, y desliza la ventana, como se representa en la Figura 5 y en la tabla a continuación:

Mín:	Créditos actuales:	Créditos:	Válido: [2,6]	Máx:
2	(5,1)	(5,1)	excepto {}	[2,12]

El cliente ahora tiene una ventana válida de [2,6]. Considérese que cuando tiene lugar una recepción fuera de orden, por ejemplo, el envío asíncrono sobre el transporte es de manera que el servidor recibió el comando 3 antes del comando 2 (y lo responde). La ventana 630_s de operación válida existiría entonces conceptualmente como se representa en la Figura 6 y como sigue:

Mín:	Créditos actuales:	Créditos:	Válido: [2,7]	Máx:
2	(5,1)	(5,1)	excepto {3}	[2,12]

Obsérvese que en la tabla, la ventana válida se amplía, pero la ventana máxima no se desliza. Sin embargo, cuando el servidor recibe el comando 2 y responde, la ventana 730_s se deslizará a través de ambas, como en la Figura 7 y como a continuación:

Mín:	Créditos actuales:	Créditos:	Válido: [4,8]	Máx:
4	(5,1)	(5,1)	excepto {}	[4,14]

Considérese que un siguiente cliente malicioso intente usar recursos en el servidor enviando comandos y rechazando las respuestas. En este punto el cliente ha enviado al servidor los comandos 4 y 5, y está rechazando las respuestas. El estado se hace como se representa en la Figura 8 y como a continuación:

Mín:	Créditos actuales:	Créditos:	Válido: [4,8]	Máx:
6	(3,1)	(5,1)	excepto {4,5}	[4,14]

- 5 Después de que se envíen los comandos con los identificadores de secuencia 6, 7, 8, debido al cumplimiento de crédito, el cliente está sin créditos y todos los paquetes se rechazan, como se representa en la Figura 9 y en la tabla a continuación:

Mín:	Créditos actuales:	Créditos:	Válido: [4,8]	Máx:
9	(0,1)	(5,1)	excepto {4, 5, 6, 7, 8}	[4,14]

- 10 Volviendo a los ejemplos de cumplimiento de ventana máxima, se usan las Figuras 10-13 y tablas similares. Considérese que un cliente malicioso intenta enviar un paquete N+1 y superior, sin enviarse el paquete N. Los ejemplos de las Figuras 10-13 empezarán en el estado anterior al último atacante, es decir, la Figura 10 es esencialmente la Figura 7:

Mín:	Créditos actuales:	Créditos:	Válido: [4,8]	Máx:
4	(5,1)	(5,1)	excepto {}	[4,14]

- 15 El cliente envía los comandos 5, 6, 7, 8, 9,10, y el servidor responde, pero sin enviar el comando 4. Este estado se muestra en la Figura 11 y en la tabla a continuación:

Mín:	Créditos actuales:	Créditos:	Válido: [4,14]	Máx:
4	(5,1)	(5,1)	excepto {5, 6, 7, 8, 9, 10}	[4,14]

Obsérvese que el cliente aún tiene cinco créditos puesto que hay aún cinco intervalos viables en la ventana de comandos. Sin embargo, cuando el cliente envía, y el servidor responde, el comando 11, existe el estado en la Figura 12 y en la tabla a continuación:

Mín: 4	Créditos actuales: (4,1)	Créditos: (5,1)	Válido: [4,14] excepto {5, 6, 7, 8, 9, 10, 11}	Máx: [4,14]
--------	--------------------------	-----------------	--	-------------

- 20 Obsérvese que el cliente a continuación continúa con 12, 13, 14. El número de créditos disponible del cliente se ha reducido en uno para cada comando puesto que no hay un intervalo disponible en la ventana máxima. Esto continuará para los comandos 12, 13, 14:

Mín: 4	Créditos actuales: (1,1)	Créditos: (5,1)	Válido: [4,14] excepto {5, 6, 7, 8, 9, 10, 11, 12, 13, 14}	Máx: [4,14]
--------	--------------------------	-----------------	--	-------------

- 25 Ahora el único comando que el servidor aceptará desde el cliente es el comando 4, como se representa en la Figura 13.

- 30 Como se apreciará fácilmente, los atributos de una ventana de números de secuencia son muy ventajosos en un número de escenarios deseables. Algunos de estos escenarios incluyen prevenir la denegación de servicio, permitir la calidad de servicio, proporcionar un lenguaje común para el cliente y servidor para hacer referencia a comandos que se ejecutaron a través de una conexión dada, permitir comandos de larga ejecución y comandos con múltiples respuestas desde el servidor, pero equilibrarlo con controlar cuántos recursos puede consumir el cliente, y permitir uso continuo de firmas de seguridad.

- 35 Con respecto a prevenir la denegación de servicio, el servidor puede limitar la cantidad de recursos que una conexión de cliente dado puede consumir, hasta que el servidor haya autenticado al cliente y el cliente se esté comportando correctamente. Por ejemplo, permitiendo al servidor controlar los recursos asignados a un cliente, cuando se detecta un ataque aparente, el servidor puede entrar en un "modo de pánico" donde reduce los recursos disponibles para sus clientes a un mínimo, y les concede de vuelta a una base de confianza. El servidor proporciona a cada cliente una pequeña ventana suficiente para que la función tenga lugar, pero evita que cualquier cliente único

desborde esencialmente todos los recursos. Cuando el ataque ha terminado o se ha reducido, el servidor puede empezar a volver a conceder recursos a los clientes que demuestran confianza.

Con respecto a permitir calidad de servicio, el esquema de ventana variable permite al servidor escalar las cantidades de recursos asignados a clientes basándose en su identidad y/o su comportamiento. Por ejemplo, el servidor puede permitir más recursos a un servidor web que se conecta al servidor de ficheros que a un único usuario que accede a documentos individuales. Como otro ejemplo, si otro servidor es un servidor de base de datos que está accediendo a un servidor de ficheros, el servidor de ficheros puede ponderar el número de créditos concedidos más alto que los concedidos a un usuario medio.

Además, cuando se controla la calidad de servicio, la asignación de recursos a clientes puede modificarse dinámicamente basándose en las diversas necesidades de los clientes. Esto permite que el servidor conceda créditos de una manera completamente equitativa, o que tenga en cuenta otra información. Un administrador puede configurar máquinas en base de prioridad de recursos, y esto puede utilizarse y cambiarse dinámicamente a medida que los usuarios se conectan y desconectan.

Los números de secuencia pueden proporcionar también un lenguaje común para que el cliente y servidor hagan referencia a comandos que se ejecutaron a través de una conexión dada. Esto ayuda a la creación de diversas características, incluyendo manejadores persistentes. Por ejemplo, puesto que tanto el cliente como el servidor han acordado en un lenguaje común para identificar comandos a medida que se envían y reciben, es decir, el mecanismo de número de secuencia, cuando tiene lugar una desconexión hay una manera fácil después del restablecimiento de conexión para que el servidor y cliente determinen cuáles comandos se recibieron y cuáles no. Sin un esquema de ajuste de este tipo, es más difícil rastrear, particularmente cuando se elige el identificador de comando por el cliente y potencialmente se vuelve a usar.

Los números de secuencia permiten además uso continuo de firmas de seguridad, sin los problemas de rendimiento extremo de los modelos actuales; los envíos no necesitan estar en secuencia, (aunque la suma de comprobación de todo el paquete necesitará calcularse aún, y recibirse todo el paquete antes de emitir). Con respecto a firma de paquete, la reproducibilidad no es posible. Más particularmente, un protocolo de red que hace firma necesita embeber un número de índice en el paquete para evitar la reproducibilidad del paquete firmado, de otra manera un atacante simplemente vuelve a emitir el paquete sin tener que volver a firmarlo, y el paquete permanece válido. Otros procedimientos incluyen indicaciones de tiempo y similares, sin embargo esto requiere alguna forma de sincronización entre el cliente y servidor. Si se usan números de índice, a menudo el tráfico de red entre el cliente y el servidor se hace en serie, puesto que el cliente debe asegurar que el servidor recibe el paquete 1 antes de que envíe el paquete 2.

Con un número de secuencia embebido como el identificador de comando, y una ventana válida soportada en el servidor, los envíos de comandos paralelos con números de secuencia en ellos pueden tener lugar de manera natural. El servidor hace cumplir la ventana válida, por lo que cada comando solo puede emitirse una vez, por lo que la reproducibilidad no es un problema, siempre que el protocolo asegure que la clave usada para firmar sea única para cada conexión autenticada. Obsérvese que si el identificador de comando se da la vuelta, entonces la reproducibilidad podría hacerse un problema, y por lo tanto los números de secuencia en el orden de 32 bits o 64 bits son deseables para evitar esto, 64 bits más deseable probablemente si se permite el re-establecimiento de conexiones rotas.

Aunque la invención es susceptible a diversas modificaciones y construcciones alternativas, se muestran ciertas realizaciones ilustradas de la misma en los dibujos y se han descrito anteriormente en detalle. Debería entenderse, sin embargo, que no hay intención de limitar la invención a las formas específicas desveladas, sino al contrario, la intención es cubrir todas las modificaciones, construcciones alternativas y equivalentes que caen dentro de la invención como se define en las reivindicaciones.

REIVINDICACIONES

1. Un procedimiento realizado en un entorno informático de red que tiene un cliente (202) y servidor (204), comprendiendo el procedimiento:

5 conceder al cliente al menos un crédito, correspondiendo cada crédito a un número de secuencia único que el cliente puede usar para enviar un comando al servidor, formando el conjunto de uno o más números de secuencia únicos una ventana de comando válido; y
hacer cumplir que para cada comando recibido, el comando incluya un número de secuencia único que está dentro de la ventana de comando válido y que el número de secuencia único no se haya usado antes con otro comando;
10 en el que

una vez que se usa un número de secuencia, el cliente no está autorizado a reutilizar dicho número de secuencia.

2. El procedimiento de la reivindicación 1, que comprende adicionalmente, mantener un tamaño de ventana máximo de manera que un cliente (202) que tiene créditos no puede enviar un comando con un número de secuencia por encima de un número de secuencia máximo que corresponde al tamaño de ventana máximo.

15 3. El procedimiento de la reivindicación 1, que comprende adicionalmente, en respuesta a recibir un comando, devolver un número de secuencia sin usar al cliente (202) que ajusta la ventana.

4. El procedimiento de la reivindicación 3, en el que el número de secuencia sin usar es un valor aumentando monotónicamente con relación a un último número en la ventana de comando válido.

20 5. El procedimiento de la reivindicación 1, en el que cada número de secuencia es un valor de tamaño fijo de 32 bits de longitud o mayor.

6. El procedimiento de la reivindicación 1, que comprende adicionalmente, conceder al cliente (202) al menos un crédito de operación de bloqueo, y hacer cumplir en el servidor que el cliente no supere un número máximo de créditos de operación de bloqueo usados.

25 7. El procedimiento de la reivindicación 6, que comprende adicionalmente, recibir un comando que corresponde a un crédito de operación de bloqueo, devolver datos que indican que la operación de bloqueo está en progreso, junto con un identificador para esa operación que se devolverá tras la finalización de la operación de bloqueo.

8. El procedimiento de la reivindicación 7, que comprende adicionalmente, devolver un número de secuencia sin usar al cliente (202) que ajusta la ventana en respuesta a recibir el comando de operación de bloqueo.

30 9. El procedimiento de la reivindicación 1, que comprende adicionalmente, recibir una solicitud para un crédito adicional del cliente (202), y denegar la solicitud para reducir el número de créditos disponibles para el cliente.

10. El procedimiento de la reivindicación 1, que comprende adicionalmente, revocar al menos un crédito previamente concedido al cliente (202).

11. El procedimiento de la reivindicación 10, que comprende adicionalmente, informar al cliente (202) que al menos un crédito se revocará a menos que se use antes de la revocación.

35 12. Un medio legible por ordenador que tiene instrucciones ejecutables por ordenador, que cuando se ejecutan en un servidor (204), provocan que el servidor realice las etapas, que comprenden:

conceder a un cliente (202) una pluralidad de créditos, correspondiendo cada crédito a un número de secuencia único que el cliente usa para enviar un comando a un servidor, formando los números de secuencia una ventana de comando válido, en el que una vez que se usa un número de secuencia, el cliente no está autorizado a reutilizar dicho número de secuencia; y
40 tras recibir un comando desde el cliente, en el que el comando incluye un número de secuencia único:

verificar que el número de secuencia único está dentro de la ventana y no se ha usado antes con otro comando;
eliminar el número de secuencia correspondiente de entre aquellos que el cliente puede usar y consumir un
45 crédito; y
determinar si conceder o no al cliente al menos otro crédito.

13. El medio legible por ordenador de la reivindicación 12, que tiene adicionalmente instrucciones ejecutables por ordenador, que cuando se ejecutan provocan que el servidor (204) determine conceder al cliente al menos otro crédito, y ajustar la ventana de comando válido con números de secuencia correspondientes en base a cada crédito
50 concedido.

14. El medio legible por ordenador de la reivindicación 12, que tiene adicionalmente instrucciones ejecutables por ordenador, que cuando se ejecutan provocan que el servidor (204) determine no conceder al cliente otro crédito reduciendo de esta manera dinámicamente los créditos disponibles para el cliente.
- 5 15. El medio legible por ordenador de la reivindicación 12, que tiene adicionalmente instrucciones ejecutables por ordenador, que cuando se ejecutan provocan que el servidor (204) determine no conceder al cliente otro crédito evaluando el tamaño de ventana máximo.
16. El medio legible por ordenador de la reivindicación 12, que tiene adicionalmente instrucciones ejecutables por ordenador, que cuando se ejecutan provocan que el servidor (204) revoque al menos un crédito previamente concedido al cliente.
- 10 17. El medio legible por ordenador de la reivindicación 16, que tiene adicionalmente instrucciones ejecutables por ordenador, que cuando se ejecutan provocan que el servidor (204) informe al cliente que al menos se revocará un crédito a no ser que se use antes de la revocación.
18. Un sistema del lado de servidor en un entorno informático de red que tiene un cliente (202) y servidor (204), comprendiendo el sistema del lado de servidor:
- 15 un mecanismo adaptado para limitar el uso de recursos de servidor controlando un número de créditos concedidos al cliente, estando el mecanismo adaptado para:
- establecer una ventana de operación válida, que contiene un número de secuencia único para cada crédito concedido al cliente; en el que
- 20 dicho mecanismo comprende un mecanismo de cumplimiento adaptado para asegurar que para permitir operaciones de servidor adicionales sobre un comando recibido, el comando incluye un número de secuencia único que está dentro de la ventana de operación válida y que el número de secuencia único no se ha usado antes con otro comando, en el que una vez que se usa un número de secuencia, el cliente no está autorizado a reutilizar dicho número de secuencia; y
- 25 dicho mecanismo comprende un mecanismo de asignación adaptado para controlar créditos concedidos al cliente y los números de secuencia únicos dentro de la ventana de operación válida.
19. El sistema de la reivindicación 18, en el que el mecanismo adaptado para limitar el uso de recursos de servidor está asociado con un controlador de protocolo de bloque de mensajes de servidor.
20. El sistema de la reivindicación 18, en el que el mecanismo incluye adicionalmente medios adaptados para revocar al menos un crédito previamente concedido al cliente.

30

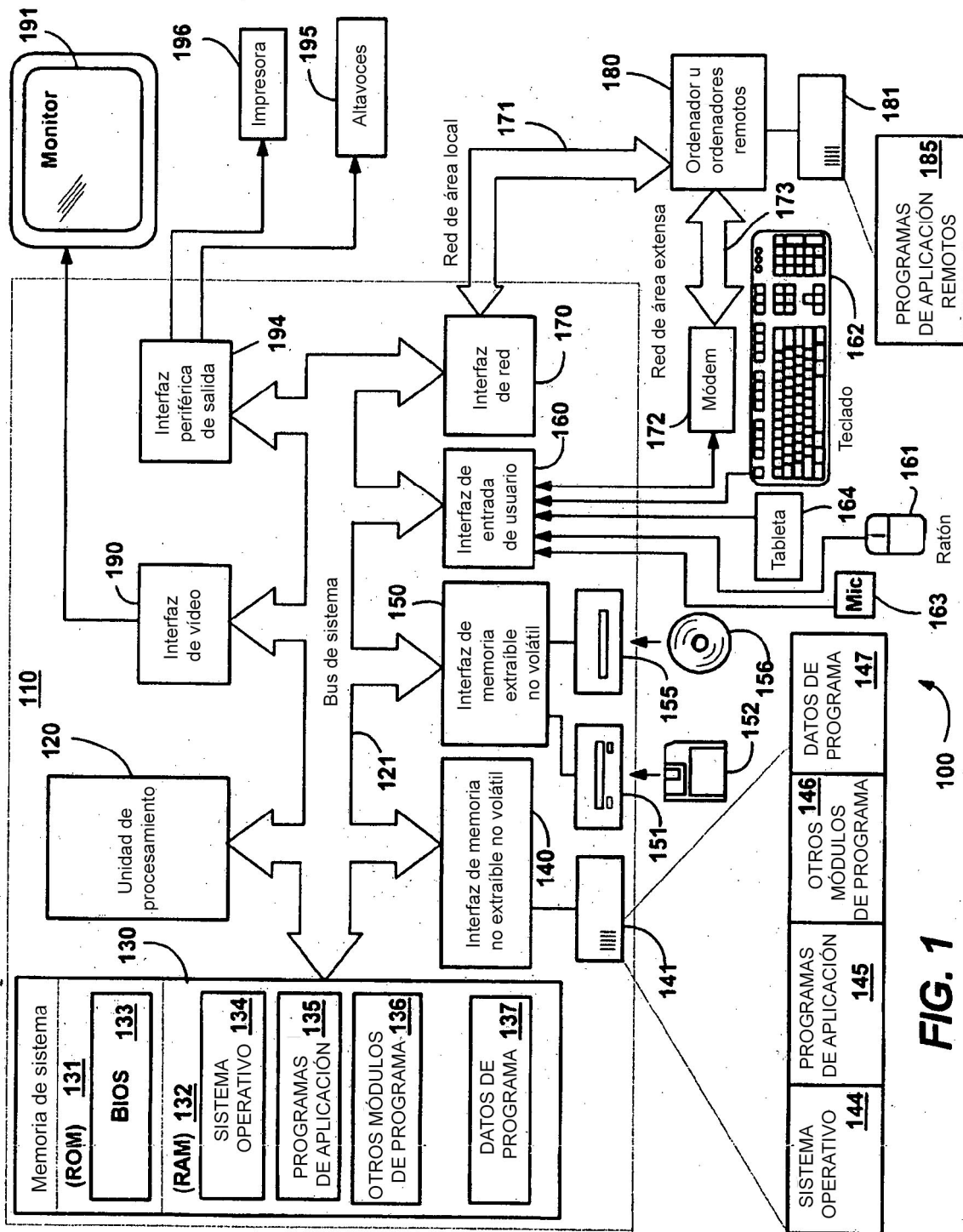


FIG. 1

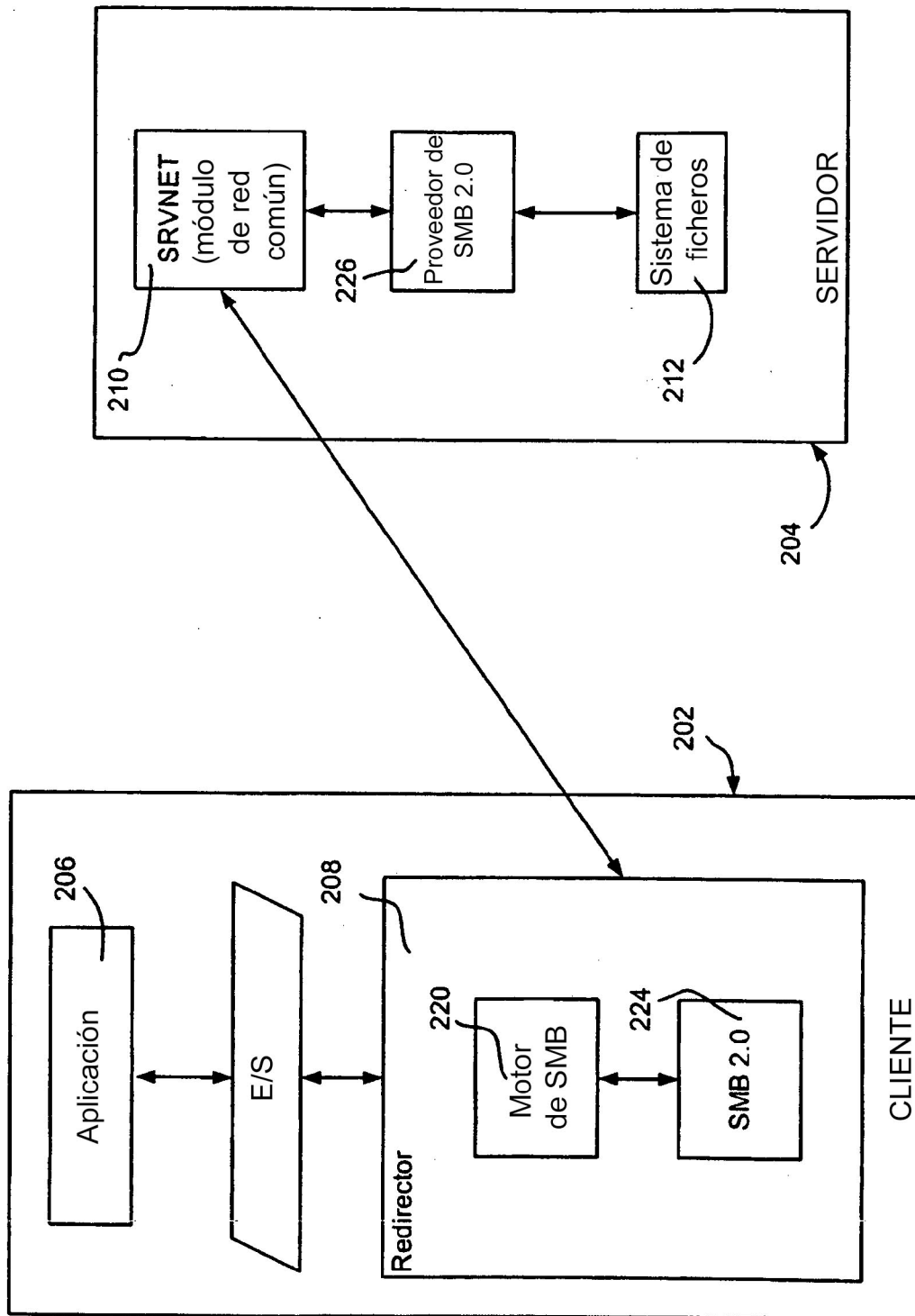


FIG. 2

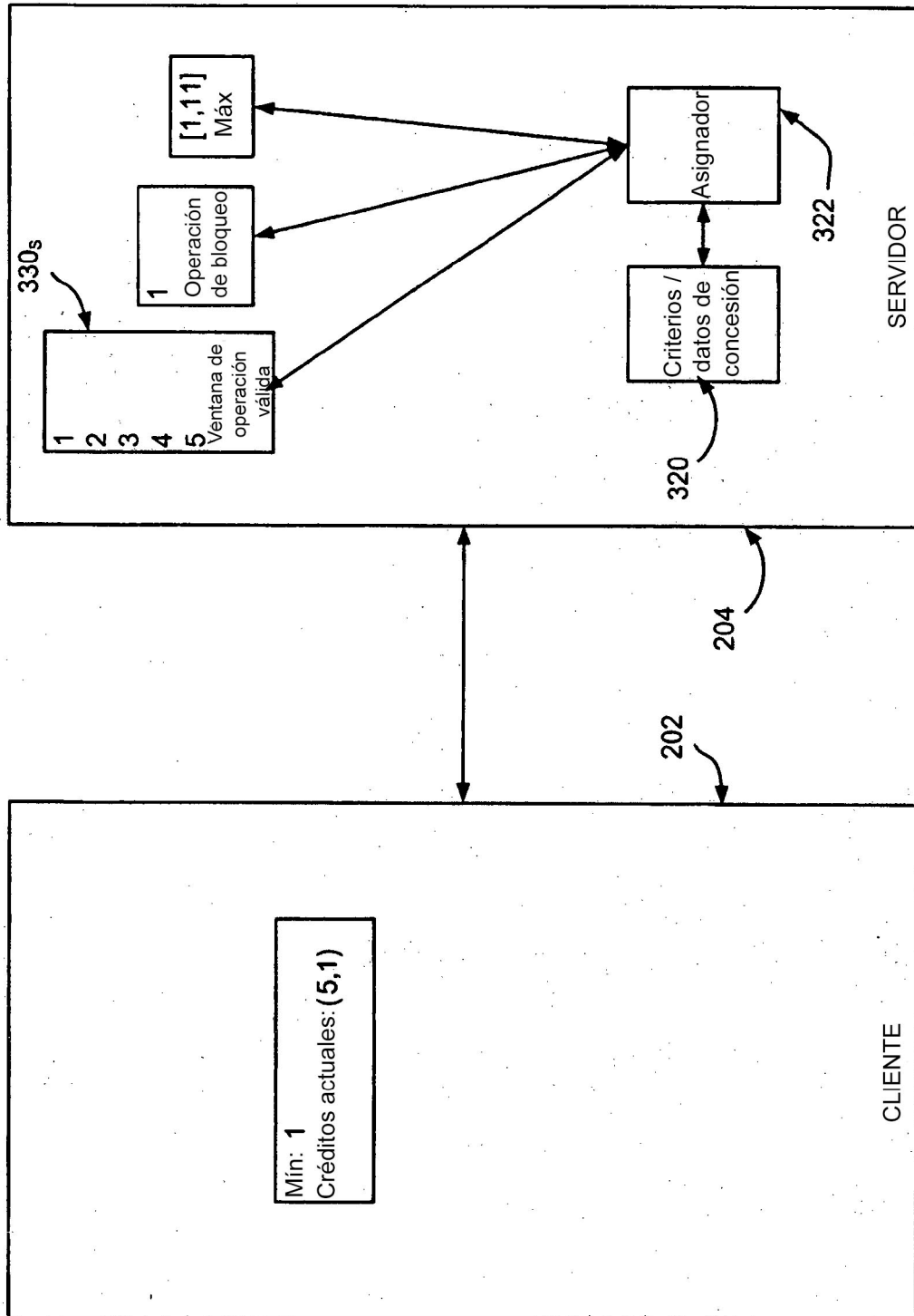


FIG. 3

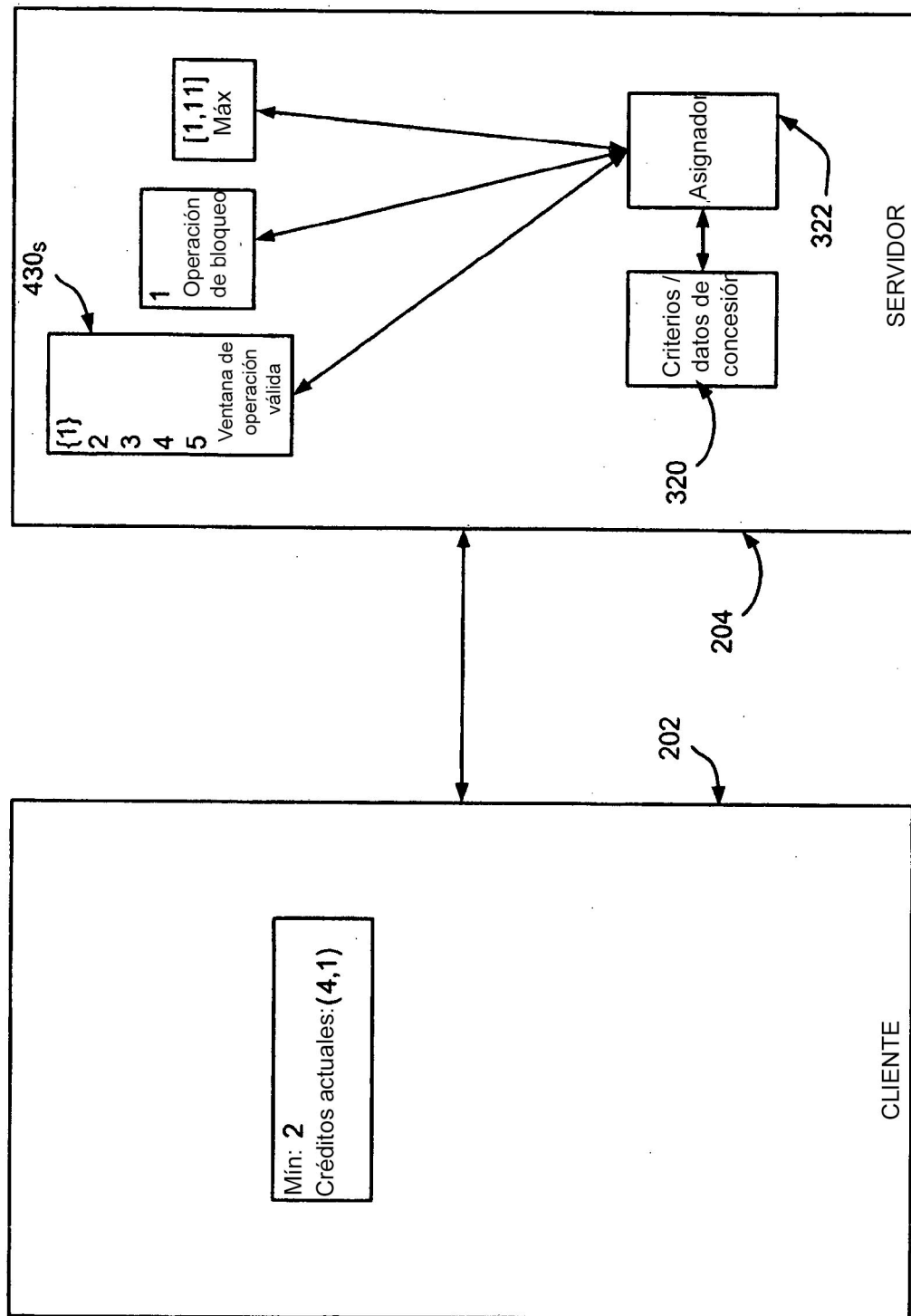


FIG. 4

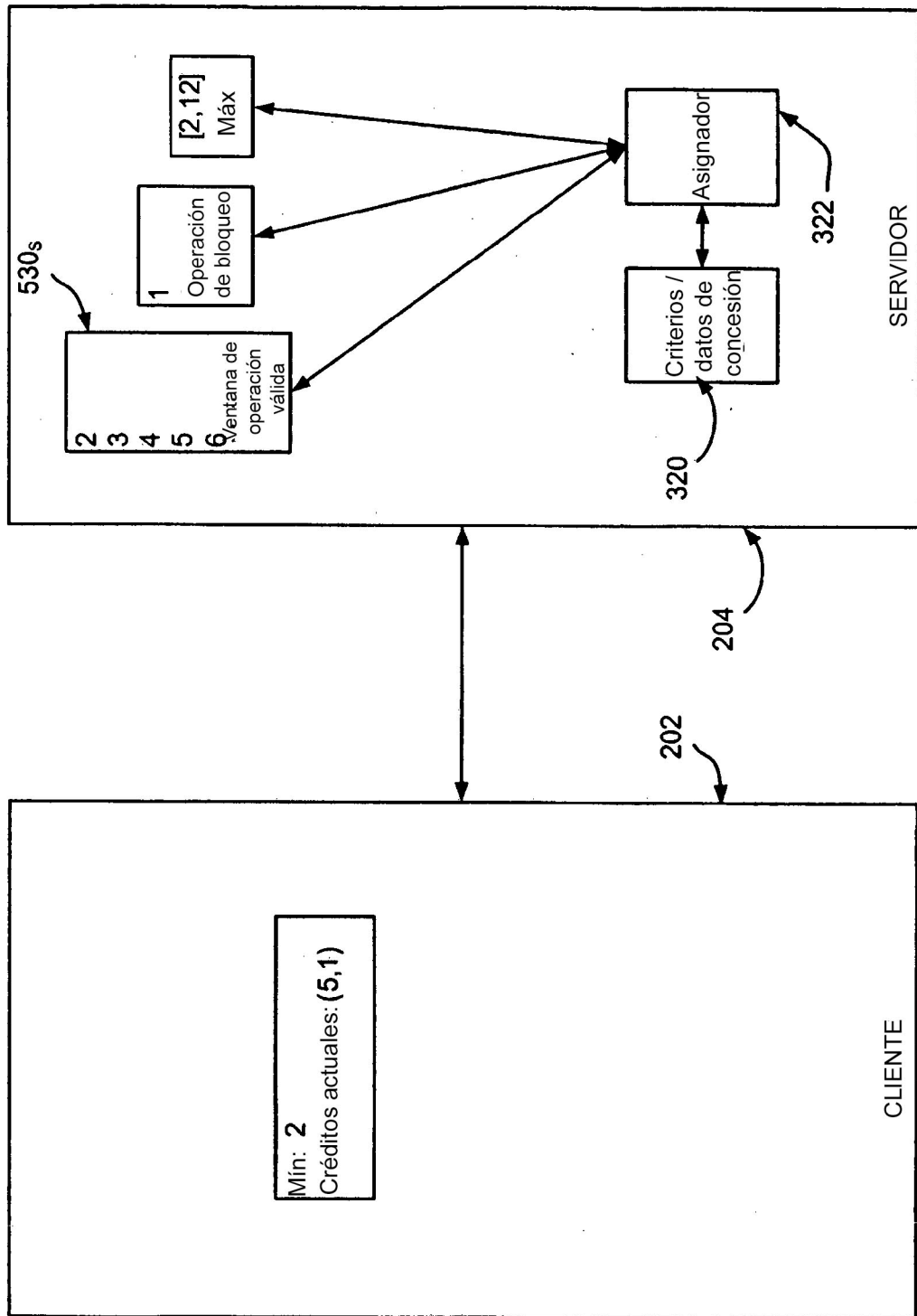


FIG. 5

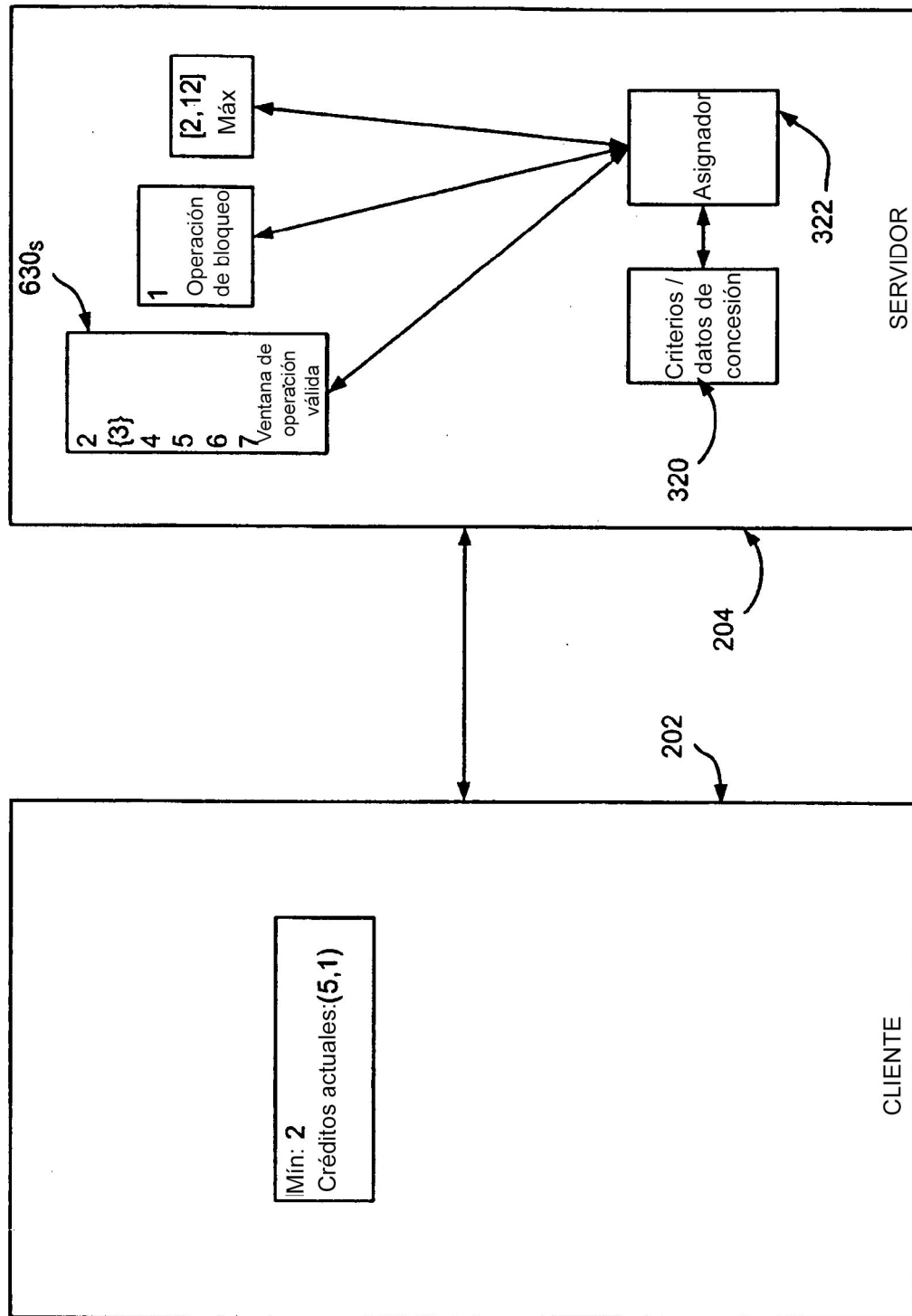


FIG. 6

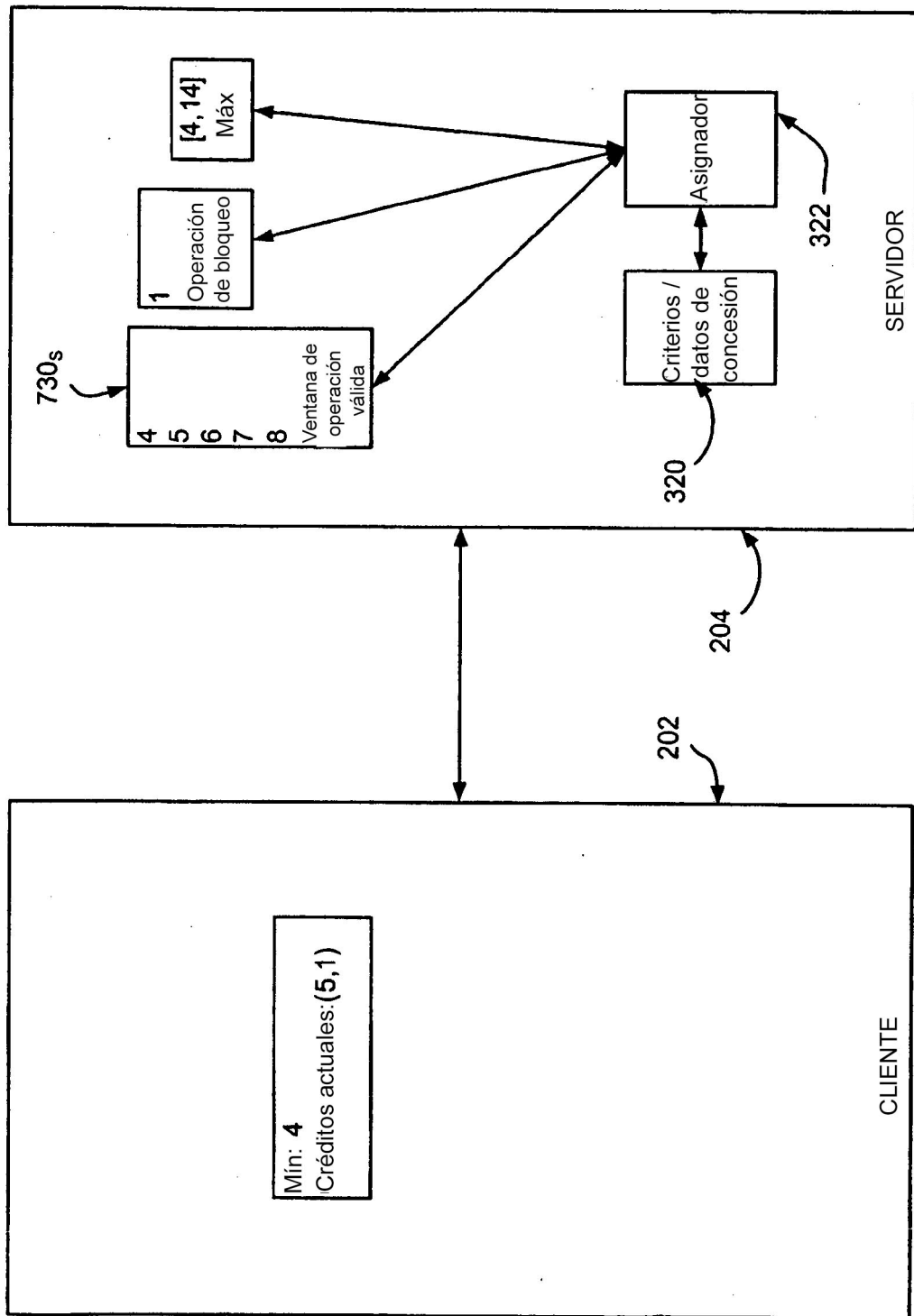


FIG. 7

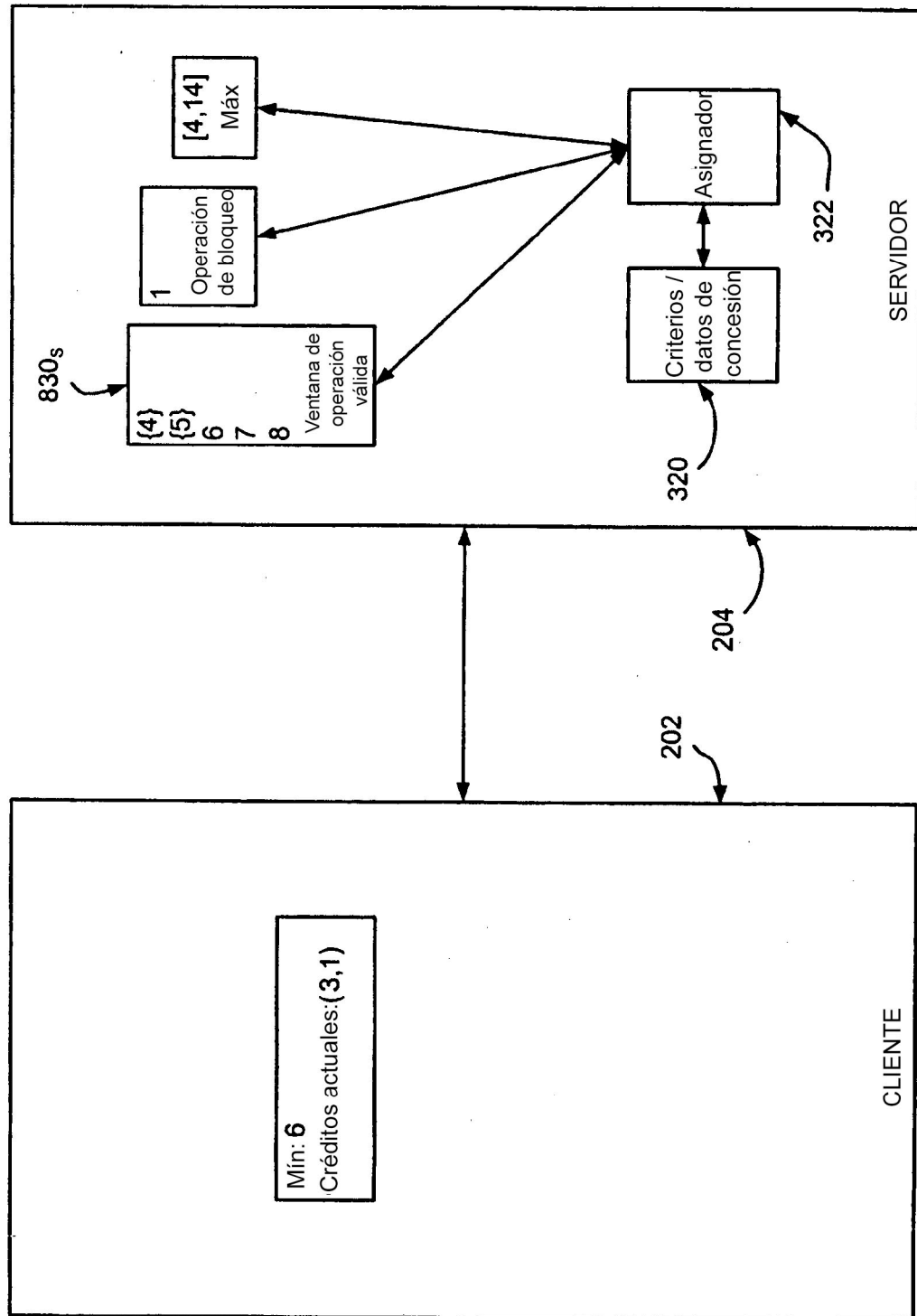


FIG. 8

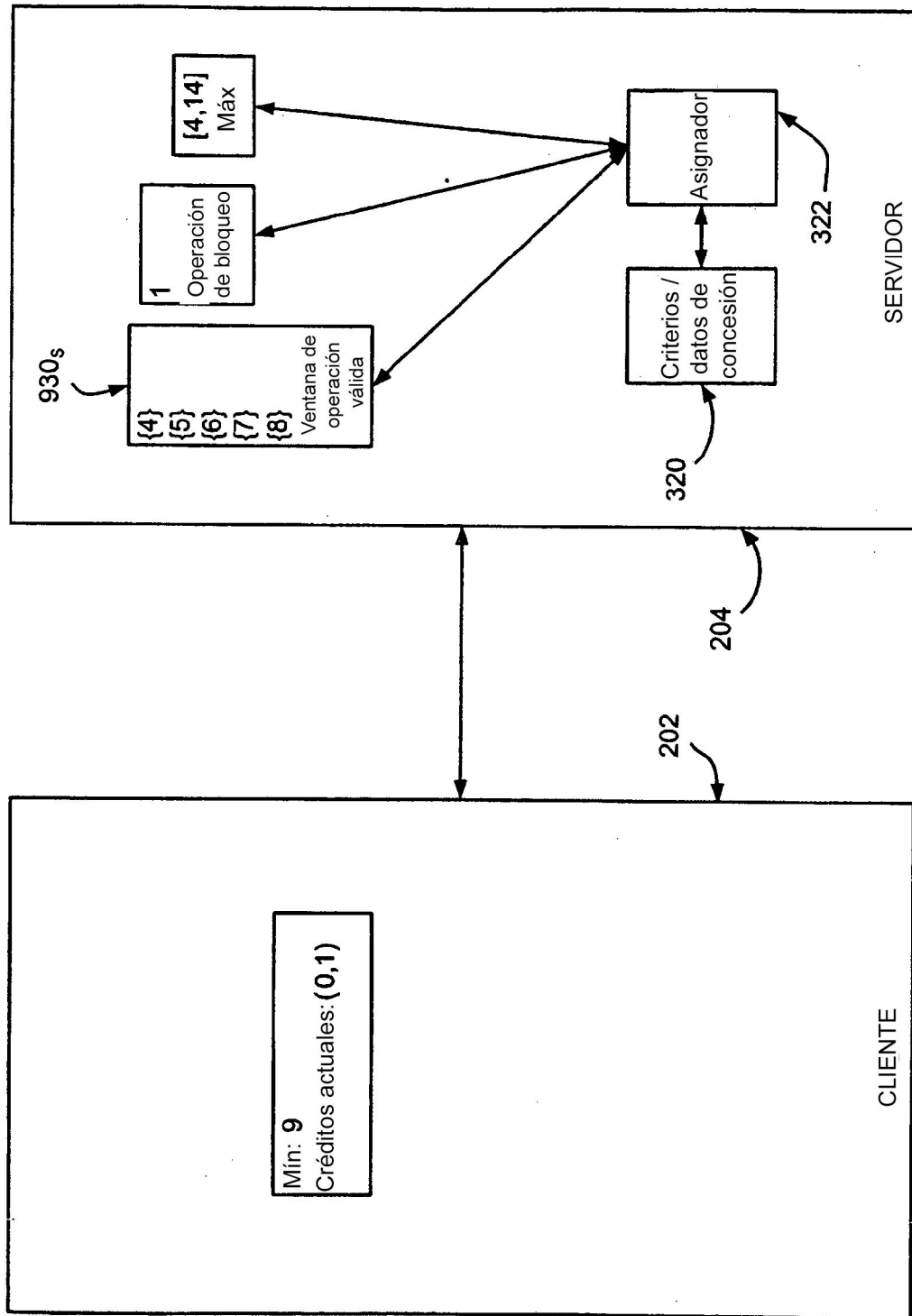


FIG. 9

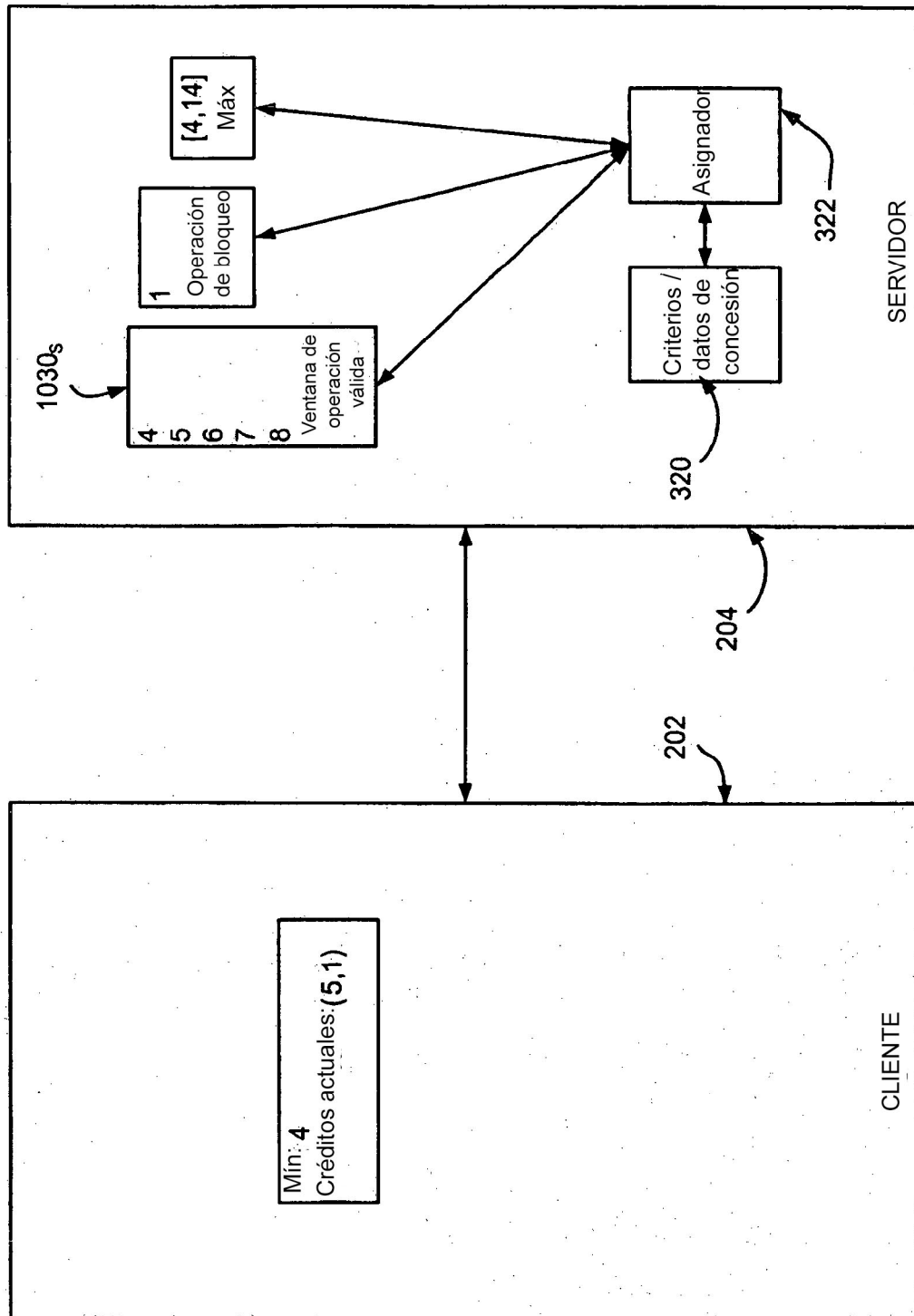


FIG. 10

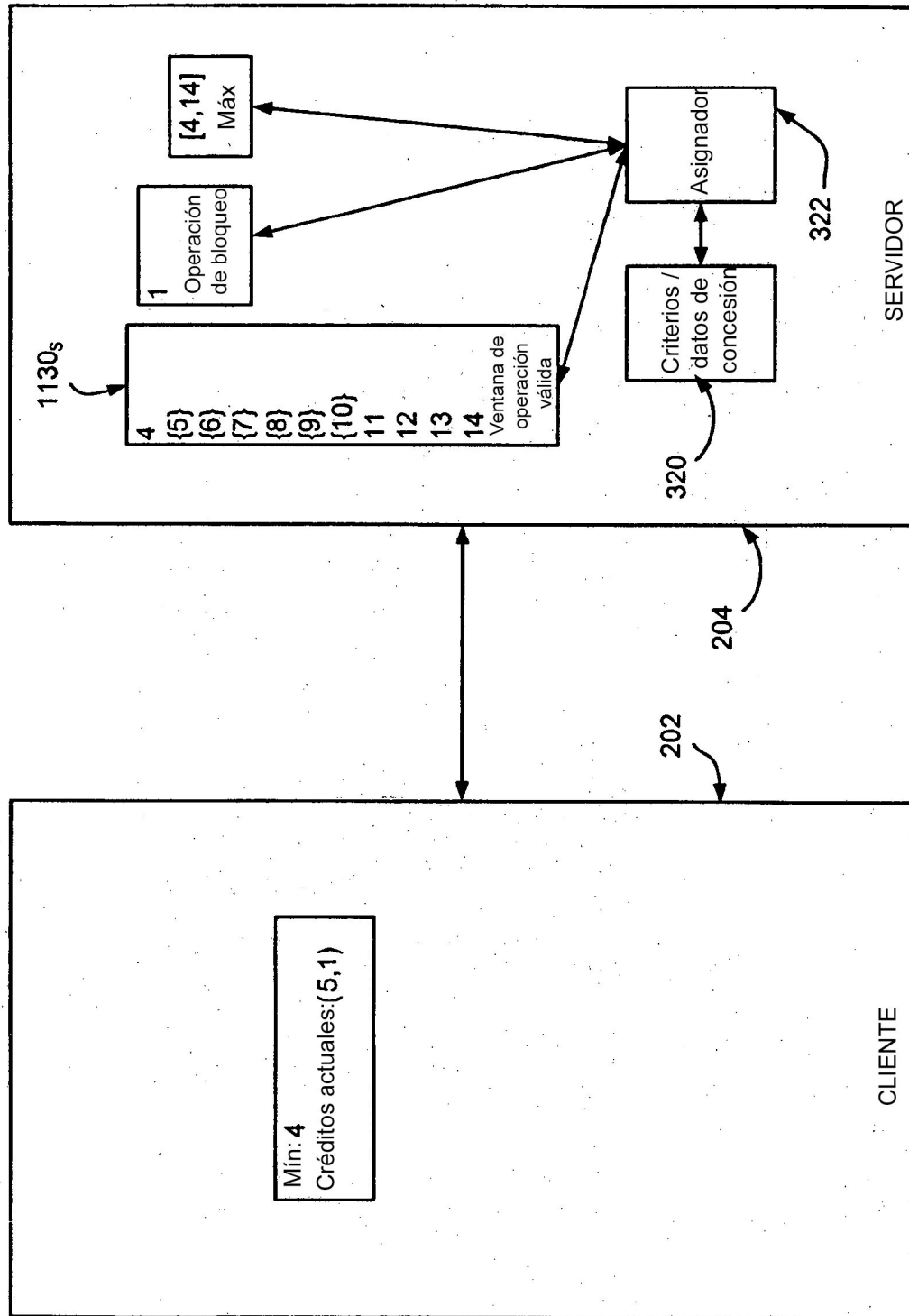


FIG. 11

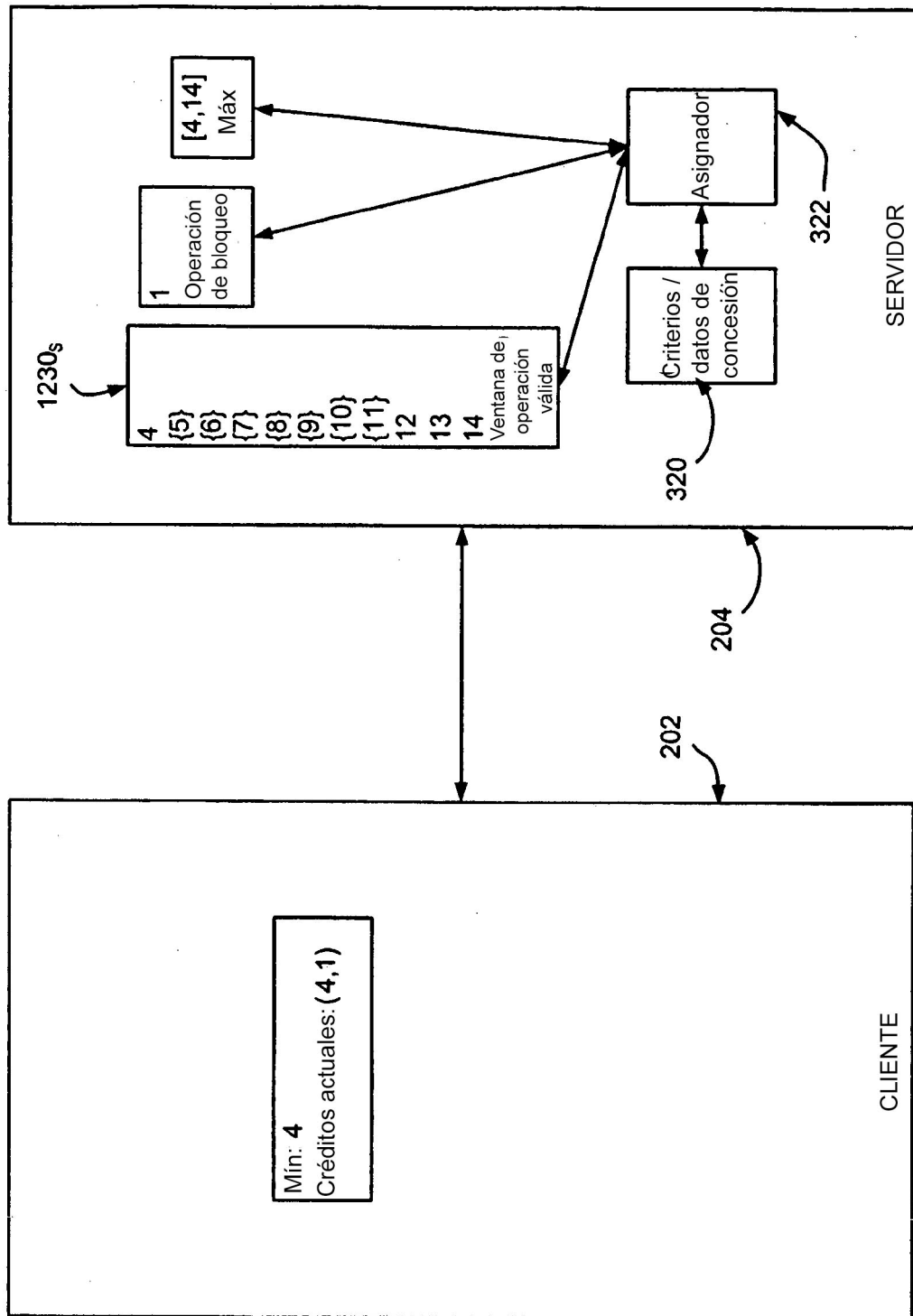


FIG. 12

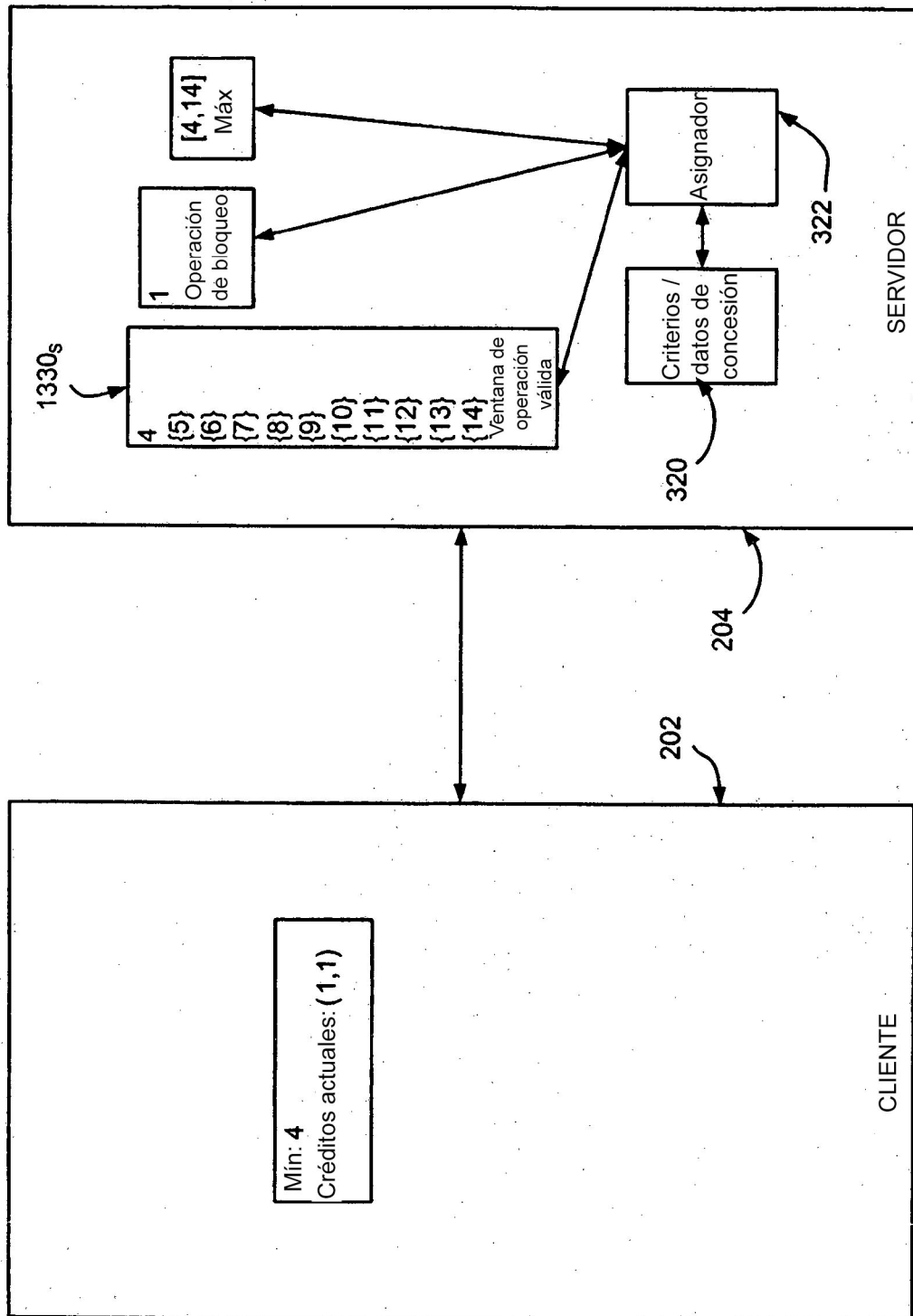


FIG. 13