(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 654 400**

(51) Int. CI.:

H04L 12/707 (2013.01)
H04L 12/803 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **16.04.2014 PCT/US2014/034423**

(87) Fecha y número de publicación internacional: **23.10.2014 WO14172497**

(96) Fecha de presentación y número de la solicitud europea: **16.04.2014 E 14785781 (7)**

(97) Fecha y número de publicación de la concesión europea: **20.12.2017 EP 2987305**

(54) Título: **Enrutamiento multitrayecto en un equilibrador de carga distribuido**

(30) Prioridad:

16.04.2013 US 201313864162

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
13.02.2018

(73) Titular/es:

AMAZON TECHNOLOGIES, INC. (100.0%)
P.O. Box 8102
Reno, NV 89507, US

(72) Inventor/es:

SORENSON, III, JAMES CHRISTOPHER y
LAURENCE, DOUGLAS STEWART

(74) Agente/Representante:

PONS ARIÑO, Ángel

DESCRIPCIÓN

Enrutamiento multitrayecto en un equilibrador de carga distribuido

ANTECEDENTES

Los equilibradores de carga convencionales son típicamente cajas únicas dedicadas que incluyen múltiples controladores de interfaz de red (NIC), por ejemplo, ocho NIC, con algunos de los NIC que gestionan tráfico entrante de/tráfico saliente a clientes y los otros NIC que manejan el tráfico saliente desde/tráfico entrante a los dispositivos anfitriones (por ejemplo, servidores tales como servidores web) que están siendo equilibrados de carga. El ancho de banda o el rendimiento en estos equilibradores de carga convencionales está típicamente en el intervalo de 40 Gigabits por segundo (Gbps) en el lado del cliente y 40 Gbps en el lado del servidor. A medida que la escala y el alcance de las aplicaciones basadas en red y los servicios basados en red tales como los servicios de computación en la nube han aumentado, los centros de datos pueden alojar cientos o incluso miles de dispositivos anfitriones (por ejemplo, servidores web) que necesitan equilibrio de carga. Los equilibradores de carga convencionales pueden no escalar bien en tales ambientes.

Además, los equilibradores de carga convencionales usan típicamente técnicas tales como las conexiones máximas (o con. máx.), round robin y/o las conexiones mínimas (con. mín.) aplicadas a los datos recogidos de los dispositivos anfitriones para seleccionar qué dispositivo anfitrión manejará una conexión. Además, los equilibradores de carga convencionales sirven típicamente como proxies para los dispositivos anfitriones que enfrentan y, por lo tanto, terminan las conexiones (por ejemplo, las conexiones del Protocolo de Control de Transmisión (TCP)) de los clientes y envían el tráfico de cliente a los dispositivos anfitriones en las conexiones TCP establecidas entre los dispositivos anfitriones y el equilibrador de carga. Por lo tanto, un dispositivo anfitrión y un cliente no se comunican a través de una conexión TCP directa al usarse estos equilibradores de carga convencionales. El documento US 2012/155266 describe métodos, sistemas y productos de programas informáticos para sincronizar el estado entre componentes de equilibrio de carga.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La figura 1 es un diagrama de bloques de un sistema de equilibrado de carga distribuido ejemplar, de acuerdo con al menos algunas realizaciones.

La figura 2 es un diagrama de flujo de alto nivel de un método de equilibrado de carga que puede implementarse por el sistema de equilibrador de carga distribuida de la figura 1, de acuerdo con al menos algunas realizaciones.

La figura 3 muestra un ejemplo de nodo de de equilibrador de carga que incluye componentes de entrada, salida y seguimiento de flujo, de acuerdo con al menos algunas realizaciones.

La figura 4 ilustra el enrutamiento y el flujo de paquetes en el equilibrador de carga distribuida, de acuerdo con al menos algunas realizaciones.

La figura 5 ilustra nodos de entrada de publicidad al enrutador periférico, de acuerdo con al menos algunas realizaciones.

La figura 6 es un diagrama de flujo de un método de enrutamiento de trayectos múltiples, de acuerdo con al menos algunas realizaciones.

La figura 7 ilustra gráficamente el flujo de paquetes asimétricos, de acuerdo con al menos algunas realizaciones.

La figura 8 ilustra el flujo de paquetes en el sistema distribuido de equilibrado de carga, de acuerdo con al menos algunas realizaciones.

Las figuras 9A y 9B proporcionan un diagrama de flujo de flujo de paquetes al establecerse conexiones en el sistema de equilibrado de carga distribuido, de acuerdo con al menos algunas realizaciones.

Las figuras 10A a 10G ilustran el flujo de paquetes en el sistema distribuido de equilibrado de carga, de acuerdo con al menos algunas realizaciones.

Las figuras 11A a 11D ilustran el manejo de eventos que realizan la pertenencia en el anillo de hash consistente con

el nodo de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

La figura 12 es un diagrama de flujo de alto nivel de un método de comprobación de salud que puede realizarse por cada nodo de equilibrador de carga de acuerdo con un intervalo de comprobación del estado, de acuerdo con al menos algunas realizaciones.

La figura 13 ilustra un método para comprobar el estado de un nodo de equilibrador de carga desde otro nodo de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

La figura 14 ilustra gráficamente una comprobación del estado del nodo de equilibrador de carga de uno o más nodos de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

La figura 15 ilustra los nodos de equilibradores de carga que comprueban el estado de los nodos servidor, de acuerdo con al menos algunas realizaciones.

La figura 16 ilustra gráficamente una vista del estado de otro nodo que puede mantenerse por un nodo de equilibrador de carga 110, de acuerdo con al menos algunas realizaciones.

La figura 17 ilustra información de estado que puede mantenerse por cada nodo de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

Las figuras 18A y 18B ilustran el manejo de un fallo del nodo del equilibrador de carga, de acuerdo con al menos algunas realizaciones.

Las figuras 19A y 19B ilustran gráficamente una técnica de publicación de conexión, de acuerdo con al menos algunas realizaciones.

La figura 20 es un diagrama de flujo de alto nivel de un método de publicación de conexión que puede realizarse por cada módulo de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

La figura 21 es un diagrama de flujo de un método para distribuir la información de conexión activa recibida en un paquete de publicación de conexión para dirigir nodos de equilibradores de carga, de acuerdo con al menos algunas realizaciones.

La figura 22 ilustra un método alternativo para distribuir la información de conexión activa recibida en un paquete de publicación de conexión a nodos de equilibrador de carga de destino, de acuerdo con al menos algunas realizaciones.

La figura 23 ilustra un ejemplo de arquitectura de bastidor de software para un nodo de equilibrador de carga de acuerdo con al menos algunas realizaciones.

La figura 24 ilustra aspectos de la tecnología de procesamiento de paquetes centrales que se pueden usar en realizaciones.

La figura 25 ilustra un procesador de paquetes de múltiples núcleos ejemplares para procesar flujos de datos en los nodos de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

La figura 26 ilustra otro procesador de paquetes de múltiples núcleos ejemplares para procesar flujos de datos en los nodos de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

La figura 27 ilustra el procesamiento de paquetes entrantes por un proceso de nodo de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

La figura 28 ilustra el procesamiento de paquetes salientes por un proceso de nodo de equilibrador de carga, de acuerdo con al menos algunas realizaciones.

La figura 29 ilustra un sistema de equilibrado de carga que incluye un equilibrador de carga distribuido en un entorno de producción, de acuerdo con al menos algunas realizaciones.

La figura 30 ilustra un sistema de prueba de equilibrador de carga distribuido que incorpora un mecanismo de bus de

mensajes que permite configurar y ejecutar múltiples componentes del sistema distribuido de equilibrado de carga en o como un único proceso, de acuerdo con al menos algunas realizaciones.

Las figuras 31 y 32 ilustran adaptadores de paquetes de bus de mensajes y conductos de paquetes, de acuerdo con al menos algunas realizaciones.

La figura 33A ilustra un entorno de red de proveedor ejemplar, de acuerdo con al menos algunas realizaciones.

La figura 33B ilustra una implementación de equilibrador de carga distribuida en un entorno de red proveedor ejemplar, como se muestra en la figura 33A, de acuerdo con al menos algunas realizaciones.

La figura 34A ilustra una implementación de bastidor físico ejemplar del equilibrador de carga distribuido y nodos servidor de acuerdo con al menos algunas realizaciones.

La figura 34B ilustra otra implementación de bastidor físico ejemplar del equilibrador de carga distribuido y nodos servidor de acuerdo con al menos algunas realizaciones.

La figura 35 ilustra un entorno de red ejemplar en el que uno, dos o más equilibradores de carga distribuidos se implementan en una red, de acuerdo con al menos algunas realizaciones.

La figura 36 es un diagrama de bloques que ilustra un sistema de ordenador ejemplar que se puede utilizar en algunas realizaciones.

Aunque se describen en el presente documento realizaciones a modo de ejemplo para varias realizaciones y dibujos ilustrativos, los expertos en la técnica reconocerán que las realizaciones no están limitadas a las realizaciones o dibujos descritos. Debe entenderse que los dibujos y la descripción detallada de los mismos no pretenden limitar las realizaciones a la forma particular descrita, sino que, por el contrario, la intención es cubrir todas las modificaciones, equivalentes y alternativas que caen dentro del alcance como se define por las reivindicaciones adjuntas. Los encabezados usados en el presente documento son sólo para fines de organización y no se pretende que se usen para limitar el alcance de la descripción o de las reivindicaciones. Como se usa a lo largo de esta solicitud, la palabra "puede" se usa en un sentido permisivo (es decir, que significa tener el potencial de), en lugar del sentido obligatorio (es decir, que significa debe). De manera similar, las palabras "incluir", "que incluye" e "incluye" significan incluyendo, pero sin limitación.

DESCRIPCIÓN DETALLADA

Se describen diversas realizaciones de métodos y sistemas para equilibrar la carga distribuida en entornos de red. Se describen realizaciones de un método y sistema de equilibrado de carga distribuida que pueden implementarse de acuerdo con realizaciones de un equilibrador de carga distribuido en diversos entornos de red. Por ejemplo, pueden utilizarse realizaciones del equilibrador de carga distribuido para facilitar y mantener flujos de paquetes, por ejemplo, flujos de paquetes de tecnología de Protocolo de Control de Transmisión (TCP), entre clientes en una red externa tal como Internet y destinos, típicamente servidores (por ejemplo, servidores web, servidores de aplicaciones, servidores de datos, etc.) en una red local, tal como una red de proveedor 1900 como se ilustra en las figuras 33A y 33B. Aunque las realizaciones se describen principalmente en el presente documento en relación con el procesamiento de flujos de paquetes TCP, ha de apreciarse que las realizaciones pueden aplicarse a protocolos de comunicaciones de datos diferentes de TCP y a aplicaciones diferentes que el procesamiento de flujos de paquetes.

El equilibrador de carga distribuido puede actuar para facilitar y mantener los flujos de paquetes TCP entre clientes particulares y servidores seleccionados (por ejemplo, servidores web). Sin embargo, el equilibrador de carga distribuido no termina los flujos TCP de los clientes y no actúa como un proxy para los servidores como se hace en equilibradores de carga convencionales. En cambio, los nodos del equilibrador de carga de los paquetes TCP de la ruta del equilibrador de carga distribuido recibidos de los clientes a los servidores de destino, y los servidores utilizan sus bastidores TCP para administrar las conexiones TCP a los clientes. En otras palabras, los servidores terminan los flujos de paquetes TCP de los clientes.

Además, en lugar del nodo o nodos del equilibrador de carga que toma decisiones en cuanto a qué servidor dará servicio a una petición de conexión basada en una técnica o algoritmo de equilibrado de carga que se aplica a la información recogida de los servidores como se hace en la tecnología de equilibrador de carga convencional, los nodos del equilibrador de carga pueden seleccionar aleatoriamente un servidor para recibir una nueva solicitud de

conexión, y un componente del equilibrador de carga distribuido que reside en el nodo servidor toma la decisión localmente de si el servidor seleccionado aceptará o rechazará la nueva solicitud de conexión basada en una o más métricas del estado actual del servidor respectivo. Por lo tanto, las decisiones sobre qué servidores deben aceptar solicitudes de conexión se trasladan desde el uno o más nodos de equilibrio de carga a los nodos servidor que manejarán las conexiones. En otras palabras, la decisión se mueve más cerca de dónde y cuándo se atenderá la solicitud de conexión.

Para facilitar y mantener los flujos de paquetes entre los clientes y los servidores, las realizaciones del equilibrador de carga distribuido pueden emplear diversas técnicas o tecnologías que incluyen, pero sin limitación, tecnología de enrutamiento multitrayecto, tecnología de hashing consistente, tabla de hash distribuida (DHT), tecnología de protocolo de puerta de enlace de frontera (BGP), seguimiento de pertenencia, comprobación de estado, publicación de conexiones, y encapsulación y decapsulación de paquetes. Estos, así como otros aspectos del sistema de equilibrado de carga distribuido, se describen a continuación en relación con las figuras.

Sistema de equilibrio de carga distribuido

La figura 1 es un diagrama de bloques de un sistema de equilibrado de carga distribuido ejemplar, de acuerdo con al menos algunas realizaciones. Pueden implementarse realizaciones del equilibrador de carga distribuido en una red 100, por ejemplo, una red de proveedor 1900 de un proveedor de servicios como se ilustra en las figuras 33A y 33B. Como una visión general de alto nivel de la manipulación de paquetes de cliente en el sistema de equilibrador de carga distribuido, uno o más clientes 160 de la red 100 pueden conectarse a un enrutador de frontera 102 de la red 100, por ejemplo a través de una red externa 150 tal como Internet. El enrutador de frontera 102 puede enrutar los paquetes entrantes (por ejemplo, paquetes TCP) desde los clientes 160 hasta un componente del enrutador periférico 104 del equilibrador de carga distribuido que encamina los paquetes entrantes a los nodos del equilibrador de carga (LB) 110 en una capa de nodo de equilibrador de carga del distribuido equilibrador de carga. En al menos algunas realizaciones, el enrutador periférico 104 puede tomar las decisiones de enrutamiento de acuerdo con una técnica de enrutamiento multitrayecto de hash por flujo, por ejemplo una técnica de hashing multitrayecto de igual coste (ECMP). Los nodos de equilibrador de carga 110 encapsulan a su vez los paquetes (por ejemplo, de acuerdo con el protocolo de datagramas de usuario (UDP)) y encaminan los paquetes encapsulados a los módulos de equilibrador de carga local 132 en los nodos servidor 130 a través de un tejido de red 120 (por ejemplo, una red L3) en la red 100. El tejido 120 puede incluir uno o más dispositivos o componentes de red que incluyen, pero sin limitación, interruptores, enrutadores y cables. En los nodos servidor 130, los módulos de equilibrio de carga local 132 decapsulan los paquetes y envían los paquetes TCP de cliente a los bastidores TCP de los servidores 134. Los servidores 134 de los nodos servidor 130 utilizan entonces sus bastidores TCP para gestionar las conexiones a los clientes 160.

La figura 2 es un diagrama de flujo de alto nivel de un método de equilibrado de carga que puede implementarse por el sistema de equilibrador de carga distribuida de la figura 1, de acuerdo con al menos algunas realizaciones. Las realizaciones del sistema de equilibrador de carga distribuido pueden no resolver el difícil problema de asignar carga entre múltiples destinos (por ejemplo, servidores web) como se hace en equilibradores de carga convencionales. Por ejemplo, los equilibradores de carga convencionales usan típicamente técnicas o algoritmos tales como conexiones máximas, round robin, y/o técnicas de conexiones mínimas para seleccionar qué servidor debe manejar una conexión. Sin embargo, estas técnicas tienen inconvenientes y, en particular, son difíciles de llevar a cabo satisfactoriamente en un sistema distribuido en el que los datos utilizados para tomar decisiones de equilibrio de carga son a menudo casi inmediatamente obsoletos. En al menos algunas realizaciones del sistema equilibrador de carga distribuido, en lugar de intentar seleccionar un nodo servidor 130 para satisfacer una petición de conexión utilizando una o más de las técnicas de equilibrado de carga como se hace en equilibradores de carga convencionales, un nodo de equilibrador de carga 110 en la capa de nodo de equilibrador de carga puede determinar aleatoriamente un nodo servidor 130 para recibir una solicitud de conexión de cliente. Si dicho nodo servidor 130 se considera sobrecargado, el nodo servidor 130 puede enviar de nuevo la petición de conexión al nodo de equilibrador de carga 110, informando así al nodo de equilibrador de carga 110 que el nodo servidor 130 no puede gestionar la conexión actualmente. La capa de nodo de equilibrador de carga puede entonces determinar aleatoriamente otro nodo servidor 130 para recibir la petición de conexión o, como alternativa, puede devolver un mensaje de error al cliente solicitante 160 para informar al cliente 160 que la conexión no se puede establecer actualmente.

Como se indica en 10 de la figura 2, la capa de nodo de equilibrador de carga del sistema de equilibrador de carga distribuido recibe una solicitud de una sesión de comunicación (por ejemplo, una conexión TCP) de una fuente. La fuente puede ser, por ejemplo, un cliente 160 en una red externa 150 a la red 100 que implementa el sistema de equilibrador de carga distribuido. En al menos algunas realizaciones, la petición puede ser recibida desde el cliente

160 en un enrutador de frontera 102 de la red 100, y enrutarse a un enrutador periférico 104 que encamina los paquetes entrantes a los nodos de equilibrador de carga (LB) 110 en una capa de nodo de equilibrador de carga, por ejemplo, utilizando una técnica de hashing multitrayecto de coste igual por flujo (ECMP) para seleccionar de forma pseudoaleatoria un nodo de equilibrador de carga 110 al que se ha de encaminar una petición de conexión particular de un cliente 160.

Como se indica en 20, la capa de nodo de equilibrador de carga selecciona aleatoriamente un nodo de destino y envía la solicitud de conexión al nodo de destino seleccionado. El nodo de destino puede, por ejemplo, ser uno de una pluralidad de nodos servidor 130 enfrentados por el equilibrador de carga. En al menos algunas realizaciones, un nodo de equilibrador de carga 110 en la capa de equilibrador de carga puede seleccionar aleatoriamente un nodo servidor 130 para recibir una solicitud de conexión entre todos los nodos servidor 130 conocidos. Sin embargo, otros métodos distintos de la selección puramente aleatoria entre todos los nodos servidor conocidos 130 pueden usarse en algunas realizaciones para seleccionar nodos servidor 130 para recibir las peticiones de conexión. Por ejemplo, en algunas realizaciones, la información sobre los nodos servidor 130 puede utilizarse por los nodos de equilibrador de carga 110 para ponderar la selección aleatoria de nodos servidor 130. A modo de ejemplo, si los nodos de equilibrador de carga 110 saben que diferentes nodos servidor 130 son diferentes tipos de dispositivos o se configuran con diferentes CPU y, por lo tanto, tienen diferentes capacidades, la información puede usarse para polarizar la selección aleatoria hacia (o alejarse) de uno o más tipos o configuraciones particulares del nodo servidor 130.

Como se indica en 30, el nodo de destino determina si puede aceptar la sesión de comunicaciones. En al menos algunas realizaciones, un módulo de equilibrador de carga local (LB) 132 en el nodo servidor 130 determina si el servidor respectivo 134 en el nodo servidor 130 puede aceptar la nueva conexión basada en una o más métricas del estado actual del servidor respectivo 134.

En 40, si se acepta la solicitud de conexión, entonces como se indica en 50, el nodo de destino informa a la capa de nodo de equilibrador de carga que el nodo de destino puede manejar la conexión. Como se indica en 60, se establece entonces una sesión de comunicaciones entre la fuente (por ejemplo, un cliente 160) y el nodo de destino (por ejemplo, un servidor 134 en un nodo servidor 130) a través de la capa de nodo de equilibrador de carga. En al menos algunas realizaciones, el servidor 134 en el nodo servidor 130 utiliza un bastidor TCP para gestionar la conexión con el cliente 160.

En 40, si la solicitud de conexión no es aceptada, entonces como se indica en 70, el nodo de destino notifica a la capa de nodo de equilibrador de carga, y el método puede volver al elemento 20. La capa de nodo de equilibrador de carga puede entonces seleccionar aleatoriamente otro nodo de destino en 20, o como alternativa, puede informar al cliente solicitante 160 que la conexión no se puede establecer actualmente. Obsérvese que el cliente 160 puede, pero no necesariamente, reenviar la petición de conexión para comenzar de nuevo el método en el elemento 10.

Haciendo referencia de nuevo a la figura 1, al menos algunas realizaciones del sistema de equilibrador de carga distribuido pueden usar hardware de consumo para enrutar el tráfico de cliente recibido en un enrutador periférico 104 en la red 100 a nodos servidor 130 en la red 100. Al menos algunas realizaciones del equilibrador de carga distribuido puede incluir una capa de nodo de equilibrador de carga que incluye múltiples nodos de equilibrador de carga 110. En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede servir en una o más de múltiples funciones en la capa de nodo de equilibrador de carga. Estos roles de los nodos de equilibrador de carga 110 pueden incluir las funciones de un nodo de entrada y un nodo de salida y un nodo de rastreador de flujo (como un rastreador de flujo primario o un rastreador de flujo secundario para un flujo de paquetes dado). En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede ser implementado en la capa de nodo de equilibrador de carga como o en un dispositivo informático separado, tal como un dispositivo informático montado en bastidor de consumo. En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede servir en cada una de las tres funciones de nodo de entrada, nodo de salida y nodo de rastreador de flujo (como un rastreador de flujo primario o secundario para un flujo de paquetes), sirviendo el nodo de equilibrador de carga 110 generalmente en solo uno (pero posiblemente en dos o tres) de las funciones para determinados flujos de paquetes. Ha de apreciarse, sin embargo, que en al menos algunas realizaciones, no se permite que un nodo de equilibrador de carga 110 sirva tanto como el rastreador de flujo primario como el seguidor de flujo secundario para un flujo de paquetes particular. Como alternativa, en algunas realizaciones, cada nodo de equilibrador de carga 110 puede servir en sólo una de las tres funciones. En esta realización, se pueden implementar conjuntos separados de dispositivos de cálculo en la capa de nodos de equilibrador de carga específicamente como nodos de entrada, nodos de salida y nodos de rastreador de flujo.

En al menos algunas realizaciones, se puede aplicar un hashing consistente y una tecnología de anillo hash

consistente para determinar los rastreadores de flujo primarios y secundarios para los flujos de paquetes. Cada flujo de paquetes de un cliente puede identificarse de manera única, por ejemplo, por una tupla de 4 que consiste en: la dirección IP cliente, el puerto cliente, la dirección IP del servidor (público), y el puerto servidor. Este identificador puede abreviarse como CP o CcPp que indica el par de puntos finales de cliente y público. Pueden aparecer paquetes asociados con cualquier flujo TCP dado (o par CP) en cualquier nodo de equilibrador de carga 110 que funcione como un servidor de entrada 112 debido a la distribución de flujo multitrayecto en hash (por ejemplo, ECMP) desde el enrutador periférico 104. Se usa el hashing consistente de manera que, cuando un paquete llega a un nodo de equilibrador de carga 110 que sirve como nodo de entrada, el nodo de entrada puede determinar qué nodo de equilibrador de carga 110 es responsable de mantener el estado del flujo de paquetes (es decir, el nodo de rastreador de flujo primario). El par CP puede estar en hash por el nodo de entrada en un anillo hash consistente para determinar qué nodo de equilibrador de carga 110 es responsable de mantener la información de estado para el flujo de paquetes. El nodo 110 determinado de acuerdo con el hash consistente del par CP para el flujo de paquetes en el anillo hash consistente es el nodo 110 que sirve como el rastreador de flujo primario para el flujo de paquetes. En al menos algunas realizaciones, el nodo sucesor en el anillo hash consistente sirve como el seguidor de flujo secundario para el flujo de paquetes.

La figura 3 muestra un nodo de equilibrador de carga (LB) ejemplar 110 que incluye componentes que implementan las tres funciones (entrada, salida y rastreador de flujo), de acuerdo con al menos algunas realizaciones. En este ejemplo, un componente del servidor de entrada 112 realiza la función de entrada de recibir paquetes TCP entrantes desde el cliente o los clientes y enviar los paquetes TCP como paquetes encapsulados al uno o más servidores. Un componente de servidor de salida 114 desempeña el papel de salida de recibir paquetes encapsulados salientes desde el servidor o servidores y enviar los paquetes TCP decapsulados al cliente o clientes. Un componente de rastreador de flujo 116 actúa como un rastreador de flujo primario o secundario para uno o más flujos de paquetes que se establecen entre el uno o más clientes 160 y el uno o más servidores 134. El servidor de entrada 112 también puede comunicarse con el rastreador de flujo 116 en el nodo de equilibrador de carga 110, o con el seguidor de flujo 116 en otro nodo de equilibrador de carga 110 para iniciar una conexión TCP entre un cliente y uno de los servidores 134 en respuesta a una solicitud de conexión recibida del cliente respectivo 160, o para obtener información de mapeo para el flujo de paquetes.

Nodos del equilibrador de carga

Haciendo referencia de nuevo a la figura 1, en al menos algunas realizaciones, los nodos de equilibradores de carga 110 en la capa de nodo de equilibrador de carga reciben tráfico de cliente (paquetes, por ejemplo paquetes TCP) de uno o más enrutadores 104 en la red y encapsulan los paquetes según a un protocolo (por ejemplo, el protocolo de datagrama de usuario (UDP)) utilizado por el sistema de equilibrador de carga distribuido en el tejido 120. La capa de nodo de equilibrador de carga envía entonces los paquetes encapsulados a los nodos servidor de destino 130 en el tejido 120. Cada nodo servidor 130 incluye un módulo local 132 que es un componente del sistema de equilibrador de carga. El módulo 132 puede denominarse en el presente documento un módulo de equilibrador de carga o simplemente un módulo LB, y puede implementarse en software, hardware o una combinación de los mismos en el nodo servidor 130. En cada nodo servidor 130, el módulo de equilibrador de carga respectivo 132 decapsula los paquetes y envía los paquetes TCP a un bastidor TCP local para el procesamiento TCP normal. En al menos algunas realizaciones, la capa de nodo de equilibrador de carga puede mantener información de estado para cada flujo TCP de cliente-servidor; sin embargo, los nodos de equilibradores de carga 110 en la capa de nodo del equilibrador de carga no pueden interpretar nada sobre el flujo TCP. Cada flujo se gestiona entre el servidor 134 en el nodo servidor 130 respectivo y el cliente 160. El sistema de equilibrador de carga distribuido asegura que los paquetes TCP lleguen al servidor de destino correcto 134. El módulo de equilibrador de carga 132 en cada nodo servidor 130 toma la decisión en cuanto a si el respectivo servidor 134 aceptará o rechazará una nueva conexión en respuesta a una petición de conexión de cliente recibida desde un nodo de equilibrador de carga 110.

En al menos algunas realizaciones, el sistema de equilibrado de carga distribuido puede utilizar tecnología de hashing consistente para, por ejemplo, determinar qué nodo o nodos de equilibrador de carga 110 deben recordar qué nodo servidor 130 es responsable de un flujo de paquetes TCP particular. Usando una tecnología de hashing consistente, los nodos de equilibrador de carga 110 en la capa de nodo de equilibrador de carga pueden verse como un anillo hash consistente y los nodos de equilibradores de carga 110 pueden seguir la pista de la pertenencia al anillo y determinar miembros particulares en el anillo que son responsables de flujos de paquetes particulares de acuerdo con una función de hashing consistente. En al menos algunas realizaciones, hay dos nodos de equilibradores de carga 110 que son responsables de rastrear cada flujo de paquetes entre los clientes 160 y los servidores 134; estos nodos 110 pueden denominarse ser como el nodo de rastreador de flujo primario (PFT) y el nodo de rastreador de flujo secundario (SFT). En al menos algunas realizaciones, el rastreador de flujo primario es el primer nodo de equilibrador de carga 110 en el anillo hash consistente para el flujo, y el rastreador de flujo

secundario es el nodo de equilibrador de carga siguiente o subsiguiente 110 en el anillo hash consistente distinto del nodo de rastreador de flujo primario. En esta disposición, si el nodo de rastreador de flujo primario falla, entonces el nodo de rastreador de flujo secundario puede convertirse en el nuevo rastreador de flujo primario, y otro nodo de equilibrador de carga 110 (por ejemplo, el siguiente nodo 110 en el anillo hash consistente) puede asumir la función del rastreador de flujo secundario. Ha de apreciarse que, en al menos algunas realizaciones, no se permite que un nodo de equilibrador de carga 110 sirva tanto como el rastreador de flujo primario como el seguidor de flujo secundario para un flujo de paquetes dado. Este y otros cambios de pertenencia en el anillo hash consistente se analizan más adelante en este documento. En al menos algunas realizaciones, la información de configuración para la implementación del equilibrador de carga (por ejemplo, lista o listas autorizadas de los nodos de equilibrador de carga 110 y nodos servidor 130 que están actualmente en la implementación) puede mantenerse por un componente de servicio de configuración 122 del sistema de equilibrado de carga distribuido, que puede implementarse, por ejemplo, en uno o más dispositivos de servidor acoplados a los nodos de equilibradores de carga 110 a través del tejido 120.

En al menos algunas realizaciones, además de servir como nodos de rastreador de flujo primario y secundario, los nodos de equilibradores de carga 110 pueden actuar también en uno de los otros dos roles para un flujo dado: el papel de un nodo de entrada y el papel de un nodo de salida. Un nodo de entrada para un flujo de paquetes es el nodo de equilibrador de carga 110 que recibe el flujo de paquetes respectivo desde el enrutador periférico 104 y envía el flujo de paquetes (como paquetes encapsulados) a un servidor seleccionado 134 en un nodo servidor 130 a través del tejido 120. Un nodo de entrada es el único nodo de equilibrador de carga 110 que mueve datos de cliente reales (paquetes de datos TCP) al nodo servidor de destino respectivo 130. El nodo de entrada mantiene una asignación del flujo TCP a un módulo de equilibrador de carga respectivo 132 en el nodo servidor de destino 130 de manera que el nodo de entrada conozca el módulo de equilibrador de carga 132 para transmitir el tráfico del cliente. Un nodo de salida es un nodo de equilibrador de carga 110 que es responsable de reenviar el tráfico de respuesta para un flujo de paquetes recibido desde el nodo servidor 130 a través del tejido 120 al cliente respectivo 160 a través de la red de frontera. El módulo de equilibrador de carga 132 encapsula paquetes de respuesta obtenidos del servidor 134 de acuerdo con un protocolo de equilibrador de carga (por ejemplo, UDP) y envía los paquetes de respuesta encapsulados al respectivo nodo de salida para el flujo a través del tejido 120. Los nodos de salida no tienen estado y simplemente decapsulan los paquetes y envían los paquetes de respuesta (por ejemplo, paquetes TCP) a la red de frontera a un enrutador periférico 102 para su entrega al respectivo cliente 160 a través de la red externa 150.

Como se ha mencionado anteriormente, en al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede realizar las funciones de un nodo de entrada, un nodo de salida y/o un nodo de rastreador de flujo (como un rastreador de flujo primario o secundario) para diferentes flujos de paquetes. Un único nodo de equilibrador de carga 110 en la capa de nodo de equilibrador de carga puede actuar en cualquiera de las funciones dependiendo de qué flujo de paquetes está procesando el nodo. Por ejemplo, en al menos algunas realizaciones, un nodo de equilibrador de carga 110 puede actuar como un nodo de entrada para un flujo de paquetes, como un rastreador de flujo primario o secundario para otro flujo de paquetes, y como nodo de salida para otro flujo de paquetes. Además, al menos en algunas realizaciones, un nodo de equilibrador de carga 110 puede realizar múltiples funciones para el mismo flujo de paquetes, por ejemplo como nodo de entrada y como nodo de rastreador de flujo primario (o secundario) para un flujo de paquetes dado. Sin embargo, en al menos algunas realizaciones, con fines de redundancia y recuperación, no se permite que un nodo de equilibrador de carga 110 sirva como nodo de rastreador de flujo primario y secundario para el mismo flujo de paquetes.

Lo anterior describe realizaciones en las que cada nodo de equilibrador de carga 110 puede servir en cualquiera de las tres funciones de servidor de entrada, servidor de salida y rastreador de flujo. Sin embargo, en algunas realizaciones, pueden asignarse diferentes grupos de dispositivos informáticos a las diferentes funciones en el sistema de equilibrado de carga. Por ejemplo, en algunas realizaciones, pueden existir conjuntos distintos de nodos de entrada, nodos de salida y nodos de rastreador de flujo implementados cada uno en un dispositivo informático separado. Como otro ejemplo, en algunas realizaciones, un conjunto de dispositivos de cálculo puede servir como nodos de entrada y nodos de rastreador de flujo, mientras que otro conjunto de dispositivos informáticos puede servir solamente como nodos de salida.

Módulos de equilibrio de carga

Como se ha mencionado previamente, cada nodo servidor 130 incluye un módulo de equilibrador de carga local 132 que es un componente del sistema de equilibrador de carga. El módulo 132 puede implementarse en software, hardware o una combinación de los mismos en el nodo servidor 130. En al menos algunas realizaciones, el módulo de equilibrador de carga 132 en un nodo servidor 130 puede realizar tres funciones principales: encapsular paquetes

salientes y decapsular paquetes entrantes, tomando decisiones de equilibrio de carga local para el servidor 134 en el nodo 130, y publicación de conexión. Estos tres roles se describen brevemente a continuación, y se describen con más detalle más adelante en este documento.

Al menos algunas realizaciones del sistema de equilibrado de carga distribuido no terminan las conexiones TCP y no redireccionan paquetes; las direcciones IP de origen y de destino de todos los paquetes enviados a través de la capa de nodo de equilibrador de carga son las direcciones IP reales de los extremos (es decir, los clientes 160 y los servidores 134) implicados en los flujos de paquetes. En lugar del redireccionamiento, estas realizaciones encapsulan todos los paquetes enviados entre los nodos equilibradores de carga 110 y los nodos servidor 130 en el tejido 120, por ejemplo como paquetes UDP. Dado que los paquetes entrantes en un flujo de paquetes que llegan a un nodo servidor 130 desde un nodo de equilibrador de carga 110 que actúa como nodo de entrada para el flujo se encapsulan por el nodo de equilibrador de carga 110, los paquetes necesitan ser decapsulados y redireccionados a un flujo TCP de máquina local para el servidor 134 en el nodo 130. El módulo de equilibrador de carga 132 en el nodo 130 realiza esta decapsulación. De forma similar, los paquetes salientes para un flujo de paquetes desde el servidor 134 son encapsulados por el módulo 132 de equilibrador de carga y enviados a través del tejido 120 a un nodo de equilibrador de carga 110 que actúa como nodo de salida para el flujo de paquetes.

En al menos algunas realizaciones, los módulos de equilibrador de carga 132 en los nodos servidor 130 también toman decisiones locales relacionadas con el equilibrio de carga para los servidores 134 en los nodos servidor respectivos 130. Específicamente, el módulo de equilibrador de carga 132 en un nodo 130 decide si el respectivo servidor 134 aceptará otro flujo TCP en respuesta a la recepción de una solicitud para una nueva conexión TCP. Como se ha indicado previamente, los nodos de equilibradores de carga 110 encapsulan todos los paquetes enviados al módulo de equilibrador de carga 132, de modo que el módulo de equilibrador de carga 132 no recibe realmente un paquete de sincronización TCP (SYN) desde el cliente 160; en su lugar, el módulo de equilibrador de carga 132 recibe un mensaje de solicitud de conexión de acuerdo con el protocolo de encapsulación (por ejemplo, UDP) desde un rastreador de flujo 116 que el módulo de equilibrador de carga 132 puede aceptar o rechazar. Si el módulo de equilibrador de carga 132 acepta el mensaje de solicitud de conexión, el módulo de equilibrador de carga 132 crea un paquete SYN destinado para la máquina local. Cuando la máquina local acepta la conexión, esto se convierte en el bastidor TCP real que maneja la respectiva conexión de cliente.

En al menos algunas realizaciones, para tomar la decisión de si se debe aceptar un mensaje de solicitud de conexión, el módulo de equilibrador de carga 132 examina una o más métricas con respecto al consumo de recursos actual en el nodo servidor 130, y si hay suficientes recursos disponibles para manejar la nueva conexión, el módulo de equilibrador de carga 132 acepta la conexión. En al menos algunas realizaciones, las métricas de recursos que pueden ser consideradas por el módulo de equilibrador de carga 132 pueden incluir uno o más de, pero sin limitación, utilización de CPU, consumo reciente de ancho de banda y número de conexiones establecidas. Otras métricas pueden considerarse en lugar de o además de estas métricas en algunas realizaciones. Por ejemplo, en algunas realizaciones, el módulo de equilibrador de carga puede considerar la latencia del servidor (es decir, la cantidad de tiempo que las solicitudes están gastando en el atraso de conexión del servidor) como una métrica, y puede rechazar la solicitud de conexión si la latencia del servidor está por encima de un umbral. Usando estas y/u otras métricas, el módulo de equilibrador de carga 132 puede decidir para el servidor respectivo 134 si el servidor 134 acepta o rechaza nuevos flujos de paquetes. En al menos algunas realizaciones, se puede determinar una tasa de utilización de recursos (por ejemplo, utilización de *% de N*) a partir de la métrica o métricas individualmente o en combinación y en comparación con un umbral (por ejemplo, utilización del 90 %). Si la tasa de utilización de recursos determinada está en o por encima del umbral, o si la adición de la conexión moverá la tasa por encima del umbral, entonces la solicitud de conexión puede ser rechazada.

En al menos algunas realizaciones, los módulos de equilibrador de carga 132 pueden implementar un método probabilístico para determinar si se deben rechazar mensajes de petición de conexión. En lugar de rechazar todas las peticiones de conexión si la utilización de recursos está en o por encima de un umbral como se ha descrito anteriormente, en este método puede rechazar peticiones de conexión en diferentes probabilidades en dos o más niveles diferentes de utilización. Por ejemplo, si la utilización de recursos es del 80 %, un módulo de equilibrador de carga 132 puede rechazar solicitudes de conexión con una probabilidad del 20 %; si la utilización de recursos es del 90 %>, el módulo de equilibrador de carga 132 puede rechazar solicitudes de conexión con una probabilidad del 25 %; si la utilización de recursos es del 95 %, el módulo de equilibrador de carga 132 puede rechazar solicitudes de conexión con una probabilidad del 50 %; y al 98 % o superior, el módulo de equilibrador de carga 132 puede rechazar todas las peticiones de conexión.

En al menos algunas realizaciones, cada mensaje de petición de conexión puede incluir una indicación de cuántas veces el mensaje de petición de conexión ha sido rechazado por los módulos de equilibrador de carga 132. Si un

mensaje de solicitud de conexión recibido por un módulo de equilibrador de carga 130 indica que ha sido rechazado sobre un número de umbral de veces, el módulo de equilibrador de carga 130 puede aceptar la conexión aunque las métricas de rendimiento del nodo servidor 130 indican que la petición de conexión debe ser rechazada.

En algunos casos, es posible que todos los módulos de equilibrador de carga 132 a los que se envía un mensaje de petición de conexión puedan rechazar la solicitud de conexión. En al menos algunas realizaciones, para evitar que un mensaje de solicitud de conexión se rebote desde el módulo de equilibrador de carga 132 al módulo de equilibrador de carga 132 durante un período indefinido, se puede dar a cada mensaje de petición de conexión un tiempo de vida. Si este tiempo de vida expira, el nodo de rastreador de flujo puede terminar la solicitud y notificar al cliente respectivo 160 que la solicitud no puede ser atendida actualmente.

En al menos algunas realizaciones, los módulos de equilibrador de carga 132 en los nodos servidor 130 realizan también publicación de conexión a los nodos de equilibrador de carga 110. En al menos algunas realizaciones, para realizar publicación de conexión, periódica o aperiódicamente (por ejemplo, una vez por segundo), cada módulo de equilibrador de carga 132 examina la tabla de enrutamiento (por ejemplo, una tabla de enrutamiento netstat) en el nodo servidor 130 y publica una lista de conexiones activas (flujos TCP) de nuevo a los nodos de equilibradores de carga 110. Los nodos de equilibrador de carga 110 que necesitan para ser informados acerca de la existencia de un flujo de paquetes dado son los nodos de equilibradores de carga 110 que sirven como nodo de entrada y como los rastreadores de flujo primario y secundario para el flujo de paquetes respectivo. En algunas realizaciones, el módulo de equilibrador de carga 132 puede usar una técnica de hashing consistente para filtrar la lista de nodos de equilibradores de carga 110 que necesitan ser informados sobre los flujos TCP activos en el nodo servidor 130. Por ejemplo, el módulo de equilibrador de carga 132 puede determinar qué nodos de equilibradores de carga 110 sirven como rastreadores de flujo primario y secundario para un flujo de paquetes dado de acuerdo con el anillo hash consistente. En algunas realizaciones, el módulo de equilibrador de carga 132 rastrea qué nodo de equilibrador de carga 110 envió por última vez un paquete de datos al módulo de equilibrador de carga 132 para cada flujo de paquetes, y utiliza esta información para determinar qué nodos de equilibradores de carga 110 sirven como nodos de entrada para los flujos de paquetes, ya que sólo los nodos de entrada envían los datos del cliente al módulo de equilibrador de carga 132. En algunas realizaciones, el módulo de equilibrador de carga 132 formula entonces mensajes para cada uno de los nodos de equilibradores de carga 110 que ha determinado que necesitan ser informados sobre los flujos de paquetes y envía los mensajes a los nodos de equilibradores de carga 110 para informar a los nodos 110 de que el nodo servidor respectivo 130 sigue manteniendo la conexión o conexiones para el cliente o clientes 160. Esta publicación de conexión a los nodos de equilibradores de carga 110 por los módulos de equilibrador de carga 132 puede visualizarse como extendiendo una concesión en los nodos de equilibradores de carga 110. Si un nodo de equilibrador de carga 110 no ha recibido un mensaje de publicación de conexión que indica un flujo de paquetes particular dentro de un pe (por ejemplo, diez segundos), entonces el nodo de equilibrador de carga 110 es libre de olvidar el flujo de paquetes respectivo.

Enrutamiento multitrayecto a nodos de equilibrador de carga

La figura 4 ilustra aspectos de enrutamiento y el flujo de paquetes en el equilibrador de carga distribuida, de acuerdo con al menos algunas realizaciones. En al menos algunas realizaciones, cada nodo de entrada (los nodos de entrada se muestran en la figura 4 como servidores de entrada 112) anuncia su capacidad para enrutar uno o más puntos finales públicos (por ejemplo, dirección IP y puerto) al enrutador periférico 104 para el equilibrador de carga distribuido, por ejemplo a través del protocolo de puerta de enlace de frontera (BGP). En al menos algunas realizaciones, en lugar de cada anuncio del propio nodo de entrada al enrutador periférico 104 a través de una sesión BGP, uno o más nodos de entrada diferentes, por ejemplo, dos nodos vecinos, pueden establecer sesiones BGP con el enrutador periférico 104 para anunciar el nodo de entrada, como se muestra en la figura 5.

Típicamente, los equilibradores de carga convencionales sólo dan servicio a un único extremo público. Por el contrario, las realizaciones del equilibrador de carga distribuido permiten que múltiples nodos de equilibradores de carga 110 den servicio a un único extremo público. Dependiendo de las capacidades del enrutador, esto permite configuraciones en las que una única dirección IP pública enrutada a todos los servidores de entrada 112 puede manejar todo el ancho de banda (por ejemplo, 160 Gbps) a través del uno o más enrutadores periféricos 104. En al menos algunas realizaciones, para conseguir esto, el enrutador o enrutadores periféricos 104 pueden utilizar una técnica de enrutamiento por trayectos múltiples de capa 4 por flujo, por ejemplo una técnica de enrutamiento de múltiples trayectos de igual coste (ECMP), para distribuir tráfico a través de múltiples servidores de entrada 112, anunciando cada uno la misma dirección IP pública. La distribución de paquetes entrantes a todos los servidores de entrada 112 que utilizan puertos de origen y destino de capa 4 para los flujos como parte del hash de flujo de uno o más enrutadores periféricos 104 puede mantener generalmente los paquetes para cada conexión enrutada al mismo nodo de equilibrador de carga 110 que sirve como el servidor de entrada 112 para evitar paquetes fuera de orden.

Ha de apreciarse que, sin embargo, el enrutador o enrutadores periféricos 104 pueden usar otras técnicas para distribuir tráfico a través de los servidores de entrada 112 en algunas realizaciones.

La figura 4 también muestra que dos o más equilibradores de carga distribuidos pueden ser implementados en una red 100. Los dos o más equilibradores de carga distribuidos pueden actuar cada uno como un equilibrador de carga independiente que enfrenta a una pluralidad de servidores 130 y que cada uno anuncia un diferente la dirección IP pública o, como alternativa, como se muestra en la figura 4, dos o más equilibradores de carga distribuidos pueden anunciar la misma dirección IP, y una técnica de hashing (por ejemplo, una técnica de enrutamiento multitrayecto de hash de capa 4 por flujo) puede usarse en el enrutador o enrutadores de frontera 102 para dividir los flujos de paquetes hacia los enrutadores periféricos 104, los cuales a su vez distribuyen los flujos de paquetes a sus respectivos servidores de entrada 112.

La figura 5 ilustra el uso del Protocolo de Puerta de Enlace de Frontera (BGP) para anunciar los nodos de entrada al enrutador periférico, según al menos algunas realizaciones. En este ejemplo, hay cuatro nodos de equilibradores de carga que actúan como nodos de entrada 110A a 110D en la implementación del equilibrador de carga. El enrutador periférico 104 envía paquetes entrantes de clientes (no mostrados) a los nodos de equilibradores de carga 110. En al menos algunas realizaciones, el enrutador periférico 104 puede tomar las decisiones de enrutamiento de acuerdo con una técnica de enrutamiento multitrayecto de hash de capa 4 por flujo, por ejemplo una técnica de enrutamiento multitrayecto de igual coste (ECMP).

En al menos algunas realizaciones, el enrutador periférico 104 aprende acerca de los nodos de entrada 110 que están actualmente disponibles en la implementación del equilibrador de carga para recibir tráfico del cliente a través de las sesiones de publicidad de tecnología del Protocolo de Puerta de Enlace de Frontera (BGP) iniciadas por los nodos de entrada 110. Cada nodo de entrada 110 podría usar BGP para anunciarse a sí mismo al enrutador periférico 104. Sin embargo, el BGP normalmente tarda un tiempo relativamente largo en converger (tres segundos o más). Utilizando esta técnica en la que cada nodo de entrada 110 se anuncia a través del BGP, si un nodo de entrada 110 desciende, puede llevar un tiempo considerable en términos de red (tres segundos o más) para que la sesión de BGP en el enrutador periférico 104 termine y, por lo tanto, para que el enrutador periférico 104 aprenda sobre el cierre del fallo y redirigir el flujo TCP actual hacia el nodo de entrada 110.

Para evitar el problema de convergencia con BGP y recuperarse más rápidamente después del fallo del nodo 110, en al menos algunas realizaciones, en lugar de un nodo de entrada 110 que se anuncia al enrutador periférico 104 a través de una sesión BGP, al menos otro nodo de entrada 110 en la implementación del equilibrador de carga toma la responsabilidad de anunciar el nodo de entrada 110 al enrutador periférico 104 a través de BGP. Por ejemplo, en algunas realizaciones, como se muestra en la figura 5, los nodos de entrada vecinos izquierdo y derecho 110 de un nodo de entrada dado 110, por ejemplo, los vecinos izquierdo y derecho en una lista ordenada de los nodos 110, por ejemplo, un anillo hash consistente formado por los nodos 110, pueden anunciar el nodo de entrada dado 110 al enrutador periférico 104. Por ejemplo, en la figura 5, el nodo de entrada 110A anuncia los nodos de entrada 110B y 110D, el nodo de entrada 110B anuncia los nodos de entrada 110A y 110C, el nodo de entrada 110C anuncia los nodos de ingreso 110B y 110D, y el nodo de entrada 110D anuncia los nodos de entrada 110C y 110A. Los nodos de ingreso 110 comprueban y hablan sobre el estado de cada uno como se describe más adelante en este documento. Utilizando el método de comprobación del estado como se describe, se pueden detectar nodos sin estado y la información puede propagarse entre los nodos 110 en menos de un segundo, por ejemplo, en 100 milisegundos (ms). Tras determinar que un nodo de entrada 110 no tiene estado, los nodos de entrada 110 que anuncian el nodo sin estado pueden dejar inmediatamente de anunciar el nodo no sin estado 110. En al menos algunas realizaciones, los nodos de entrada 110 terminan las sesiones BGP con el enrutador periférico 104 enviando un mensaje de cierre de TCP o similar para la sesión BGP al enrutador periférico 104. Por lo tanto, en lugar de tener que esperar a que una sesión BGP establecida por un nodo fallido 110 termine para detectar el fallo del nodo 110, el enrutador periférico 104 puede descubrir el nodo de fallo 110 cuando los otros nodos de entrada 110 que anuncian en nombre del nodo fallido 110 terminan las sesiones BGP con el enrutador periférico 104 que anuncia el nodo 110 tras detectar que el nodo 110 no tiene estado. El tratamiento de los fallos de los nodos del equilibrador de carga se analiza adicionalmente en relación con las figuras 18A y 18B más adelante en este documento.

La figura 6 es un diagrama de flujo de un método de enrutamiento de trayectos múltiples, de acuerdo con al menos algunas realizaciones del sistema de equilibrado de carga distribuido. Como se indica en 900, los nodos de entrada 110 en una implementación de equilibrador de carga anuncian sus nodos vecinos 110 al enrutador periférico 104. En al menos algunas realizaciones, los nodos de entrada 110 pueden determinar sus nodos vecinos 110 de acuerdo con una lista ordenada de los nodos 110 tal como un anillo hash consistente. En al menos algunas realizaciones, los nodos de entrada 110 anuncian sus nodos vecinos 110 al enrutador periférico 104 usando sesiones BGP, con una sesión BGP establecida para el enrutador periférico 104 para cada nodo anunciado 110.

Como se indica en 902, el enrutador periférico 104 distribuye el tráfico recibido desde los clientes 160 a los nodos de entrada activos (anunciados) 110 de acuerdo con una técnica de enrutamiento multitrayecto de hash por flujo, por ejemplo, una técnica de enrutamiento multitrayecto de igual coste (ECMP). En al menos algunas realizaciones, el enrutador periférico 104 expone una dirección IP pública a los clientes 160; los nodos de entrada 110 anuncian todos la misma dirección IP pública al enrutador periférico 104. El enrutador periférico usa los puertos de origen y de destino de la capa 4 como parte del hash de flujo del enrutador periférico 104 para distribuir los paquetes entrantes entre los nodos de entrada 110. Esto generalmente mantiene los paquetes para cada conexión enrutada al mismo nodo de entrada 110.

Como se indica en 902, los nodos de entrada transmiten los flujos de datos a los nodos servidor de destino 130. En al menos algunas realizaciones, los nodos de entrada 110 interactúan con los nodos de rastreador de flujo primario y secundario para los flujos de datos para asignar los flujos de datos a los nodos servidor de destino 130. Por lo tanto, cada nodo de entrada 110 puede mantener asignaciones de flujos de datos activos a través del nodo 110 que pueden usarse para transmitir apropiadamente los paquetes recibidos a los nodos servidor de destino 130.

Los elementos 906 a 910 se refieren a la detección y recuperación de fallos del nodo de entrada 110. Como se indica en 906, los nodos de entrada 110 pueden detectar que un nodo de entrada 110 está inactivo, por ejemplo, de acuerdo con una técnica de comprobación de estado como se describe en el presente documento. Tras detectar que el nodo 110 está inactivo, sus nodos vecinos 110 dejan de anunciar el nodo 110 al enrutador periférico 104. En al menos algunas realizaciones, esto implica enviar un cierre de TCP al enrutador periférico 104 para la respectiva sesión BGP.

Como se indica en 908, el enrutador periférico 104, tras detectar que el nodo de entrada 110 está inactivo a través del cierre de las sesiones BGP, redistribuye el tráfico entrante de los clientes 160 a los nodos de entrada restantes 110 de acuerdo con la técnica de enrutamiento multitrayecto de hash por flujo. Por lo tanto, al menos algunos flujos de datos pueden enrutarse a diferentes nodos de entrada 110.

Como se indica en 910, los nodos de entrada 110 pueden recuperar las asignaciones según sea necesario y enviar los flujos de datos a los nodos servidor de destino apropiados. Los métodos para recuperar fallos del nodo 110 en los nodos de entrada 110 se analizan en otra parte de este documento. Como un ejemplo, un nodo de entrada 110, tras recibir un paquete para el cual no tiene una asignación actual, puede usar una función de hash consistente para determinar un nodo de rastreador de flujo para el flujo de datos de acuerdo con un anillo hash consistente y recuperar la asignación del nodo de rastreador de flujo.

Flujo de paquetes asimétrico

En al menos algunas realizaciones, para utilizar eficientemente el ancho de banda del nodo de entrada y el uso de CPU cuando la relación de tráfico saliente con respecto a datos entrantes es mayor de 1, el sistema de equilibrado de carga distribuido envía paquetes salientes desde los nodos servidor 130 a múltiples nodos de salida como se muestra en la figura **7.** En al menos algunas realizaciones, para cada conexión, el módulo de equilibrador de carga 132 en el nodo servidor respectivo 130 direcciona la tupla de extremo de cliente/extremo público y utiliza un algoritmo de hash consistente para seleccionar un nodo de equilibrador de carga 110 para dar servicio como el servidor de salida 114 para el respectivo flujo de paquetes de salida. Sin embargo, en algunas realizaciones pueden usarse otros métodos y/o datos para seleccionar los servidores de salida 114 para conexiones. El servidor de salida seleccionado 114 puede ser, pero no necesariamente, un nodo de equilibrador de carga diferente 110 del nodo de equilibrador de carga 110 que da servicio como el servidor de entrada 112 para la conexión. En al menos algunas realizaciones, a menos que haya un fallo de ese nodo de equilibrador de carga 110/servidor de salida 114, todos los paquetes salientes para la conexión particular serán reenviados al mismo servidor de salida 114 para evitar paquetes fuera de orden.

En al menos algunas realizaciones, el método y los datos utilizados para seleccionar un servidor de salida 114 por los nodos servidor 130 pueden ser diferentes del método y datos utilizados para seleccionar un servidor de entrada 112 realizado por el enrutador o enrutadores periféricos 104. El uso de los diferentes métodos y datos puede dar como resultado generalmente un nodo de equilibrador de carga diferente 110 que es seleccionado como nodo de salida para una conexión dada que el nodo de equilibrador de carga 110 seleccionado como nodo de entrada para la conexión, y también puede dar lugar a múltiples nodos de equilibrador de carga 110 que se seleccionan como nodos de salida para manejar el tráfico saliente para conexiones que pasan a través de un único nodo de equilibrador de carga 110 que sirve como nodo de entrada.

La figura 7 ilustra gráficamente el flujo de paquetes asimétricos, de acuerdo con al menos algunas realizaciones. Se ha establecido al menos una conexión desde los clientes 160 en la red externa 150 a través del servidor de entrada 112 a cada uno de los nodos servidor 130A, 130B, 130C y 130D. En al menos algunas realizaciones, para seleccionar los nodos de salida para las conexiones, para cada conexión, el módulo de equilibrador de carga 132 en el nodo servidor respectivo 130 direcciona la tupla de extremo de cliente/extremo público y utiliza un algoritmo de hash consistente para seleccionar un nodo de equilibrador de carga 110 para dar servicio como el servidor de salida 114 para el respectivo flujo de paquetes de salida. Por ejemplo, el nodo servidor 130A ha seleccionado el servidor de salida 114A para una conexión, y el nodo servidor 130B ha seleccionado el servidor de salida 114A para un servidor de conexión y salida 114B para otra conexión. Sin embargo, en algunas realizaciones pueden usarse otros métodos y/o datos para seleccionar los nodos de salida para conexiones.

Recuperación de los fallos de los nodos del equilibrador de carga sin dejar caer las conexiones del cliente

Aunque es posible que los nodos de equilibradores de carga 110 utilicen hashing consistente para determinar qué nodo servidor 130 debe recibir tráfico de cliente, debido a la larga vida útil de algunas conexiones, este enfoque puede no mantener flujos existentes en casos en los que un nuevo nodo servidor 130 se une al miembro de hash consistente y hay un fallo del nodo del equilibrador de carga de entrada posterior 110. En este escenario, un nodo de equilibrador de carga 110 que asume un flujo desde el nodo fallido 110 puede no ser capaz de determinar la asignación original seleccionada, ya que el anillo hash consistente para los servidores 130 tendrá miembros diferentes. Por lo tanto, en al menos algunas realizaciones, la tecnología de tabla de hash distribuida (DHT) puede usarse por los nodos de equilibradores de carga 110 para seleccionar nodos servidor 130 para conexiones y para enrutar paquetes a los nodos servidor seleccionados 130. Una vez que se ha seleccionado un nodo servidor 130 de acuerdo con la DHT para recibir una conexión particular, y asumiendo que el nodo servidor 130 permanece bien y que el módulo de equilibrador de carga 132 en el nodo servidor 130 continúa extendiendo el arrendamiento transmitiendo periódicamente el estado de esa conexión activa a la DHT (por ejemplo, a través de la publicación de conexión), la DHT conservará la asignación hasta que se complete la conexión. Un fallo del nodo de entrada 110 afecta a la distribución de paquetes desde el enrutador periférico 104 a los nodos de equilibrador de carga restantes 110, dando lugar a que los nodos de equilibradores de carga 110 reciban tráfico desde un conjunto diferente de conexiones de cliente. Sin embargo, puesto que la DHT rastrea todas las conexiones activas, los nodos de equilibrador de carga 110 pueden consultar a la DHT para obtener concesiones para cualquier asignación activa. Como resultado, todos los nodos de equilibradores de carga 110 pasarán tráfico a los nodos servidor correctos 130, evitando así el fallo de las conexiones de cliente activas incluso en el caso de un fallo del nodo del equilibrador de carga de entrada 110.

Flujo de paquetes en el sistema de equilibrado de carga distribuido

La figura 8 ilustra el flujo de paquetes en el sistema distribuido de equilibrado de carga, de acuerdo con al menos algunas realizaciones. Obsérvese que las líneas continuas con flechas en la figura 8 representan paquetes TCP, mientras que las líneas discontinuas con flechas representan paquetes UDP. En la figura 8, un servidor de entrada 112 recibe paquetes TCP de uno o más clientes 160 a través del enrutador periférico 104. Tras recibir un paquete TCP, el servidor de entrada 112 determina si tiene una asignación para el flujo de paquetes TCP a un nodo servidor 130. Si el servidor de entrada 112 tiene una asignación para el flujo de paquetes TCP, entonces el servidor 112 encapsula el paquete TCP (por ejemplo de acuerdo con UDP) y envía el paquete encapsulado al nodo servidor de destino 130. Si el servidor de entrada 112 no tiene una asignación para el flujo de paquetes TCP, entonces el servidor de entrada 112 puede enviar un mensaje UDP que incluye información sobre el flujo de paquetes TCP extraído del paquete TCP al rastreador de flujo primario 116A para establecer una conexión a un nodo servidor 130 y/o obtener una asignación para el flujo de paquetes TCP. Las figuras 9A y 9B y las figuras 10A a 10G ilustran métodos para establecer una conexión entre un cliente 160 y un nodo servidor 130. El módulo de equilibrador de carga 132 en un nodo servidor 130 selecciona aleatoriamente uno o más nodos de equilibrador de carga 110 para dar servicio como el servidor o servidores de salida 114 para la conexión o conexiones TCP en el nodo servidor 130 y envía paquetes de respuesta TCP encapsulados de UDP al cliente o clientes 160 a través del servidor o servidores de salida 114.

Las figuras 9A y 9B proporcionan un diagrama de flujo de flujo de paquetes al establecerse conexiones en el sistema de equilibrado de carga distribuido, de acuerdo con al menos algunas realizaciones. Como se indica en 200 de la figura 9A, un servidor de entrada 112 recibe un paquete TCP desde un cliente 160 a través del enrutador periférico 104. En 202, si el servidor de entrada 112 tiene una asignación para el flujo TCP a un nodo servidor 130, entonces el servidor de entrada 112 encapsula y envía el paquete TCP al respectivo nodo servidor 130 como se indica en 204. Observe que el servidor de entrada 112 puede recibir continuamente y procesar paquetes para uno, dos o más flujos TCP de uno, dos o más clientes 160.

En 202, si el servidor de entrada 112 no tiene una asignación para el flujo TCP, el paquete puede ser un paquete de sincronización TCP (SYN) desde un cliente 160. Como se indica en 206, tras la recepción de un paquete SYN, el servidor de entrada 112 extrae datos del paquete SYN y envía los datos al rastreador de flujo primario 116A, por ejemplo en un mensaje UDP. En al menos algunas realizaciones, el servidor de entrada 112 puede determinar el rastreador de flujo primario 116A y/o el rastreador de flujo secundario 116B para el flujo de TCP de acuerdo con una función de hash consistente. En 208, el rastreador de flujo primario 116A almacena los datos, por ejemplo en una tabla de hash, genera un número de secuencia TCP inicial para el lado del nodo servidor 130 de la conexión TCP, y reenvía los datos y el número de secuencia TCP al rastreador de flujo secundario 116B. En 210, el rastreador de flujo secundario 116B también puede almacenar los datos, y fabrica y envía un paquete SYN/ACK al cliente 160, el paquete SYN/ACK que contiene al menos el número de secuencia TCP.

Como se indica en 212, el servidor de entrada 112 recibe un paquete de acuse de recibo TCP (ACK) del cliente 160 a través del enrutador periférico 104. El servidor de entrada 112 no tiene en este momento una asignación para el flujo TCP a un nodo servidor 130, por lo que en 214, el servidor de entrada 112 envía un mensaje que incluye datos extraídos del paquete ACK al rastreador de flujo primario 116A. Como se indica en 216, tras recibir el mensaje, el rastreador de flujo primario 116A confirma el flujo TCP de acuerdo con los datos almacenados, y confirma que el número de secuencia reconocido (+1) del paquete ACK coincide con el valor enviado en SYN/ACK. El rastreador de flujo primario 116A selecciona entonces un nodo servidor 130 para recibir el flujo TCP, y envía un mensaje que contiene los datos, el número de secuencia TCP y la dirección IP del módulo de equilibrador de carga local 132 en el nodo servidor seleccionado 130 al rastreador de flujo secundario 116B. Como se indica en 218, el rastreador de flujo secundario 116B también confirma los datos y el número de secuencia TCP, fabrica un mensaje SYN, y envía el mensaje SYN fabricado al módulo de equilibrador de carga local 132 en el nodo servidor seleccionado 130. El método continúa en el elemento 220 de la figura 9B.

Como se indica en 220 de la figura 9B, en respuesta al mensaje SYN fabricado, el módulo de equilibrador de carga 132 puede examinar una o más métricas del nodo servidor 130 para determinar si el nodo servidor 130 puede aceptar la conexión. En 222, si el módulo de equilibrador de carga 132 determina que el nodo servidor 130 no puede aceptar actualmente la conexión, entonces en 224 el módulo de equilibrador de carga 132 envía mensajes al rastreador de flujo secundario 116B. El rastreador de flujo secundario 116B puede suprimir la información para el flujo que almacenó previamente. En 226, el rastreador de flujo secundario 116B envía mensajes al rastreador de flujo primario 116A. El rastreador de flujo primario 116A puede entonces seleccionar un nuevo nodo servidor de destino 130 y enviar un mensaje al rastreador de flujo secundario 116B como se indica en 216 de la figura 9A.

En 222, si el módulo de equilibrador de carga 132 determina que el nodo servidor 130 puede aceptar la conexión, entonces como se indica en 228 de la figura 9B, el módulo de equilibrador de carga local 132 construye un paquete TCP SYN a partir del SYN fabricado y envía el paquete TCP SYN al servidor 134 en el nodo servidor 130. La dirección IP de origen del paquete TCP SYN se rellena con la dirección IP real del cliente 160 de modo que el servidor 134 cree que ha recibido una conexión TCP directa al cliente 160. El módulo de equilibrador de carga 132 almacena detalles relevantes sobre el flujo TCP, por ejemplo, en una tabla de hash local. Como se indica en 230, el servidor 134 responde con un paquete SYN/ACK que el módulo de equilibrador de carga 132 intercepta. Como se indica en 232, el módulo de equilibrador de carga 132 envía entonces un mensaje que incluye información de conexión al rastreador de flujo secundario 116B para indicar que la conexión ha sido aceptada. Tras la recepción de este mensaje, en 234, el rastreador de flujo secundario 116B registra la asignación al servidor 134, y envía un mensaje similar al rastreador de flujo primario 116A, que también registra la información de asignación. Como se indica en 236, el rastreador de flujo primario 116A envía entonces un mensaje de asignación al servidor de entrada 112. El servidor de entrada 112 tiene ahora una asignación para el flujo TCP del cliente 160 al servidor 130.

En 238, el servidor de entrada 112 encapsula y envía cualesquiera paquetes de datos almacenados en búfer para el flujo de datos al módulo de equilibrador de carga local 132 en el nodo servidor 130. Los paquetes entrantes adicionales para el flujo de datos del cliente 160 recibidos por el servidor de entrada 112 se encapsulan y se reenvían directamente al módulo de equilibrador de carga 132, que decapsula los paquetes y envía los paquetes de datos al servidor 134.

En 240, el módulo de equilibrador de carga 132 selecciona aleatoriamente un servidor de salida 114 para el flujo de datos. Los paquetes TCP salientes posteriores del servidor 134 se interceptan por el módulo de equilibrador de carga 132, se encapsulan de acuerdo con UDP y se reenvían al servidor de salida seleccionado arbitrariamente 114. El servidor de salida 114 decapsula los paquetes salientes y envía los paquetes TCP al cliente 160.

Como se ha indicado anteriormente, en 202, si el servidor de entrada 112 no tiene una asignación para el flujo TCP

de un paquete recibido, el paquete puede ser un paquete de sincronización TCP (SYN) desde un cliente 160. Sin embargo, el paquete puede no ser paquete TCP SYN. Por ejemplo, si cambia la pertenencia del nodo del equilibrador de carga 110 debido a la adición o fallo de un nodo de equilibrador de carga 110, el enrutador periférico 104 puede iniciar el enrutamiento de paquetes para uno o más flujos TCP al servidor de entrada 112 que los que el servidor de entrada 112 no tiene asignaciones. En al menos algunas realizaciones, tras recibir tal paquete para el cual el servidor de entrada 112 no tiene una asignación, el servidor de entrada 112 puede usar la función de hash consistente para determinar el rastreador de flujo primario 116A y/o el rastreador de flujo secundario 116B para el flujo TCP de acuerdo con el anillo hash coherente y envían mensajes al rastreador de flujo primario 116A o al rastreador de flujo secundario 116B para solicitar la asignación. Tras recibir la asignación para el flujo TCP desde un rastreador de flujo 116, el servidor de entrada 112 puede almacenar la asignación y comenzar a encapsular y reenviar el uno o más paquetes TCP para el flujo TCP al nodo servidor de destino correcto 130.

Detalles del nodo del equilibrador de carga

En al menos algunas realizaciones, los nodos de equilibradores de carga 110 tienen cada uno tres funciones:

• Entrada - Recepción de todos los paquetes entrantes desde un cliente 160 en una conexión cliente, enrutando los paquetes a un nodo servidor 130 si se conoce la asignación, o enviando mensajes a un rastreador de flujo si la asignación no se conoce. Los paquetes salientes de un nodo de entrada están encapsulados (por ejemplo, de acuerdo con UDP) por el nodo de entrada.
• Seguimiento de flujo - Control de los estados de conexión (por ejemplo, qué nodo servidor 130/servidor 134 se ha asignado para dar servicio a cada conexión de cliente). Los rastreadores de flujo también participan en el establecimiento de conexiones entre los clientes 160 y los servidores 134.
• Salida - Decapsulación y reenvío de paquetes salientes recibidos de un servidor 134 a un cliente 160.

En al menos algunas realizaciones, en la función de entrada, un nodo de equilibrador de carga 110 es responsable de reenviar paquetes a los servidores 134 cuando se conoce una asignación cliente-> servidor, o reenvía una solicitud a un rastreador de flujo cuando la asignación es desconocida. En al menos algunas realizaciones, un nodo de equilibrador de carga 110 que da servicio como un nodo de entrada para una conexión de cliente/flujo de datos particular también puede dar servicio como el rastreador de flujo primario o el rastreador de flujo secundario para la conexión de cliente, pero no ambos.

En al menos algunas realizaciones, en el rol de rastreador de flujo, un nodo de equilibrador de carga 110 es responsable de mantener el estado de las conexiones que todavía se están estableciendo, así como mantener la asignación cliente-> servidor para conexiones establecidas. Dos rastreadores de flujo están involucrados con cada conexión de cliente individual, conocida como el rastreador de flujo primario y el rastreador de flujo secundario. En al menos algunas realizaciones, los rastreadores de flujo asociados con las conexiones de cliente pueden determinarse usando un algoritmo de hash consistente. Los rastreadores de flujo también realizan funcionalidad de equilibrado de carga, que incluye, pero sin limitación, una selección pseudo-aleatoria de un nodo servidor 130 para cada nueva conexión de cliente. Obsérvese que el módulo de equilibrador de carga local 132 en un nodo servidor seleccionado 130 puede rechazar una petición de conexión si determina que el servidor 134 no puede manejar la conexión. Si esto sucede, entonces los rastreadores de flujo pueden seleccionar otro nodo servidor 130 y enviar la solicitud de conexión al otro nodo servidor 130. En al menos algunas realizaciones, la función de rastreador de flujo primario y la función de rastreador de flujo secundario para una conexión dada se realizan por diferentes nodos de equilibrador de carga 110.

En al menos algunas realizaciones, en el papel de salida, un nodo de equilibrador de carga 110 no tiene estado y decapsula paquetes entrantes recibidos de nodos servidor 130, realiza cierta validación, y envía los paquetes TCP salientes a clientes respectivos 160. En al menos algunas realizaciones, un módulo de equilibrador de carga local 132 en un nodo servidor 130 puede seleccionar arbitrariamente un nodo de equilibrador de carga 110 para una conexión dada.

Topología de anillo hash consistente con el nodo de equilibrador de carga

En al menos algunas realizaciones, los nodos de equilibrador de carga 110 forman una topología de anillo basada en el hashing consistente del espacio de claves de entrada (extremo del cliente, extremo público). El espacio de claves de entrada puede dividirse entre los nodos de rastreador de flujo disponibles, y cada nodo de rastreador de flujo puede ser responsable de responder a las consultas correspondientes a su espacio de claves. En al menos algunas realizaciones, los datos pueden replicarse a los nodos de rastreador de flujo primario y secundario basados en el sucesor en el anillo hash consistente (por ejemplo, el nodo de rastreador de flujo secundario es el nodo sucesor, o el

nodo siguiente en el anillo hash consistente, con respecto al nodo de rastreador de flujo primario). Si un nodo de rastreador de flujo se inactiva por alguna razón, el siguiente nodo de equilibrador de carga en el anillo hash consistente adquiere el espacio de claves del nodo fallido. Cuando un nuevo nodo de rastreador de flujo se une, el nodo registra su extremo (por ejemplo, con un servicio de configuración 122 como se muestra en la figura 1) para que otros nodos de equilibrador de carga puedan aprender sobre el cambio de configuración en la implementación de equilibrador de carga y, por lo tanto, en el anillo hash consistente. La manipulación de las adiciones y fallos de los rastreadores de flujo en el anillo hash consistente se analiza con más detalle con referencia a las figuras 11A a 11D.

Comunicaciones de nodo de entrada <-> nodo de rastreador de flujo

En al menos algunas realizaciones, los nodos de equilibradores de carga 110 que dan servicio como nodos de entrada pueden aprender acerca de los nodos de equilibradores de carga 110 que dan servicio como nodos de rastreador de flujo desde el servicio de configuración 122. Los nodos de entrada pueden controlar el servicio de configuración 122 para cambios de pertenencia en la implementación del equilibrador de carga y, por lo tanto, en el anillo hash consistente. Cuando un nodo de entrada recibe un paquete desde un cliente 160 para el que el nodo de entrada no tiene una asignación, el nodo de entrada puede utilizar una función de hash consistente para determinar qué nodo de rastreador de flujo debe dar servicio al paquete. En al menos algunas realizaciones, la entrada a la función de hash es el par (extremo del cliente, extremo público) del paquete. En al menos algunas realizaciones, los nodos de entrada y los nodos de rastreador de flujo se comunican usando mensajes UDP.

Cuando un nodo de rastreador de flujo primario recibe un mensaje desde un nodo de entrada para un nuevo flujo de paquetes, el nodo de rastreador de flujo primario determina aleatoriamente un número de secuencia TCP y envía otro mensaje al nodo de rastreador de flujo secundario. El nodo de rastreador de flujo secundario genera un mensaje TCP SYN/ACK para el cliente. Ambos rastreadores de flujo recuerdan el par de extremos de conexión de cliente y el número de secuencia TCP, y conservan esta información hasta que la presión de memoria o la expiración hacen que el estado sea purgado.

Cuando el nodo de rastreador de flujo primario recibe un mensaje desde un nodo de entrada que ha recibido un paquete TCP ACK, el nodo de rastreador de flujo primario verifica que el número de secuencia TCP reconocido coincide con el valor almacenado que fue enviado en el paquete SYN/ACK, selecciona un nodo servidor 130 para dar servicio a la solicitud, y envía un mensaje al nodo de rastreador de flujo secundario. El nodo de rastreador de flujo secundario envía un mensaje al módulo de equilibrador de carga 132 en el nodo servidor seleccionado 130 para iniciar una conexión TCP real con el bastidor TCP en el nodo servidor 130, y luego espera una respuesta de acuse de recibo del nodo servidor 130.

Cuando el nodo de rastreador de flujo secundario recibe un acuse de recibo de conexión del módulo de equilibrador de carga 132 en el nodo servidor 130, se desencadena un flujo de mensaje inverso a través del rastreador de flujo primario al nodo de entrada que almacena información sobre el nodo servidor asociado 130 en ambos nodos. Desde este punto hacia adelante, los paquetes TCP adicionales recibidos en el nodo de entrada se reenvían directamente al módulo de equilibrador de carga 132 en el nodo servidor 130.

Comunicaciones del módulo de equilibrador de carga <-> nodo de equilibrador de carga

En al menos algunas realizaciones, cada módulo de equilibrador de carga 132 registra su extremo con el servicio de configuración 122 y controla el servicio de configuración 122 continuamente para cambios de pertenencia en la capa de nodo de equilibrador de carga. A continuación se describen las funciones del módulo de equilibrador de carga 132, de acuerdo con al menos algunas realizaciones:

• Publicación de la conexión - Publicar periódicamente (por ejemplo, una vez por segundo) o aperiódicamente el conjunto de conexiones activas (extremo del cliente, extremo público) en el nodo servidor respectivo 130 tanto para los nodos de rastreadores de flujo primarios como secundarios responsables de esas conexiones, así como para los nodos de entrada que enviaron por último paquetes al módulo de equilibrador de carga 132 para esas conexiones. La función de publicación de conexión renueva la concesión para los estados de conexión en los nodos de equilibradores de carga responsables 110.
• Supervisar los cambios de pertenencia en la capa de equilibrador de carga. Si cambia la pertenencia, los módulos de equilibrador de carga 132 pueden usar esta información de cambio para enviar inmediatamente conexiones activas a los nodos de equilibrador de carga que son ahora responsables de las conexiones.

Flujo de paquetes en el sistema de equilibrado de carga distribuido - detalles

El sistema de equilibrado de carga distribuido puede incluir múltiples nodos de equilibradores de carga 110. En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 en el sistema de equilibrado de carga distribuido puede dar servicio en las funciones de un nodo de rastreador de flujo, un nodo de salida y un nodo de entrada para las conexiones del cliente 160 con los servidores 134. El sistema de equilibrado de carga distribuido también puede incluir un módulo de equilibrador de carga 132 en cada nodo servidor 130.

Las figuras 10A a 10G ilustran el flujo de paquetes en el sistema distribuido de equilibrado de carga, de acuerdo con al menos algunas realizaciones. En las figuras 10A a 10G, los paquetes intercambiados entre los nodos de equilibrador de carga 110 y los paquetes intercambiados entre los nodos de equilibrador de carga 110 y los nodos servidor 130 son mensajes UDP o paquetes TCP de cliente encapsulados de UDP. En al menos algunas realizaciones, los paquetes TCP de cliente sólo existen en la red 100 de forma decapsulada en el lado norte de los nodos de equilibradores de carga 110 en tránsito hacia y desde el enrutador de frontera 102 (véase la figura 1). Obsérvese que las líneas continuas con flechas en las figuras 10A-10G representan paquetes TCP, mientras que las líneas discontinuas con flechas representan paquetes UDP.

En al menos algunas realizaciones, el sistema de equilibrado de carga distribuido puede intentar preservar las conexiones establecidas en el caso de un fallo único del nodo del equilibrador de carga 110. En al menos algunas realizaciones, esto puede lograrse replicando detalles de conexión en un nodo de rastreador de flujo primario y un nodo de rastreador de flujo secundario de manera que, si uno de estos nodos falla, la asignación cliente-> servidor de una conexión puede restaurarse por el nodo de rastreador de flujo restante. En al menos algunas realizaciones, puede producirse alguna pérdida de paquetes en el caso de un fallo de nodo; sin embargo, las retransmisiones de paquetes TCP cliente/servidor pueden recuperar los paquetes perdidos.

Cada conexión TCP de un cliente puede denominarse como un flujo TCP, y se identifica de manera única por una tupla de 4 que consiste en: la dirección IP del cliente, el puerto del cliente, la dirección IP del servidor (pública) y el puerto del servidor. Este identificador puede abreviarse como CP o CcPp que indica el par de puntos finales de cliente y público. Pueden aparecer paquetes asociados con cualquier flujo TCP dado (o par CP) en cualquier nodo de equilibrador de carga 110 que funcione como un servidor de entrada 112 debido a la distribución de flujo multitrayecto de hash de igual coste (ECMP) desde el enrutador periférico 104. Sin embargo, los paquetes para un flujo TCP generalmente pueden seguir llegando al mismo nodo de equilibrador de carga 110 a menos que haya un fallo de enlace o de nodo de equilibrador de carga 110 que hace que los flujos TCP se redireccionen. El nodo de equilibrador de carga 110 que recibe paquetes para un flujo TCP del enrutador aguas arriba 104 se denomina nodo de entrada para el flujo TCP.

En al menos algunas realizaciones, se utiliza hashing consistente de manera que cuando los paquetes llegan a un nodo de equilibrador de carga 110 que sirve como un nodo de entrada para el flujo TCP, el nodo de entrada puede determinar qué nodo de equilibrador de carga 110 contiene el estado para el flujo TCP (es decir, el nodo de rastreador de flujo). El par CP puede estar en hash por el nodo de entrada en un anillo hash consistente para determinar qué nodo de equilibrador de carga 110 es responsable de mantener el estado con respecto al flujo TCP. Este nodo sirve como el rastreador de flujo primario para el flujo TCP. El nodo sucesor en el anillo hash consistente sirve como el seguidor de flujo secundario para el flujo TCP.

En al menos algunas realizaciones, todos los nodos de equilibradores de carga 110 pueden servir como nodos de entrada, nodos de rastreador de flujo primario y nodos de rastreador de flujo secundarios. Dependiendo del resultado de hash consistente para un flujo TCP, un nodo de equilibrador de carga 110 que sirve como nodo de entrada para el flujo TCP también puede servir como nodo de rastreador de flujo primario o secundario para el flujo TCP. Sin embargo, en al menos en algunas realizaciones, diferentes nodos de equilibradores de carga física 110 realizan las funciones de rastreador de flujo primario y secundario para el flujo TCP.

Establecimiento de conexiones

Haciendo referencia a la figura 10A, las nuevas conexiones desde un cliente 160 pueden desencadenarse por un paquete de sincronización de TCP (SYN) del cliente. Los nodos de equilibradores de carga 110 no establecen realmente una conexión con un nodo servidor 130 tras la recepción del paquete SYN, ni seleccionan inmediatamente un nodo servidor 130 para recibir la conexión. En cambio, los nodos de equilibradores de carga 110 almacenan datos relevantes del paquete SYN del cliente, y generan un paquete SYN/ACK en nombre del nodo servidor aún no elegido 130. Haciendo referencia a la figura 10C, una vez que el cliente 160 responde con el primer paquete ACK en el intercambio de tres vías TCP, los nodos de equilibrador de carga 110 seleccionan un nodo servidor 130, generan un paquete SYN equivalente para ese nodo servidor 130, e intentan establecer una conexión TCP real con el nodo servidor 130.

Haciendo referencia de nuevo a la figura 10A, tras la recepción de un paquete SYN de cliente en el nodo de equilibrador de carga 110 que sirve como servidor de entrada 112 para el flujo TCP, el servidor de entrada 112 extrae los campos de datos del paquete SYN y envía los datos al rastreador de flujo primario 116A para el flujo TCP. El rastreador de flujo primario 116A almacena los datos, por ejemplo en una tabla de hash, genera un número de secuencia TCP inicial (para el lado servidor de la conexión TCP), y envía los mismos datos al rastreador de flujo secundario 116B. El rastreador de flujo secundario 116B fabrica un paquete SYN/ACK para el cliente 160 que contiene ese número de secuencia TCP del servidor.

En la figura 10A, el servidor de entrada 112, el rastreador de flujo primario 116A, y el rastreador de flujo secundario 116B se realizan cada uno por diferentes nodos de equilibrador de carga 110. Sin embargo, en algunos casos, el nodo de equilibrador de carga 110 que da servicio como el servidor de entrada 112 para un flujo TCP puede ser el mismo nodo 110 que da servicio como el rastreador de flujo primario 116A o el rastreador de flujo secundario 116B para el flujo TCP (pero no ambos). La razón por la que el servidor de entrada 112 para un flujo de paquetes puede estar en el mismo nodo 110 como un rastreador de flujo 116 para el flujo es que el enrutador periférico 104 selecciona de manera pseudoaleatoria el servidor de entrada 112 para el flujo de acuerdo con una técnica de enrutamiento multitrayecto de hash por flujo (por ejemplo, una técnica de enrutamiento ECMP), mientras que los rastreadores de flujo 116 para el flujo de paquetes se determinan en un anillo hash consistente de acuerdo con una función de hash consistente aplicada a la información de dirección del flujo de paquetes. Si el servidor de entrada 112 para un flujo de paquetes está en el mismo nodo 110 como un rastreador de flujo 116 para el flujo de paquetes, los datos del paquete SYN sólo pueden ser reenviados desde el nodo 110 que implementa el servidor de entrada 112 al otro nodo 110 de rastreador de flujo 116. Por ejemplo, en la figura 10B, el rastreador de flujo primario 116A está en el mismo nodo de equilibrador de carga 110A que el servidor de entrada 112 para el flujo TCP, mientras que el rastreador de flujo secundario 116B está en un nodo de equilibrador de carga diferente 110B, y por lo tanto, los datos del paquete SYN se reenvían desde el nodo 110A (por el rastreador de flujo 116A) al rastreador de flujo secundario 116B en el nodo de equilibrador de carga 110B.

Haciendo referencia a la figura 10C, cuando los paquetes no SYN llegan a un servidor de entrada 112, el servidor de entrada 112 sabe o no sabe a qué nodo servidor 130 debe enviar los paquetes. El primer paquete no SYN para llegar a un servidor de entrada 112 para un flujo TCP debería ser el primer paquete de acuse de recibo TCP (ACK) en el intercambio de tres vías TCP (o posiblemente un paquete de datos posterior), donde el número de acuse de recibo TCP coincide con el número de secuencia del servidor (+1) que se envió en el paquete SYN/ACK en la figura 10A. Cuando el servidor de entrada 112 recibe un paquete no SYN para el cual no tiene asignación de servidor, envía un mensaje al rastreador de flujo primario 116A para el flujo TCP, incluyendo el mensaje la información del paquete ACK tal como un número de secuencia, o como alternativa, que contiene el propio paquete ACK. En al menos algunos casos, el rastreador de flujo primario 116A recuerda los datos almacenados para el flujo TCP y confirma que el número de secuencia de acuse de recibo (+1) coincide con el valor que se envió al cliente 160 en el paquete SYN/ACK. El controlador de flujo primario selecciona entonces un nodo servidor 130 para el flujo TCP y envía otro mensaje que contiene los datos previamente almacenados para el flujo TCP, el número de secuencia del servidor, y una dirección IP para el módulo de equilibrador de carga 132 en el nodo servidor seleccionado 130 al rastreador de flujo secundario 116B. El rastreador de flujo secundario 116B confirma el número de secuencia del servidor, registra la información, y envía un mensaje SYN fabricado al módulo de equilibrador de carga 132 en el nodo servidor seleccionado 130. El par de extremos CP del flujo TCP ahora está asignado al módulo de equilibrador de carga 132/nodo servidor 130. El módulo de equilibrador de carga 132 en el nodo servidor 130 es responsable de crear un paquete TCP SYN legítimo para el servidor 134 en el nodo servidor 130 cuando recibe el mensaje SYN fabricado desde el rastreador de flujo secundario 116B. En la creación del paquete SYN, la dirección IP de origen se rellena con la dirección IP real del cliente 160 para que el servidor 134 crea que ha recibido una solicitud de conexión TCP directa desde el cliente 160. El módulo de equilibrador de carga 132 almacena los detalles relevantes sobre el flujo TCP, por ejemplo, en una tabla de hash local, y envía el paquete SYN TCP al servidor 134 (por ejemplo, inyecta el paquete SYN en el núcleo Linux del servidor 134).

En la figura 10C, el servidor de entrada 112, el rastreador de flujo primario 116A, y el rastreador de flujo secundario 116B se realizan cada uno por diferentes nodos de equilibrador de carga 110. Sin embargo, en algunos casos, el nodo de equilibrador de carga 110 que da servicio como el servidor de entrada 112 para un flujo TCP será el mismo nodo 110 que da servicio como el rastreador de flujo primario 116A o el rastreador de flujo secundario 116B para el flujo TCP (pero no ambos). Por ejemplo, en la figura 10D, el rastreador de flujo secundario 116B está en el mismo nodo de equilibrador de carga 110A que el servidor de entrada 112 para el flujo TCP, mientras que el rastreador de flujo primario 116A está en un nodo de equilibrador de carga 110B diferente.

Haciendo referencia a la figura 10E, el servidor 134 (por ejemplo, el núcleo de Linux) responde con un paquete

SYN/ACK que el módulo de equilibrador de carga 132 también intercepta. El paquete SYN/ACK puede contener un número de secuencia TCP diferente que el suministrado originalmente al cliente 160 en el SYN/ACK generado desde el rastreador de flujo secundario 116B (véase la figura 10A). El módulo de equilibrador de carga 132 es responsable de aplicar el número de secuencia delta a paquetes entrantes y salientes. El paquete SYN/ACK del servidor 134 también desencadena un mensaje (por ejemplo, un mensaje UDP) desde el módulo de equilibrador de carga 132 de nuevo al rastreador de flujo secundario 116B para indicar que la conexión con el nodo servidor seleccionado 130/módulo de equilibrador de carga 132/servidor 134 ha tenido éxito. Tras la recepción de este mensaje, el rastreador de flujo secundario 116A puede registrar la asignación del par de extremos de cliente y público (CP) entre el cliente 160 y el servidor 134 como comprometido, y enviar un mensaje similar al rastreador de flujo primario 116A que también registrará la asignación CP. El rastreador de flujo primario 116A puede entonces reenviar un mensaje de asignación de CP al servidor de entrada 112, lo que hace que el servidor de entrada 112 reenvíe paquetes de datos almacenados en memoria intermedia para la conexión al módulo de equilibrador de carga local 132 en el nodo servidor 130 como paquetes de datos encapsulados.

Con referencia a la figura 10F, la asignación de CP para la conexión es conocida por el servidor de entrada, por lo que los paquetes TCP entrantes recibidos por el servidor de entrada 112 para la conexión pueden encapsularse (por ejemplo según el UDP) y enviarse directamente al módulo de equilibrador de carga local 132 en el nodo servidor 130 como paquetes de datos encapsulados. El módulo de equilibrador de carga 132 decapsula los paquetes de datos y envía los paquetes TCP al servidor 134 en el nodo servidor 130, por ejemplo, inyectando los paquetes TCP en un bastidor TCP del núcleo. Los paquetes salientes del servidor 134 son interceptados por el módulo de equilibrador de carga 132 en el nodo servidor 130, encapsulados (por ejemplo según UDP) y reenviados a un nodo de equilibrador de carga arbitrario 110 que el módulo de equilibrador de carga 132 selecciona aleatoriamente como servidor de salida 114 para esta conexión. El servidor de salida 114 decapsula los paquetes y envía los paquetes decapsulados al cliente 116. La función de salida del nodo de equilibrador de carga seleccionado 110 no tiene estado, por lo que puede seleccionarse un nodo de equilibrador de carga diferente 110 como servidor de salida 114 para la conexión en el caso de fallo del nodo de equilibrador de carga 110 que sirve como servidor de salida. Sin embargo, generalmente se utiliza el mismo nodo de equilibrador de carga 110 como servidor de salida 114 durante la duración de la conexión para reducir o eliminar el reordenamiento de los paquetes salientes.

Haciendo referencia a la figura 10G, en al menos algunas realizaciones, si el módulo de equilibrador de carga 132A en un nodo servidor 130A que se selecciona por el rastreador de flujo primario 116A (véase la figura 10C) determina que está sobrecargado, tiene la opción de rechazar el mensaje SYN fabricado recibido del rastreador de flujo secundario 116B (véase la figura 10C). En al menos algunas realizaciones, el mensaje SYN fabricado incluye un valor o contador de tiempo de vida (TTL) que permite un número máximo de rechazos. En al menos algunas realizaciones, si este valor TTL llega a cero, el módulo de equilibrador de carga 132A puede aceptar la conexión o dejar caer la conexión a la carga desprendida. Si el módulo de equilibrador de carga 132A decide rechazar la conexión, disminuye el valor TTL y envía un mensaje de rechazo al rastreador de flujo secundario 116B. El rastreador de flujo secundario 116B restablece la asignación de CP y envía un mensaje de liberación al rastreador de flujo primario 116A para hacer lo mismo. El rastreador de flujo primario 116A elige un nuevo módulo de equilibrador de carga 132B en otro nodo servidor 130B y envía un nuevo mensaje de destino de nuevo al rastreador de flujo secundario 116B, que envía un nuevo mensaje SYN fabricado al módulo de equilibrador de carga recién elegido 132B. Ha de apreciarse que las caídas de paquetes pueden hacer que esta secuencia no se complete; sin embargo, una retransmisión desde el cliente 160 puede desencadenar el proceso de selección de módulo de equilibrador de carga de nuevo en el rastreador de flujo primario 116A, que puede, pero no necesariamente, elegir el mismo módulo de equilibrador de carga 132 para la conexión si no ha aprendido sobre el anterior rechazo del paquete SYN fabricado.

En al menos algunas realizaciones, el contador TTL puede utilizarse para evitar el envío continuo de peticiones de conexión a los nodos servidor 130, lo cual puede ocurrir, por ejemplo, si todos los nodos servidor 130 están ocupados. En al menos algunas realizaciones, cada vez que un módulo de equilibrador de carga 132 rechaza una petición de conexión en nombre de un nodo servidor respectivo 130, el módulo de equilibrador de carga 132 disminuye el contador TTL. Los nodos de rastreador de flujo 116 pueden controlar el contador TTL y, mientras el contador TTL no sea cero (o esté por encima de un cierto umbral especificado), puede seleccionar otro nodo servidor 130 y volver a intentarlo. Si el contador TTL llega a cero (o alcanza el umbral especificado), la solicitud de conexión se elimina y no se hacen más intentos por los nodos de rastreador de flujo 116 para enviar una solicitud de conexión a uno seleccionado de los nodos servidor 130 para esa conexión. En al menos algunas realizaciones, se puede enviar un mensaje de error al cliente respectivo 160.

En al menos algunas realizaciones, el sistema de equilibrador de carga distribuido soporta múltiples direcciones IP públicas. Como tal, es posible que un cliente 160 pueda iniciar dos conexiones TCP desde el mismo número de

puerto de cliente a dos direcciones IP públicas diferentes. Estas conexiones TCP son distintas desde el punto de vista del cliente 160, pero internamente el equilibrador de carga distribuido puede asignar las conexiones al mismo nodo servidor 130, lo que daría lugar a una colisión. En al menos algunas realizaciones, para detectar y manejar posibles colisiones, el módulo de equilibrador de carga 132, tras recibir el paquete SYN fabricado desde el rastreador de flujo secundario 116B como se muestra en las figuras 10C y 10D, puede comparar la información de dirección con sus conexiones activas y, si esta conexión causa una colisión, rechazar la solicitud de conexión como se muestra en la figura 10G.

Manejo de fallos y adiciones de nodos de equilibrador de carga

En muchos equilibradores de carga convencionales, se pierden algunas o todas las conexiones existentes en el caso de un fallo del equilibrador de carga. En al menos algunas realizaciones, en el caso de fallo de un único nodo de equilibrador de carga 110, el sistema de equilibrado de carga distribuido puede mantener al menos algunas de las conexiones establecidas de manera que los clientes y servidores puedan continuar intercambiando paquetes a través de las conexiones hasta que las conexiones se completen normalmente. Además, el sistema de equilibrado de carga distribuido puede continuar dando servicio a las conexiones que estaban en proceso de establecerse en el momento del fallo.

En al menos algunas realizaciones del sistema de equilibrado de carga distribuido, se puede implementar un protocolo de recuperación de fallos que puede recuperar las conexiones de cliente existentes en el caso de un fallo único del nodo de equilibrador de carga 110. Sin embargo, los múltiples fallos del nodo de equilibrador de carga 110 pueden causar la pérdida de conexiones del cliente. En al menos algunas realizaciones, las retransmisiones TCP entre un cliente 160 y un servidor 134 pueden usarse como un medio de recuperación después de un fallo del nodo del equilibrador de carga 110.

Además de fallos potenciales del nodo del equilibrador de carga 110, se pueden añadir nuevos nodos de equilibradores de carga 110 al sistema de equilibrador de carga distribuido. Estos nuevos nodos 110 pueden añadirse a la capa de equilibrador de carga y, por lo tanto, al anillo hash consistente, y las funciones del nodo de equilibrador de carga 110 con respecto a las conexiones de cliente existentes pueden ajustarse de acuerdo con el cambio, según sea necesario.

Manejo de fallos y adiciones de nodos de rastreador de flujo

En al menos algunas realizaciones, a medida que se establece cada conexión (véanse, por ejemplo, las figuras 10A a 10G), la información de estado de conexión se hace pasar a través de dos nodos de equilibradores de carga 110, denominados rastreadores de flujo primario y secundario, que pueden determinarse usando un algoritmo de hash consistente que, por ejemplo, usa la tupla (IP de cliente:puerto, IP pública:puerto) como entrada de función de hash. En el caso de un fallo del nodo del equilibrador de carga único 110, al menos uno de los nodos de equilibrador de carga supervivientes 110 puede continuar siendo asignado a través de la función de hash consistente y puede contener la información de estado necesaria para una conexión a paquetes directos al nodo servidor seleccionado 130 para una conexión. Además, en el caso de una adición de un nodo de equilibrador de carga 110 al anillo hash consistente, la información de estado para conexiones puede actualizarse para los rastreadores de flujo apropiados.

Las figuras 11A a 11D ilustran el manejo de eventos que realizan la pertenencia en el anillo de hash consistente con el nodo de equilibrador de carga, de acuerdo con al menos algunas realizaciones. Estos eventos pueden incluir, pero sin limitación, añadir un nuevo nodo de rastreador de flujo primario, añadir un nuevo nodo de rastreador de flujo secundario, un fallo de un nodo de rastreador de flujo primario y un fallo de un nodo de rastreador de flujo secundario.

La figura 11A ilustra el manejo de la adición de un nuevo nodo de rastreador de flujo primario al anillo hash consistente. La fila superior de la figura 11A muestra el rastreador de flujo 116A como el rastreador de flujo primario para una o más conexiones de cliente y el nodo de rastreador de flujo 116B como el rastreador de flujo secundario para la misma conexión o conexiones. En la fila inferior de la figura 11A, se ha añadido un nuevo nodo de rastreador de flujo 116C, y se convierte en el rastreador de flujo primario para la conexión o conexiones de cliente. El nodo de rastreo de flujo 116A, anteriormente el rastreador de flujo primario, se convierte en el rastreador de flujo secundario, mientras que el nodo de rastreador de flujo 116B, anteriormente el rastreador de flujo secundario, se convierte en un rastreador de flujo posterior en el anillo hash consistente. La información de estado para la conexión o conexiones de cliente que se mantuvo por los rastreadores de flujo 116A y 116B puede proporcionarse al nuevo rastreador de flujo primario 116C. Además, el rastreador de flujo 116B puede "olvidar" sus conexiones rastreadas anteriormente en el papel de rastreador de flujo secundario.

La figura 11B ilustra el manejo de la adición de un nuevo nodo de rastreador de flujo secundario al anillo hash consistente. La fila superior de la figura 11B muestra el rastreador de flujo 116A como el rastreador de flujo primario para una o más conexiones de cliente y el nodo de rastreador de flujo 116B como el rastreador de flujo secundario para la misma conexión o conexiones. En la fila inferior de la figura 11B, se ha añadido un nuevo nodo de rastreador de flujo 116C, y se convierte en el rastreador de flujo secundario para la conexión o conexiones de cliente. El nodo de rastreador de flujo 116A permanece como el rastreador de flujo primario para la conexión o conexiones, mientras que el nodo de rastreador de flujo 116B, anteriormente el rastreador de flujo secundario, se convierte en un rastreador de flujo posterior en el anillo hash consistente. La información de estado para la conexión o conexiones de cliente que se mantuvo por los rastreadores de flujo 116A y 116B puede proporcionarse al nuevo rastreador de flujo secundario 116C. Además, el rastreador de flujo 116B puede "olvidar" sus conexiones rastreadas anteriormente en el papel de rastreador de flujo secundario.

La figura 11C ilustra el manejo del fallo de un nodo de rastreador de flujo primario en el anillo hash consistente. La fila superior de la figura 11C muestra el rastreador de flujo 116A como el rastreador de flujo primario para una o más conexiones de cliente, el nodo de rastreador de flujo 116B como el rastreador de flujo secundario para la misma conexión o conexiones, y el nodo de rastreador de flujo 116C como el siguiente rastreador de flujo en el anillo hash consistente. En la fila inferior de la figura 11C, el nodo de rastreador de flujo primario 116A ha fallado. El nodo de rastreador de flujo 116B se convierte en el rastreador de flujo primario para la conexión o conexiones, mientras que el nodo de rastreador de flujo 116C se convierte en el rastreador de flujo secundario para la conexión o conexiones. La información de estado para la conexión o conexiones de cliente es mantenida por el rastreador de flujo 116B y puede proporcionarse al nuevo rastreador de flujo secundario 116C.

La figura 11D ilustra el manejo del fallo de un nodo de rastreador de flujo secundario en el anillo hash consistente. La fila superior de la figura 11D muestra el rastreador de flujo 116A como el rastreador de flujo primario para una o más conexiones de cliente, el nodo de rastreador de flujo 116B como el rastreador de flujo secundario para la misma conexión o conexiones, y el nodo de rastreador de flujo 116C como el siguiente rastreador de flujo en el anillo hash consistente. En la fila inferior de la figura 11D, el nodo de rastreador de flujo secundario 116B ha fallado. El nodo de rastreador de flujo 116A permanece como el rastreador de flujo primario para la conexión o conexiones, mientras que el nodo de rastreador de flujo 116C se convierte en el rastreador de flujo secundario para la conexión o conexiones. La información de estado para la conexión o conexiones de cliente es mantenida por el rastreador de flujo 116B y puede proporcionarse al nuevo rastreador de flujo secundario 116C.

En al menos algunas realizaciones, los módulos de equilibrador de carga 132 en los nodos servidor 130 realizan el anuncio de conexión a los nodos de equilibrador de carga 110. En al menos algunas realizaciones, la publicación de conexiones empuja periódica (por ejemplo, una vez por segundo) o aperiódicamente la información de estado de conexión actual desde los nodos servidor 130 a los nodos de equilibrador de carga 110 que dan servicio como nodos de seguimiento de flujo y nodos de entrada, que actúa para actualizar o restaurar las asignaciones de conexión tanto a los nodos de rastreador de flujo primario como secundario para las conexiones. En al menos algunas realizaciones, un módulo de equilibrador de carga 132 puede detectar un cambio de pertenencia de rastreador de flujo, por ejemplo, como se ilustra en las figuras 11A a 11D. En respuesta, el módulo de equilibrador de carga 132 puede realizar una publicación de conexión para rellenar la información de estado para las conexiones en los nodos de rastreador de flujo primario y secundario, que pueden haber cambiado para las conexiones cuando se cambió la pertenencia. Cabe apreciar que la publicación de conexiones puede permitir que al menos algunas conexiones establecidas se recuperen en el caso de fallos de múltiples nodos del equilibrador de carga.

Flujo de mensajes relacionados con fallos

En al menos algunas realizaciones, el protocolo entre los nodos de rastreador de flujo primario y secundario puede incluir una funcionalidad de corrección o sincronización. Por ejemplo, haciendo referencia a la figura 11A, cuando un nuevo nodo de rastreador de flujo primario 116C se une al anillo hash consistente, el nuevo nodo 116C puede reivindicar el espacio de claves de hash consistente para cierto número de conexiones (~1/N) y comenzar a recibir tráfico relacionados con estas conexiones desde el enrutador periférico 104. Sin embargo, el nuevo nodo de rastreador de flujo primario 116C no tiene ningún estado almacenado para las conexiones, por lo que puede operar en cada paquete como si fuera el primer paquete recibido del cliente 160. El rastreador de flujo primario es responsable de generar números de secuencia TCP de servidor en respuesta a paquetes SYN (véase, por ejemplo, la figura 10A) y de seleccionar nodos servidor 130 en respuesta al primer paquete ACK de un cliente 160 (véase, por ejemplo, la figura 1), y estos valores generados pueden estar en desacuerdo con los valores elegidos por el anterior rastreador de flujo primario (nodo de rastreador de flujo 116A en la figura 11A). Sin embargo, en al menos algunas realizaciones, el algoritmo de hash consistente asigna al rastreador de flujo primario anterior (nodo de rastreador de

flujo 116A en la figura 11A) al rol de rastreador de flujo secundario, y este rastreador de flujo todavía retiene el estado previamente almacenado para las conexiones. Por lo tanto, en al menos en algunas realizaciones, cuando el rastreador de flujo secundario (nodo de rastreador de flujo 116A en la figura 11A) detecta una discrepancia en la información recibida del rastreador de flujo primario 116C, puede enviar mensajes de actualización de nuevo al rastreador de flujo primario 116C para llevar los dos nodos de equilibradores de carga 110 que sirven como rastreadores de flujo para las conexiones en sincronización. Pueden usarse métodos similares para sincronizar los rastreadores de flujo después de otros cambios en la pertenencia al anillo hash consistente.

Detalles del módulo del equilibrador de carga

En al menos algunas realizaciones, el módulo de equilibrador de carga 132 es un componente del sistema de equilibrador de carga distribuido que reside en cada uno de los nodos servidor 130. Las funciones del nodo de equilibrador de carga 132 incluyen, pero sin limitación, decapsular paquetes recibidos de los nodos de equilibrador de carga 110 y enviar los paquetes decapsulados al servidor 134 en el nodo servidor 130, y encapsular los paquetes salientes del servidor 134 y enviar los paquetes encapsulados a un nodo de equilibrador de carga 110.

En al menos algunas realizaciones, los paquetes entrantes a los módulos de equilibrador de carga 132 en los nodos servidor 130 desde los nodos de equilibradores de carga 110 que sirven como servidores de entrada 112 son paquetes de protocolo sin estado (por ejemplo, UDP) que encapsulan los paquetes de datos de cliente reales. Cada paquete de datos de cliente encapsulado tiene la relación IP de cliente original:puerto de un cliente respectivo 160 como la dirección de origen y la IP pública:puerto del servidor 134 como la dirección de destino. Los módulos de equilibrador de carga 132 separan la encapsulación de los paquetes de datos de cliente y envían los paquetes a los respectivos servidores 134 en los nodos servidor 130, por ejemplo redirigiendo los paquetes a un flujo TCP de máquina local.

En al menos algunas realizaciones, los paquetes salientes desde los servidores 134 a los nodos de equilibradores de carga 110 que sirven como servidores de salida 114 son paquetes de protocolo sin estado (por ejemplo, UDP) que encapsulan los paquetes IP salientes. Los módulos de equilibrador de carga 132 encapsulan los paquetes IP salientes y envían los paquetes encapsulados a los servidores de salida 114 a través del tejido 120. Cada paquete IP saliente encapsulado tiene la IP pública:puerto del servidor 134 como la dirección de origen y la IP de cliente:puerto de un cliente respectivo 160 como la dirección de destino.

Funcionalidad del módulo de equilibrador de carga

En al menos algunas realizaciones, las funciones del módulo de equilibrador de carga 132 en un nodo servidor 130 pueden incluir uno o más de, pero sin limitación:

- Terminación de los túneles UDP del nodo o nodos del equilibrador de carga 110, por ejemplo, desde el servidor de entrada 112 que maneja una conexión a un cliente 160. Esto incluye separar la encapsulación UDP de los paquetes de datos de cliente entrantes recibidos de los servidores de entrada 112.
- Selección de un servidor de salida 114 para recibir tráfico saliente para una conexión.
- Interceptación de paquetes IP salientes en una conexión al respectivo servidor 134, encapsulación de los paquetes IP salientes para la conexión, y envío de los paquetes encapsulados al servidor de salida 114.
- Manejar el número de secuencia en los paquetes entrantes y salientes de manera que el número de secuencia se alinea con el número de secuencia generado por los nodos de rastreador de flujo 116 cuando los nodos de rastreador de flujo 116 envían un SYN/ACK al cliente 160.
- Tomar la decisión de aceptar o rechazar una conexión para el servidor respectivo 134, por ejemplo, basándose en una o más métricas que indiquen la carga actual del servidor 134 respectivo.
- Detención y rechazo de conexiones desde la misma dirección IP de cliente:puerto al servidor respectivo 134 si hay una conexión activa para esa dirección IP de cliente:puerto para evitar colisiones.
- Seguimiento de conexiones y publicación de conexiones.

Información de configuración del módulo de equilibrador de carga

En al menos algunas realizaciones, cada módulo de equilibrador de carga 132 puede adquirir y almacenar localmente uno o más de, pero sin limitación, los siguientes conjuntos de información para su configuración: un conjunto de extremos del nodo de equilibrador de carga 110; un conjunto de direcciones IP públicas válidas que debe dar servicio; y el número o números de puerto en los que el servidor respectivo 134 acepta conexiones entrantes. En al menos algunas realizaciones, esta información puede adquirirse o actualizarse accediendo o consultando un componente de servicio de configuración 122 del sistema de equilibrador de carga distribuido, como

se ilustra en la figura 1. Otros métodos de adquisición de la información pueden utilizarse en algunas realizaciones.

Manejo de paquetes de módulos de equilibrador de carga

A continuación se describen las operaciones del módulo de equilibrador de carga 132 para tráfico entrante y tráfico saliente según al menos algunas realizaciones. En al menos algunas realizaciones, cuando un paquete de datos entrante es recibido por el módulo de equilibrador de carga 132, el paquete de datos se decapsula del paquete UDP, y la dirección de destino en el paquete TCP decapsulado se valida primero contra un conjunto de direcciones IP públicas válidas configuradas. Si no hay correspondencia, se descarta o se omite el paquete. En al menos algunas realizaciones, el módulo de equilibrador de carga 132 puede ajustar el número de secuencia en el encabezado TCP mediante una delta constante de manera que el número de secuencia coincida con el número de secuencia elegido al azar generado por los nodos de rastreador de flujo 116 que enviaron el paquete SYN/ACK a el cliente 160. El módulo de equilibrador de carga 132 registra la asignación desde el extremo [Client:Public] al extremo [Client:Server] como un estado interno.

En al menos algunas realizaciones, para los paquetes TCP salientes del servidor 134, el módulo de equilibrador de carga 132 comprueba primero su estado interno para determinar si el paquete es para una conexión activa que el módulo de equilibrador de carga está gestionando. Si no es así, el módulo de equilibrador de carga 132 simplemente pasa el paquete a través. Si es así, el módulo de equilibrador de carga 132 encapsula el paquete TCP saliente, por ejemplo de acuerdo con UDP, y reenvía el paquete encapsulado a un nodo de equilibrador de carga 110 que se seleccionó como servidor de salida 114 para esta conexión. En al menos algunas realizaciones, el módulo de equilibrador de carga 134 puede ajustar el número de secuencia TCP en el paquete TCP saliente por una delta constante de manera que se alinea con el número de secuencia generado por los nodos de rastreador de flujo 116 que enviaron el paquete SYN/ACK al cliente 160.

Rastreo de la conexión

En al menos algunas realizaciones, el módulo de equilibrador de carga 132 en cada nodo servidor 130 gestiona una tabla de hash que contiene detalles de conexión para cada conexión cliente activa al servidor respectivo 134. En al menos algunas realizaciones, la clave para la tabla de hash es la tupla (clientIp:port, publicIp:port). En al menos algunas realizaciones, el estado de conexión para cada conexión de cliente puede incluir uno o más de, pero sin limitación:

- La IP del cliente: Puerto
- La IP pública: Puerto
- El número de secuencia TCP del servidor inicial proporcionado por los nodos del rastreador de flujo 116.
- El número de secuencia TCP del servidor delta.
- La dirección IP original del rastreador de flujo primario.
- La dirección IP original del rastreador de flujo secundario.
- La dirección IP del último servidor de entrada detectado 112.
- Una hora de caducidad para esta entrada
- Índices de uso menos reciente(LRU)/colisión.

En al menos algunas realizaciones, cada módulo de equilibrador de carga 132 genera periódicamente mensajes de publicación de conexión a los nodos de rastreador de flujo primario y secundario para todas las conexiones de cliente activas. En al menos algunas realizaciones, el contenido de/proc/net/tcp se explora y se cruza con las conexiones activas en la tabla de hash del módulo de equilibrador de carga de manera que seguirán siendo publicadas en los nodos de rastreador de flujo hasta que el núcleo de Linux deje de rastrear la conexión. La publicación de conexiones se analizará más detalladamente más adelante en este documento.

Gestión del número de secuencia

Como se ha descrito previamente, en al menos algunas realizaciones, los nodos de equilibrador de carga 110 generan paquetes SYN/ACK en respuesta a los paquetes SYN de cliente 160 en nombre del servidor 134. Sólo después de que el cliente 160 envíe un paquete ACK (el intercambio de tres vías TCP) si un módulo de equilibrador de carga 110 envía cualquier dato a un módulo de equilibrador de carga 132 en un nodo servidor 130. Cuando se ordena en primer lugar al módulo de equilibrador de carga 132 que establezca una conexión cliente, el módulo de equilibrador de carga 132 fabrica localmente un paquete SYN para iniciar una conexión TCP con el servidor 134 en el nodo servidor 130, e intercepta el paquete SYN/ACK correspondiente del servidor 134. Típicamente, el servidor 134 (por ejemplo, el núcleo de Linux en el nodo servidor 130) selecciona un número de secuencia TCP

completamente diferente al que el cliente recibió en el paquete SYN/ACK de los nodos de equilibradores de carga 110. Por lo tanto, en al menos en algunas realizaciones, el módulo de equilibrador de carga 132 puede corregir los números de secuencia en todos los paquetes de la conexión TCP entre el cliente 160 y el servidor 134. En al menos algunas realizaciones, el módulo de equilibrador de carga 132 calcula la diferencia entre el número de secuencia generado por los nodos de equilibradores de carga 110 y el número de secuencia generado por el servidor 134 y almacena la diferencia como un valor delta en la entrada de la tabla de hash para la conexión TCP. Cuando los paquetes de datos entrantes llegan desde el cliente 160 en la conexión, el encabezado TCP contendrá números de acuse que no se alinearán con el número de secuencia utilizado por el servidor 134, por lo que el módulo de equilibrador de carga 132 resta el valor delta (por ejemplo, usando el complemento de los dos) del valor del número de secuencia en el encabezado TCP. El módulo de equilibrador de carga también añade el valor delta al número de secuencia en paquetes salientes del servidor 134 al cliente 130 en la conexión.

Comprobación de estado en el sistema equilibrador de carga distribuido

En al menos algunas realizaciones del sistema equilibrador de carga distribuido, cada nodo de equilibrador de carga 110 requiere una vista consistente de los miembros con estado la implementación del equilibrador de carga (es decir, de los nodos de equilibradores de carga con estado 110 y nodos servidor 130) por al menos las siguientes razones.

• Equilibrado de carga - Los nodos del equilibrador de carga 110 necesitan detectar fallos en el nodo servidor 130 y converger en un conjunto de nodos servidor con estado 130 que pueden aceptar tráfico del cliente.
• Gestión de estado distribuida - El equilibrador de carga es un sistema distribuido con el estado compartido/replicado en varios nodos de equilibrador de carga 110 (por ejemplo, de acuerdo con un mecanismo de hashing consistente). Para manejar adecuadamente el tráfico de cliente, cada nodo de equilibrador de carga 110 necesita tener una vista eventualmente consistente de los nodos miembros con estado 110 en la implementación del equilibrador de carga.

Para conseguir esto, al menos algunas realizaciones del sistema de equilibrador de carga distribuido pueden implementar realizaciones de un protocolo de comprobación de estado que supervisa los nodos en la implementación de equilibrador de carga y detecta nodos sin estado tan pronto como sea posible. El protocolo de comprobación de estado puede propagar información de estado entre los nodos en la implementación del equilibrador de carga, y puede proporcionar métodos que permiten a los nodos converger en un conjunto de nodos con estado. Además, el protocolo de comprobación de estado puede proporcionar mecanismos para indicar nodos con estado/sin estado y cambios de estado en la implementación del equilibrador de carga.

En al menos algunas realizaciones, el protocolo de comprobación de estado puede estar basado en uno o más de, pero sin limitación, las siguientes suposiciones:

• Todos los nodos de la implementación del equilibrador de carga se conocen. (Es decir, el protocolo de comprobación de estado puede no realizar el descubrimiento).
• Todas los fallos de nodo son fallo-parada.
• Todos los mensajes entre nodos son mensajes de protocolo sin estado (por ejemplo, UDP), y los mensajes pueden eliminarse, retrasarse, duplicarse o dañarse. No hay garantías en la entrega de mensajes.

En al menos algunas realizaciones, un nodo en una implementación de equilibrador de carga (por ejemplo, un nodo de equilibrador de carga 110 o nodo servidor 130) puede considerarse con estado bajo las siguientes condiciones:

• Todos los componentes internos del nodo están listos (listos para manejar el tráfico del cliente).
• Los enlaces de red entrante/saliente del nodo tienen estado (al menos para los controladores de interfaz de red (NIC) en los que fluye el tráfico del cliente).

La figura 12 es un diagrama de flujo de alto nivel de un método de comprobación de salud que puede realizarse por cada nodo de equilibrador de carga de acuerdo con un intervalo de comprobación del estado, de acuerdo con al menos algunas realizaciones. Como se indica en 1000, en cada intervalo de equilibrador de carga, por ejemplo cada 100 milisegundos, cada nodo de equilibrador de carga 110 (LB) puede comprobar el estado en al menos otro nodo LB 110 y al menos un nodo servidor 130. Como se indica en 1002, el nodo de equilibrador de carga 110 puede actualizar su información de estado almacenada localmente de acuerdo con las comprobaciones de estado. Como se indica en 1004, el nodo de equilibrador de carga 110 puede entonces seleccionar al azar al menos otro nodo de equilibrador de carga 110 y enviar su información de estado al nodo o nodos 110 de equilibrador de carga seleccionados. En al menos algunas realizaciones, el nodo 110 también puede enviar una lista de nodos de

equilibrador de carga con estado 110 a uno o más nodos servidor 130, por ejemplo, al mismo nodo o nodos servidor 130 que son comprobados en estado por el nodo 110. Los elementos de la figura 12 se explican con más detalle en el siguiente análisis.

En al menos algunas realizaciones del protocolo de comprobación de estado, un nodo de equilibrador de carga 110 no afirma su propio estado a los otros nodos de equilibradores de carga 110. En su lugar, uno o más nodos de equilibradores de carga 110 pueden comprobar el estado del nodo 110. Por ejemplo, en al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede seleccionar de forma periódica o aperiódicamente al azar uno o más nodos 110 para comprobar el estado. Como otro ejemplo, en al menos algunas realizaciones, uno o más nodos de equilibradores de carga 110, por ejemplo, los dos vecinos más próximos de un nodo de equilibrador de carga dado 110 en una lista ordenada de nodos 110 tales como un anillo hash consistente, pueden comprobar cada uno periódicamente o aperiódicamente el estado del nodo dado 110. En al menos algunas realizaciones, la comprobación de estado de un nodo 110 puede incluir el uso de pings de estado enviados a los NIC 1114 en el nodo 110 como se ilustra en la figura 23. En al menos algunas realizaciones, si un primer nodo 110 determina que un segundo nodo 110 tiene estado a través de una comprobación de estado, el primer nodo 110 puede actualizar (por ejemplo, aumentar) el contador de señales para el segundo nodo 110 almacenado en información de estado local para los nodos de equilibradores de carga 110. El primer nodo 110 periódicamente o de manera aperiódica envía su información de estado local a uno o más nodos de equilibradores de carga 110 en la implementación del equilibrador de carga, los cuales pueden actualizar su propia información de estado local por consiguiente (por ejemplo, aumentando el contador de señales para el segundo nodo) y enviar su información de estado local actualizada a uno o más nodos diferentes 110. La información de señal del segundo nodo 110 puede así propagarse a los otros nodos 110 en la implementación del equilibrador de carga. Siempre y cuando el segundo nodo 110 tenga estado, todos los demás nodos 110 que son alcanzables desde el segundo nodo 110 deberían ver así que el contador de señales del segundo nodo 110 se aumenta de forma consistente, por ejemplo, una vez por segundo o una vez cada diez segundos. Si el segundo nodo 110 es detectado como sin estado por el nodo o los nodos 110 que comprueban su estado, ninguna señal para el nodo 110 se envía por los nodos de comprobación de estado 110 y, después de algún umbral de tiempo, los otros nodos 110 en la implementación de equilibrador de carga 110 consideran que el nodo 110 en cuestión no tiene estado, o está inactivo.

En al menos algunas realizaciones, un nodo de equilibrador de carga 110 puede comprobar uno o más aspectos de su propio estado interno y, si el nodo 110 detecta que no tiene estado por alguna razón, el nodo 110 puede dejar de responder a los pings de estado desde otros nodos 110 que comprueban su estado. Por lo tanto, los nodos 110 que comprueban el estado del nodo 110 sin estado pueden considerar el nodo 110 como sin estado, y pueden no propagar aumentos de señales en nombre del nodo 110.

Detalles del protocolo de comprobación de estado

En al menos algunas realizaciones, el protocolo de comprobación de estado puede aprovechar una técnica de contador de señales y una tecnología de protocolo de intercambio. El protocolo de comprobación de estado puede considerarse con dos partes principales - la comprobación del estado y la detección de intercambios/fallos.

Comprobación de estado - Cada nodo de equilibrador de carga 110 en la implementación del equilibrador de carga puede comprobar de forma periódica o aperiódica el estado de uno o más nodos 110 en la implementación. Los métodos mediante los cuales se determinan uno o más nodos se analizan más adelante. Una idea central de la comprobación de estado es que si un estado del nodo 110 comprueba otro nodo 110 y determina que el otro nodo 110 tiene estado, el nodo de comprobación 110 afirma que el otro nodo 110 tiene estado aumentando y propagando un contador de señales para el otro nodo 110. En otras palabras, los nodos 110 no afirman su propio estado a los otros nodos; en su lugar, uno o más nodos 110 comprueban y aseguran el estado de cada nodo 110 en la implementación del equilibrador de carga.

Detección de intercambio/fallos - En al menos algunas realizaciones, el protocolo de comprobación de estado puede aprovechar un protocolo de intercambio para propagar la información de estado del nodo de equilibrador de carga 110 entre los nodos de equilibrador de carga de miembros 110 en la implementación de equilibrador de carga. El protocolo de intercambio converge rápidamente, y proporciona garantías de consistencia eventuales que son suficientes para los propósitos del sistema de equilibrado de carga distribuido. En al menos algunas realizaciones, utilizando el protocolo de intercambios, cada nodo de equilibrador de carga 110 mantiene un contador de señales para cada otro nodo 110 en la implementación del equilibrador de carga, por ejemplo en una lista de señales. Cada nodo de equilibrador de carga 110 realiza periódica o aperiódicamente una verificación de estado de al menos otro nodo de equilibrador de carga 110 como se ha descrito anteriormente, y aumenta el contador de señales para un nodo 110 tras determinar a través de la comprobación de estado que el nódulo comprobado 110 tiene estado. En al

menos algunas realizaciones, cada nodo de equilibrador de carga 110 selecciona de forma periódica o aperiódica al azar al menos otro nodo 110 en la implementación del equilibrador de carga a la que envía su lista de señales actual. Tras la recepción de una lista de señales desde otro nodo 110, un nodo de equilibrador de carga 110 fusiona la información de señales en la lista recibida con su propia lista de señales determinando el contador de señales máximo para cada nodo 110 en las dos listas (las listas recibidas y su propia lista) y usando el contador de señales máximas determinado en su propia lista de señales. A su vez, esta lista de señales se envía a otro nodo seleccionado aleatoriamente 110, que actualiza su propia lista de señales en consecuencia, etc. Usando esta técnica, la información de señales para cada nodo con estado 110 se propaga eventualmente (por ejemplo, en pocos segundos) a todos los otros nodos de equilibradores de carga 110 en la implementación del equilibrador de carga. Mientras el contador de señales sigue aumentando para un nodo de equilibrador de carga dado 110, se considera que tiene estado por los otros nodos 110. Si un contador de señales del nodo del equilibrador de carga 110 no se aumenta durante un período especificado por el método de comprobación de estado y de intercambio, entonces otros nodos de equilibradores de carga 110 pueden converger en el nodo de equilibrador de carga 110 que se considera sin estado.

Comprobación de estado de los nodos de equilibradores de carga

A continuación, se describe un método para comprobar el estado de un nodo de equilibrador de carga 110 que puede realizarse por otro nodo de equilibrador de carga 110, de acuerdo con al menos algunas realizaciones. Con referencia a la figura 23, en al menos algunas realizaciones, un nodo de equilibrador de carga 110 puede considerarse con estado si se determina una o más de las siguientes condiciones para el nodo 110:

- Los hilos de procesador (por ejemplo, hilos de código de procesamiento de paquete de núcleo 1108) del nodo 110 están en estado listo (interno).
- El nodo 110 conoce la dirección IP y/o la dirección MAC (interna) del enrutador periférico 104.
- Todos los hilos y/o controladores de protocolo del nodo 110 están en estado listo (interno).
- Los enlaces entrantes y salientes desde el lado norte (enrutador periférico 104/red de frontera) y desde el lado sur (servidores 130/red de producción) están activos (externos).
- El nodo 110 puede recibir y enviar paquetes a través de los controladores de interfaz de red (NIC) utilizados en la implementación del equilibrador de carga. Por ejemplo, en una realización del nodo del equilibrador de carga ejemplar 110 como se muestra en la figura 23, el nodo 110 debería recibir y enviar con éxito paquetes a través del NIC orientado hacia el norte 1114A y el NIC orientado hacia el sur 1114B.

Si una o más de estas condiciones de estado no se sostienen para un nodo dado 110, el nodo 110 puede considerarse sin estado. Obsérvese que, en algunas realizaciones, un nodo 110 sólo se considera con estado si todas las condiciones anteriores se mantienen para el nodo 110.

En al menos algunas realizaciones, además de las condiciones de estado anteriores, un tercer NIC, mostrado en la figura 23 como NIC 1114C, en cada nodo de equilibrador de carga 110 que se puede utilizar, por ejemplo, para comunicaciones de plano de control también puede comprobarse por un nodo de comprobación de estado 110 enviando paquetes a y recibiendo paquetes del NIC y, si la comprobación del tercer NIC falla, el nodo 110 que se comprueba puede considerarse sin estado.

La figura 13 ilustra un método ejemplar para comprobar el estado de un nodo de equilibrador de carga desde otro nodo de equilibrador de carga, de acuerdo con al menos algunas realizaciones. En este ejemplo, el nodo de equilibrador de carga 110A es el nodo de equilibrador de carga de comprobación de estado 110B. Cada nodo 110A y 110B tiene un NIC orientado hacia el norte (NIC 1114A en la figura 23) y un NIC orientado hacia el sur (NIC 1114B en la figura 23). En 1, el nodo 110A envía un paquete (por ejemplo, un paquete de ping) desde su NIC orientado hacia el norte al NIC orientado hacia el norte del nodo 110B a través del enrutador periférico 104. El nodo 110B recibe el paquete en su NIC hacia el norte, y en 2 envía una respuesta desde su NIC orientado hacia el norte al NIC orientado hacia el norte del nodo 110A a través del tejido 120, siempre que se cumplan las condiciones dadas en la lista anterior. Después de recibir la respuesta en su NIC orientado al norte, en 3, el nodo 110A envía un paquete (por ejemplo, un paquete de ping) desde su NIC orientado al sur al NIC orientado al sur del nodo 110B a través del tejido 120. El nodo 110B recibe el paquete en su NIC hacia el sur, y en 4 envía una respuesta desde su NIC orientado hacia el sur al NIC orientado hacia el sur del nodo 110A a través del enrutador periférico 104, siempre que se cumplan las condiciones dadas en la lista anterior. Tras recibir la respuesta en su NIC orientado hacia el sur, el nodo 110A considera que el nodo 110B tiene estado y aumenta el contador de señales local del nodo 110B, que puede entonces propagarse a otros nodos 110 de acuerdo con un protocolo de intercambio tal como se ha descrito previamente.

Como alternativa a lo anterior, en algunas realizaciones, el nodo de equilibrador de carga 110B puede responder al primer mensaje de ping, recibido en su NIC orientado al norte, a través de su NIC orientado al sur al NIC orientado al sur del nodo 110A, y responder al segundo mensaje de ping, recibido en su NIC orientado al sur, a través de su NIC orientado al norte con respecto al NIC orientado al norte del nodo 110A.

Además, en algunas realizaciones, el nodo 110A también puede comprobar el estado de un tercer NIC del nodo 110B que se usa para las comunicaciones del plano de control (mostrado como NIC 1114C en la figura 23) haciendo ping en el tercer NIC del nodo 110B desde su propio tercer NIC y recibir una respuesta al mensaje de ping en su tercer NIC del tercer NIC del nodo 110B si el nodo 110B tiene estado. El mensaje de ping y la respuesta pueden pasar a través de uno o más dispositivos de plano de control 170, por ejemplo, un conmutador de red

El mecanismo de comprobación de estado descrito anteriormente ejerce todos los enlaces entrantes y salientes y las rutas de datos del nodo 110B en todas las direcciones (norte, sur y a través del plano de control), así como todos los NIC de 110B de nodo, y también verifica el estado interno del nodo 110B según los paquetes ping atraviesan las colas internas y el envío del nodo 110B como lo haría un paquete cliente.

Asignación de responsabilidades de comprobación de estado a los nodos del equilibrador de carga

En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 en una implementación de equilibrador de carga tiene acceso a una lista (por ejemplo, una lista ordenada) de todos los otros nodos de equilibradores de carga 110 en la implementación del equilibrador de carga, por ejemplo, a través de una función de configuración y/o a través de un componente de servicio de configuración 122 como se muestra en la figura 1. En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede seleccionar aleatoriamente uno o más nodos diferentes 110 de la lista para comprobar su estado en cada intervalo de comprobación, aumentando su contador de señales si se determina con estado. Obsérvese que la lista incluye todos los nodos de equilibradores de carga 110 en la implementación del equilibrador de carga, si se consideran actualmente con estado o sin estado a través del mecanismo de comprobación de estado, y los nodos actualmente sin estado 110 pueden seleccionarse aleatoriamente de la lista, así como los nodos con estado 110. Por lo tanto, se puede determinar que un nodo actualmente sin estado 110 puede determinarse por uno o más nodos 110 que comprueban el estado del nodo 110, su contador de señales puede aumentarse y propagarse a los otros nodos 110, y el nodo sin estado 110 puede volver así a un estado con estado.

Como alternativa, en algunas realizaciones, cada nodo de equilibrador de carga 110 puede asumir la responsabilidad de comprobar el estado de uno o más nodos 110 en la lista y aumentar su contador de señales si se determina con estado. Por ejemplo, en algunas realizaciones, cada nodo 110 puede asumir la responsabilidad de otros dos nodos, por ejemplo, sus nodos vecinos más próximos "izquierdos" (o anteriores) y "derechos" (o próximos) 110 en la lista. Obsérvese que la lista puede considerarse circular y un nodo 110 en el "extremo" de la lista puede asumir la responsabilidad de la comprobación de estado de un nodo 110 al "comienzo" de la lista, y viceversa. En algunas realizaciones, los otros dos nodos 110 pueden seleccionarse de otro modo, por ejemplo, como los dos vecinos más próximos en la lista. En algunas realizaciones, cada nodo 110 puede asumir la responsabilidad de la comprobación de estado de más de dos otros nodos 110 de la lista, por ejemplo, tres o cuatro nodos diferentes 110. En al menos algunas realizaciones, si un nodo vecino 110 que está siendo verificado por un nodo 110 se determina como sin estado, entonces el nodo 110 puede asumir la responsabilidad de la comprobación de estado de al menos un nodo en la lista de que el nodo nocivo vecino 110 fue responsable de la comprobación. En al menos algunas realizaciones, además de la comprobación de estado de sus nodos vecinos 110 (por ejemplo, un nodo vecino "izquierdo" y "derecho"), cada nodo de equilibrador de carga 110 puede seleccionar también periódica o aperiódicamente al azar un nodo 110 en el anillo y realizar una comprobación de estado de ese nodo seleccionado aleatoriamente 110 y, si tiene estado, aumentar y propagar la señal del nodo aleatorio 110. En al menos algunas realizaciones, todos los otros nodos 110 en la lista ordenada se consideran para la selección aleatoria y la comprobación de estado independientemente de si el otro nodo 110 se consideraba previamente con estado o sin estado.

En al menos algunas realizaciones, cada nodo 110 realiza la comprobación de estado de uno o más nodos seleccionados aleatoriamente 110, o como alternativa, de sus nodos vecinos 110 y de un nodo seleccionado aleatoriamente, en un intervalo regular, que se puede denominar como el intervalo de comprobación de estado. Por ejemplo, en algunas realizaciones, el intervalo de señales puede ser de 100 milisegundos, aunque pueden usarse intervalos más cortos o más largos. Además, en al menos algunas realizaciones, cada nodo 110 envía o "intercambia" su lista de señales actual al menos a otro nodo seleccionado aleatoriamente 110 en un intervalo regular, que puede denominarse intervalo de intercambio. En algunas realizaciones, el intervalo de comprobación de estado y el intervalo de intercambio pueden iguales, aunque no son necesariamente los mismos.

La figura 14 ilustra gráficamente una comprobación del estado del nodo de equilibrador de carga de uno o más nodos de equilibrador de carga, de acuerdo con al menos algunas realizaciones. En este ejemplo, hay ocho nodos de equilibradores de carga 110A - 110H en la implementación del equilibrador de carga. El círculo de líneas discontinuas representa una lista ordenada de todos los nodos 110 en la implementación. En algunas realizaciones, cada nodo 110 puede seleccionar aleatoriamente uno o más nodos 110 de la lista para comprobar su estado en cada intervalo. Como alternativa, en algunas realizaciones, cada nodo de equilibrador de carga 110 puede asumir la responsabilidad de comprobar uno o más nodos particulares 110 en la lista ordenada, por ejemplo, el nodo 110A puede asumir la responsabilidad de comprobar el estado de sus dos nodos vecinos más cercanos 110B y 110H de acuerdo con la lista ordenada como se muestra en la figura 14. Además, el nodo de equilibrador de carga también puede seleccionar aleatoriamente otro nodo 110 de la lista ordenada en cada intervalo de comprobación de estado. Como se muestra en este ejemplo, el nodo 110A también ha seleccionado aleatoriamente el nodo 110 F para comprobar el estado. En el intervalo de intercambio, el nodo 110A selecciona aleatoriamente algún otro nodo con estado 110, por ejemplo, el nodo 110D, y envía su lista de señales actual al otro nodo seleccionado 110, por ejemplo en un mensaje UDP. Un nodo 110, tras recibir una lista de señales del otro nodo 110, puede actualizar su propia lista de señales en consecuencia y propagar la lista de señales a uno o más nodos seleccionados aleatoriamente 110 en el siguiente intervalo de intercambio.

Comprobación de estado de los nodos servidor

Además de comprobar el estado de los nodos de equilibradores de carga 110 como se ha descrito anteriormente, las realizaciones del protocolo de comprobación de estado pueden realizar comprobación de estado de los nodos servidor 130 que incluyen los módulos de equilibrador de carga 132 y los servidores 134 en estos nodos 130. En al menos algunas realizaciones, un nodo servidor 130 puede considerarse con estado si se determina una o ambas de las siguientes condiciones para el nodo 130:

- El módulo de equilibrador de carga 132 tiene estado.
- El nodo servidor 130 responde satisfactoriamente a pings de estado (por ejemplo, pings de estado L7).

La figura 15 ilustra los nodos de equilibradores de carga que comprueban el estado de los nodos servidor, de acuerdo con al menos algunas realizaciones. En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 en una implementación de equilibrador de carga tiene acceso a una lista de todos los demás nodos de equilibradores de carga 110 en la implementación de equilibrador de carga, así como una lista de todos los nodos servidor 130 en la implementación de equilibrador de carga. La lista o listas pueden obtenerse y actualizarse, por ejemplo, a través de una función de configuración y/o a través de un componente de servicio de configuración 122 como se muestra en la figura 1. En al menos algunas realizaciones, los nodos servidor 130 pueden estar en hash consistente contra los nodos de equilibradores de carga con estado 110 para formar un anillo hash consistente como se ilustra en la figura 15. En al menos algunas realizaciones, para cada nodo servidor 130 en el anillo se comprueba el estado por dos nodos de equilibradores de carga con estado 110 en el anillo. Por ejemplo, en la figura 15, se comprueba el estado del nodo servidor 130A por los nodos de equilibrador de carga 110A y 110C. Estos dos nodos 110 pueden denominarse como el primer (nodo 110A) y el segundo (nodo 110B) nodos de comprobación de estado 110 para el nodo servidor 130 en el anillo hash consistente. Obsérvese que un nodo de equilibrador de carga con estado dado 110 puede comprobar el estado de más de un nodo servidor 130. Por ejemplo, en la figura 15, el nodo de equilibrador de carga 110A también comprueba el estado de los nodos servidor 130B y 130C. Además, un nodo de equilibrador de nodos dado 110 puede ser un primer nodo de comprobación de estado 110 para uno o más nodos servidor 130 y un segundo nodo de comprobación de estado 110 para uno o más nodos servidor 130. Por ejemplo, en la figura 15, el nodo de equilibrador de carga 110A es el primer nodo comprobador de estado para los nodos servidor 130A y 130B, y el segundo nodo de comprobador de estado para los nodos servidor 130C y 130D.

En al menos algunas realizaciones, si falla un nodo de equilibrador de carga 110, cambia la pertenencia a los cambios de anillo hash consistentes, y uno o más de los nodos de equilibradores de carga 110 que todavía tienen estado y, por lo tanto, todavía están en el anillo hash consistente pueden asumir la responsabilidad de comprobar el estado de los nodos servidor 130 previamente comprobados en estado por el nodo fallido 110.

En al menos algunas realizaciones, cada nodo con estado 110 realiza la comprobación de estado de sus nodos servidor asignados 130 en un intervalo regular, que puede denominarse intervalo de comprobación de servidor. En al menos algunas realizaciones, el intervalo de comprobación de servidor puede ser mayor o igual que el intervalo de intercambio mencionado anteriormente.

En al menos algunas realizaciones, para realizar una comprobación de estado de un nodo servidor 130, un nodo de

equilibrador de carga saludable 110 (por ejemplo, el nodo 110A en la figura 15) inicia un mensaje de ping de estado (por ejemplo, un mensaje de ping de estado L7 HTTP) a un nodo servidor 130 (por ejemplo, nodo servidor 130A en la figura 15). Si tiene estado, el nodo servidor 130 envía una respuesta de ping al nodo de equilibrador de carga 110. En al menos algunas realizaciones, el mensaje de ping es recibido y procesado por el módulo de equilibrador de carga 132 en el nodo servidor 130, por lo que el ping de comprobación de estado, si tiene éxito, establece que el módulo 132 en el nodo servidor 130 tiene estado. Tras recibir la respuesta al ping, el nodo de equilibrador de carga 110 considera el nodo servidor 130 con estado, y aumenta un contador de señales para el nodo servidor 130.

En al menos algunas realizaciones, los contadores de señales para todos los nodos servidor 130 comprobados en estado por un nodo de equilibrador de carga con estado 110 pueden propagarse a los demás nodos de equilibradores de carga 110, por ejemplo, de acuerdo con la técnica de intercambio descrita previamente para los contadores de señales del nodo de equilibrador de carga 110 en los que cada nodo 110 envía su lista de señales al menos a otro nodo 110 seleccionado aleatoriamente en un intervalo regular (el intervalo de intercambio), y el nodo receptor 110 actualiza su propia lista de señales según los valores máximos en la dos listas.

Detección de fallos e intercambio

En al menos algunas realizaciones, la información obtenida a través de las comprobaciones de estado del nodo de equilibrador de carga 110 y las comprobaciones de estado del nodo servidor 130 descritas anteriormente pueden necesitar propagarse a todos los nodos 110 en la implementación del equilibrador de carga para que todos los nodos de equilibrador de carga 110 puedan mantener una visión coherente de la implementación del equilibrador de carga. Como se ha descrito anteriormente, en al menos algunas realizaciones, los nodos de equilibradores de carga 110 pueden comunicarse entre sí de acuerdo con un protocolo de intercambio para intercambiar y propagar esta información de estado y para detectar fallos del nodo de equilibrador de carga 110 y el nodo servidor 130.

En al menos algunas realizaciones, a intervalos regulares (denominados intervalos de intercambio), cada nodo de equilibrador de carga 110 selecciona aleatoriamente otro nodo de equilibrador de carga 110 y envía al otro nodo 110 su vista de nodos de equilibradores de carga con estado 110 y nodos servidor 130 junto con los contadores de señales para los nodos de equilibrador de carga 110 y los nodos servidor 130. Siempre que un nodo de equilibrador de carga o nodo servidor 130 tenga estado, el nodo pasará sus comprobaciones de estado y su contador de señales continuará aumentando. Si el contador de señales para un nodo no cambia durante un intervalo especificado (que se puede denominar un intervalo de tiempo de fallo), entonces se sospecha que el nodo ha fallado por los nodos del equilibrador de carga 110. Una vez que se sospecha que un nodo ha fallado, los nodos de equilibradores de carga 110 pueden esperar un intervalo especificado (que puede denominarse intervalo de tiempo sin estado) antes de determinar que el nodo no tiene estado. Este intervalo de tiempo sin estado permite que los nodos de equilibradores de carga 110 esperen hasta que todos los nodos de equilibradores de carga 110 se den cuenta de que el nodo ha fallado.

La figura 16 ilustra gráficamente un estado para, o una vista, del estado de otro nodo (ya sea un nodo equilibrador de carga 110 o un nodo servidor 130) que puede mantenerse por un nodo de equilibrador de carga 110, de acuerdo con al menos algunas realizaciones. Se supone que el nodo de equilibrador de carga 110 empieza con una vista del nodo en cuestión como con estado, como se indica en 300. Esto indica que el contador de señales del nodo ha estado aumentando. Sin embargo, si el contador de señales del nodo no aumenta durante un intervalo especificado (el intervalo de tiempo de fallo) como se indica en 302, entonces el nodo de equilibrador de carga 110 sospecha que el nodo ha fallado, como se indica en 304. Si el contador de señales del nodo no aumenta durante un intervalo especificado (el intervalo de tiempo sin estado) como se indica en 306, entonces el nodo de equilibrador de carga 110 considera el nodo sin estado, como se indica en 308. Sin embargo, si el contador de señales del nodo aumenta antes de que expire el intervalo de tiempo sin estado como se indica en 310, el nodo de equilibrador de carga 110 considera de nuevo el nodo con estado 300. De manera similar, recibir un aumento de señal para un nodo sin estado como se indica en 312 puede hacer que el nodo sea considerado como con estado 300.

Determinar que un nodo no tiene estado puede implicar acciones diferentes por el nodo o nodos de equilibrador de carga 110 dependiendo de si el nodo sin estado es un nodo de equilibrador de carga 110 o un nodo servidor 130, y también dependiendo de la relación del nodo de equilibrador de carga 110 con el nódulo sin estado, como se describe en otra parte en el presente documento.

Datos del nodo de equilibrador de carga

En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede mantener datos sobre el estado de la implementación del equilibrador de carga. En al menos algunas realizaciones, estos datos pueden mantenerse

en una o más estructuras de datos en cada nodo de equilibrador de carga 110 incluyendo, pero sin limitación, una lista de nodos de equilibradores de carga con estado, una lista de nodos de equilibradores de carga sospechosos, y una lista de señales. La figura 17 ilustra un nodo de equilibrador de carga ejemplar 110 que mantiene una lista de nodos de equilibradores de carga con estado 320, una lista de nodos de equilibradores de carga sospechosos 322, una lista de nodos de equilibradores de carga sin estado 324, y una lista de señales de nodos de equilibradores de carga 326.

En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede mantener una lista de nodos de equilibradores de carga con estado 320, que es una lista de nodos de equilibradores de carga con estado 110 que puede, por ejemplo, usarse para determinar qué nodos 110 tienen estado y, por lo tanto, están participando en el protocolo de intercambio. Sólo los nodos 110 de la lista 320 están implicados en la propagación de la información del equilibrador de carga a través del protocolo de intercambio, sólo los nodos 110 de la lista 320 se consideran en el anillo hash consistente, y sólo los nodos 110 de esta lista, comprueban el estado de los nodos servidor 130. Un nodo 110 puede seleccionar aleatoriamente otro nodo 110 de esta lista 320 a la que se envía su información de señal. Además, los contadores de señales se intercambian sólo por los nodos 110 que están actualmente en la lista de nodos de equilibradores de carga con estado 320. En al menos algunas realizaciones, un nodo de equilibrador de carga *N* puede añadirse a la lista de nodos de equilibradores de carga con estado 320 de otro nodo de equilibrador de carga 110 si el nodo *N* pasa una comprobación de estado por el nodo de equilibrador de carga 110 o si el nodo de equilibrador de carga 110 recibe un mensaje de intercambio sobre el nodo *N* de algún otro nodo de equilibrador de carga 110 en la lista 320.

En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede mantener una lista de nodos de equilibradores de carga sospechosos 322, que es una lista de nodos de equilibradores de carga cuyo contador de señales (véase la lista de señales 326) no ha aumentado durante un intervalo especificado (denominado como el intervalo de tiempo de fallo). Si un nodo de equilibrador de carga *E* está en la lista de nodos de equilibrador de carga sospechosos 322 de un nodo de equilibrador de carga 110, entonces el nodo de equilibrador de carga 110 no hará intercambios sobre el nodo *E.* Si algún otro nodo de equilibrador de carga 110 en la lista con estado 320 hace intercambios con el nodo de equilibrador de carga 110 sobre el nodo *E* con un contador de señales mayor que el contador para el nodo *E* en la lista de señales 326 del nodo 110, entonces el nodo *E* no se moverá de la lista de sospecha 322 a la lista con estado 320. Si el nodo *E* permanece en la lista de sospecha 322 del nodo de equilibrador de carga 110 durante un intervalo especificado (denominado como el intervalo de tiempo sin estado), el nodo *E* se considera sin estado por el nodo de equilibrador de carga 110 y se mueve en una lista de nodos sin estado 324. Un nodo 110 en la lista de nodos sin estado 324 (en este ejemplo, el nodo *G)* puede moverse a la lista de nodos con estado 320 de un nodo de equilibrador de carga 110 tras pasar el *G* una comprobación de estado por el nodo 110 o tras recibir un contador de señales actualizado para el nodo *G* de otro nodo 110.

En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 puede mantener una lista de señales 326 para todos los nodos de equilibradores de carga conocidos 110. Para cada nodo 110, esta lista 326 puede incluir un contador de señales y una marca de tiempo que indica cuando el contador de señales cambió por última vez.

En al menos algunas realizaciones, cada nodo de equilibrador de carga 110 también puede mantener una lista de señales para todos los nodos servidor conocidos, no mostrados en la figura 17. Esta lista puede ser similar a la lista de señales de nodo de equilibrador de carga 326. En algunas realizaciones, las dos listas pueden combinarse. En al menos algunas realizaciones, la información de señales para los nodos servidor 130 puede propagarse entre los nodos de equilibrador de carga 110, por ejemplo, de acuerdo con un protocolo de intercambio, junto con o además de la información de señales para los nodos de equilibrador de carga 110.

Aunque la figura 17 muestra cuatro listas separadas, se ha de apreciar que dos o más de las listas pueden combinarse en una única lista. Por ejemplo, en algunas realizaciones, puede mantenerse una única lista de todos los nodos 110 en cada nodo de equilibrador de carga 110, y se pueden usar banderas de bits u otras estructuras de datos para indicar si cada nodo tiene actualmente estados, es sospechoso o no tiene estado.

Datos del nodo servidor

En al menos algunas realizaciones, los nodos servidor 130 y los módulos de equilibrador de carga local 132 en los nodos 130 no participan en el protocolo de intercambio con los nodos de equilibrador de carga 110. Los nodos de equilibrador de carga 110 intercambian la información de señales sobre los otros nodos de equilibradores de carga 110 obtenidos por el método de comprobación de estado del nodo del equilibrador de carga y la información de señales sobre los nodos servidor 130 obtenidos por el método de comprobación de estado del nodo servidor

solamente entre ellos (específicamente, cada nodo de equilibrador de carga 110 hace intercambios solamente con los nodos actualmente en su lista de nodos de equilibrador de carga con estado 320).

.Sin embargo, cada nodo servidor 130/módulo de equilibrador de carga 132 puede necesitar información sobre nodos de equilibradores de carga con estado 110 en la implementación de equilibrador de carga de manera que el nodo servidor 130 pueda determinar nodos de equilibrador de carga 110 (específicamente, nodos de salida) a los que el nodo servidor 130 puede reenviar el tráfico de cliente saliente y determinar qué nodos de equilibrador de carga a los que se va a enviar la información de publicación de conexión. En al menos algunas realizaciones, para proporcionar esta información a los nodos servidor 130, los nodos de equilibradores de carga 110 pueden actualizar periódica o aperiódicamente los nodos servidor 130 con información que identifica los nodos de equilibradores de carga actualmente con estado 110 (por ejemplo, la lista de nodo de equilibrador de carga con estado 320 en la figura 17). En al menos algunas realizaciones, los nodos de equilibradores de carga 110 que son responsables de la comprobación de estado de un nodo servidor dado 130 (véase la figura 15) son responsables de proporcionar la información que identifica los nodos de equilibradores de carga actualmente con estado al servidor 130. Por ejemplo, haciendo referencia a la figura 15, el nodo de equilibrador de carga 110A puede enviar su lista de nodos de equilibrador de carga con estado 320 a los nodos servidor 130A, 130B, 130C y 130D, el nodo de equilibrador de carga 110B puede enviar su lista de nodos de equilibradores de carga con estado 320 a los nodos servidor 130C, 130D y 130E, etc.

Manejo de fallos de nodos de equilibradores de carga

Las figuras 18A y 18B ilustran el manejo de un fallo del nodo del equilibrador de carga, de acuerdo con al menos algunas realizaciones. La figura 18A muestra una implementación de equilibrador de carga ejemplar. Existen cuatro nodos de equilibradores de carga 110A a 110D actualmente en la implementación del equilibrador de carga. El enrutador periférico 104 envía paquetes entrantes de clientes (no mostrados) a los nodos de equilibradores de carga 110. En al menos algunas realizaciones, el enrutador periférico 104 puede tomar las decisiones de enrutamiento de acuerdo con una técnica de enrutamiento multitrayecto de hash de capa 4 por flujo, por ejemplo una técnica de enrutamiento multitrayecto de igual coste (ECMP). En al menos algunas realizaciones, el enrutador periférico 104 conoce los nodos de equilibradores de carga 110 que están actualmente disponibles en la implementación del equilibrador de carga para recibir tráfico de cliente a través de avisos del nodo de equilibrador de carga 110, por ejemplo, avisos a través de sesiones de tecnología del protocolo de puerta de enlace de frontera (BGP) iniciadas por los nodos de equilibradores de carga 110. Sin embargo, en al menos algunas realizaciones, en lugar de un nodo de equilibrador de carga 110 que se anuncia al enrutador periférico 104 a través de una sesión BGP, al menos otro nodo 110 en la implementación de equilibrador de carga asume la responsabilidad de anunciar el nodo 110 al enrutador periférico 104 a través de BGP. Por ejemplo, en algunas realizaciones, como se muestra en la figura 18A, los nodos vecinos izquierdo y derecho 110 de un nodo dado 110 anuncian el nodo dado 110 al enrutador periférico 104. Por ejemplo, el nodo de equilibrador de carga 110A anuncia los nodos 110B y 110D, el nodo de equilibrador 110B anuncia los nodos 110A y 110C, y el nodo de equilibrador de carga 110C anuncia los nodos 110B y 110D.

Como se muestra en el ejemplo de la figura 18A, cada nodo de equilibrador de carga 110 también comprueba el estado periódicamente de uno o más nodos de equilibradores de carga diferentes 110, por ejemplo, uno o más nodos seleccionados aleatoriamente 110, uno o más nodos vecinos 110 determinados por una lista ordenada de nodos del equilibrador de carga, o uno o más nodos vecinos y uno o más nodos seleccionados aleatoriamente. Además, cada nodo de equilibrador de carga 110 puede comprobar el estado periódicamente de al menos un nodo servidor 130 y también puede enviar su lista de nodos de equilibrador de carga con estado 110 al nodo o nodos servidor de los que comprueba el estado. La información de estado para los nodos de equilibradores de carga 110 y los nodos servidor 130 puede propagarse entre los nodos 110, por ejemplo, de acuerdo con un protocolo de intercambio.

La figura 18B ilustra el manejo del fallo de un único nodo de equilibrador de carga 110 en la implementación de equilibrador de carga ejemplar de la figura 18A. En este ejemplo, el nodo del equilibrador de carga 110B ha fallado por alguna razón. Por ejemplo, los nodos 110A y 110C pueden comprobar el estado del nodo 110B, y ambos pueden detectar que el nodo 110B falla sus comprobaciones de estado. Por lo tanto, los nodos 110A y 110C no aumentan el contador de señales para el nodo 110B. La información de señales de los dos nodos 110A y 110B se propaga a los otros nodos de equilibradores de carga con estado 110 (en este ejemplo, el único otro nodo de equilibrador de carga es el nodo 110D) de acuerdo con el protocolo de intercambio. Tan pronto como todos los nodos de equilibradores de carga con estado 110 (en este ejemplo, los nodos 110A, 110C y 110D) convergen en el fallo del nodo 110B, puede aparecer uno o más de, pero sin limitación, los siguientes eventos. Ha de apreciarse que estos eventos no se producen necesariamente en este orden.

• Los nodos 110A y 110C detienen el nodo de anuncio 110B al enrutador periférico 104. En al menos algunas realizaciones, esto implica finalizar la sesión BGP que el nodo 110 estableció con el enrutador periférico 104 para anunciar el nodo 110B. Obsérvese que cada nodo 110 establece una sesión BGP separada con el enrutador periférico 104 para cada otro nodo 110 que anuncia, por lo que finalizar la sesión BGP para el nodo 110B no afecta a otros nodos 110 que se anuncian. En al menos algunas realizaciones, un nodo 110 termina una sesión BGP con el enrutador periférico 104 enviando un mensaje de cerrar TCP o similar para la sesión BGP al enrutador periférico 104.

• En respuesta a la detección de que el nodo 110B ya no está siendo anunciado por ninguno de los nodos, el enrutador periférico 104 deja de enrutar los paquetes de datos del cliente al nodo 110B. El enrutador periférico 104 también ajusta el hashing multitrayecto (por ejemplo, ECMP) para redistribuir los flujos de paquetes de los clientes a los nodos de equilibradores de carga con estado restantes 110, específicamente a los servidores de entrada 112 en los nodos 110. Para cualquier flujo de paquetes enrutado a un servidor de entrada 112 para el que el servidor de entrada 112 no tiene una asignación cliente -> servidor, la asignación puede obtenerse de un nodo de rastreador de flujo para la conexión cliente -> servidor o, como alternativa, puede establecerse una nueva conexión cliente -> servidor de acuerdo con la técnica como se ilustra en las figuras 10A a 10G.

• Los nodos 110A y 110C pueden abrir cada uno una sesión BGP al enrutador periférico 104 para anunciarse entre sí. Obsérvese que, puesto que ambos nodos 110A y 110C se anuncian al enrutador periférico 104 por el nodo de equilibrador de carga 110D, así como el nodo 110B, el hecho de que el nodo 110B pueda detener los nodos de anuncio 110A y 110B al enrutador periférico 104 cuando falla, no hace que el enrutador periférico 104 detenga los paquetes de enrutamiento a estos dos nodos 110.

• En al menos algunas realizaciones, los nodos 110A y 110C pueden asumir la responsabilidad de verificar el estado entre sí, ya que ahora son nodos vecinos 110. Ha de apreciarse que el nodo 110B, aunque considerado sin estado, puede ser comprobado en estado aleatoriamente por uno o más de los otros nodos 110.

• Uno o más de los nodos de equilibradores de carga con estado restantes 110 pueden asumir la responsabilidad de las conexiones de rastreador de flujo anteriormente el flujo rastreado por el nodo 110B. Por ejemplo, el nodo 110C y/o el nodo 110D pueden actuar como rastreadores de flujo primario o secundario como se ilustra en las figuras 11C y 11D para una o más conexiones para las cuales el nodo 110B era un rastreador de flujo primario o secundario.

• Uno o más de los nodos de equilibradores de carga con estado restantes 110 pueden asumir la responsabilidad de verificar el estado de los nodos servidor 130 previamente comprobados en estado por el nodo 110B. Los nodos servidor 130 se actualizan con la lista de nodos de equilibrador de carga con estado (que no incluye el nodo 110B) por los nodos de equilibrador de carga restantes 110. Por ejemplo, en la figura 18B, el nodo de equilibrador de carga 110A comienza la comprobación de estado y la actualización del nodo servidor 130C, el nodo de equilibrador de carga 110C comienza la comprobación de estado y la actualización del nodo servidor 130B.

• En el enrutador periférico 104, las sesiones BGP del nodo fallido 110B eventualmente se agotan. Como alternativa, el enrutador periférico 104 puede terminar las sesiones BGP al reconocer que el nodo 110B ha fallado.

Es posible que dos nodos de equilibradores de carga 110 puedan fallar en o cerca del mismo tiempo. Si los dos nodos de equilibrador de carga fallidos no son adyacentes entre sí, entonces los fallos son independientes y pueden manejarse como fallos individuales de nodos separados 110 de acuerdo con el método ilustrado en la figura 18B. Sin embargo, si los dos nodos fallidos son adyacentes entre sí (por ejemplo, los nodos 110B y 110C en la figura 18A, entonces tan pronto como todos los nodos de equilibradores de carga con estado 110 (en este ejemplo, los nodos 110A y 110D) detecten y converjan en el fallo, puede producirse uno o más de, pero sin limitación, los siguientes eventos. Ha de apreciarse que estos eventos no se producen necesariamente en este orden.

• El nodo 110A finaliza la sesión BGP al enrutador periférico 104 para el nodo 110B.
• El nodo 110D finaliza la sesión BGP al enrutador periférico 104 para el nodo 110C.
• Los nodos 110A y 110D inician la sesión BGP con el enrutador periférico 104 para anunciarse entre sí.
• Los nodos 110A y 110D pueden comenzar a comprobar su estado entre sí. Obsérvese que los nodos 110A y 110D también pueden continuar comprobando el estado de los nodos fallidos 110.
• Los nodos con estado restantes 110 actualizan los nodos servidor 130 con las listas de nodos del equilibrador de carga con estado.
• El tráfico puede seguir fluyendo desde el enrutador periférico 104 al nodo 110B y/o nodo 110C, puesto que estos dos nodos 110 pueden continuar anunciándose entre sí al enrutador periférico 104. Sin embargo, estas sesiones de BGP eventualmente se agotarán, y el enrutador periférico 104 redistribuirá los flujos a los nodos anunciados restantes 110 en consecuencia.
• Los nodos 110B y 110C pueden cerrar sus sesiones BGP con el enrutador periférico 104 en el que anuncian los nodos 110A y 110D, respetuosamente, si los nodos 110B y 110C piensan que todavía tienen estado.

Publicación de conexiones

Haciendo referencia de nuevo a la figura 1, en al menos algunas realizaciones, los nodos de equilibradores de carga 110 en una implementación de equilibrador de carga mantienen información de estado para conexiones TCP de cliente a los servidores 130. Esta información de estado permite que los nodos de equilibradores de carga 110 enruten tráfico de cliente entrante desde el enrutador periférico 104 hasta los nodos servidor 130 responsables de las conexiones TCP. Los módulos de equilibrador de carga 132 en los nodos servidor 130 mantienen listas de conexiones TCP activas a sus respectivos servidores 134. La publicación de conexiones es un mecanismo a través del cual los módulos de equilibrador de carga 132 en los nodos servidor 130 pueden publicar sus listas de conexiones TCP de cliente activo a los nodos de equilibradores de carga 110. En al menos algunas realizaciones, los paquetes de publicación de conexión se forman y se publican en los nodos de equilibradores de carga 110 por los módulos de carga 132 en un intervalo regular, que se puede denominar intervalo de publicación de conexión.

En al menos algunas realizaciones, la información de estado de conexión mantenida por los nodos de equilibradores de carga 110 puede verse como una forma de caché, y el mantenimiento de la información de estado para una conexión particular puede verse como un mantenimiento de una cesión en el nodo de equilibrador de carga 110 para esa conexión. A menos que se renueven las entradas de caché, los nodos de equilibradores de carga 110 pueden no ser capaces de enrutar los flujos de datos del cliente hacia los nodos servidor 130 que están gestionando los flujos de datos. El mecanismo de publicación de conexión periódicamente renueva los cachés y, por lo tanto, las cesiones, en los nodos de equilibradores de carga 110 con información de estado de conexión actual desde los nodos servidor 130 para mantener así los paquetes TCP fluyendo de los clientes 160 a los nodos servidor adecuados 130. Cuando un cliente 160 termina una conexión TCP con un servidor 134, el módulo de equilibrador de carga 132 en el nodo servidor 130 asociado con esa conexión hará caer la conexión de su lista de conexiones activas y, por lo tanto, no dejará de publicar la conexión TCP a través del mecanismo de publicación de conexiones. Por lo tanto, la información de estado de conexión para esa conexión (la entrada o entradas de caché) en los nodos de equilibrador de carga 110 asociados con dicha conexión (específicamente, el servidor de entrada 112 y los rastreadores de flujo primario y secundario 116 para la conexión) ya no se renueva, y la conexión es eliminada por los nodos de equilibradores de carga 110. En al menos algunas realizaciones, la entrada o entradas de caché para la conexión pueden permanecer en el caché en un nodo de equilibrador de carga 110 hasta que se requiera la memoria para alguna otra conexión activa.

Por lo tanto, el mecanismo de publicación de conexión amplía periódicamente o de forma aperiódica las cesiones de conexión en los servidores de entrada 112 y los rastreadores de flujo primario y secundario 116 para mantener el tráfico del cliente fluyendo. Además, el mecanismo de publicación de conexiones puede ayudar a recuperar al menos algunos fallos del nodo del equilibrador de carga 110. Cuando uno o más nodos de equilibradores de carga 110 que mantienen información de estado para una conexión de cliente falla, la información de conexión activa proporcionada a los nodos de equilibrador de carga restantes 110 por la publicación de conexión puede en algunos casos usarse para recuperar la conexión.

Usando el mecanismo de publicación de conexión, los nodos servidor 130 son las fuentes autorizadas para los estados de las conexiones entre los servidores 134 y los clientes 160. Además, el cierre de las conexiones a los servidores 134 es manejado pasivamente por los módulos de equilibrador de carga 132 en los nodos servidor 130 y los nodos de equilibrador de carga 110. No es necesario un intercambio entre los nodos servidor 130 y los nodos de equilibrador de carga 110. En otras palabras, los módulos de equilibrador de carga 132 no tienen que enviar mensajes a los nodos de equilibrador de carga 110 para informar activamente a los nodos de que determinadas conexiones se han cerrado. Cuando un servidor 134 cierra una conexión, el servidor 134 borra su estado interno para la conexión. El módulo de equilibrador de carga 132 utiliza el estado interno del servidor 134 para rellenar el paquete de publicación de conexión. Dado que la conexión ya no está en el estado interno del servidor 134, la conexión no se publica en los nodos de equilibrador de carga 110. Por lo tanto, la cesión para la conexión en el nodo de equilibrador de carga 110 expira, y los nodos de equilibradores de carga 110 olvidan pasivamente la conexión. La memoria en el caché del nodo del equilibrador de carga 110 que se usó para la conexión puede usarse entonces para otras conexiones según sea necesario.

En algunas realizaciones, las cesiones para conexiones mantenidas por los nodos de equilibradores de carga 110 pueden implicar entradas de registro de tiempo para las conexiones en la memoria caché. Cuando la cesión de una conexión se renueva mediante un paquete de publicación de conexión, se puede actualizar la marca de tiempo. Si la cesión de una conexión no se renueva porque la conexión ya no se publica por el módulo de equilibrador de carga 132 en el nodo servidor 130, entonces la marca de tiempo ya no se actualiza. En al menos algunas realizaciones, se puede usar un método de recogida de basura perezosa en el que la entrada para la conexión puede permanecer en el caché hasta que se necesite la memoria. Por ejemplo, en al menos algunas realizaciones, las marcas de tiempo en entradas de memoria caché pueden compararse con un umbral de tiempo de renovación de concesión; si la marca de tiempo para una entrada de caché es anterior al umbral, entonces la entrada está obsoleta y puede volver

a utilizarse. Sin embargo, en algunas realizaciones, las entradas obsoletas pueden ser basura recogida activamente.

Destinatarios de publicación de conexiones

En al menos algunas realizaciones, para cada conexión TCP de cliente, hay tres nodos de equilibradores de carga 110 que mantienen un estado de conexión - el nodo 110 que da servicio como el servidor de entrada 112, el nodo 110 que da servicio como el rastreador de flujo primario 116 y el nodo que sirve como el rastreador de flujo secundario 116. Para un flujo TCP dado, los rastreadores de flujo primario y secundario 116 pueden determinarse, por ejemplo, mediante un nodo de equilibrador de carga 110, aplicando una función de hash consistente al flujo TCP para encontrar el nodo de rastreador de flujo primario 116 y su nodo sucesor en el anillo hash consistente. El nodo de equilibrador de carga 110 que sirve como el servidor de entrada 112 para un flujo TCP es el nodo 110 que recibe tráfico para ese flujo desde el enrutador periférico 104 basado en la función de hash multitrayecto interna (por ejemplo, ECMP) del enrutador periférico 104. Si hay un fallo o adición del nodo 110, el nodo de equilibrador de carga 110 que da servicio como el servidor de entrada 112 puede cambiar para muchos de los flujos TCP activos; y los nodos de equilibradores de carga 110 que sirven como rastreadores de flujo para al menos algunos flujos TCP activos pueden cambiar (véase, por ejemplo, las figuras 11A a 11D). Para cada flujo TCP al servidor 132 en un nodo servidor 130, el módulo de equilibrador de carga 132 en ese nodo servidor 130 mantiene información de estado que indica cuál de los nodos de equilibradores de carga 110 es el servidor de entrada 112 para ese flujo TCP, ya que recibe tráfico desde ese nodo de equilibrador de carga 110. Sin embargo, en al menos algunas realizaciones, el módulo de equilibrador de carga 132 puede no saber y puede no ser capaz de determinar qué nodos de equilibradores de carga 110 sirven como rastreadores de flujo primario y secundario para un flujo TCP, ya que el módulo de equilibrador de carga 132 puede no conocer la función de hash consistente que se utiliza. En otras palabras, en al menos algunas realizaciones, los módulos de equilibrador de carga 132 no hacen un hash consistente.

Publicación de información de conexión activa

Las figuras 19A y 19B ilustran gráficamente una técnica de publicación de conexión, de acuerdo con al menos algunas realizaciones. La figura 19A ilustra los módulos de equilibrador de carga (LB) que publican información de conexión activa a los nodos del equilibrador de carga. En al menos algunas realizaciones, cada módulo de equilibrador de carga 132 recopila información para cada flujo TCP activo en el nodo servidor 130 y forma un paquete de publicación de conexiones. La información para un flujo TCP dado incluye información que identifica el nodo de equilibrador de carga 110 que sirve como el servidor de entrada 112 para el flujo. Cuando un paquete de publicación de conexión está listo (por ejemplo, cuando se ha alcanzado el intervalo de publicación de conexión), el módulo de equilibrador de carga 132 selecciona aleatoriamente un nodo de equilibrador de carga 110, por ejemplo, de la lista de nodos de equilibradores de carga con estado 110 que se envían periódicamente a los nodos servidor 130 desde los nodos de equilibradores de carga 110 que comprueban el estado de los nodos servidor 130 como se ha descrito anteriormente. El módulo de equilibrador de carga 132 envía entonces el paquete de publicación de conexión al nodo seleccionado 110. Por ejemplo, en la figura 19A, el módulo de equilibrador de carga 132A ha enviado un paquete de publicación de conexión al nodo de equilibrador de carga 110A, y envía posteriormente otro paquete de publicación de conexión al nodo de equilibrador de carga 110B.

La figura 20 es un diagrama de flujo de alto nivel de un método de publicación de conexión que puede realizarse por cada módulo de equilibrador de carga 132, de acuerdo con al menos algunas realizaciones. Como se indica en 500, el módulo de equilibrador de carga 132 (LB) crea una entrada de publicación de conexión para cada flujo TCP activo en el nodo servidor 130 respectivo. En al menos algunas realizaciones, el módulo de equilibrador de carga 132 recupera el conjunto de conexiones TCP activas que maneja el servidor 134 en el nodo servidor 130, por ejemplo, desde /proc/net/tcp en el nodo servidor 130. Para cada conexión TCP activa, el módulo de equilibrador de carga 132 busca (por ejemplo, en una tabla mantenida localmente de conexiones activas) el nodo de equilibrador de carga 110 que sirve como el servidor de entrada 112 para el flujo TCP y crea una entrada de publicación de conexión que indica la tupla TCP para la conexión (por ejemplo, una tupla de 4 que consiste en: la dirección IP del cliente, el puerto del cliente, la dirección IP del servidor (pública), y el puerto del servidor) y el servidor de entrada 112 para la conexión. Obsérvese que cada módulo de equilibrador de carga 132 mantiene información para cada conexión TCP activa que indica el último nodo de equilibrador de carga 110 desde el que se recibió un paquete para la conexión, y esta información puede ser utilizada por el módulo de equilibrador de carga 132 para identificar el nodo de entrada 110 para cada conexión activa.

Como se indica en 502, el módulo de equilibrador de carga 132 selecciona aleatoriamente un nodo de equilibrador de carga 110 al que se va a enviar el paquete de publicación de conexión (que contiene una o más entradas de publicación de conexión, con una entrada para cada conexión TCP activa). En al menos algunas realizaciones, el

módulo de equilibrador de carga 110 puede seleccionarse aleatoriamente cuando el módulo de equilibrador de carga 132 determina que el paquete de publicación de conexión está listo para ser enviado. En al menos algunas realizaciones, esta determinación se hace de acuerdo con un intervalo de publicación de conexión. Como ejemplos no limitantes, el intervalo de publicación de conexión puede ser de 100 milisegundos (ms), o un segundo. En al menos algunas realizaciones, el módulo de equilibrador de carga 110 se selecciona de una lista de nodos de equilibrador de carga con estado 110 que se ha recibido previamente desde uno de los nodos de equilibrador de carga 110. Como se indica en 504, el módulo de equilibrador de carga publica entonces el paquete de publicación de conexión al nodo de equilibrador de carga seleccionado 110. En al menos algunas realizaciones, el paquete de publicación de conexión es un paquete sin estado, por ejemplo, un paquete UDP. En algunas realizaciones, el paquete de publicación de conexión puede comprimirse antes de enviar los paquetes al nodo de equilibrador de carga de destino 110. En al menos alguna realización, la información de publicación de conexión puede enviarse al nodo de equilibrador de carga de destino 110 en dos o más paquetes.

Como se indica por la flecha que regresa del elemento 504 al elemento 500, el módulo de equilibrador de carga 132 puede construir continuamente paquetes de publicación de conexión, seleccionar nodos aleatorios 110, y enviar los paquetes a los nodos seleccionados. Como se ha indicado anteriormente, esto puede realizarse de acuerdo con un intervalo de publicación de conexión de manera que los nodos de equilibradores de carga 110 se refrescan relativamente de forma regular con la información de conexión activa actual para mantener las concesiones de conexión en los nodos de equilibradores de carga 110.

En al menos algunas realizaciones, puesto que los paquetes de publicación de conexión son distribuidos aleatoriamente a los nodos de equilibradores de carga 110 por los módulos equilibradores de carga, los nodos de equilibradores de carga 110 que reciben los paquetes de publicación de conexión son responsables de distribuir la información de conexión activa en los paquetes de publicación de conexión a los nodos de entrada/primario/secundario correctos 110 para las conexiones. La figura 19B y las figuras 21 y 22 ilustran métodos para distribuir la información de conexión activa que se puede usar en al menos algunas realizaciones.

La figura 19B ilustra la distribución de la información de conexión activa entre los nodos de equilibradores de carga 110, de acuerdo con al menos algunas realizaciones. Cuando un nodo de equilibrador de carga 110 recibe un paquete de publicación de conexión de un módulo de equilibrador de carga 132, el nodo de equilibrador de carga 110 puede analizar la información para cada flujo TCP indicado en el mismo para determinar el nodo de entrada y los nodos de rastreador de flujo primario y secundario para ese flujo. Si el nodo de equilibrador de carga 110 sirve en una de esas funciones para un flujo, el nodo de equilibrador de carga 110 consume la información para el flujo (por ejemplo, actualizando su caché de información de estado). En al menos algunas realizaciones, el nodo de equilibrador de carga 110 puede también poner la información para que el flujo en paquete(s) sea enviado a uno o más nodos 110 que sirven en las otras funciones para el flujo. Para los flujos restantes indicados por el paquete de publicación de conexión, el nodo de equilibrador de carga 110 divide la información de conexión activa en dos o más paquetes más pequeños y envía cada paquete a uno o más nodos de equilibrador de carga 110. Por ejemplo, en al menos algunas realizaciones, puede enviarse un paquete que contiene información de conexión activa para uno o más flujos a los nodos de equilibradores de carga 110 que sirven como el servidor de entrada 112, rastreador de flujo primario 116A y rastreador de flujo secundario 116B para el uno o más flujos.

La figura 21 es un diagrama de flujo de un método para distribuir la información de conexión activa recibida en un paquete de publicación de conexión para dirigir nodos de equilibradores de carga 110, de acuerdo con al menos algunas realizaciones. Como se indica en 520, un nodo de equilibrador de carga 110 recibe un paquete de publicación de conexión de un módulo de equilibrador de carga 132. El módulo de equilibrador de carga 132 generó el paquete y seleccionó el nodo de equilibrador de carga 110 para recibir el paquete, por ejemplo, como se ha descrito anteriormente con referencia a las figuras 19A y 20. El paquete de publicación de conexión puede incluir información que identifica el nodo servidor 130 desde el que se recibió el paquete (por ejemplo, una dirección IP del módulo de equilibrador de carga 132 en el nodo servidor 130) y una lista de entradas que identifican conexiones TCP activas (por ejemplo, una tupla de 4 que consiste en: la dirección IP del cliente, el puerto del cliente, la dirección IP del servidor (pública), y el puerto del servidor para cada conexión).

En los elementos 522-530 de la figura 21, el módulo de equilibrador de carga 110 procesa iterativamente la información de conexión TCP activa indicada en el paquete de publicación de conexión recibida. Como se indica en 522, el nodo de equilibrador de carga 110 analiza la entrada para un flujo TCP siguiente en el paquete para determinar el nodo de entrada 110 y los nodos de rastreador de flujo primario y secundario 110 para el flujo TCP respectivo. En al menos algunas realizaciones, el nodo de equilibrador de carga 110 obtiene la identidad del nodo de entrada 110 desde la entrada de publicación de conexión. En al menos algunas realizaciones, los nodos de rastreador de flujo primario y secundario 110 para el flujo de TCP pueden determinarse de acuerdo con la función de

hash consistente. En 524, si el nodo de equilibrador de carga 110 da servicio en una de las funciones para el flujo TCP que se está examinando, entonces en 526, el nodo de equilibrador de carga 110 consume la información para el flujo, por ejemplo, actualizando su caché de información de estado. Como se indica en 528, el nodo de equilibrador de carga 110 puede añadir la entrada de publicación de conexión para el flujo TCP a un paquete que se está construyendo que se va a enviar a otro nodo de equilibrador de carga 110. En 530, si hay más entradas de publicación de conexión para flujos en el paquete de publicación de conexión, entonces el método vuelve a 522 para procesar la siguiente entrada. De lo contrario, el nodo de equilibrador de carga envía el paquete o paquetes recién construidos que contienen cada uno un subconjunto de las entradas de publicación de conexión desde el paquete de publicación de conexión original a nodos de equilibrador de carga de destino 110 para los paquetes, como se indica en 532. En al menos algunas realizaciones, los paquetes enviados a los nodos de equilibradores de carga de destino 110 son paquetes sin estado, por ejemplo, paquetes UDP. En algunas realizaciones, los paquetes pueden comprimirse antes de enviar los paquetes a los nodos de equilibradores de carga objetivo 110.

Por lo tanto, en al menos algunas realizaciones, en los elementos 522-528 de la figura 21, el nodo de rastreador de flujo 110 construye uno o más paquetes (por ejemplo, paquetes UDP) cada uno para ser enviado a uno particular de los otros nodos 110 de acuerdo con la información determinada en 522 desde las entradas de publicación de conexión en el paquete de publicación de conexión recibido. En al menos algunas realizaciones, un paquete enviado a otro nodo 110 contiene entradas para flujos TCP para los cuales el nodo objetivo 110 sirve como nodo de entrada 110, nodo de rastreador de flujo primario 110, o nodo de rastreador de flujo secundario 110. Obsérvese que en algunas realizaciones un nodo de equilibrador de carga 110 dado puede servir como nodo de rastreador de flujo de entrada y primario para un flujo de TCP, o tanto como nodo de rastreador de flujo de entrada como secundario para un flujo TCP.

La figura 22 ilustra un método alternativo para distribuir la información de conexión activa recibida en un paquete de publicación de conexión a nodos de equilibrador de carga de destino 110, de acuerdo con al menos algunas realizaciones. Como se indica en 550, un nodo de equilibrador de carga 110 recibe un paquete de publicación de conexión de un módulo de equilibrador de carga 132. En este método, como se indica en 552, un proceso en el módulo de equilibrador de carga 110 analiza las entradas de publicación de conexión en el paquete y divide el paquete recibido en uno o más paquetes más pequeños en consecuencia. El módulo de equilibrador de carga 110 no consume localmente la información de flujo durante este proceso. Una vez que el paquete de publicación de conexión se ha dividido en uno o más paquetes, los paquetes se procesan entonces como se indica en 554-560. En 554, si el nodo objetivo 110 para el paquete es este nodo de equilibrador de carga 110, entonces el nodo de equilibrador de carga 110 consume localmente el paquete como se indica en 556. De lo contrario, el paquete se envía al nodo de equilibrador de carga objetivo 110. En 560, si hay más paquetes a procesar, entonces el método vuelve a 554. De lo contrario, se realiza el método.

Por lo tanto, el nodo de equilibrador de carga 110 que recibe un paquete de publicación de conexión de un módulo de equilibrador de carga 132 puede dividir el paquete de publicación de conexión en dos o más paquetes más pequeños que son específicos para los partículas de los otros nodos de equilibrador de carga 110 y distribuir los paquetes en consecuencia, mientras que internamente se consume información de flujo para cualquier flujo TCP que esté siendo manejado actualmente por el nodo de equilibrador de carga 110. Entretanto, otros nodos de equilibrador de carga 110 también pueden recibir paquetes de publicación de conexión desde los módulos de equilibrador de carga 132, dividir las entradas de publicación de conexión en múltiples paquetes más pequeños, y enviar los paquetes más pequeños a los nodos de destino 110 para distribuir de este modo la información de conexión activa entre los nodos 110.

Desencadenantes de publicación de conexión

En al menos algunas realizaciones, se puede desencadenar una publicación de conexión en un módulo de equilibrador de carga 132 mediante uno o más eventos diferentes. Como se ha indicado anteriormente, en algunas realizaciones, puede generarse y enviarse un paquete de publicación de conexión a un nodo de equilibrador de carga 110 seleccionado aleatoriamente de acuerdo con un intervalo de publicación de conexión, por ejemplo, a 100 ms o intervalos de un segundo, para renovar las cesiones para las conexiones TCP en los nodos de equilibradores de carga 110. En algunas realizaciones, un cambio en la pertenencia de los nodos de equilibradores de carga 110 puede desencadenar un evento de publicación de conexión inmediata. En al menos algunas realizaciones, el módulo de equilibrador de carga 132 puede aprender acerca del cambio de la lista de nodos de equilibradores de carga con estado 110 enviados desde uno de los nodos de equilibradores de carga 110 que comprueba el estado del respectivo nodo servidor 130. Tras detectar el cambio de acuerdo con la lista (ya sea una deleción o una adición), el módulo de equilibrador de carga 132 puede generar un paquete de publicación de conexión y enviarlo a un nodo de equilibrador de carga 110 para que las conexiones TCP afectadas por el cambio puedan recuperarse más

rápidamente por los nodos de equilibrador de carga 110.

Prevención de bucles de paquetes

Los bucles de paquete de publicación de conexión pueden producirse si la pertenencia a la capa de equilibrador de carga cambia mientras se procesa un paquete de publicación de conexión. Un primer nodo 110 puede recibir un paquete de publicación de conexión desde un módulo de equilibrador de carga 132 y enviar un paquete más pequeño a un segundo nodo 110. Sin embargo, si la pertenencia ha cambiado, el segundo nodo 110 puede determinar que el paquete debe ir al primer nodo 110 y, por lo tanto, puede reenviar el paquete al primer nodo 110. En al menos algunas realizaciones, para evitar que suceda este bucle, pueden utilizarse números de puerto diferentes para los paquetes de publicación de conexión recibidos de los módulos de equilibrador de carga 132 y los recibidos desde los nodos del equilibrador de carga 110, y los nodos de equilibradores de carga 110 no redistribuyen paquetes de publicación de conexión recibidos de otros nodos de equilibradores de carga 110.

Alternativas de distribución de paquetes de publicación de conexión

En los métodos de publicación de conexión descritos anteriormente, el módulo de equilibrador de carga 132 selecciona aleatoriamente un nodo de equilibrador de carga 110 al que se envía un paquete de publicación de conexión. Sin embargo, en algunas realizaciones, pueden usarse otros métodos para seleccionar un nodo de equilibrador de carga 110. Por ejemplo, en algunas realizaciones, el nodo de equilibrador de carga 132 puede construir uno o más paquetes de publicación de conexión que están dirigidos cada uno a un nodo de entrada particular 110 que maneja uno o más de los flujos TCP activos y envía el paquete o paquetes al nodo o nodos de entrada objetivo 110. El nodo o nodos de entrada 110 redestribuirán entonces la información de conexión activa a los rastreadores de flujo primario y secundario para la conexiones. Como otro ejemplo, en algunas realizaciones, en lugar de enviar el paquete de publicación de conexión a un único nodo seleccionado al azar 110, cada paquete de publicación de conexión puede ser enviado por el módulo de equilibrador de carga 132 a dos o más de los nodos con estado 110, o a cada uno de los nódulos con estado 110.

Arquitectura del nodo del equilibrador de carga

La figura 23 ilustra un ejemplo de arquitectura de bastidor de software para un nodo de equilibrador de carga 110 de acuerdo con al menos algunas realizaciones, y no pretende ser limitante. En este ejemplo de arquitectura de bastidor de software, el nodo de equilibrador de carga 110 se ejecuta dentro de un único proceso de tecnología Java™ 1102 que utiliza la tecnología Java Native Interface (JNI™) 1104 para gestionar una capa de código nativo que puede incluir el código nativo servidor de equilibrador de carga 1106 y el código de procesamiento de paquetes de núcleo 1108, por ejemplo, la tecnología del código de kit de desarrollo de plano de datos tecnología Intel™ (DPDK). El código nativo puede interactuar con dos controladores de interfaz de red (NIC 1114A y 1114B). Un primer NIC (NIC 1114A) puede mirar hacia el "norte"; es decir, hacia el enrutador periférico 104. Un segundo NIC (NIC 1114B) puede mirar hacia el "sur"; es decir, hacia los nodos servidor 130. En al menos alguna realización, los NIC 1114A y 1114B pueden no mantener bastidores TCP. Por lo tanto, al menos algunas realizaciones pueden incluir un tercer NIC 1114C que soporta conexiones TCP de manera que el nodo de equilibrador de carga 110 puede comunicarse con procesos a través de un plano de control, y viceversa. Como alternativa, en algunas realizaciones, sólo el primer NIC orientado hacia el norte 1114A y el segundo NIC orientado hacia el sur 111B pueden implementarse en el nodo de equilibrador de carga 110, y el segundo NIC orientado hacia el sur 1114B puede implementar un bastidor TCP a través de la cual el nodo de equilibrador de carga 110 puede comunicarse con procesos a través del plano de control. El nodo de equilibrador de carga 110 también incluye software de tecnología de sistema operativo (OS) 1112, por ejemplo, un núcleo Linux™, y una capa de software de tecnología de máquina virtual Java (JVM™) 1110 por encima del software de tecnología OS 1112 y la tecnología JNI 1104.

En al menos algunas realizaciones, los nodos de equilibradores de carga 110 en el sistema de equilibrado de carga distribuido pueden necesitar cada uno procesar simultáneamente muchos flujos de datos a altas velocidades de paquetes. En al menos algunas realizaciones, para alcanzar el nivel requerido de rendimiento, los nodos de equilibradores de carga 110 pueden aprovechar la tecnología del kit de desarrollo de plano de datos Intel™ (DPDK) para el procesamiento de paquetes de alto rendimiento. La tecnología DPDK permite a un programa de espacio de usuario leer/escribir paquetes directamente a y desde un controlador de interfaz de red (NIC) y deriva las múltiples capas del bastidor de red del núcleo de Linux (excepto el controlador NIC de base Linus ixgbe). El enfoque DPDK para el procesamiento de paquetes rechaza la entrada basada en el procesador de interrupciones a favor de los núcleos dedicados de la CPU que realizan un sondeo directamente del hardware del NIC en un bucle ocupado. Este enfoque puede permitir tasas de paquetes mucho más altas, a expensas del aumento de la salida térmica mediante la ejecución continua de núcleos de CPU dedicados en un bucle ocupado. La tecnología DPDK también puede

proporcionar herramientas para el procesamiento de paquetes, incluyendo la gestión del núcleo de la CPU, las colas sin bloqueo, las agrupaciones de memoria y las primitivas de sincronización. Como se muestra en la figura 24, en la tecnología DPDK, un núcleo de CPU dedicado 600 se puede utilizar para cada tarea particular, y el trabajo se pasa desde un núcleo de CPU 600A a otro núcleo de CPU 600B usando colas sin bloqueo 602.

Las colas de DPDK 602 se pueden implementar usando búferes rápidos de potencia de dos anillos, y pueden soportar variantes individuales y múltiples de productores/consumidores. Las múltiples variantes de productor/consumidor no están realmente libres de bloqueo, ya que contienen un ciclo de comparación e intercambio (CAS) para sincronizar el acceso. Todas las memorias de memoria intermedia de paquetes pueden estar preasignadas en agrupaciones de memoria, de manera que sólo se lean y escriban punteros de las memorias intermedias en las colas 602. Las agrupaciones de memoria se pueden implementar como colas, pueden optimizarse para distribuir memoria a través del canal de memoria y el rango, y pueden soportar la asignación optimizada de acceso de memoria no uniforme (NUMA). En al menos algunas realizaciones, las memorias intermedias de paquetes pueden utilizar un método tal como un paradigma de Mbuf que asigna en exceso espacio suficiente y espacio de cola en cada memoria intermedia de paquetes para soportar operaciones de encapsulación/decapsulación que pueden añadir/eliminar encabezamientos de capa de red externa sin requerir copias de memoria intermedia.

En al menos algunas realizaciones de los nodos de equilibradores de carga 110, puede implementarse una arquitectura de procesamiento de paquetes de núcleo que aprovecha la tecnología DPDK. Cada nodo de equilibrador de carga 110 puede incluir al menos un procesador de paquetes multinúcleo implementado de acuerdo con la arquitectura de procesamiento de paquetes de núcleo. La arquitectura de procesamiento de paquetes de núcleo puede utilizar un paradigma de productor único/consumidor único para el flujo de paquetes a través de las colas y núcleos del procesador de paquetes multinúcleo. En este paradigma, cada cola introduce a uno y sólo un núcleo, y cada núcleo se transmite a uno y sólo un núcleo para cada núcleo diferente al que se suministran los paquetes. Además, la memoria utilizada por los núcleos en el procesador de paquetes multinúcleo no se comparte; cada núcleo tiene su propia región de memoria separada. Por lo tanto, no hay memoria ni compartición de cola entre núcleos, no hay contención de memoria o cola, y no hay necesidad de mecanismos de compartición de memoria o cola tal como solicitud de propiedad (RFO) o comparar y intercambiar (CAS). Las figuras 25 y 26 ilustran procesadores de paquetes multinúcleo ejemplares implementados de acuerdo con la arquitectura de procesamiento de paquetes de núcleo.

La figura 25 ilustra un procesador de paquetes multinúcleo ejemplar implementado de acuerdo con la arquitectura de procesamiento de paquetes de núcleo que aprovecha la tecnología DPDK para procesar flujos de datos, de acuerdo con al menos algunas realizaciones. La arquitectura de procesamiento de paquetes de núcleo se puede implementar como un procesador de paquetes multinúcleo según un paradigma de único productor/único consumidor. En al menos algunas realizaciones, como se ilustra en las figuras 23, los nodos de equilibradores de carga 110 tienen cada uno dos controladores de interfaz de red (NIC) - un NIC orientado al norte 1114A que está orientado hacia la red de frontera/enrutador periférico 104 y un NIC orientado al sur 1114B que está orientado hacia los nodos de red/servidor de producción 130. En al menos algunas realizaciones, los NIC 1114 pueden ser NIC de 10 Gpbs. La mayoría de los paquetes que fluyen a través de un nodo de equilibrador de carga 110 se reciben en uno de estos dos NIC (NIC 1114A o 1114B), se procesan (por ejemplo, se encapsulan o se decapsulan) y se transmiten al otro NIC (NIC 1114B o 1114A).

Haciendo referencia a la figura 25, en al menos algunas realizaciones, un nodo de equilibrador de carga 110 hace girar dos núcleos de CPU, un núcleo de recepción (RX) 610 y un núcleo de transmisión (TX) 630, para cada NIC 1114. El nodo de equilibrador de carga 110 también gira varios núcleos de trabajo 620 que procesan paquetes para ambos NIC 1114 en ambas direcciones; en este ejemplo se usan cuatro núcleos de trabajo 620A a 620D. Los núcleos de recepción 610 leen lotes de paquetes entrantes desde sus colas de entrada a medida que llegan al NIC 1114 y distribuyen los paquetes a los núcleos de trabajo 620 que realizan la mayor parte del trabajo para cada paquete, suministrando cada núcleo de recepción 610 paquetes en una cola de entrada de trabajo respectivo 612 para cada núcleo de trabajo 620. En al menos alguna realización, un núcleo de recepción 610 puede realizar una técnica de "flujo-hash" de capa 4 en cada paquete entrante (similar a la técnica de enrutamiento multitrayecto de hash por flujo que puede usarse por el enrutador periférico 104 como se ha descrito previamente) para distribuir los paquetes a los núcleos de trabajo 620 mientras se asegura que cualquier conexión de cliente particular (distinguida por su dirección IP y puerto) será procesada por el mismo núcleo de trabajo 620. Esto significa que cada núcleo de trabajado 620 siempre puede ver el mismo subconjunto de paquetes, y elimina la contención en los datos de estado administrados por el núcleo de trabajo 620 de manera que no se requieren bloqueos. Los punteros a los paquetes recibidos pueden distribuirse a través de las colas de trabajo 622 que los núcleos de trabajo 620 supervisan continuamente para una nueva entrada. Los núcleos de trabajo 620 son responsables de gestionar el estado (por

ejemplo, el nodo servidor asignado 130) para cada conexión, y pueden realizar una encapsulación UDP o decapsulación en el paquete antes de reenviar el paquete a una de sus colas de salida 632. Los núcleos de transmisión 630 se repiten a través de las colas salientes 632 del núcleo de trabajo 620 y escriben los paquetes de salida a su correspondiente NIC 1114 como aparecen en las colas 632.

La figura 26 ilustra otro procesador de paquetes multinúcleo ejemplar implementado de acuerdo con la arquitectura de procesamiento de paquetes de núcleo que aprovecha la tecnología DPDK para procesar flujos de datos, de acuerdo con al menos algunas realizaciones. La arquitectura de procesamiento de paquetes de núcleo se puede implementar como un procesador de paquetes multinúcleo según un paradigma de único productor/único consumidor. En al menos algunas realizaciones, además de procesar los flujos TCP de cliente de alto rendimiento, la arquitectura de núcleo DPDK en un nodo de equilibrador de carga 110 también puede usarse para enviar y recibir paquetes en los NIC orientados al norte y al sur 1114 para otros protocolos tales como ARP, DHCP y BGP. En la realización mostrada en la figura 26, un núcleo de trabajo 620A está dedicado a manejar los paquetes para estos otros protocolos. Este núcleo de trabajo 620A puede denominarse como un núcleo de trabajo "lento", ya que el procesamiento de estos paquetes generalmente se produce a una velocidad más lenta que los flujos TCP de cliente, mientras que los otros núcleos de trabajo 620B-620D que procesan solamente los flujos TCP de cliente pueden denominarse como núcleos de trabajo rápidos. Los núcleos de recepción 610A y 610B que manejan los paquetes entrantes en los NIC orientados al norte y al sur 1114, respectivamente, pueden identificar paquetes que han de manejarse por el núcleo de trabajo lento 620A y dirigir los paquetes a las colas de entrada 622 para el núcleo de trabajo lento 620A. El núcleo de trabajo lento 620A también puede controlar una cola de entrada 622 para paquetes generados por Java/JNI y una cola de salida 634 para paquetes de salida a Java/JNI. El núcleo de trabajo lento 620A también transmite a una cola de entrada 622 para cada uno de los núcleos de trabajo rápido 620B a 620D de manera que el núcleo de trabajo lento 620A puede enviar paquetes a cada uno de los núcleos de trabajo rápido 620B a 620D, por ejemplo, paquetes de publicación de conexión. El núcleo de trabajo lento 620A también tiene una cola de salida 632 que alimenta cada uno de los núcleos de transmisión 630A y 630B.

En al menos algunas realizaciones, la tercera cola de entrada 622 de cada núcleo de trabajador rápido 620B a 620D es una cola de salida del núcleo de trabajo lento 620A. En al menos algunas realizaciones, esta tercera cola de entrada 622 puede usarse, por ejemplo, para recibir y procesar paquetes de publicación de conexión, conteniendo cada uno información de estado de conexión, por las colas de trabajo rápido 620B a 620D. Para al menos algunos de estos paquetes de publicación de conexión, puede no haber salida a los núcleos de transmisión 630. En su lugar, la información de estado de conexión en los paquetes puede consumirse por el núcleo de trabajo rápido 620, por ejemplo, actualizando el estado almacenado para uno o más flujos de paquetes que el respectivo núcleo de trabajo rápido 620 mantiene. Por lo tanto, las colas de salida del núcleo de trabajo lento 620A que entran en los núcleos de trabajo rápido 620B a 620D pueden proporcionar una trayectoria distinta de una cola de entrada 622 directamente desde un núcleo de recepción 610 para actualizar los estados almacenados de los núcleos de trabajo rápido.

En al menos algunas realizaciones, los procesadores de paquetes multinúcleo de las figuras 25 y 26 pueden filtrar paquetes entrantes y sólo procesar y transmitir paquetes que son válidos. Por ejemplo, en al menos algunas realizaciones, los núcleos de recepción 610 pueden filtrar paquetes que son de un protocolo no soportado por ninguno de los núcleos de trabajo 620 y, por lo tanto, no envían los paquetes a los núcleos de trabajo 620. En al menos algunas realizaciones, los núcleos de trabajo 620, al procesar paquetes, pueden cada uno analizar primero los paquetes leídos de sus respectivas colas de entrada de trabajo 622 para determinar si los paquetes deben ser aceptados para su procesamiento y transmisión adicional a los núcleos de transmisión 630, y sólo pueden completar el procesamiento y transmisión de paquetes a los núcleos de transmisión 630 que se aceptan; los paquetes no aceptados pueden ser desechados. Por ejemplo, los núcleos de trabajo 620 pueden mirar la información de dirección para cada paquete y aceptar sólo paquetes que están dirigidos a direcciones válidas que están siendo equilibradas en carga, descartando cualquier otro paquete.

Gestión de datos de protocolo de puerta de enlace de frontera (BGP)

En al menos algunas realizaciones, los flujos de paquetes asociados con un cliente BGP dentro y fuera de la arquitectura de núcleo pueden manejarse como se indica a continuación. Puesto que los NIC 1114A y 1114B no están unidos al núcleo de Linux, la conexión TCP con el enrutador periférico 104 se intercepta por la arquitectura de núcleo como se ilustra en la figura 26 y se procesa por el núcleo de trabajo lento 622A, que pasa los paquetes BGP al espacio Java a través de la cola de salida 634. Estos paquetes TCP se procesan adicionalmente por uno o más módulos en el nodo del equilibrador de carga 110 antes de entregarse al cliente BGP, incluyendo el procesamiento por el núcleo de Linux para gestionar la conexión TCP y trasladar efectivamente los paquetes en una corriente TCP. Este diseño permite que el cliente BGP se escriba utilizando bibliotecas estándar de socket Java TCP.

La figura 27 ilustra el procesamiento de paquetes TCP de BGP entrantes mediante un proceso de nodo de equilibrador de carga (LB) 650, de acuerdo con al menos algunas realizaciones. Un paquete del enrutador periférico 104 llega al NIC orientado al norte 640 y entra en la cola de entrada 640 para el núcleo de recepción 652. El núcleo de recepción 652 lee el paquete de la cola 640, identificado el paquete como un paquete BGP, y pone el paquete en una cola de entrada 654 para el núcleo de trabajador lento 656. El núcleo de trabajo lento 656 valida el paquete y lo coloca en la cola de salida JNI 658. El receptor del paquete JNI 660 lee el paquete de la cola 658 a través de JNI, gestiona las direcciones de origen/destino, y escribe el paquete en un socket básico 644. El núcleo de Linux 646 recibe el paquete básico, lo maneja de acuerdo con el protocolo TCP y adjunta los datos de carga útil al socket TCP InputStream. Los datos del paquete se envían entonces al socket TCP Java en el cliente BGP 662.

La figura 28 ilustra el procesamiento de paquetes TCP de BGP salientes mediante un proceso de nodo de equilibrador de carga (LB) 650, de acuerdo con al menos algunas realizaciones. El cliente BGP 662 escribe datos en un socket TCP Java del núcleo de Linux 646. El núcleo de Linux 646 maneja los datos de acuerdo con el protocolo TCP y convierte los datos en uno o más paquetes TCP. En al menos algunas realizaciones, el paquete o paquetes TCP corresponden a una regla iptables de 127.x.x.x. El uno o más paquetes TCP se colocan en una cola de salida 648, por ejemplo una cola LOCAL_OUT de Netfilter. Un hilo Java del receptor de paquetes JNI 670 que supervisa la cola 648 a través de JNI recibe el paquete o paquetes TCP y marca cada NF_STOLEN para hacer que el núcleo 646 se olvide de ellos. El hilo de Java maneja las direcciones de origen/destino y añade el paquete o paquetes a una cola de entrada JNI 672 para el núcleo de trabajo lento 656 a través de JNI. El núcleo de trabajo lento 656 recibe el paquete o paquetes TCP de su cola de entrada JNI 672 y pone los paquetes en la cola de salida 664 para el núcleo de transmisión 666 del NIC orientado al norte 640. El núcleo de transmisión 666 lee el paquete o paquetes TCP de su cola de entrada 664 y los escribe en el NIC orientado al norte 640. Los paquetes TCP se envían por el NIC 640 al enrutador periférico 104.

Simulación y pruebas de equilibradores de carga distribuidos

El equilibrador de carga descrito en el presente documento es un sistema distribuido que requiere la interacción de muchos componentes independientes (por ejemplo, enrutadores, nodos de equilibrador de carga, módulos de equilibrador de carga, etc.). Para realizar pruebas de los componentes distribuidos, la lógica y los protocolos, así como para simular escenarios tales como fallos de nodos, caídas de mensajes y retrasos, se describen realizaciones de un sistema de prueba que permiten que el equilibrador de carga distribuido se ejecute en un único proceso donde las interacciones pueden probarse sin requerir que el código se despliegue a múltiples anfitriones en una topología de red compleja (por ejemplo, una red de producción). Para ello, se describe un mecanismo de software denominado bus de mensajes que permite configurar y ejecutar múltiples componentes de equilibrador de carga en o como un único proceso; el único proceso se puede ejecutar en un único sistema anfitrión. El mecanismo del bus de mensajes permite que el sistema de equilibrador de carga distribuido se pruebe como un único proceso, por ejemplo, en un único sistema anfitrión, mientras que a los componentes del equilibrador de carga (por ejemplo, los nodos de equilibradores de carga y módulos de equilibrador de carga) parece que se están ejecutando en una red de producción real.

El bus de mensajes proporciona una estructura que permite que el equilibrador de carga distribuido funcione como un único proceso. Cada una de una o más capas de bus de mensajes en el proceso simula un segmento de red (por ejemplo, Ethernet) entre componentes del equilibrador de carga distribuido. Los componentes de software del sistema de equilibrador de carga distribuido no tienen que escribirse de una manera especial para permitir que los componentes funcionen dentro del entorno del bus de mensajes. En cambio, la estructura de bus de mensajes proporciona un componente (que puede denominarse NIC de bus de mensajes o adaptador de paquetes) que intercepta los paquetes que producen los componentes del sistema de equilibrador de carga distribuido, dirige los paquetes a la red simulada proporcionada por una capa de bus de mensaje en lugar de en una red física real, y entrega los paquetes a los componentes de destino. Las capas de bus de mensajes no implementan el bastidor o bastidores TCP/IP para las comunicaciones entre los componentes. En su lugar, las capas de bus de mensajes interactúan con el sistema operativo (SO) del sistema anfitrión y utilizan el bastidor TCP/IP del sistema anfitrión. Las capas del bus de mensajes aprovechan el bastidor TCP/IP proporcionada por el sistema operativo para convertir las corrientes TCP que esperan los clientes y servidores y de los paquetes individuales que el bus de mensajes intercepta y entrega.

En al menos algunas realizaciones, para interactuar con el bus de mensajes, se pueden proporcionar componentes equilibradores de carga con al menos un controlador de interfaz de red de bus de mensajes (NIC), cada uno con una dirección válida de control de acceso a medios (MAC), que envía paquetes a y recibe paquetes desde el entorno de red simulada de bus de mensajes en lugar de hacia y desde una red física. Un NIC de bus de mensajes es un controlador de interfaz de red virtual que se conecta al bus de mensajes en lugar de a una red física. Cada

componente de equilibrador de carga que necesita comunicarse a través del bus de mensajes requiere al menos un NIC de bus de mensajes. Un NIC de bus de mensajes sirve como una salida de tubería al bus de mensajes y como una entrada de conducto al componente. Los componentes pueden instanciar múltiples interfaces de red de bus de mensajes a cada NIC de bus de mensajes.

Una interfaz de red de bus de mensajes es un mecanismo para que los componentes se conecten a un bus de mensajes a través de un NIC de bus de mensajes. Una interfaz de red de bus de mensajes puede ser también una interfaz de configuración de interfaz (ifconfig) en tecnología Linux, con la diferencia de que la interfaz de red de bus de mensajes se conecta al bus de mensajes en lugar de a una red física. Una interfaz de red de bus de mensajes tiene una dirección IP, y se encuentra encima de un NIC de bus de mensajes. La interfaz de red de bus de mensajes expone una interfaz de fuente de paquetes, que puede usarse por el componente para recibir paquetes del bus de mensajes, y una interfaz colectora de paquetes que puede ser utilizada por el componente para enviar paquetes al bus de mensajes.

Cada nodo de equilibrador de carga procesa paquetes de red individuales que se entregan y se enfrían a través de una implementación de las interfaces de origen de paquetes y colectoras de paquetes. Cuando se ejecutan en el entorno del bus de mensajes, estas interfaces se implementan mediante la interfaz de red del bus de mensajes que añade o elimina los encabezados de Ethernet de la capa 2 (para los nodos del equilibrador de carga que esperan que esto sea realizado por el bastidor de red del núcleo). En un entorno de producción como se muestra en la figura 29, la implementación de las interfaces de origen de paquetes y colectoras de paquetes recibe y transmite paquetes en una interfaz de red real. En un entorno de bus de mensajes como se muestra en la figura 30, la implementación de las interfaces de origen de paquetes y colectoras de paquetes reciben paquetes desde y transmiten paquetes a una capa o capas de bus de mensajes.

En aras de la simplicidad, un NIC de bus de mensajes y una interfaz de bus de mensajes se pueden denominar colectivamente como un adaptador de paquete de bus de mensajes, o simplemente un adaptador de paquetes. Véanse, por ejemplo, las figuras 31 y 32.

La figura 29 ilustra un sistema de equilibrado de carga que incluye un equilibrador de carga distribuido 700 en un entorno de producción, de acuerdo con al menos algunas realizaciones. El equilibrador de carga 700 se ha simplificado para esta descripción. El equilibrador de carga 700 puede conectarse a los clientes 742 en una red externa 740 a través de un enrutador de frontera 702 de una instalación de red, tal como un centro de datos que implementa el equilibrador de carga 700. El equilibrador de carga 700 incluye varios tipos de componentes - al menos un enrutador periférico 704, dos o más nodos de equilibrador de carga (LB) 710, dos o más módulos de equilibrador de carga (LB) 732 implementados cada uno en un nodo servidor separado (no mostrado), uno o más componentes de red que forman el tejido 720, tal como enrutadores o conmutadores, y al menos en algunas realizaciones un servicio de configuración 722. En al menos algunas realizaciones, cada componente del equilibrador de carga 700 puede implementarse como o en un dispositivo informático separado, tal como un dispositivo informático montado en bastidor de productos básicos.

La figura 30 ilustra un sistema de prueba de equilibrador de carga distribuido 800 que incorpora un mecanismo de bus de mensajes que permite configurar y ejecutar múltiples componentes del sistema distribuido de equilibrado de carga en o como un único proceso, de acuerdo con al menos algunas realizaciones. En el equilibrador de carga 700 mostrado en la figura 29, cada componente de software de equilibrador de carga se instala y se ejecuta en un dispositivo informático separado (por ejemplo, el software de equilibrador de carga en los nodos de equilibrador de carga 710, y los módulos de equilibrador de carga 732 en los nodos servidor). Para permitir que estos componentes de software de equilibrador de carga se ejecuten en un único proceso, cada componente de software de equilibrador de carga (mostrado como nodos de equilibrador de carga (LB) 810 y módulos de equilibrador de carga (LB) 832 en la figura 30) puede incluir código que abstrae la conectividad de red de los componentes para que los paquetes dentro y fuera del componente de software de balanceo de carga también puedan ser interceptados y enrutados a través del mecanismo de bus de mensajes en lugar de ser enviados y recibidos en una red física.

En al menos algunas realizaciones, en el sistema de prueba de equilibrador de carga distribuido 800, el mecanismo de bus de mensajes no implementa un bastidor o bastidores TCP para comunicaciones entre los componentes. En su lugar, el mecanismo del bus de mensajes se interconecta con el sistema operativo (SO) del sistema anfitrión y utiliza el bastidor TCP del sistema anfitrión. En al menos algunas realizaciones, la funcionalidad de bus de mensajes se vincula al núcleo (por ejemplo, el núcleo de Linux) del SO del sistema anfitrión por debajo de la capa de usuario a través de las tablas de IP, una funcionalidad del núcleo. La funcionalidad del bus de mensajes se conecta a las tablas IP en el nivel del núcleo, intercepta los paquetes y envía los paquetes al proceso del bus de mensajes para su enrutamiento.

Como se muestra por el enrutador periférico simulado 862 y el tejido simulado 864 en la figura 30, la funcionalidad de los componentes físicos de red (por ejemplo, el enrutador periférico 704 y el tejido 720 de la figura 29) puede simularse en software, como clientes 860, servidores 834 y servicio de configuración 866. Sin embargo, se debe observar que en al menos algunas realizaciones se pueden usar servidores reales 834 en lugar de servidores simulados en los sistemas de prueba de equilibrador de carga distribuido 800. Las capas de bus de mensajes 850 en la figura 30 reemplazan la infraestructura de red física. Por lo tanto, los componentes de software del equilibrador de carga (nodos de equilibrador de carga 810 y módulos de equilibrador de carga 832) pueden ejecutarse en el sistema de prueba del equilibrador de carga 800 sin tener en cuenta que no están ejecutándose en un entorno de red de producción como se muestra en la figura 29.

Algunos componentes (por ejemplo, enrutadores simulados) pueden estar conectados a más de una capa de bus de mensajes 850 para pasar paquetes a y recibir paquetes de diferentes capas de bus de mensajes 850 que simulan segmentos de red.

El mecanismo de bus de mensajes implementado en las capas de bus de mensajes 850 del sistema de prueba de equilibrado de carga distribuido 800 simula el "cable" de un segmento de red. En al menos algunas realizaciones, el mecanismo de bus de mensajes entrega paquetes a componentes de destino en el sistema de prueba de equilibrado de carga distribuida 800 basado en las direcciones MAC de los componentes. Por lo tanto, cada componente de software de equilibrador de carga (nodos de equilibrador de carga 810 y módulos de equilibrador de carga 832) proporciona una dirección MAC a la capa o capas de bus de mensajes 850 a las que está conectada para que el componente de software de equilibrador de carga pueda recibir paquetes que son enviados a éste de otros componentes en el sistema de prueba de equilibrado de carga distribuido 800. Adaptadores de paquete de bus de mensajes

Las figuras 31 y 32 ilustran adaptadores de paquetes de bus de mensajes, de acuerdo con al menos algunas realizaciones. En al menos algunas realizaciones, cada componente de software de equilibrador de carga (LB) procesa paquetes de red individuales que se entregan y se envían a través de una implementación de las interfaces PacketSource y PacketSink. Con referencia a la figura 31, cuando se ejecutan en el sistema de prueba de equilibrado de carga distribuido 800, estas interfaces (mostradas como interfaz de fuente de paquetes 862 e interfaz colectora de paquetes 864) pueden implementarse mediante un adaptador de paquetes 860 entre la capa de bus de mensajes 850 y el componente de software de equilibrador de carga 870 que añade o elimina los encabezados de Ethernet de la capa 2 para los componentes de software de equilibrador de carga 870 que esperan que esto sea realizado por el bastidor de red del núcleo. En el entorno de producción ilustrado en la figura 29, la implementación de PacketSource y PacketSink para los componentes de software de equilibrador de carga recibe y transmite los paquetes en las interfaces de red reales de los dispositivos físicos en los que se implementan los componentes.

Haciendo referencia a la figura 31, en al menos algunas realizaciones, cuando un componente de software de equilibrador de carga 870 transmite un paquete, el hilo de ejecución que llama a un método de paquete de envío de la interfaz colectora de paquetes 864 atraviesa una cadena de funciones dentro del adaptador de paquete 860 y también dentro de la capa de bus de mensajes 850 para entregar eventualmente el paquete al componente de destino añadiendo el paquete a la cola de entrada de ese componente. En al menos algunas realizaciones, cuando un componente de software equilibrador de carga 870 recibe un paquete, el componente de software equilibrador de carga 870 llama a un método de paquete de recepción de la interfaz de origen de paquetes 862 y lee paquetes desde su cola de entrada. En al menos algunas realizaciones, el mecanismo de bus de mensajes no requiere ningún hilo adicional propio para entregar paquetes.

Conductos de paquetes de bus de mensajes

Haciendo referencia a la figura 32, en al menos algunas realizaciones, el lado del bus de mensajes 850 de la interfaz de origen de paquetes 862 y la interfaz colectora de paquetes 864 proporciona una característica de canalización de paquetes. Cuando un componente de software de equilibrador de carga 870 envía un paquete a través de la interfaz colectora de paquetes 864, los datos de paquete pueden atravesar una serie de fases (canalización de paquetes 880) antes de llegar a la capa de bus de mensajes 850. Estas fases pueden modificar el paquete, eliminar el paquete, duplicar el paquete, retrasar el paquete, etc. Una vez que un paquete atraviesa la canalización de paquetes 880 y la capa de bus de mensajes 850 selecciona un componente de destino 870, también se puede atravesar una segunda serie de fases de canalización (canalización de paquetes 882) asociadas con el componente de destino 870 antes de que se añada el paquete a la cola de entrada del componente de destino 870.

Entornos de red de proveedor ejemplares

Esta sección describe entornos de red de proveedor ejemplares en los que pueden implementarse realizaciones de los métodos y aparatos de equilibrado de carga distribuidos. Sin embargo, estos entornos de red de proveedores ejemplares no pretenden ser limitantes.

La figura 33A ilustra un entorno de red de proveedor ejemplar, de acuerdo con al menos algunas realizaciones. Una red de proveedor 1900 puede proporcionar una virtualización de recursos a los clientes a través de uno o más servicios de virtualización 1910 que permiten a los clientes acceder, comprar, alquilar u obtener de cualquier otra forma las instancias 1912 de recursos virtualizados, incluyendo, pero sin limitación, recursos de computación y almacenamiento, implementados en dispositivos dentro de la red o redes de proveedores en uno o más centros de datos. Las direcciones IP privadas 1916 pueden estar asociadas con las instancias de recursos 1912; las direcciones IP privadas son las direcciones de red internas de las instancias de recursos 1912 en la red de proveedores 1900. En algunas realizaciones, la red de proveedores 1900 puede proporcionar también direcciones IP públicas 1914 y/o rangos de direcciones IP públicas (por ejemplo, direcciones de Protocolo de Internet versión 4 (IPv4) o Protocolo de Internet versión 6 (IPv6)) que los clientes pueden obtener del proveedor 1900.

Convencionalmente, la red de proveedor 1900, a través de los servicios de virtualización 1910, puede permitir a un cliente del proveedor de servicios (por ejemplo, un cliente que opera la red de cliente 1950A) asociar dinámicamente al menos algunas direcciones IP públicas 1914 asignadas o dedicadas al cliente con instancias de recursos particulares 1912 asignadas al cliente. La red de proveedor 1900 también puede permitir que el cliente reasigne una dirección IP pública 1914, previamente asignada a una instancia de recurso de computación virtualizada 1912 asignada al cliente, a otra instancia de recurso de computación virtualizada 1912 que también está asignada al cliente. Utilizando las instancias de recursos de computación virtualizadas 1912 y las direcciones IP públicas 1914 proporcionadas por el proveedor de servicios, un cliente del proveedor de servicios tal como el operador de la red de cliente 1950A puede implementar, por ejemplo, aplicaciones específicas del cliente y presentar las aplicaciones del cliente en una red intermedia 1940, tal como Internet. Otras entidades de red 1920 en la red intermedia 1940 pueden entonces generar tráfico a una dirección IP pública de destino 1914 publicada por la red de cliente 1950A; el tráfico se enruta al centro de datos del proveedor de servicios, y en el centro de datos se enruta, a través de un sustrato de red, a la dirección IP privada 1916 de la instancia de recursos de computación virtualizada 1912 asignada actualmente a la dirección IP pública de destino 1914. De forma similar, el tráfico procedente de la instancia de recursos de computación virtualizada 1912 puede enrutarse a través del substrato de red de nuevo a través de la red intermedia 1940 a la entidad de origen 1920.

Las direcciones IP privadas, como se usan en el presente documento, se refieren a las direcciones de red internas de instancias de recurso en una red de proveedor. Las direcciones IP privadas sólo se pueden enrutar dentro de la red del proveedor. El tráfico de red que se origina fuera de la red de proveedores no se enruta directamente a direcciones IP privadas; en su lugar, el tráfico utiliza direcciones IP públicas que se asignan a las instancias de recursos. La red de proveedores puede incluir dispositivos o aparatos de red que proporcionan traducción de direcciones de red (NAT) o una funcionalidad similar para realizar la asignación de direcciones IP públicas a direcciones IP privadas y viceversa.

Las direcciones IP públicas, como se usan en el presente documento, son direcciones de red enrutables por Internet que están asignadas a instancias de recursos, ya sea por el proveedor de servicios o por el cliente. El tráfico enrutado a una dirección IP pública se traduce, por ejemplo, mediante una traducción de dirección de red (NAT) 1:1, y se reenvía a la dirección IP privada respectiva de una instancia de recurso.

Algunas direcciones IP públicas pueden asignarse por la infraestructura de red de proveedores a instancias de recursos particulares; estas direcciones IP públicas pueden denominarse direcciones IP públicas estándar, o simplemente direcciones IP estándar. En al menos algunas realizaciones, la asignación de una dirección IP estándar a una dirección IP privada de una instancia de recurso es la configuración de inicio por defecto para todos los tipos de instancia de recurso.

Al menos algunas direcciones IP públicas pueden ser asignadas u obtenidas por clientes de la red de proveedores 1900; un cliente puede asignar entonces sus direcciones IP públicas asignadas a instancias de recursos particulares asignadas al cliente. Estas direcciones IP públicas pueden denominarse direcciones IP públicas de clientes, o simplemente direcciones IP de clientes. En lugar de asignarse por la red de proveedores 1900 a instancias de recursos como en el caso de direcciones IP estándar, las direcciones IP de cliente pueden ser asignadas a instancias de recursos por los clientes, por ejemplo, a través de una API proporcionada por el proveedor de servicios. A diferencia de las direcciones IP estándar, las direcciones IP de clientes se asignan a las cuentas de cliente y pueden reasignarse a otras instancias de recursos por los clientes respectivos según sea necesario o

deseado. Una dirección IP del cliente se asocia con la cuenta de un cliente, no con una instancia de recurso particular, y el cliente controla esa dirección IP hasta que el cliente decide liberarla. A diferencia de las direcciones IP estáticas convencionales, las direcciones IP del cliente permiten que el cliente enmascare instancias de recursos o fallos de de zonas de disponibilidad mediante la reasignación de las direcciones IP públicas del cliente a cualquier instancia de recurso asociada con la cuenta del cliente. Las direcciones IP de cliente, por ejemplo, permiten a un cliente tratar problemas con las instancias de recursos del cliente o con el software reasignando las direcciones IP del cliente a instancias de recursos de reemplazo.

La figura 33B ilustra una implementación de equilibrador de carga distribuida en un entorno de red proveedor ejemplar, como se muestra en la figura 33A, de acuerdo con al menos algunas realizaciones. Una red de proveedor 1900 puede proporcionar un servicio 1910 a los clientes 1960, por ejemplo un servicio de almacenamiento virtualizado. Los clientes 1960 pueden acceder al servicio 1910, por ejemplo, a través de una o más API al servicio 1910, para obtener el uso de recursos (por ejemplo, recursos de almacenamiento o recursos de computación) implementados en múltiples nodos servidor 1990 en una porción de red de producción de la red de proveedores 1900. Los nodos servidor 1990 pueden implementar cada uno un servidor (no mostrado), por ejemplo, un servidor web o un servidor de aplicaciones, así como un módulo de equilibrador de carga local (LB) 1992. Uno o más equilibradores de carga distribuidos 1980 pueden implementarse en una capa de equilibrador de carga entre la red de frontera y la red de producción. El enrutador o enrutadores de frontera 1970 pueden recibir paquetes (por ejemplo, paquetes TCP) en flujos de paquetes desde los clientes 1960 a través de una red intermedia 1940 tal como Internet, y reenviar los paquetes al enrutador o enrutadores periféricos del equilibrador o equilibradores de carga distribuidos 1980 a través de la red de frontera. Los paquetes pueden estar dirigidos a la dirección o direcciones IP pública publicadas por el enrutador o enrutadores periféricos del equilibrador o equilibradores de carga distribuidos 1980. El enrutador periférico de cada equilibrador de carga distribuido 1980 puede distribuir los flujos de paquetes entre los nodos de equilibrador de carga del equilibrador de carga distribuido respectivo 1980. En al menos algunas realizaciones, cada nodo de equilibrador de carga que sirve como nodo de entrada anuncia la misma dirección IP pública al enrutador periférico, y el enrutador periférico distribuye los flujos de paquetes de los clientes 1960 entre los servidores de entrada de acuerdo con una técnica de enrutamiento multitrayecto de hash por flujo, por ejemplo, una técnica de hashing multitrayecto de igual coste (ECMP). Los nodos de equilibradores de carga pueden utilizar el protocolo de conexión descrito en el presente documento para determinar nodos servidor objetivo 1990 para los flujos de paquetes y para facilitar las conexiones entre los servidores y los clientes 1960. Una vez establecida una conexión, los nodos de entrada encapsulan y envían paquetes recibidos para los flujos a los nodos servidor de destino 1990 en la red de producción, mientras que los nodos de rastreador de flujo mantienen el estado para las conexiones. Los módulos de equilibrador de carga 1992 en los nodos servidor 1990 pueden tomar las decisiones sobre si los respectivos servidores en los nodos servidor 1960 aceptan conexiones. Los módulos de equilibrio de carga reciben y decapsulan los paquetes de los nodos de entrada, y envían los paquetes decapsulados (por ejemplo, paquetes TCP) a los servidores respectivos en los nodos servidor 1990. Los módulos de equilibrador de carga 1992 pueden también seleccionar nodos de equilibrador de carga como nodos de salida para los flujos de paquete, y encapsular y enviar paquetes salientes para los flujos a los nodos de salida seleccionados a través de la red de producción. Los nodos de egreso a su vez decapsulan los paquetes y envían los paquetes decapsulados a través de la red de frontera para su entrega a los respectivos clientes 1960.

La figura 34A ilustra una implementación de bastidor físico ejemplar del equilibrador de carga distribuido y nodos servidor de acuerdo con al menos algunas realizaciones, y no pretende ser limitante. En al menos algunas realizaciones, pueden implementarse diversos componentes del equilibrador de carga distribuido en o como dispositivos de computación montados en bastidor de productos básicos. El bastidor 190 puede incluir múltiples dispositivos de computación que sirven cada uno como un nodo de equilibrador de carga (nodos LB 110A-110F), y múltiples dispositivos de computación que sirven cada uno como un nodo servidor (nodos servidor 130A-130L). El bastidor 190 también puede incluir al menos un enrutador periférico 104, uno o más dispositivos de red montados en bastidor (enrutadores, conmutadores, etc.) que forman el tejido 120, y uno o más componentes 180 (otros dispositivos de red, paneles de conexión, fuentes de alimentación, sistemas de refrigeración, buses, etc.). Una instalación de red 100 tal como un centro de datos o centros que implementan la red de proveedor 1900 de las figuras 33A y 33B puede incluir uno o más bastidores 190.

La figura 34B ilustra otra implementación de bastidor físico ejemplar del equilibrador de carga distribuido y nodos servidor de acuerdo con al menos algunas realizaciones, y no pretende ser limitante. La figura 34B muestra los nodos LB 110 y los nodos servidor 130 implementados como dispositivos de computación montados en bastidor, por ejemplo, servidores blade, en el bastidor 190.

La figura 35 ilustra un entorno de red ejemplar en el que uno, dos o más equilibradores de carga distribuidos pueden implementarse en una red, con los nodos servidor implementados por separado, según al menos algunas

realizaciones. En este ejemplo, se muestran dos equilibradores de carga distribuidos 1980A y 1980B. Los equilibradores de carga distribuidos 1980 pueden recibir cada uno flujos de paquetes de los clientes 1960 a través de la red de frontera y realizar los métodos de equilibrado de carga descritos en el presente documento para distribuir los flujos de paquetes a través de múltiples nodos servidor 1990. En algunas implementaciones, cada equilibrador de carga distribuido 1980 puede ser una implementación de bastidor similar a los bastidores 190 mostrados en las figuras 34A y 34B, pero sin los nodos servidor instalados en los bastidores del equilibrador de carga. Los nodos servidor 1990 pueden ser dispositivos de computación montados en bastidor tales como servidores Blade instalados en uno o más bastidores separados dentro del centro de datos. En algunas implementaciones, los nodos servidor 1990 pueden implementar dos o más servicios diferentes proporcionados por la red de proveedores, con cada servicio frente a uno diferente o más de los equilibradores de carga 1980.

Sistema ilustrativo

En al menos algunas realizaciones, un servidor que implementa una porción o todos los métodos y aparatos de equilibrado de carga distribuidos como se describe en el presente documento, puede incluir un sistema informático de uso general que incluye o está configurado para acceder a uno o más medios accesibles por ordenador, tal como el sistema informático 2000 ilustrado en la figura 36. En la realización ilustrada, el sistema informático 2000 incluye uno o más procesadores 2010 acoplados a una memoria de sistema 2020 por medio de una interfaz de entrada/salida (I/O) 2030. El sistema informático 2000 incluye además un interfaz de red 2040 acoplada a la interfaz I/O 2030.

En diversas realizaciones, el sistema informático 2000 puede ser un sistema uniprocesador que incluye un procesador 2010, o un sistema multiprocesador que incluye varios procesadores 2010 (por ejemplo, dos, cuatro, ocho u otro número adecuado). Los procesadores 2010 pueden ser cualquier procesador adecuado capaz de ejecutar instrucciones. Por ejemplo, en diversas realizaciones, los procesadores 2010 pueden ser procesadores de propósito general o integrados que implementan cualquiera de una diversidad de arquitecturas de conjuntos de instrucciones (ISA), tales como las ISA x86, PowerPC, SPARC o MIPS ISA, o cualquier otra ISA adecuada. En sistemas multiprocesador, cada uno de los procesadores 2010 puede implementar, comúnmente, pero no necesariamente, la misma ISA.

La memoria de sistema 2020 puede estar configurada para almacenar instrucciones y datos accesibles por uno o más procesadores 2010. En diversas realizaciones, la memoria de sistema 2020 puede implementarse usando cualquier tecnología de memoria adecuada, tal como memoria de acceso aleatorio estática (SRAM), dinámica síncrona RAM (SDRAM), memoria no volátil/tipo Flash, o cualquier otro tipo de memoria. En la realización ilustrada, las instrucciones de programa y los datos que implementan una o más funciones deseadas, tales como los métodos, técnicas y datos descritos anteriormente para los métodos y aparatos de equilibrado de carga distribuidos, se muestran almacenados dentro de la memoria de sistema 2020 como el código 2024 y los datos 2026.

En una realización, la interfaz I/O 2030 puede configurarse para coordinar el tráfico I/O entre el procesador 2010, la memoria de sistema 2020, y cualquier dispositivo periférico en el dispositivo, incluyendo la interfaz de red 2040 u otras interfaces periféricas. En algunas realizaciones, la interfaz I/O 2030 puede realizar cualquier protocolo necesario, sincronización u otras transformaciones de datos para convertir señales de datos de un componente (por ejemplo, memoria de sistema 2020) en un formato adecuado para su uso por otro componente (por ejemplo, el procesador 2010). En algunas realizaciones, la interfaz I/O 2030 puede incluir soporte para dispositivos conectados a través de diversos tipos de buses periféricos, tales como una variante del estándar de bus de interconexión de componentes periféricos (PCI) o el estándar de bus serie universal (USB), por ejemplo. En algunas realizaciones, la función de la interfaz I/O 2030 puede dividirse en dos o más componentes separados, tales como un puente norte y un puente sur, por ejemplo. Además, en algunas realizaciones, puede incorporarse directamente en el procesador 2010 una parte o la totalidad de la funcionalidad de la interfaz I/O 2030, tal como una interfaz con la memoria del sistema 2020.

La interfaz de red 2040 puede configurarse para permitir el intercambio de datos entre el sistema informático 2000 y otros dispositivos 2060 unidos a una red o redes 2050, tales como otros sistemas o dispositivos informáticos como se ilustra en las figuras 1 a 35, por ejemplo. En diversas realizaciones, la interfaz de red 2040 puede soportar la comunicación a través de cualesquiera redes de datos generales alámbricas o inalámbricas adecuadas, tales como tipos de red Ethernet, por ejemplo. Además, la interfaz de red 2040 puede soportar la comunicación a través de redes de telecomunicaciones/telefonía, tales como redes de voz analógicas o redes de comunicaciones de fibra digital, a través de redes de área de almacenamiento, tales como SAN de canal de fibra, o a través de cualquier otro tipo adecuado de red y/o protocolo.

En algunas realizaciones, la memoria de sistema 2020 puede ser una realización de un medio accesible por ordenador configurado para almacenar instrucciones de programa y datos como se ha descrito anteriormente para las figuras 1 a 35 para implementar realizaciones de un sistema de equilibrado de carga distribuido. Sin embargo, en otras realizaciones, se pueden recibir, enviar o almacenar instrucciones y/o datos de programas en diferentes tipos de medios accesibles por ordenador. En general, un medio accesible por ordenador puede incluir medios de almacenamiento no transitorios o medios de memoria tales como medios magnéticos u ópticos, por ejemplo, disco o DVD/CD acoplados al sistema informático 2000 a través de la interfaz I/O 2030. Un medio de almacenamiento no transitorio, accesible por ordenador también puede incluir cualquier medio volátil o no volátil tal como RAM (por ejemplo, SDRAM, DDR SDRAM, RDRAM, SRAM, etc.), ROM, etc., que puede incluirse en algunas realizaciones del sistema informático 2000 como memoria de sistema 2020 u otro tipo de memoria. Además, un medio accesible por ordenador puede incluir medios de transmisión o señales tales como señales eléctricas, electromagnéticas o digitales, transportadas a través de un medio de comunicación tal como una red y/o un enlace inalámbrico, tal como se puede implementar a través de la interfaz de red 2040.

Conclusión

Las diversas realizaciones pueden incluir además recibir, enviar o almacenar instrucciones y/o datos implementados de acuerdo con la descripción anterior sobre un medio accesible por ordenador. Generalmente hablando, un medio accesible por ordenador puede incluir medios de almacenamiento o medios de memoria tales como medios magnéticos u ópticos, por ejemplo, disco o DVD/CD-ROM, medios volátiles o no volátiles, tales como RAM (por ejemplo, SDRAM, DDR, RDRAM, SRAM, etc.), ROM, etc, así como medios de transmisión o señales tales como señales eléctricas, electromagnéticas o digitales, transportadas a través de un medio de comunicación tal como una red y/o un enlace inalámbrico.

Los diversos métodos como se ilustran en las figuras y descritos en el presente documento representan realizaciones ejemplares de los métodos. Los métodos pueden implementarse en software, hardware o una combinación de los mismos. El orden del método puede cambiarse, y pueden añadirse, reordenarse, combinarse, omitirse, modificarse, etc. diversos elementos.

Pueden hacerse diversas modificaciones y cambios como será obvio para un experto en la técnica que tiene el beneficio de esta descripción. Se pretende que incluya todas estas modificaciones y cambios y, en consecuencia, la descripción anterior se considerará en un sentido ilustrativo en lugar de restrictivo.

REIVINDICACIONES

1. Un sistema equilibrador de carga distribuido, que comprende:
un enrutador (104) configurado para recibir paquetes en flujos de paquetes desde uno o más clientes (160) de acuerdo con una única dirección IP pública del enrutador; una pluralidad de nodos servidor (130A-130m); y
una pluralidad de nodos de equilibradores de carga (10A-110n) configurados cada uno como un servidor de entrada en el sistema de equilibrador de carga distribuido, donde los servidores de entrada anuncian la misma única dirección IP pública al enrutador;
donde el enrutador está configurado además para distribuir los flujos de paquetes entre la pluralidad de servidores de entrada de acuerdo con una técnica de enrutamiento multitrayecto de hash aplicada a la información de dirección de origen y de destino de los paquetes en los flujos de paquetes; y
donde cada servidor de entrada está configurado además para recibir paquetes en uno o más flujos de paquetes desde el enrutador y distribuir los paquetes a uno o más de la pluralidad de nodos servidor que están asignados a los flujos de paquetes respectivos.

2. El sistema de equilibrador de carga distribuido según la reivindicación 1, donde la técnica de enrutamiento multitrayecto de hash es una técnica de enrutamiento multitrayecto de igual coste (ECMP).

3. El sistema de equilibrador de carga distribuido según la reivindicación 1 o la reivindicación 2, donde cada nodo de equilibrador de carga se anuncia al enrutador por uno o más de los otros nodos de equilibradores de carga.

4. El sistema de equilibrador de carga distribuido según la reivindicación 3, donde el uno o más nodos de equilibradores de carga establecen cada uno una sesión de protocolo de puerta de enlace de frontera (BGP) con el enrutador para anunciar el nodo de equilibrador de carga al enrutador.

5. El sistema de equilibrador de carga distribuido según la reivindicación 4, donde cada uno del uno o más nodos de equilibradores de carga que anuncian el nodo de equilibrador de carga al enrutador está configurado además para:
detectar que el nodo de equilibrador de carga que se anuncia al enrutador está inactivo; y
en respuesta a dicha detección, cerrar la sesión BGP que anuncia el nodo del equilibrador de carga al enrutador.

6. El sistema de equilibrador de carga distribuido según la reivindicación 5, donde el enrutador está configurado además para redistribuir los flujos de paquetes entre la pluralidad de servidores de entrada de acuerdo con la técnica de enrutamiento multitrayecto de hash en respuesta al uno o más nodos de equilibrador de carga diferentes que cierran las sesiones BGP.

7. El sistema de equilibrador de carga distribuido según una cualquiera de las reivindicaciones anteriores, donde la información de dirección de origen y de destino de un paquete incluye una dirección IP de cliente, un puerto de cliente, una dirección IP pública de servidor y un puerto de servidor.

8. Un método, que comprende:
recibir, por un enrutador, paquetes en flujos de paquetes de uno o más clientes según una única dirección IP pública del enrutador;
distribuir, por el enrutador, los flujos de paquetes entre una pluralidad de nodos de equilibradores de carga de acuerdo con una técnica de enrutamiento multitrayecto de hash aplicada a la información de dirección de origen y destino de los paquetes en los flujos de paquetes, donde la pluralidad de nodos de equilibradores de carga están configurados cada uno como un servidor de entrada en un sistema de equilibrador de carga distribuido, donde los servidores de entrada anuncian la misma única dirección IP pública al enrutador; y
distribuir, por cada uno del uno o más nodos de equilibradores de carga, los paquetes en uno o más flujos de paquetes recibidos desde el enrutador a uno o más de una pluralidad de nodos servidor que están asignados a los flujos de paquetes respectivos.

9. El método según la reivindicación 8, donde la técnica de enrutamiento multitrayecto de hash es una técnica de enrutamiento multitrayecto de igual coste (ECMP).

10. El método según la reivindicación 8 o la reivindicación 9, que comprende además cada equilibrador de carga que anuncia al menos otro nodo de equilibrador de carga al enrutador, donde cada nodo de equilibrador de carga se anuncia al enrutador por uno o más de los otros nodos de equilibradores de carga.

11. El método según la reivindicación 10, donde el uno o más nodos de equilibradores de carga diferentes que anuncian un nodo de equilibrador de carga incluyen los nodos de equilibradores de carga vecinos izquierdo y derecho del nodo de equilibrador de carga de acuerdo con un orden especificado de los nodos de equilibradores de carga.

12. El método según la reivindicación 10 o la reivindicación 11, donde dicho anuncio comprende cada nodo de equilibrador de carga que establece una sesión de protocolo de puerta de enlace de frontera (BGP) con el enrutador para cada nodo de equilibrador de carga diferente que el nodo de equilibrador de carga anuncia al enrutador.

13. El método según la reivindicación 12, que comprende además:
detectar, mediante un nodo de equilibrador de carga, que otro nodo de equilibrador de carga que se anuncia alenrutador por el nodo del equilibrador de carga está inactivo; y
en respuesta a dicha detección, cerrar la sesión BGP con el enrutador que anuncia el otro nodo de equilibrador de carga.

14. El método según la reivindicación 13, que comprende además redistribuir, por el enrutador, el paquete fluye entre la pluralidad de nodos de equilibradores de carga de acuerdo con la técnica de enrutamiento multitrayecto de hash en respuesta a determinar que una o más sesiones BGP que anuncian un nodo de equilibrador de carga se han cerrado.

15. El método según una cualquiera de las reivindicaciones 8-14, donde la información de dirección de origen y de destino de un paquete incluye una dirección IP de cliente, un puerto de cliente, una dirección IP pública de servidor y un puerto de servidor.

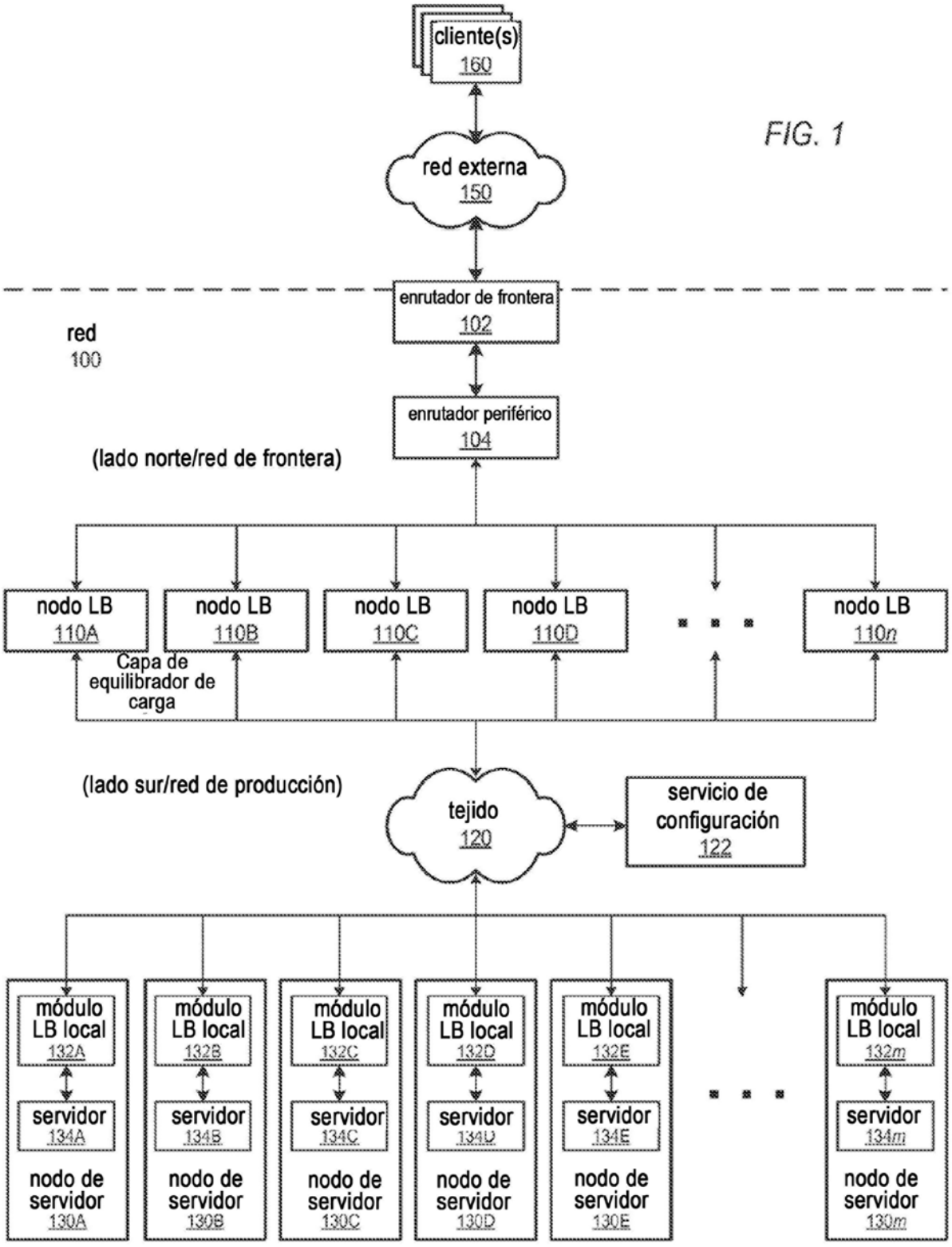
cliente(s)
160
red externa
150
enrutador de frontera
102
red
100
enrutador periférico
104
(lado norte/red de frontera)
nodo LB
110A
nodo LB
110B
nodo LB
110C
nodo LB
110D
nodo LB
110n
Capa de equilibrador de carga
(lado sur/red de producción)
tejido
120
servicio de configuración
122
módulo LB local
132A
servidor
134A
nodo de servidor
130A
módulo LB local
132B
servidor
134B
nodo de servidor
130B
módulo LB local
132C
servidor
134C
nodo de servidor
130C
módulo LB local
132D
servidor
134D
nodo de servidor
130D
módulo LB local
132E
servidor
134E
nodo de servidor
130E
módulo LB local
132m
servidor
134m
nodo de servidor
130m

FIG. 1

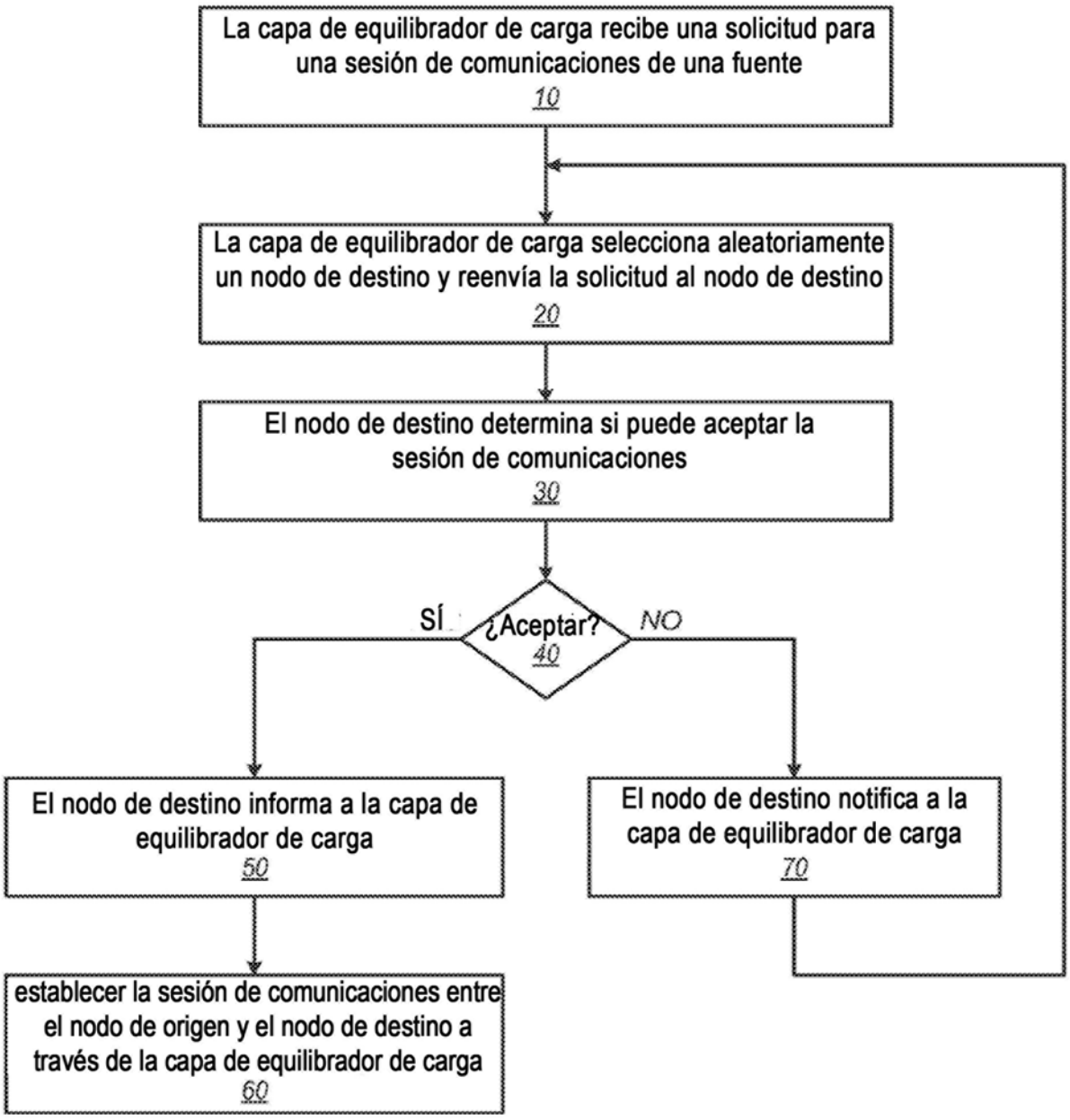
La capa de equilibrador de carga recibe una solicitud para una sesión de comunicaciones de una fuente
10
La capa de equilibrador de carga selecciona aleatoriamente un nodo de destino y reenvía la solicitud al nodo de destino
20
El nodo de destino determina si puede aceptar la sesión de comunicaciones
30
¿Aceptar?
40
SÍ
NO
El nodo de destino informa a la capa de equilibrador de carga
50
El nodo de destino notifica a la capa de equilibrador de carga
70
establecer la sesión de comunicaciones entre el nodo de origen y el nodo de destino a través de la capa de equilibrador de carga
60

FIG. 2

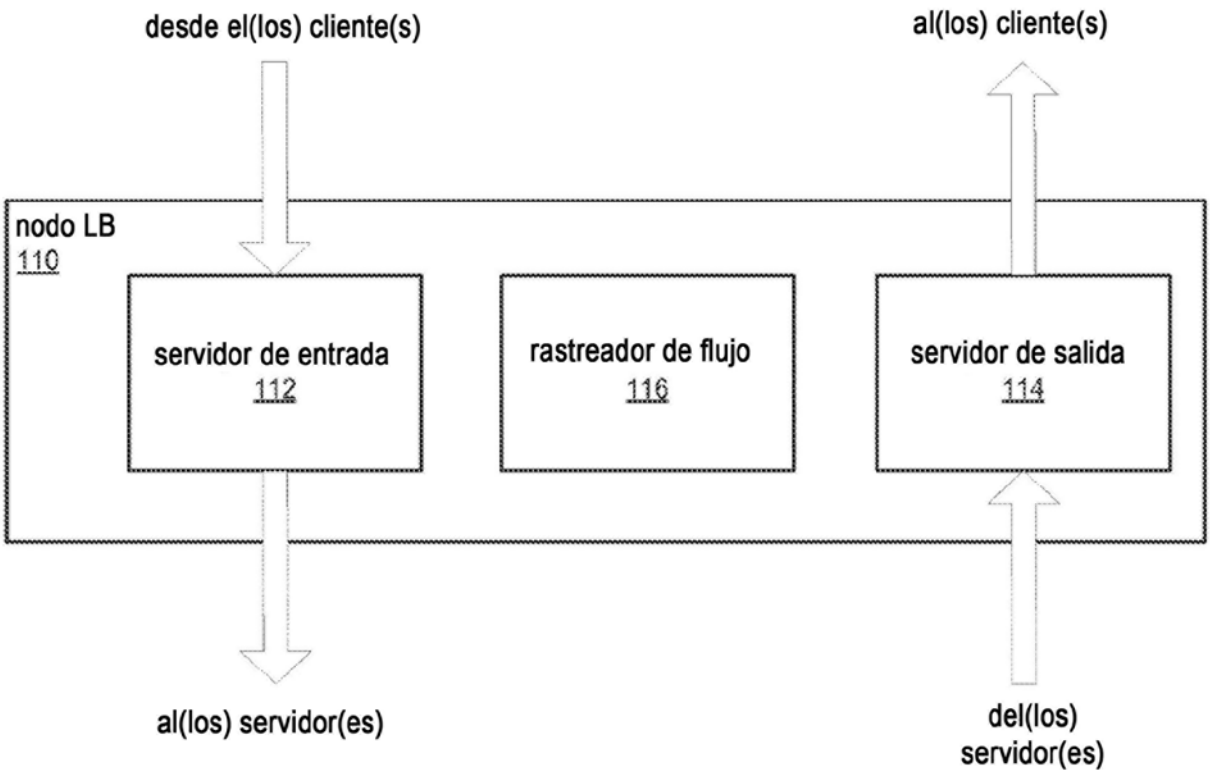
desde el(los) cliente(s)
al(los) cliente(s)
nodo LB
110
servidor de entrada
112
rastreador de flujo
116
servidor de salida
114
al(los) servidor(es)
del(los)
servidor(es)

FIG. 3

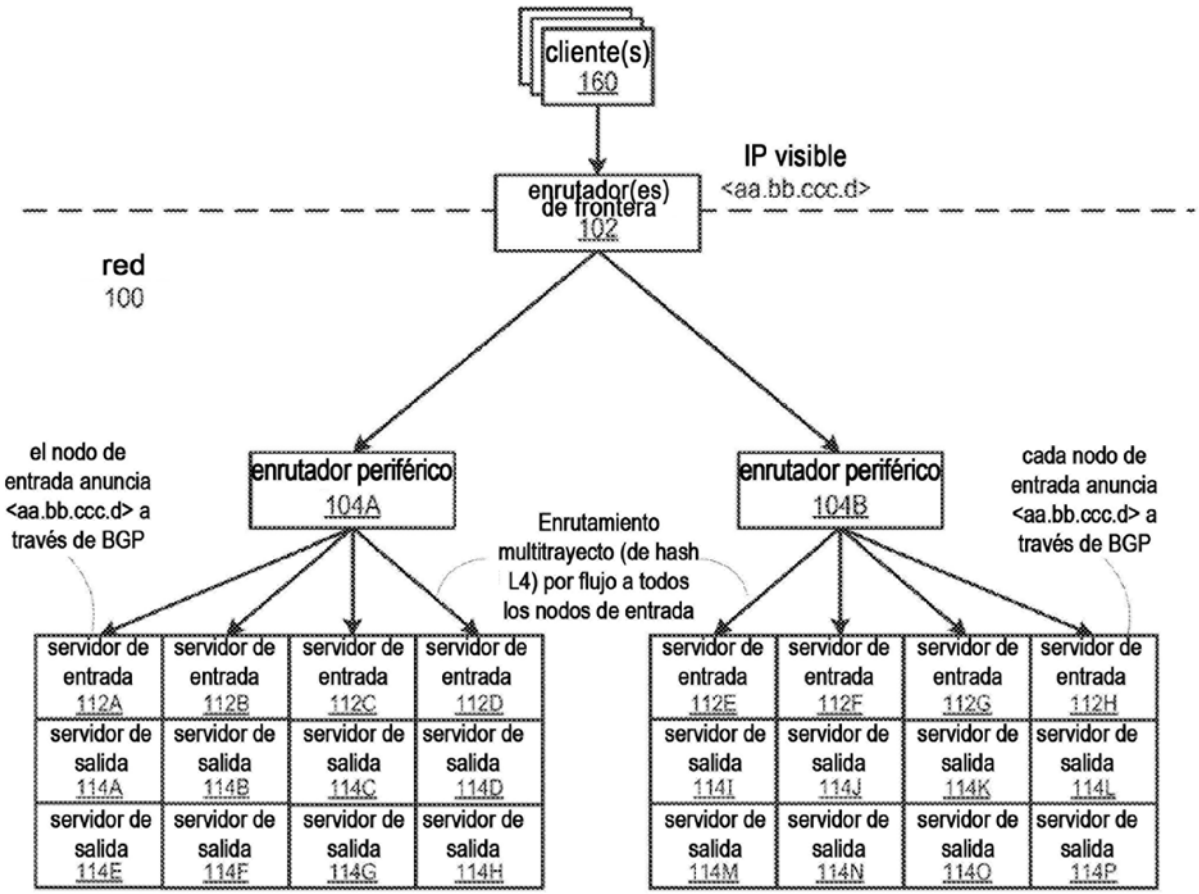
cliente(s)
160
IP visible
<aa.bb.ccc.d>
enrutador(es) de frontera
102
red
100
el nodo de entrada anuncia <aa.bb.ccc.d> a través de BGP
enrutador periférico
104A
Enrutamiento multitrayecto (de hash L4) por flujo a todos los nodos de entrada
enrutador periférico
104B
cada nodo de entrada anuncia <aa.bb.ccc.d> a través de BGP
servidor de entrada 112A
servidor de entrada 112B
servidor de entrada 112C
servidor de entrada 112D
servidor de salida 114A
servidor de salida 114B
servidor de salida 114C
servidor de salida 114D
servidor de salida 114E
servidor de salida 114F
servidor de salida 114G
servidor de salida 114H
servidor de entrada 112E
servidor de entrada 112F
servidor de entrada 112G
servidor de entrada 112H
servidor de salida 114I
servidor de salida 114J
servidor de salida 114K
servidor de salida 114L
servidor de salida 114M
servidor de salida 114N
servidor de salida 114O
servidor de salida 114P

FIG. 4

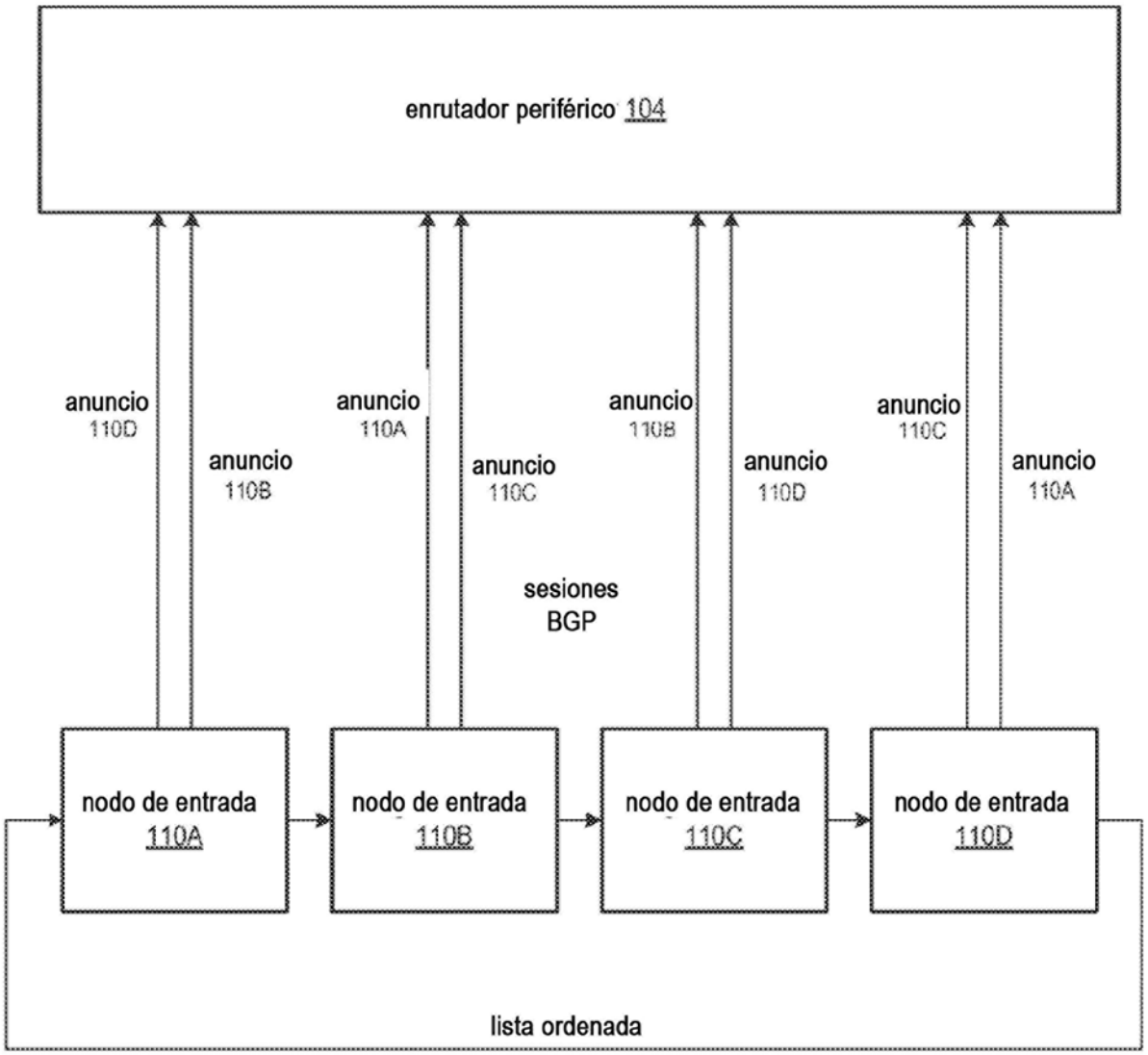
enrutador periférico 104
anuncio 110D
anuncio 110B
anuncio 110A
anuncio 110C
anuncio 110B
anuncio 110D
anuncio 110C
anuncio 110A
sesiones BGP
nodo de entrada 110A
nodo de entrada 110B
nodo de entrada 110C
nodo de entrada 110D
lista ordenada

FIG. 5

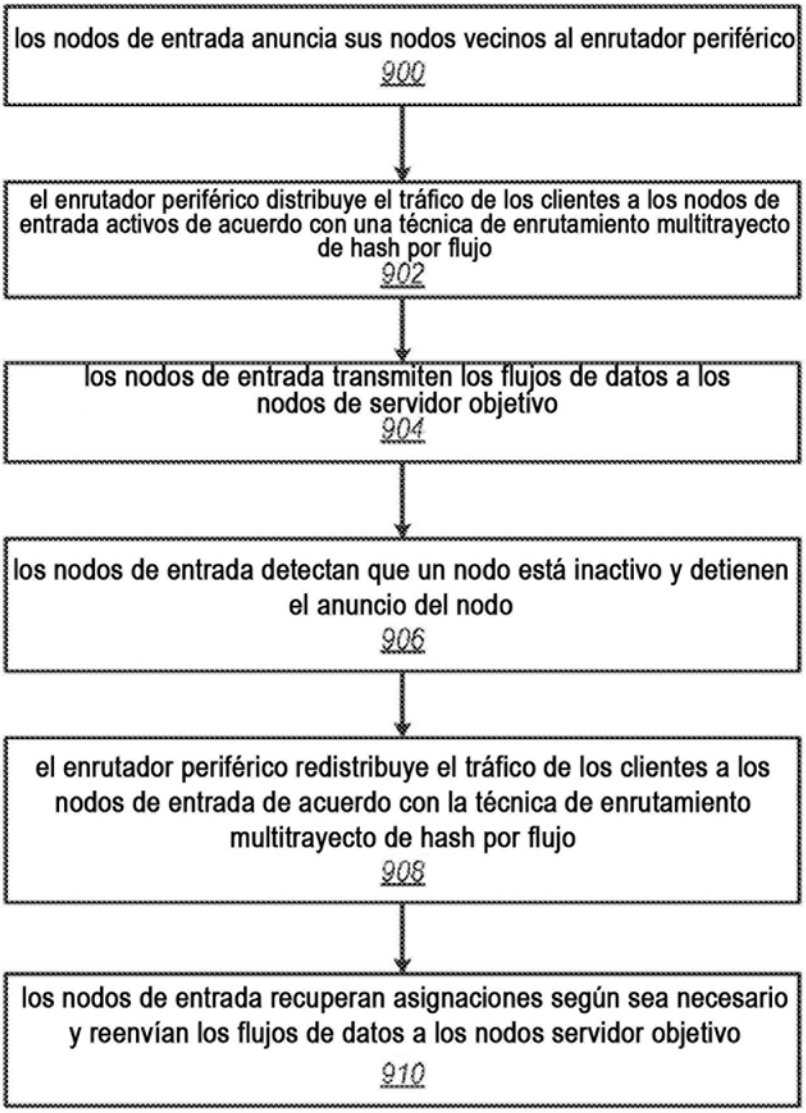
los nodos de entrada anuncia sus nodos vecinos al enrutador periférico
900
el enrutador periférico distribuye el tráfico de los clientes a los nodos de entrada activos de acuerdo con una técnica de enrutamiento multitrayecto de hash por flujo
902
los nodos de entrada transmiten los flujos de datos a los nodos de servidor objetivo
904
los nodos de entrada detectan que un nodo está inactivo y detienen el anuncio del nodo
906
el enrutador periférico redistribuye el tráfico de los clientes a los nodos de entrada de acuerdo con la técnica de enrutamiento multitrayecto de hash por flujo
908
los nodos de entrada recuperan asignaciones según sea necesario y reenvían los flujos de datos a los nodos servidor objetivo
910

FIG. 6

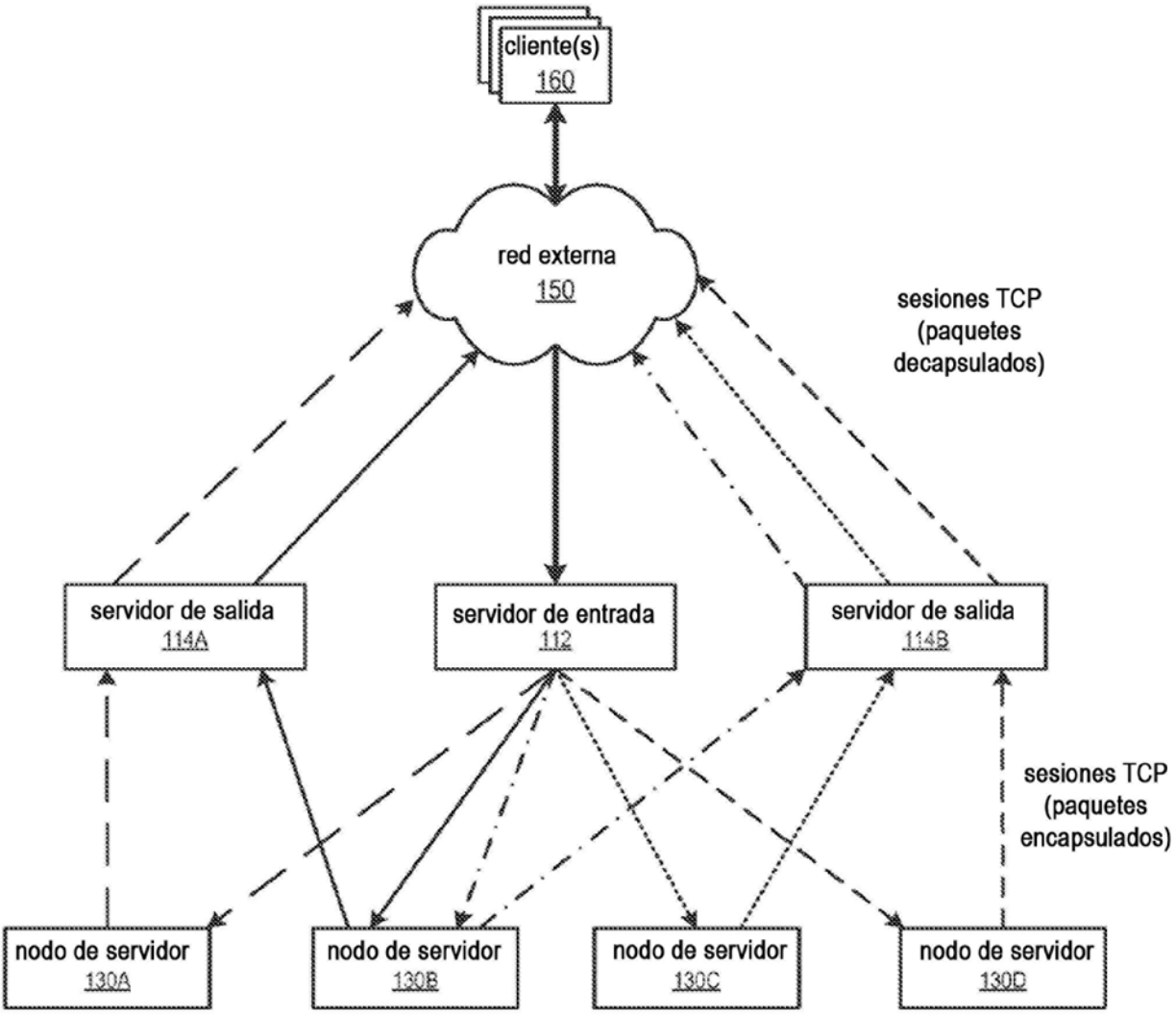
cliente(s)
160
red externa
150
sesiones TCP
(paquetes
decapsulados)
servidor de salida
114A
servidor de entrada
112
servidor de salida
114B
sesiones TCP
(paquetes
encapsulados)
nodo de servidor
130A
nodo de servidor
130B
nodo de servidor
130C
nodo de servidor
130D

FIG. 7

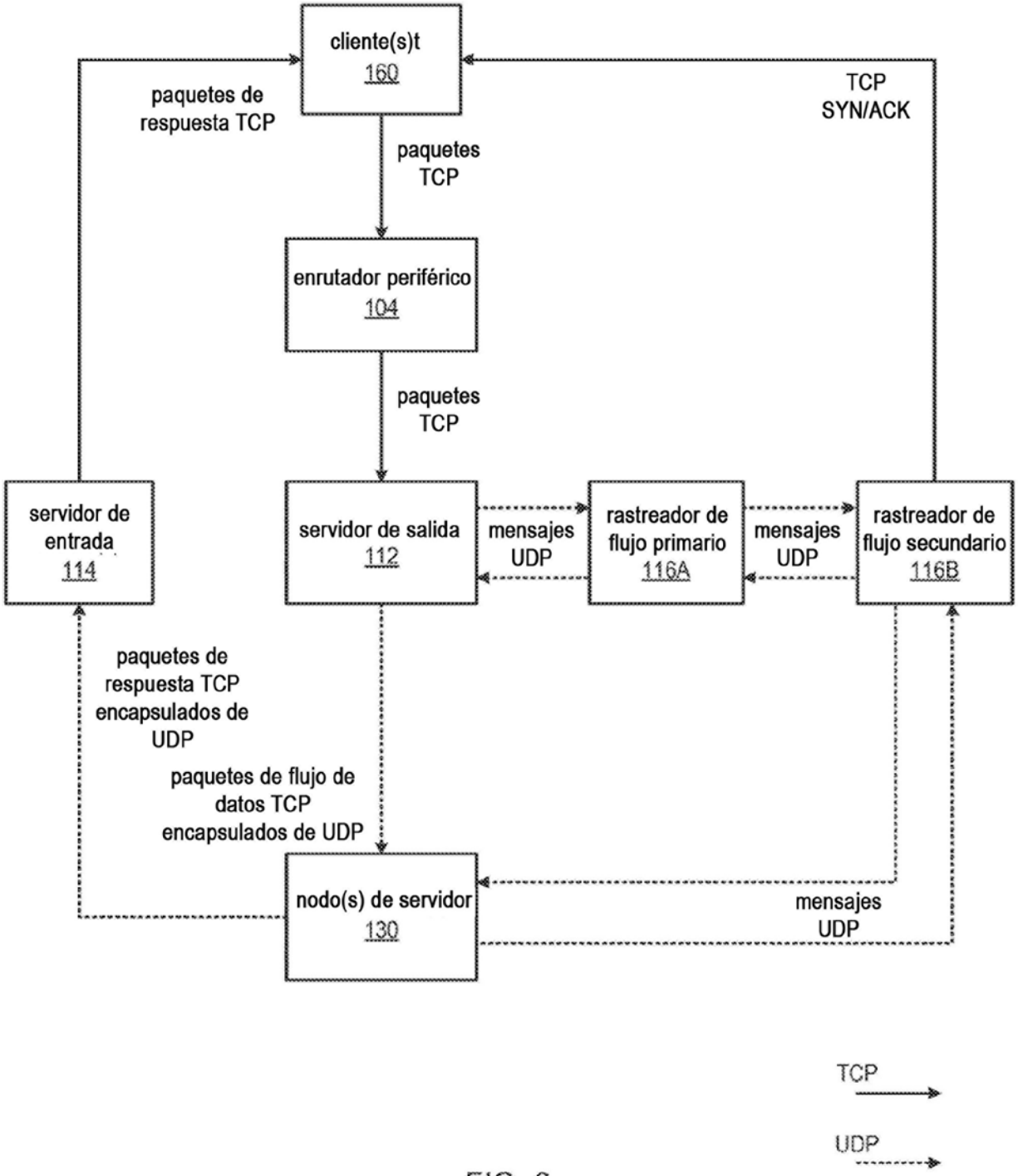
cliente(s)t
160
paquetes de respuesta TCP
TCP SYN/ACK
paquetes TCP
enrutador periférico
104
paquetes TCP
servidor de entrada
114
servidor de salida
112
mensajes UDP
rastreador de flujo primario
116A
mensajes UDP
rastreador de flujo secundario
116B
paquetes de respuesta TCP encapsulados de UDP
paquetes de flujo de datos TCP encapsulados de UDP
nodo(s) de servidor
130
mensajes UDP
TCP
UDP

FIG. 8

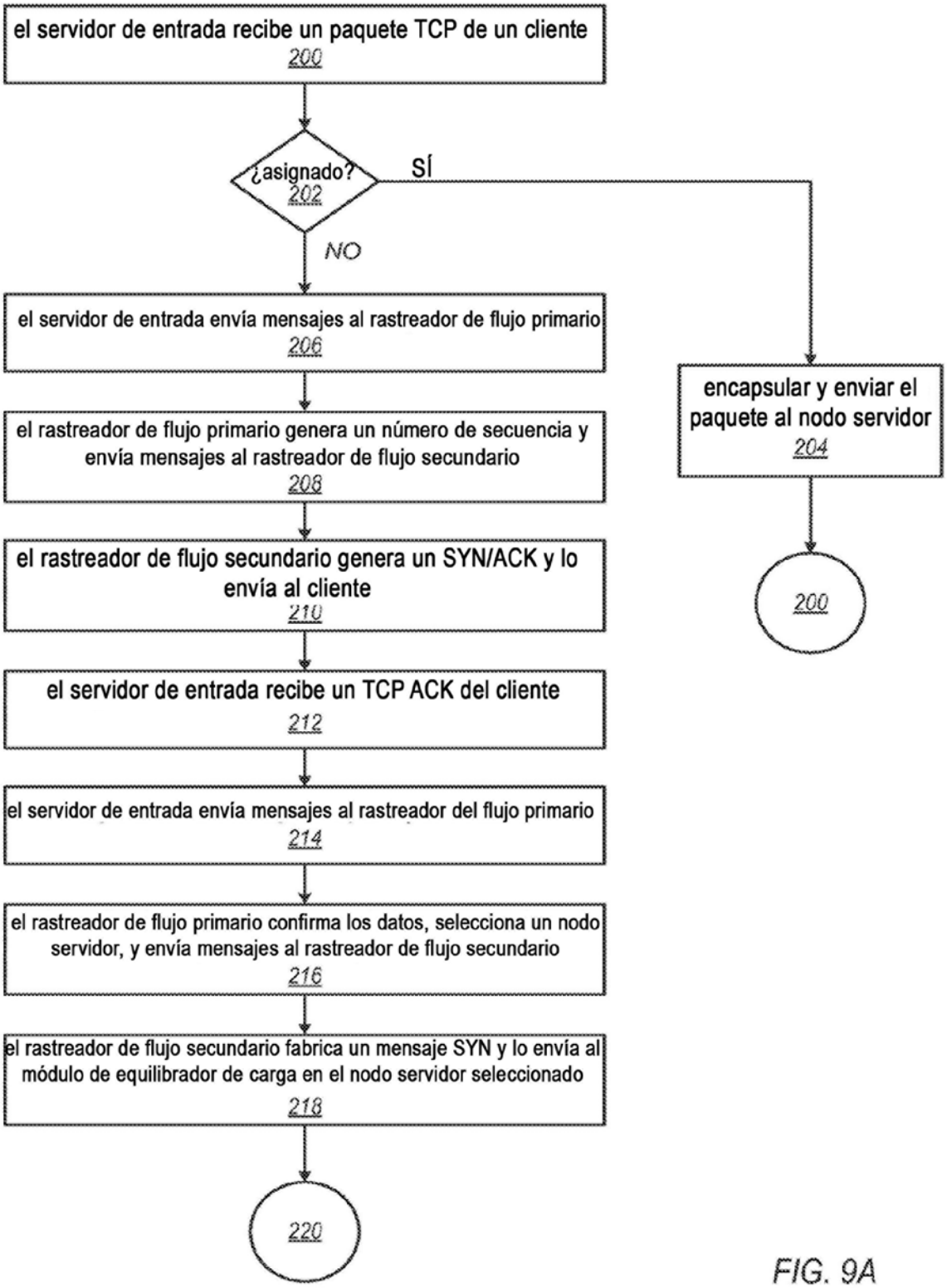
el servidor de entrada recibe un paquete TCP de un cliente
200
¿asignado?
202
SÍ
NO
el servidor de entrada envía mensajes al rastreador de flujo primario
206
el rastreador de flujo primario genera un número de secuencia y envía mensajes al rastreador de flujo secundario
208
el rastreador de flujo secundario genera un SYN/ACK y lo envía al cliente
210
el servidor de entrada recibe un TCP ACK del cliente
212
el servidor de entrada envía mensajes al rastreador del flujo primario
214
el rastreador de flujo primario confirma los datos, selecciona un nodo servidor, y envía mensajes al rastreador de flujo secundario
216
el rastreador de flujo secundario fabrica un mensaje SYN y lo envía al módulo de equilibrador de carga en el nodo servidor seleccionado
218
220
encapsular y enviar el paquete al nodo servidor
204
200

FIG. 9A

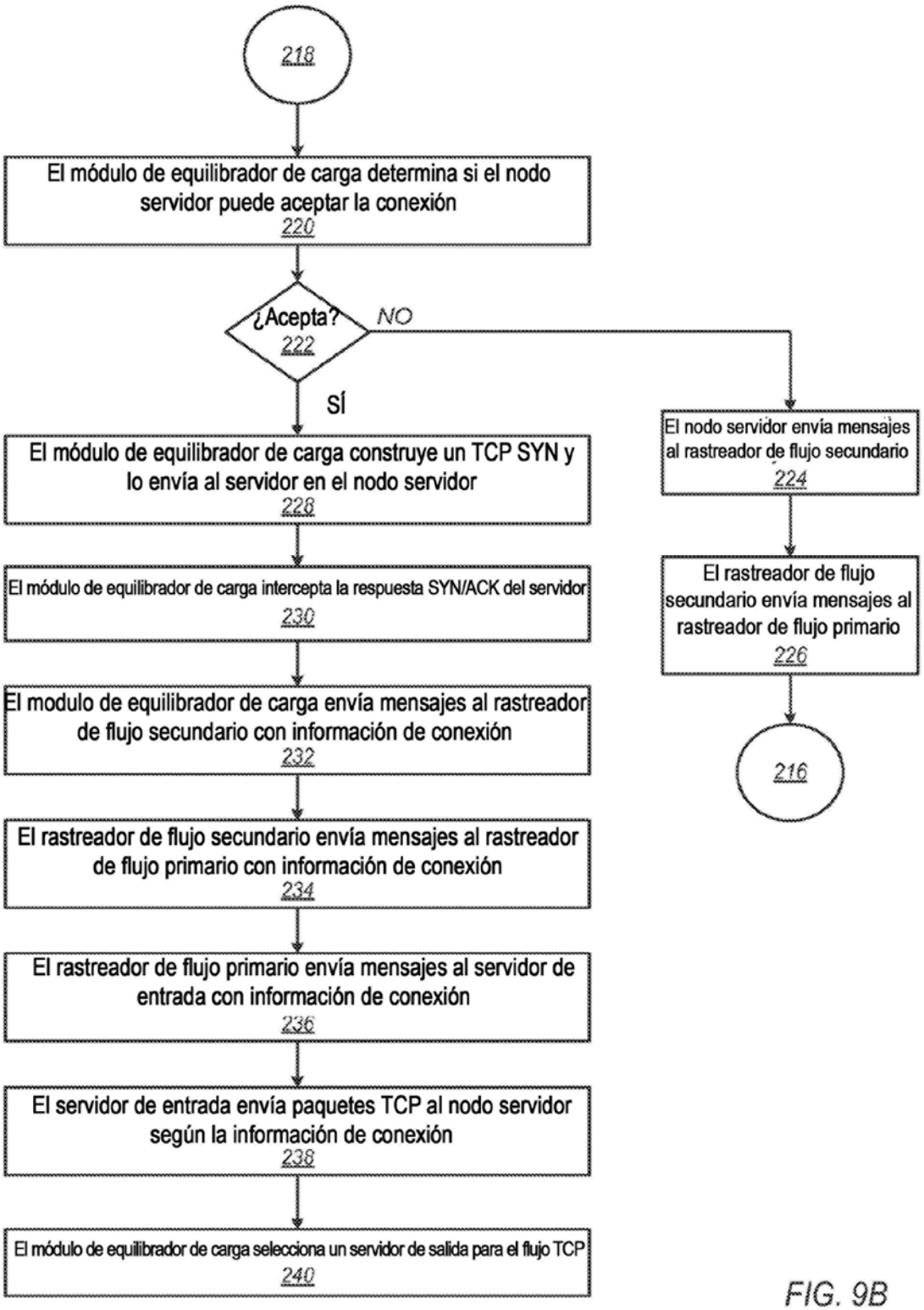
218
El módulo de equilibrador de carga determina si el nodo servidor puede aceptar la conexión
220
¿Acepta?
222
NO
SÍ
El módulo de equilibrador de carga construye un TCP SYN y lo envía al servidor en el nodo servidor
228
El módulo de equilibrador de carga intercepta la respuesta SYN/ACK del servidor
230
El modulo de equilibrador de carga envía mensajes al rastreador de flujo secundario con información de conexión
232
El rastreador de flujo secundario envía mensajes al rastreador de flujo primario con información de conexión
234
El rastreador de flujo primario envía mensajes al servidor de entrada con información de conexión
236
El servidor de entrada envía paquetes TCP al nodo servidor según la información de conexión
238
El módulo de equilibrador de carga selecciona un servidor de salida para el flujo TCP
240
El nodo servidor envía mensajes al rastreador de flujo secundario
224
El rastreador de flujo secundario envía mensajes al rastreador de flujo primario
226
216

FIG. 9B

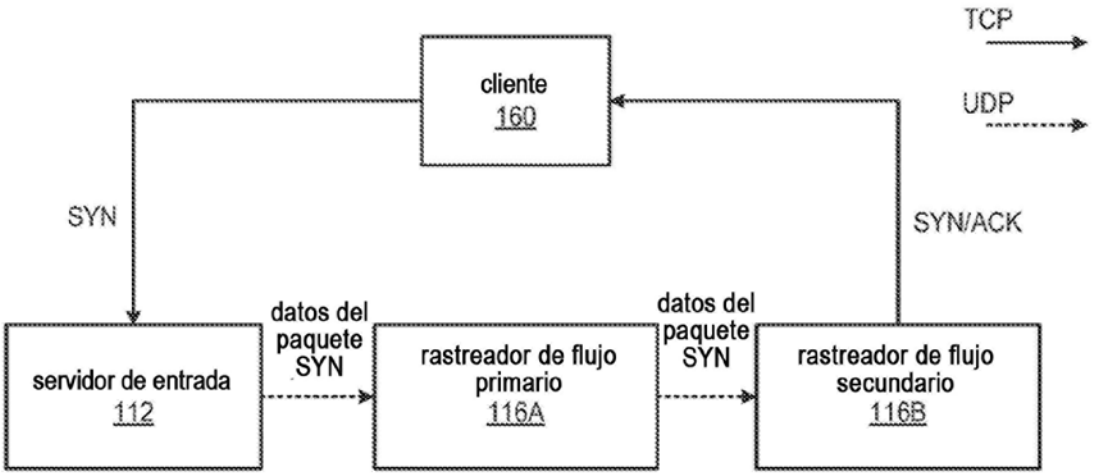

FIG. 10A

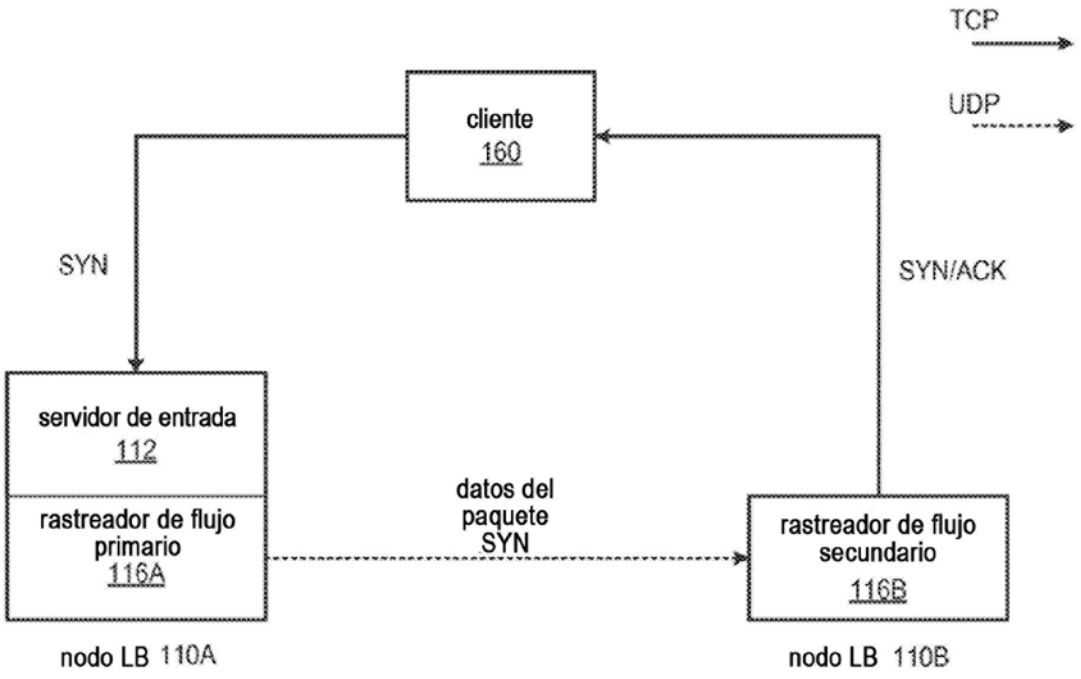

FIG. 10B

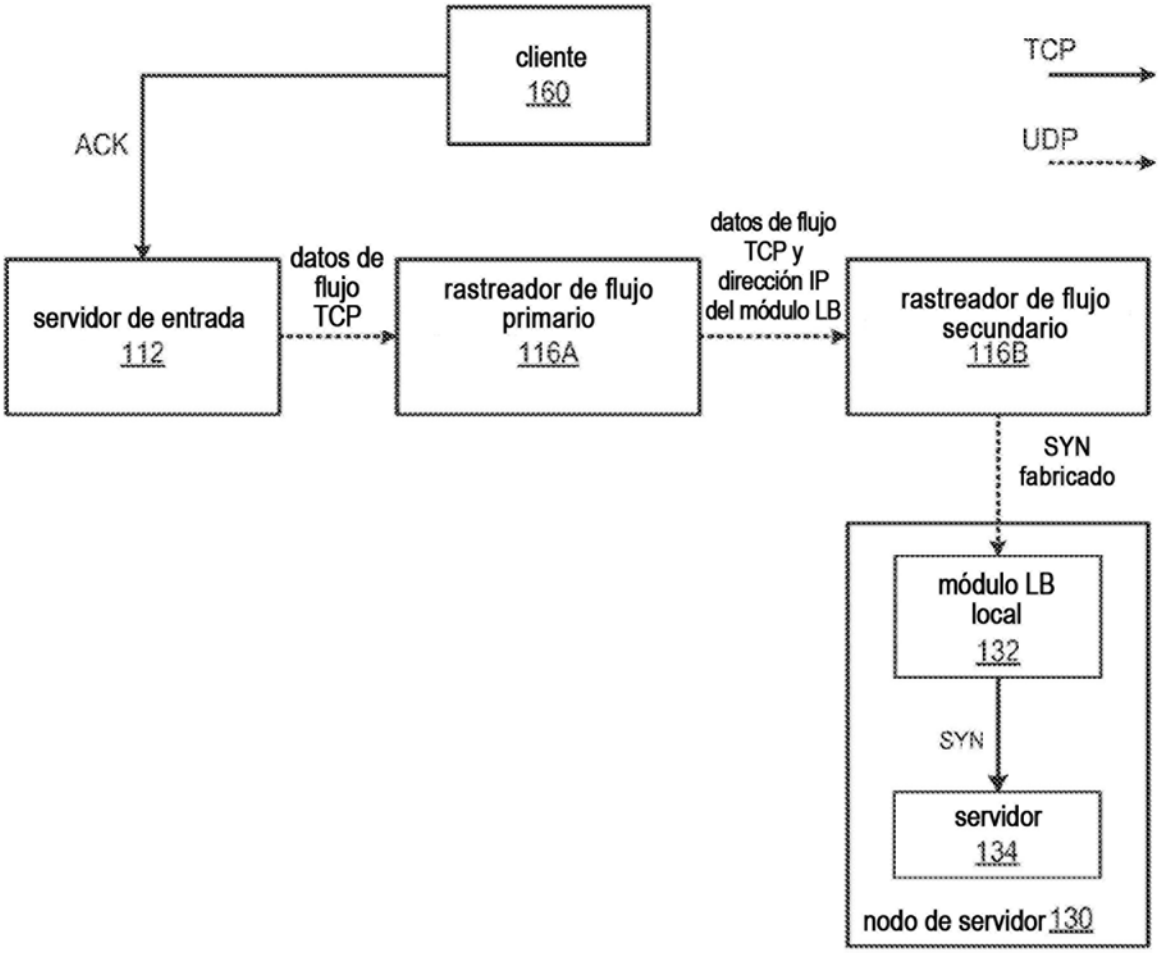
cliente
160
TCP
UDP
ACK
servidor de entrada
112
datos de flujo TCP
rastreador de flujo primario
116A
datos de flujo TCP y dirección IP del módulo LB
rastreador de flujo secundario
116B
SYN fabricado
módulo LB local
132
SYN
servidor
134
nodo de servidor 130

FIG. 10C

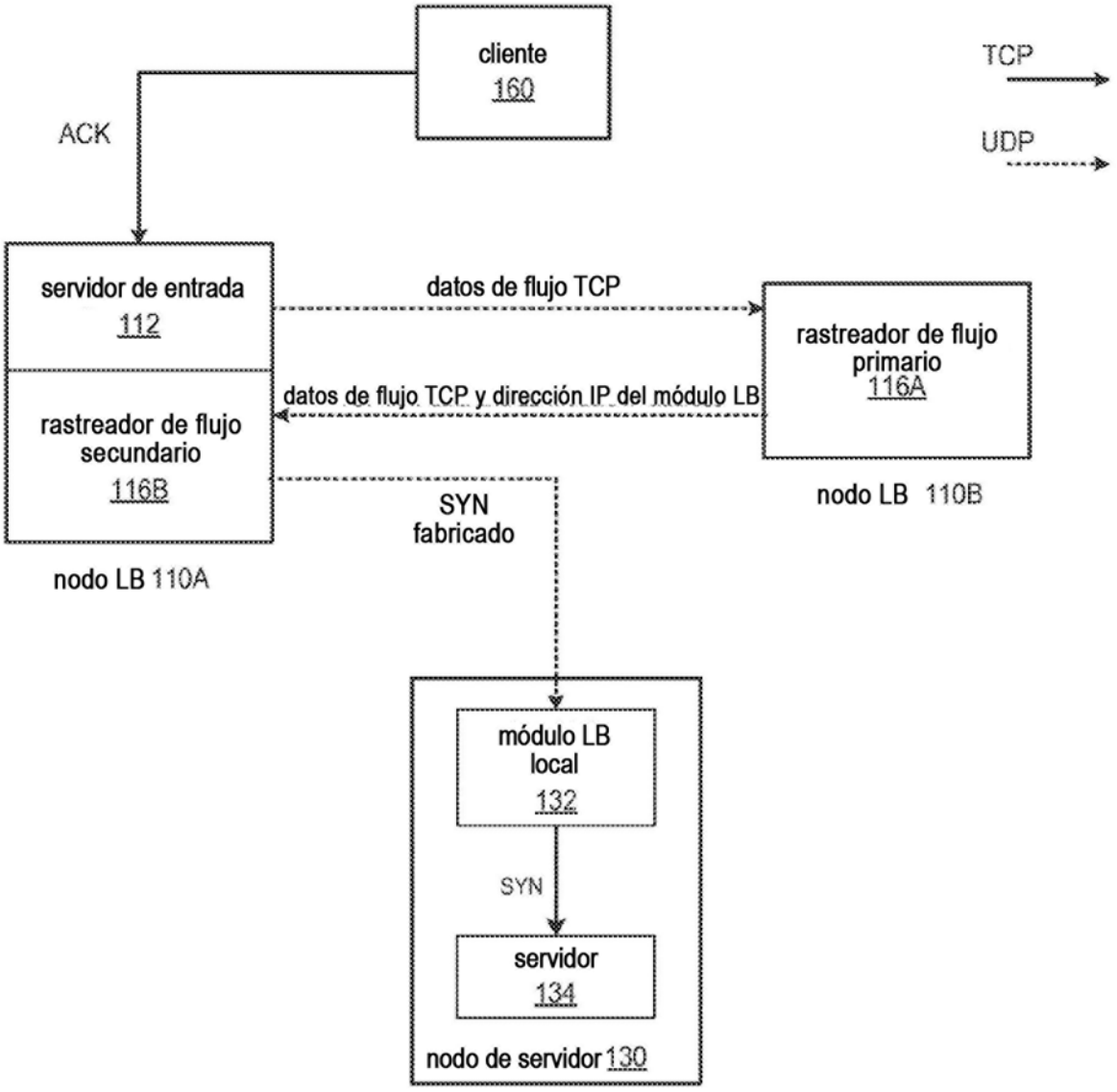
cliente
160
TCP
UDP
ACK
servidor de entrada
112
datos de flujo TCP
rastreador de flujo primario
116A
datos de flujo TCP y dirección IP del módulo LB
rastreador de flujo secundario
116B
nodo LB 110B
SYN fabricado
nodo LB 110A
módulo LB local
132
SYN
servidor
134
nodo de servidor 130

FIG. 10D

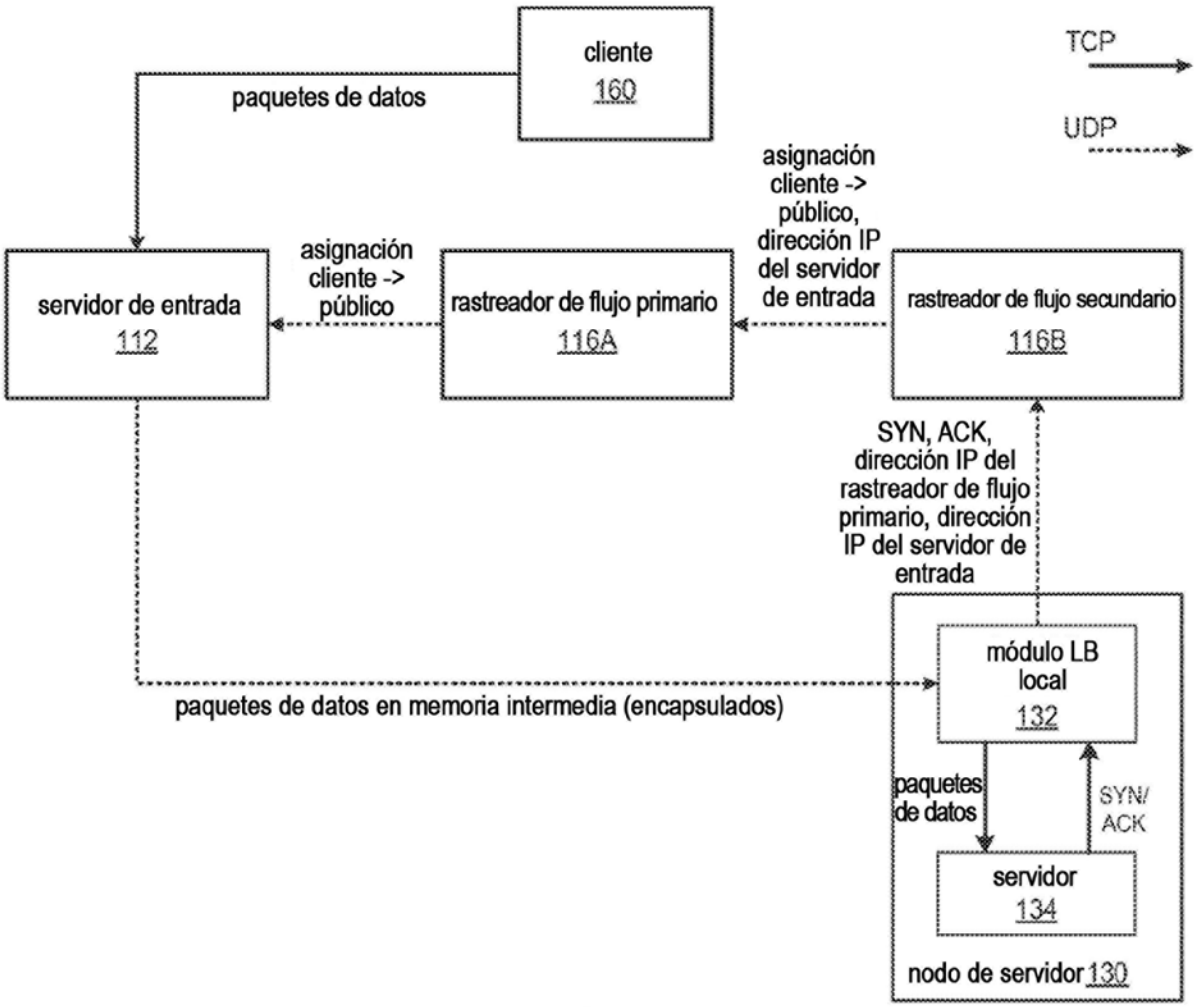
cliente
160
paquetes de datos
TCP
UDP
asignación cliente -> público, dirección IP del servidor de entrada
servidor de entrada
112
asignación cliente -> público
rastreador de flujo primario
116A
rastreador de flujo secundario
116B
SYN, ACK, dirección IP del rastreador de flujo primario, dirección IP del servidor de entrada
módulo LB local
132
paquetes de datos en memoria intermedia (encapsulados)
paquetes de datos
SYN/ ACK
servidor
134
nodo de servidor 130

FIG. 10E

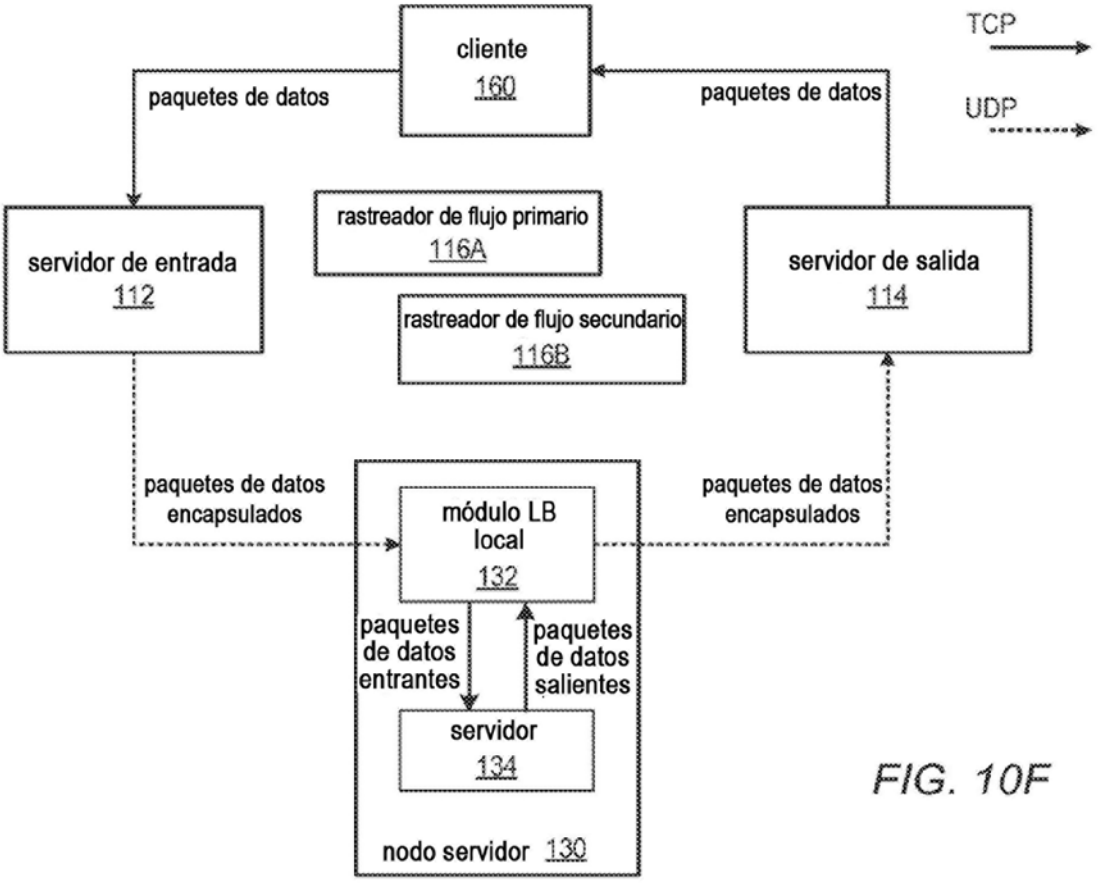

FIG. 10F

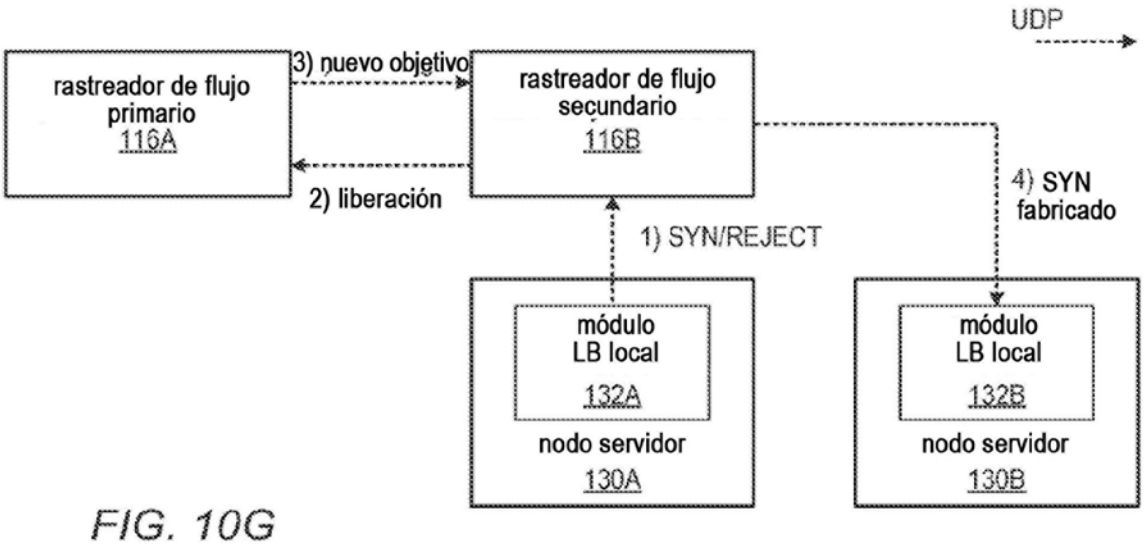

FIG. 10G

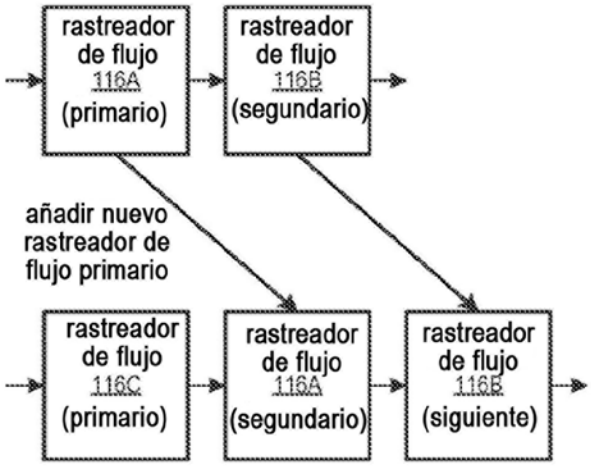

FIG. 11A

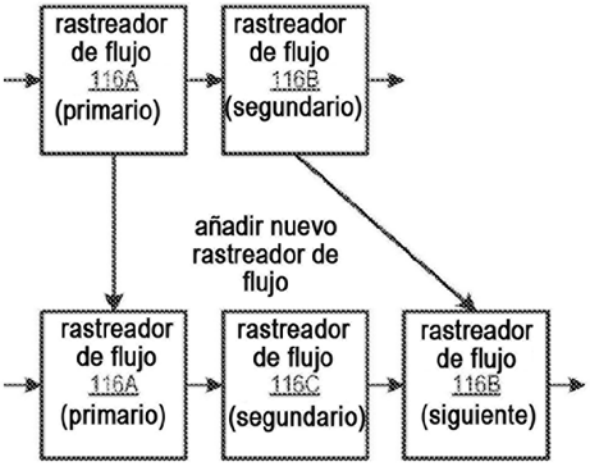

FIG. 11B

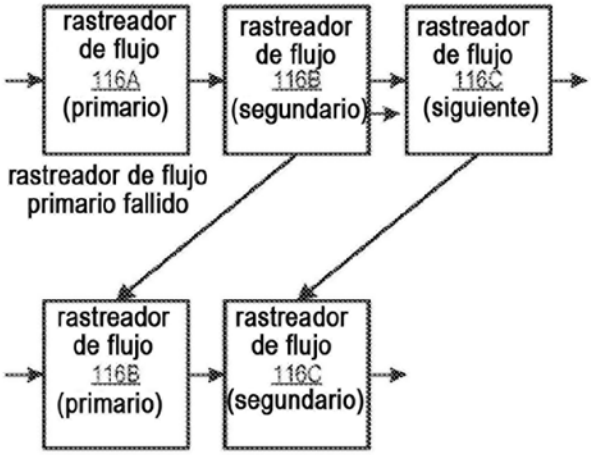

FIG. 11C

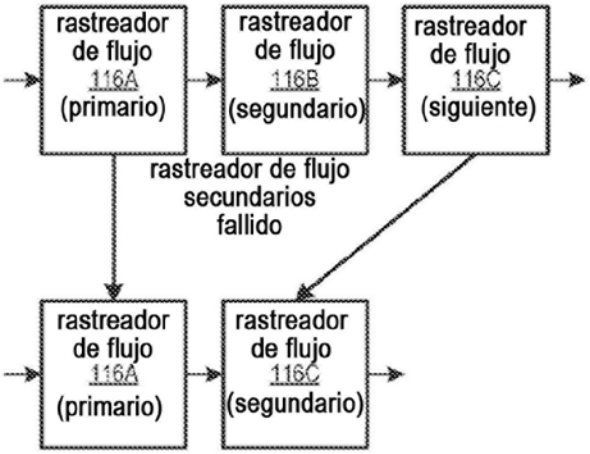

FIG. 11D

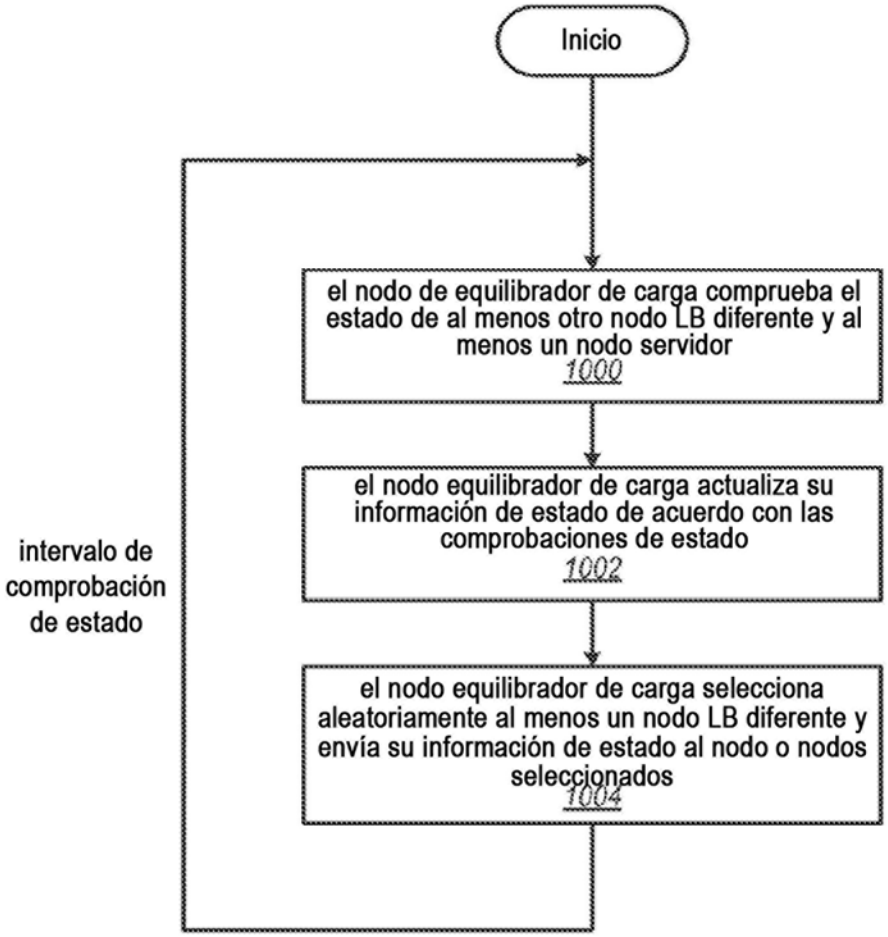
Inicio
el nodo de equilibrador de carga comprueba el estado de al menos otro nodo LB diferente y al menos un nodo servidor
1000
el nodo equilibrador de carga actualiza su información de estado de acuerdo con las comprobaciones de estado
1002
el nodo equilibrador de carga selecciona aleatoriamente al menos un nodo LB diferente y envía su información de estado al nodo o nodos seleccionados
1004
intervalo de comprobación de estado

FIG. 12

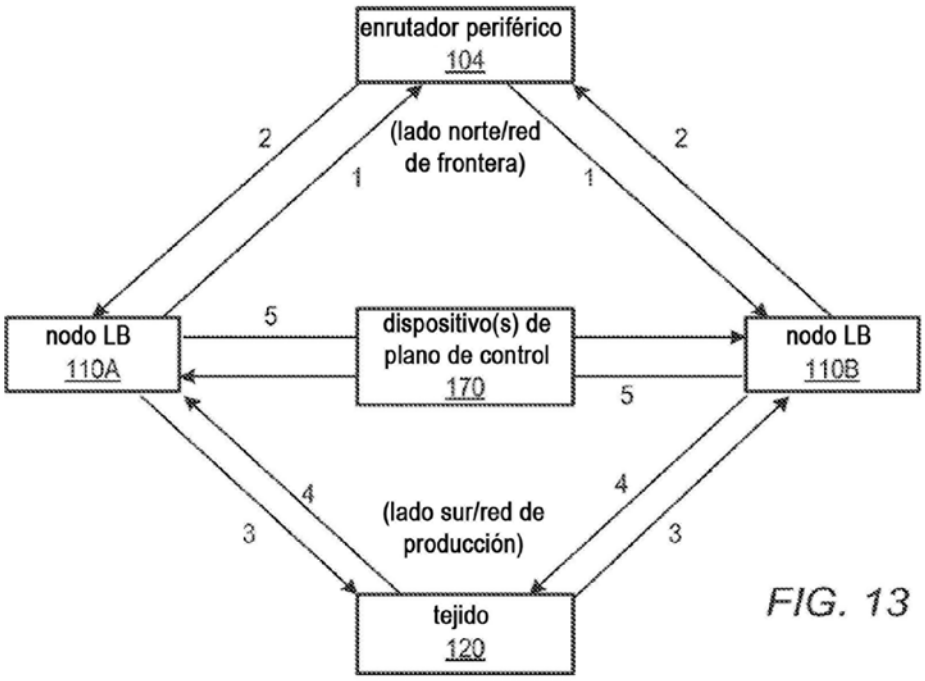
enrutador periférico
104
(lado norte/red de frontera)
2
1
1
2
nodo LB
110A
5
dispositivo(s) de plano de control
170
5
nodo LB
110B
4
3
(lado sur/red de producción)
4
3
tejido
120

FIG. 13

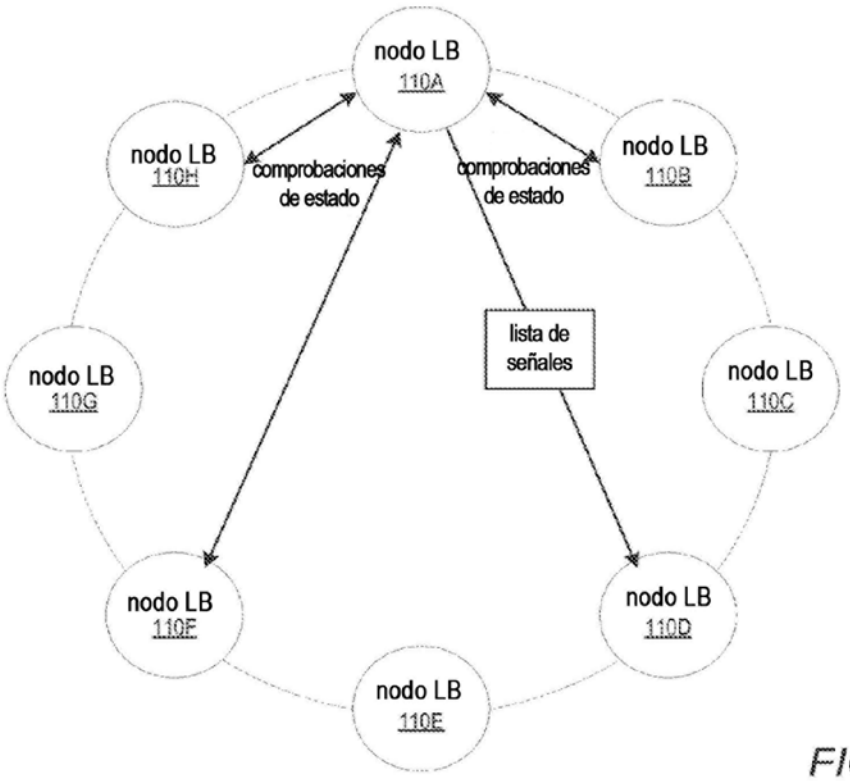
nodo LB
110A
nodo LB
110H
comprobaciones de estado
comprobaciones de estado
nodo LB
110B
lista de señales
nodo LB
110G
nodo LB
110C
nodo LB
110F
nodo LB
110D
nodo LB
110E

FIG. 14

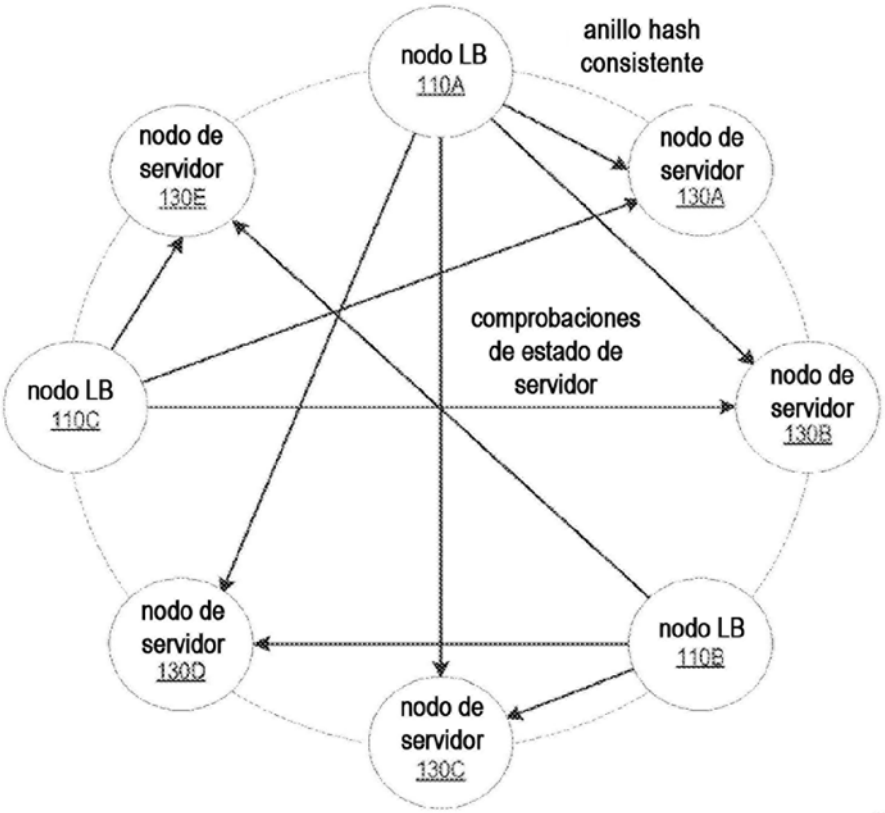
anillo hash consistente
nodo LB 110A
nodo de servidor 130A
nodo de servidor 130B
nodo de servidor 130C
nodo de servidor 130D
nodo de servidor 130E
nodo LB 110B
nodo LB 110C
comprobaciones de estado de servidor

FIG. 15

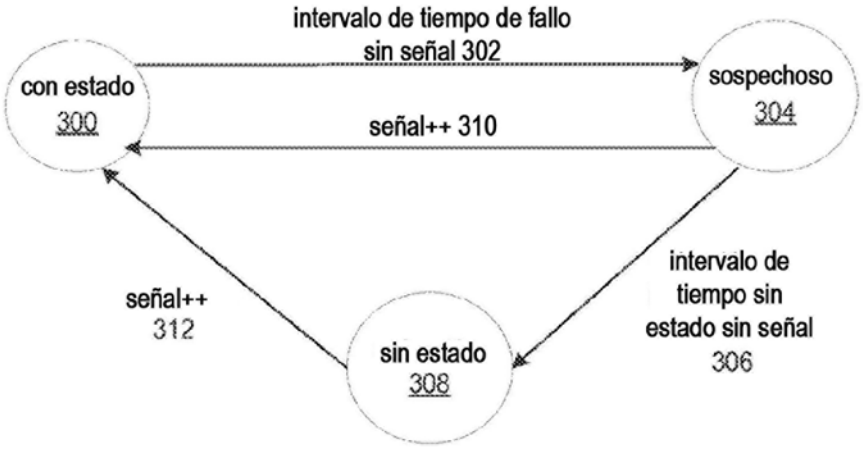
intervalo de tiempo de fallo sin señal 302
con estado 300
sospechoso 304
señal++ 310
señal++ 312
intervalo de tiempo sin estado sin señal 306
sin estado 308

FIG. 16

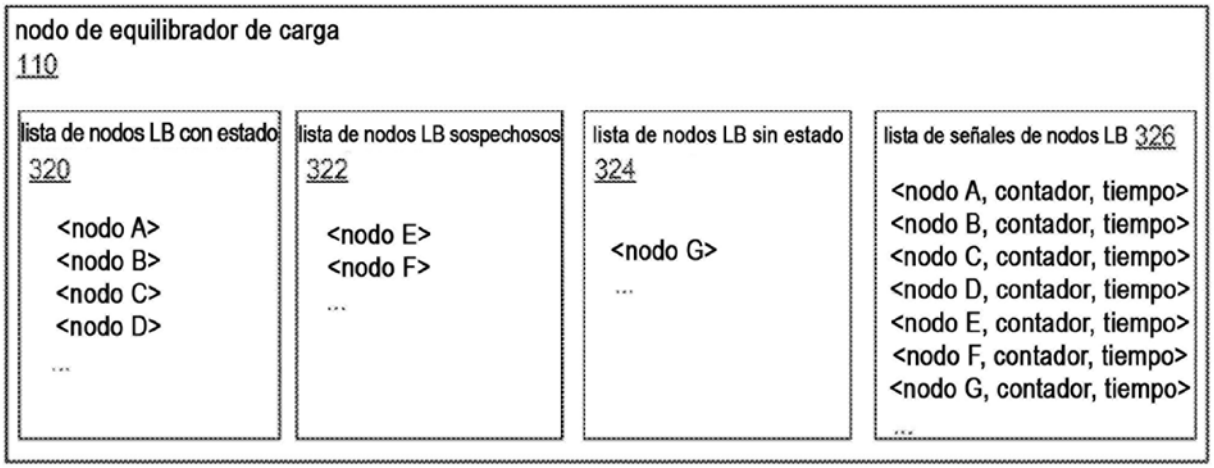
nodo de equilibrador de carga
110
lista de nodos LB con estado
320
<nodo A>
<nodo B>
<nodo C>
<nodo D>
...
lista de nodos LB sospechosos
322
<nodo E>
<nodo F>
...
lista de nodos LB sin estado
324
<nodo G>
...
lista de señales de nodos LB 326
<nodo A, contador, tiempo>
<nodo B, contador, tiempo>
<nodo C, contador, tiempo>
<nodo D, contador, tiempo>
<nodo E, contador, tiempo>
<nodo F, contador, tiempo>
<nodo G, contador, tiempo>
...

FIG. 17

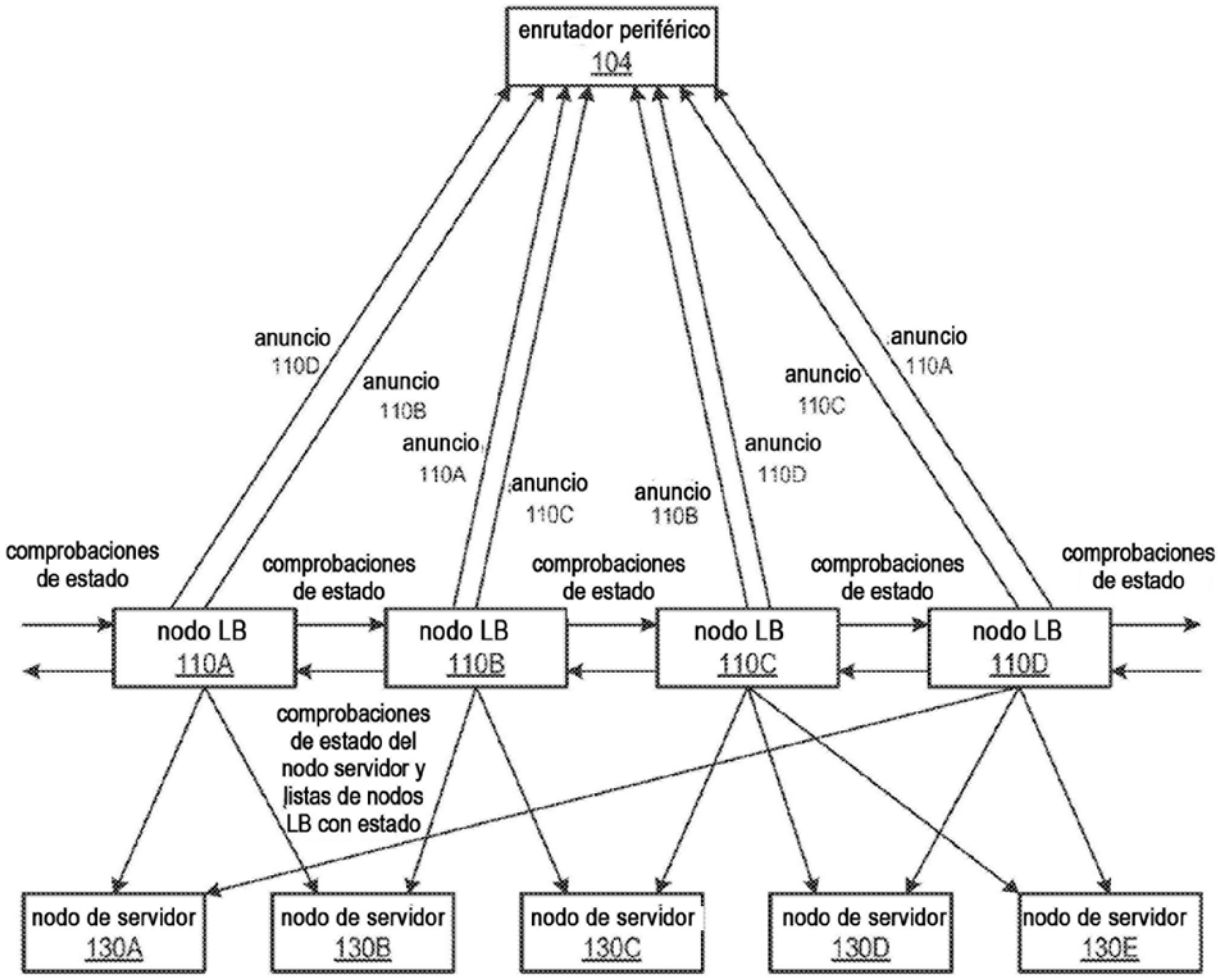
enrutador periférico
104
anuncio
110D
anuncio
110B
anuncio
110A
anuncio
110C
anuncio
110B
anuncio
110C
anuncio
110D
anuncio
110A
comprobaciones
de estado
comprobaciones
de estado
comprobaciones
de estado
comprobaciones
de estado
comprobaciones
de estado
nodo LB
110A
nodo LB
110B
nodo LB
110C
nodo LB
110D
comprobaciones
de estado del
nodo servidor y
listas de nodos
LB con estado
nodo de servidor
130A
nodo de servidor
130B
nodo de servidor
130C
nodo de servidor
130D
nodo de servidor
130E

FIG. 18A

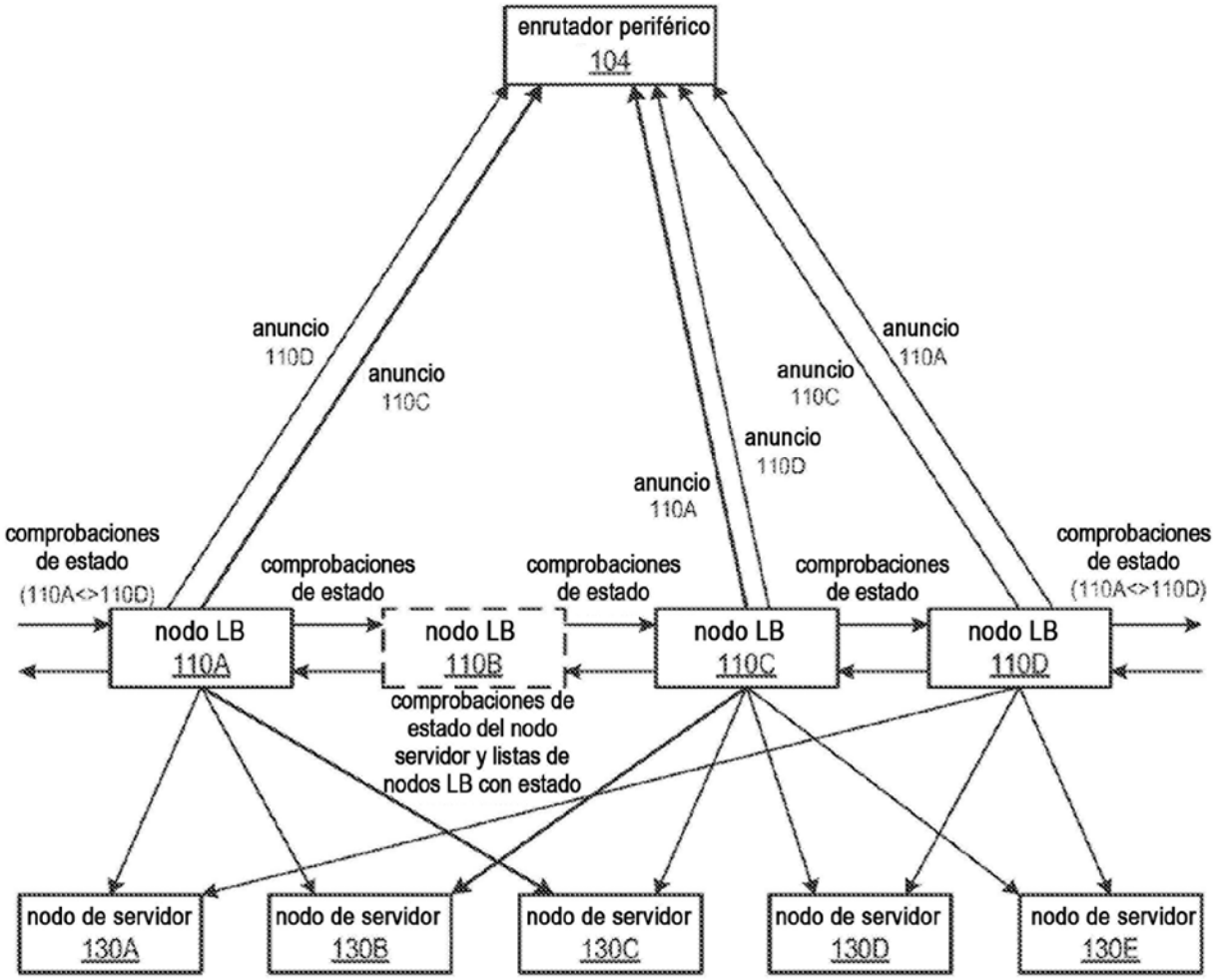
enrutador periférico
104
anuncio
110D
anuncio
110C
anuncio
110A
anuncio
110C
anuncio
110D
anuncio
110A
comprobaciones
de estado
(110A<>110D)
comprobaciones
de estado
comprobaciones
de estado
comprobaciones
de estado
comprobaciones
de estado
(110A<>110D)
nodo LB
110A
nodo LB
110B
nodo LB
110C
nodo LB
110D
comprobaciones de
estado del nodo
servidor y listas de
nodos LB con estado
nodo de servidor
130A
nodo de servidor
130B
nodo de servidor
130C
nodo de servidor
130D
nodo de servidor
130E

FIG. 18B

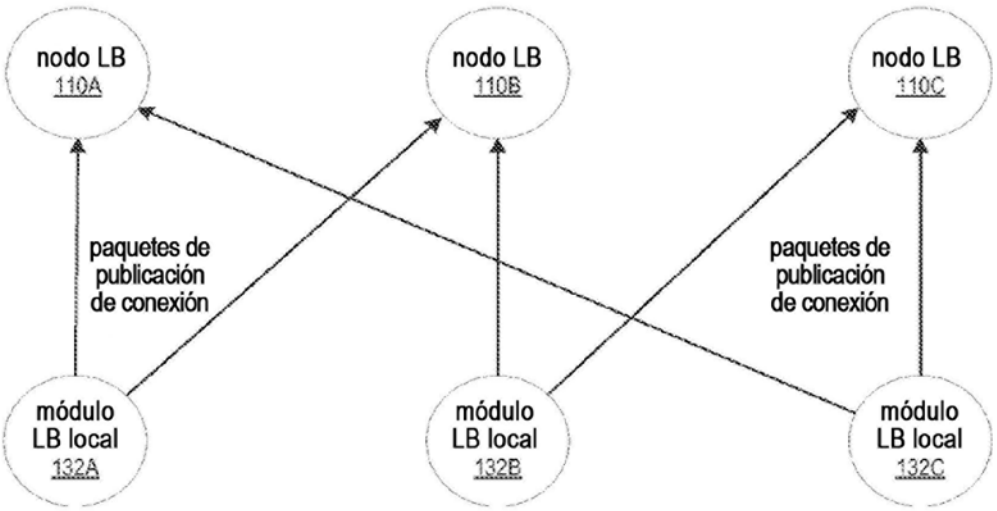

FIG. 19A

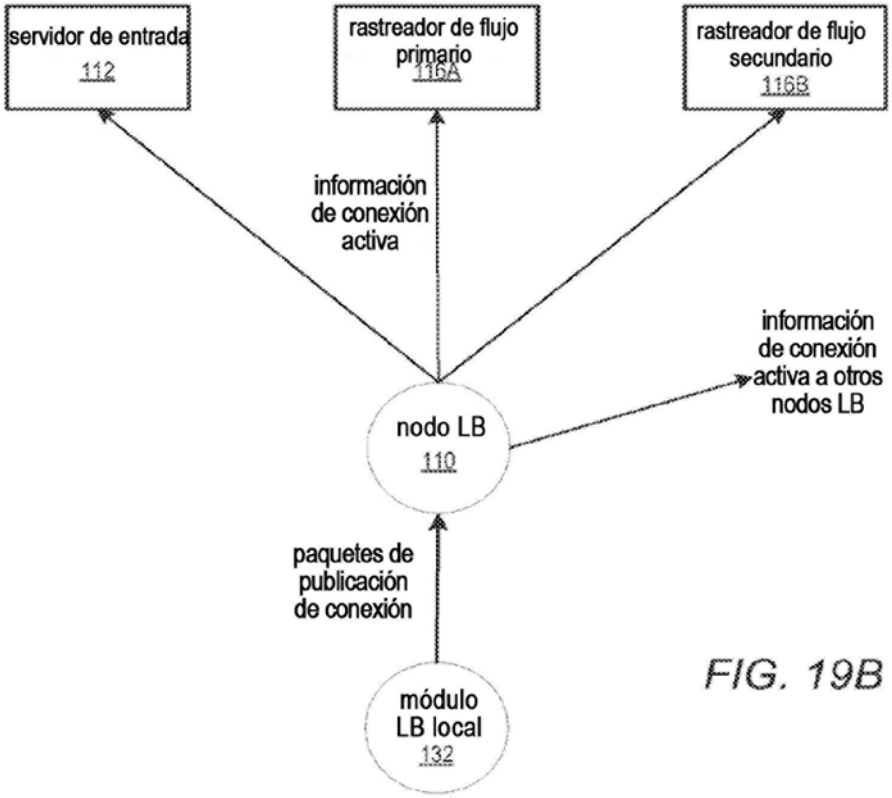

FIG. 19B

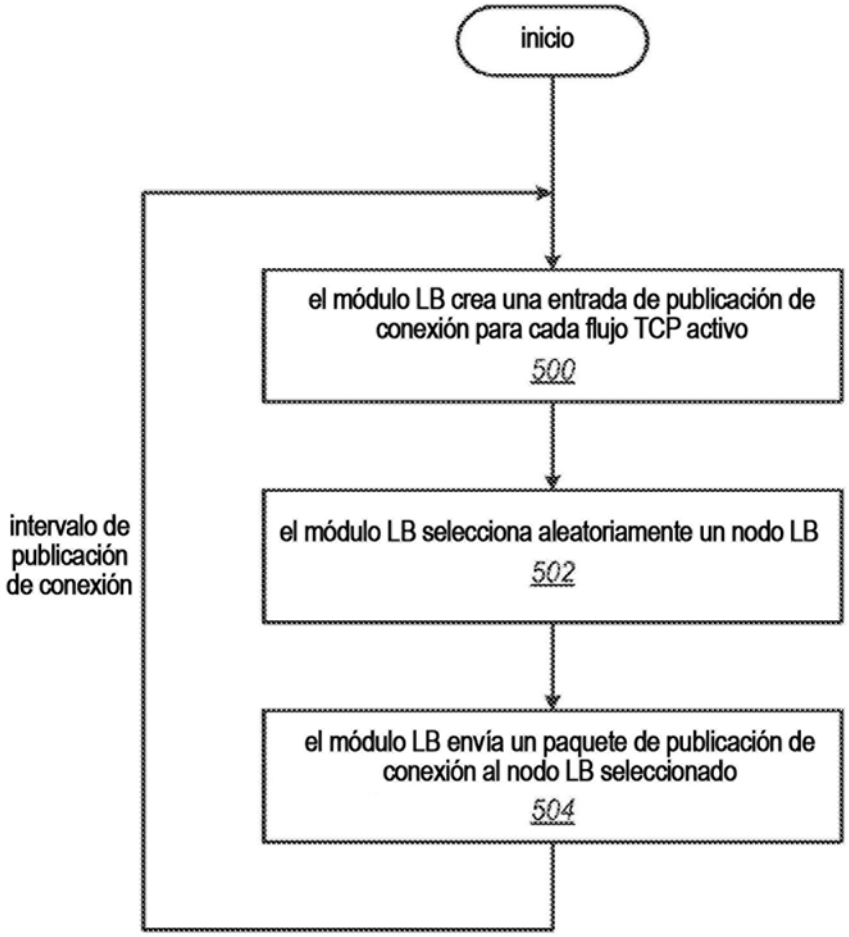
inicio
el módulo LB crea una entrada de publicación de conexión para cada flujo TCP activo
500
el módulo LB selecciona aleatoriamente un nodo LB
502
el módulo LB envía un paquete de publicación de conexión al nodo LB seleccionado
504
intervalo de publicación de conexión

FIG. 20

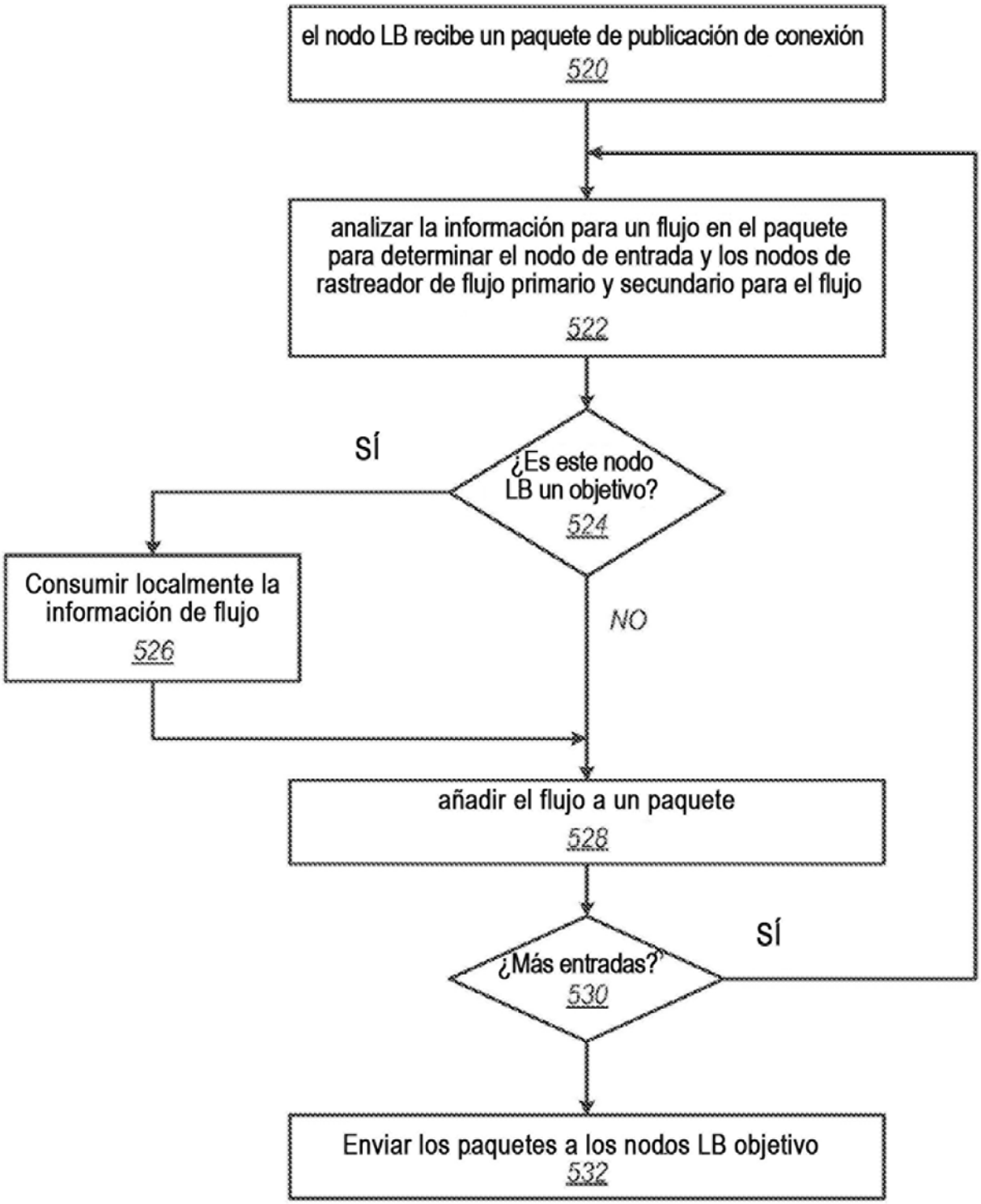
el nodo LB recibe un paquete de publicación de conexión
520
analizar la información para un flujo en el paquete para determinar el nodo de entrada y los nodos de rastreador de flujo primario y secundario para el flujo
522
¿Es este nodo LB un objetivo?
524
SÍ
NO
Consumir localmente la información de flujo
526
añadir el flujo a un paquete
528
¿Más entradas?
530
SÍ
Enviar los paquetes a los nodos LB objetivo
532

FIG. 21

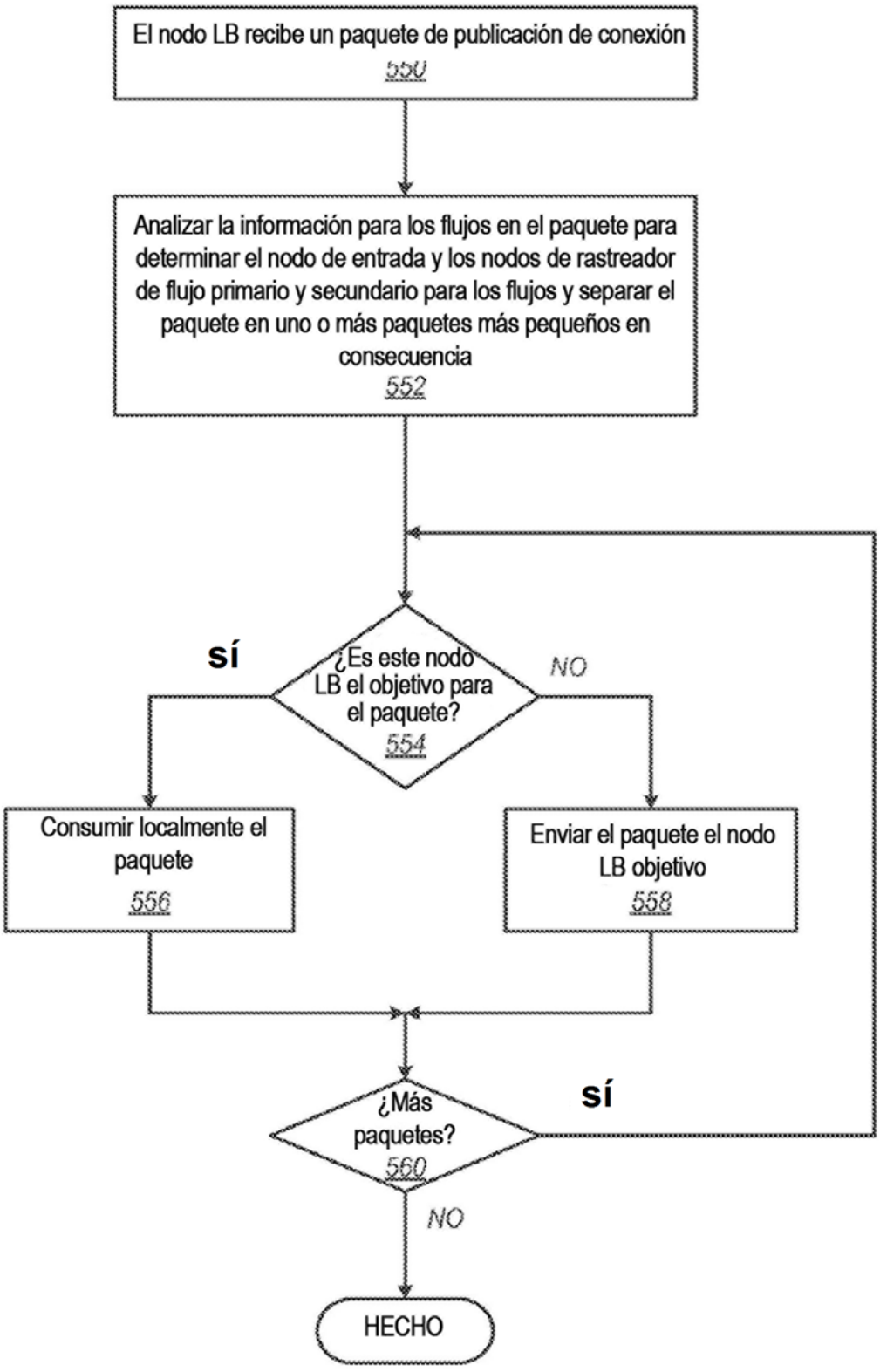
El nodo LB recibe un paquete de publicación de conexión
550
Analizar la información para los flujos en el paquete para determinar el nodo de entrada y los nodos de rastreador de flujo primario y secundario para los flujos y separar el paquete en uno o más paquetes más pequeños en consecuencia
552
¿Es este nodo LB el objetivo para el paquete?
554
sí
NO
Consumir localmente el paquete
556
Enviar el paquete el nodo LB objetivo
558
¿Más paquetes?
560
sí
NO
HECHO

FIG. 22

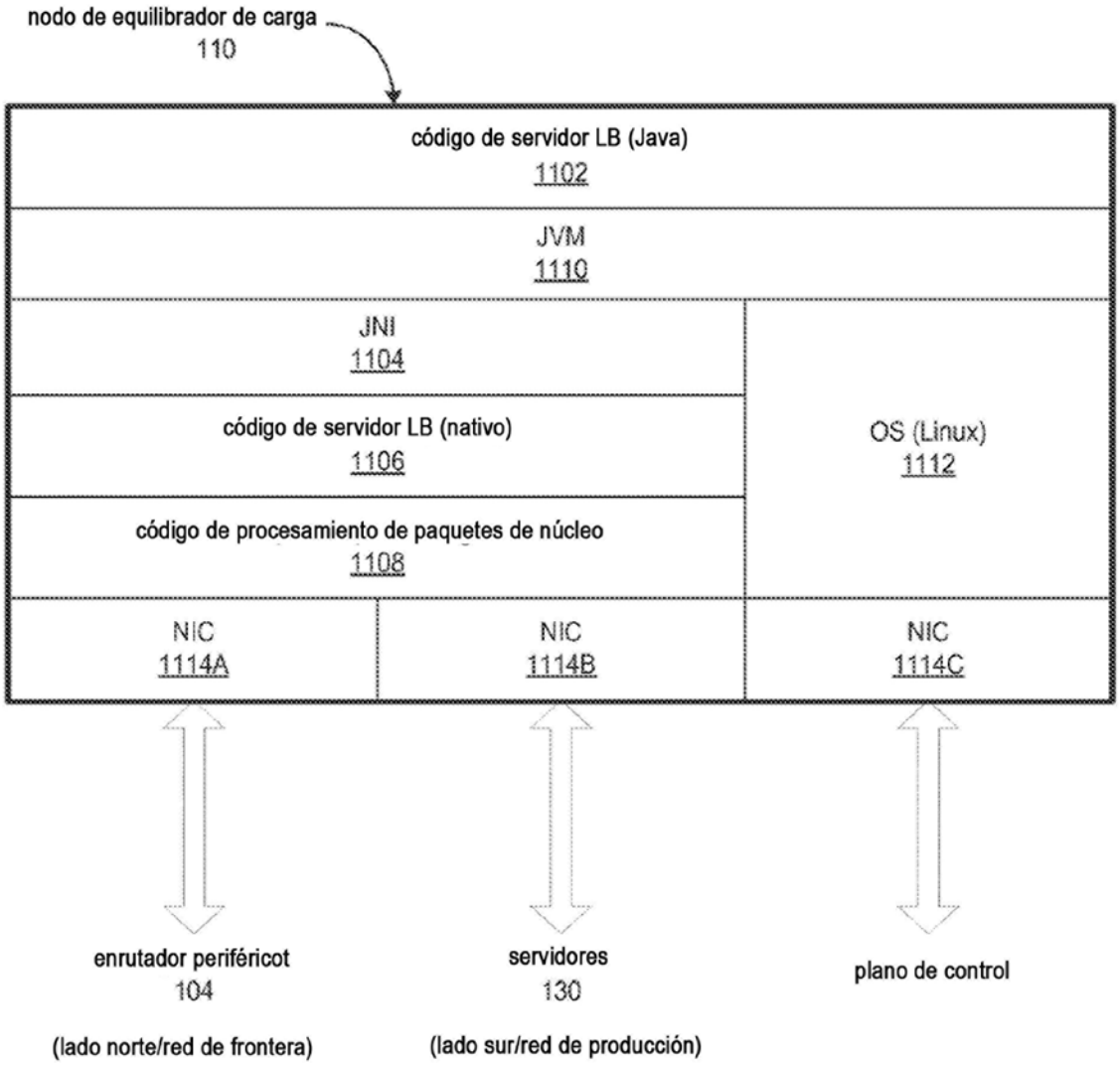
nodo de equilibrador de carga
110
código de servidor LB (Java)
1102
JVM
1110
JNI
1104
código de servidor LB (nativo)
1106
código de procesamiento de paquetes de núcleo
1108
OS (Linux)
1112
NIC
1114A
NIC
1114B
NIC
1114C
enrutador periféricot
104
(lado norte/red de frontera)
servidores
130
(lado sur/red de producción)
plano de control

FIG. 23

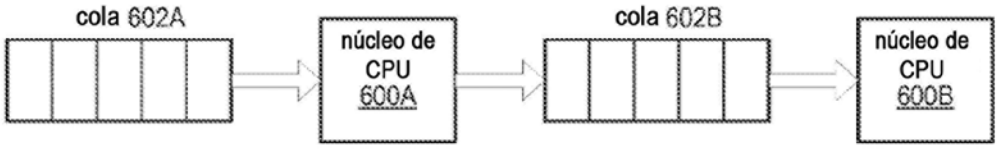
cola 602A
núcleo de CPU 600A
cola 602B
núcleo de CPU 600B

FIG. 24

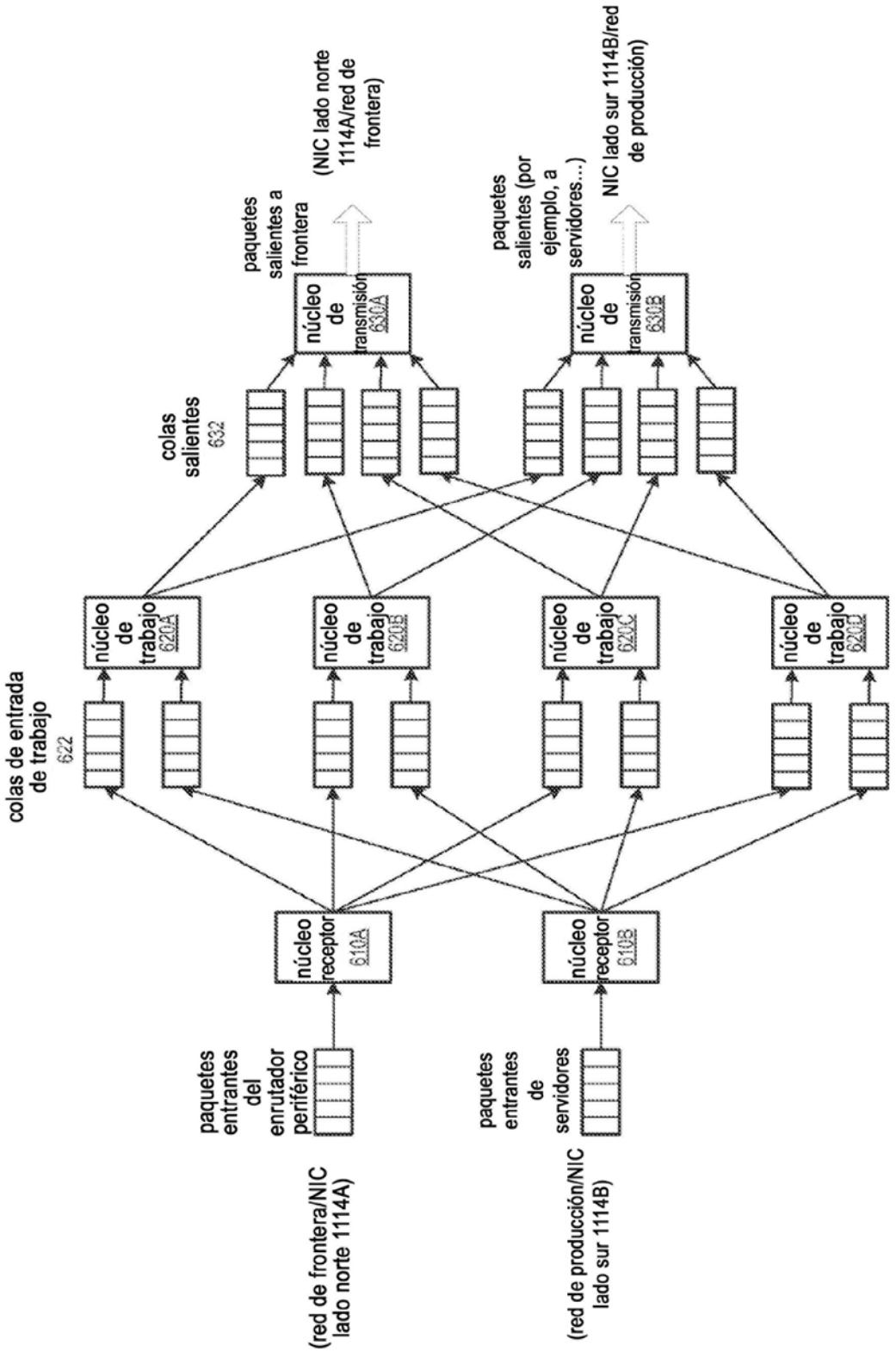
paquetes entrantes del enrutador periférico
(red de frontera/NIC lado norte 1114A)
paquetes entrantes de servidores
(red de producción/NIC lado sur 1114B)
núcleo receptor 610A
núcleo receptor 610B
colas de entrada de trabajo 622
núcleo de trabajo 620A
núcleo de trabajo 620B
núcleo de trabajo 620C
núcleo de trabajo 620D
colas salientes 632
paquetes salientes a frontera
paquetes salientes (por ejemplo, a servidores...)
núcleo de transmisión 630A
núcleo de transmisión 630B
(NIC lado norte 1114A/red de frontera)
NIC lado sur 1114B/red de producción)

FIG. 25

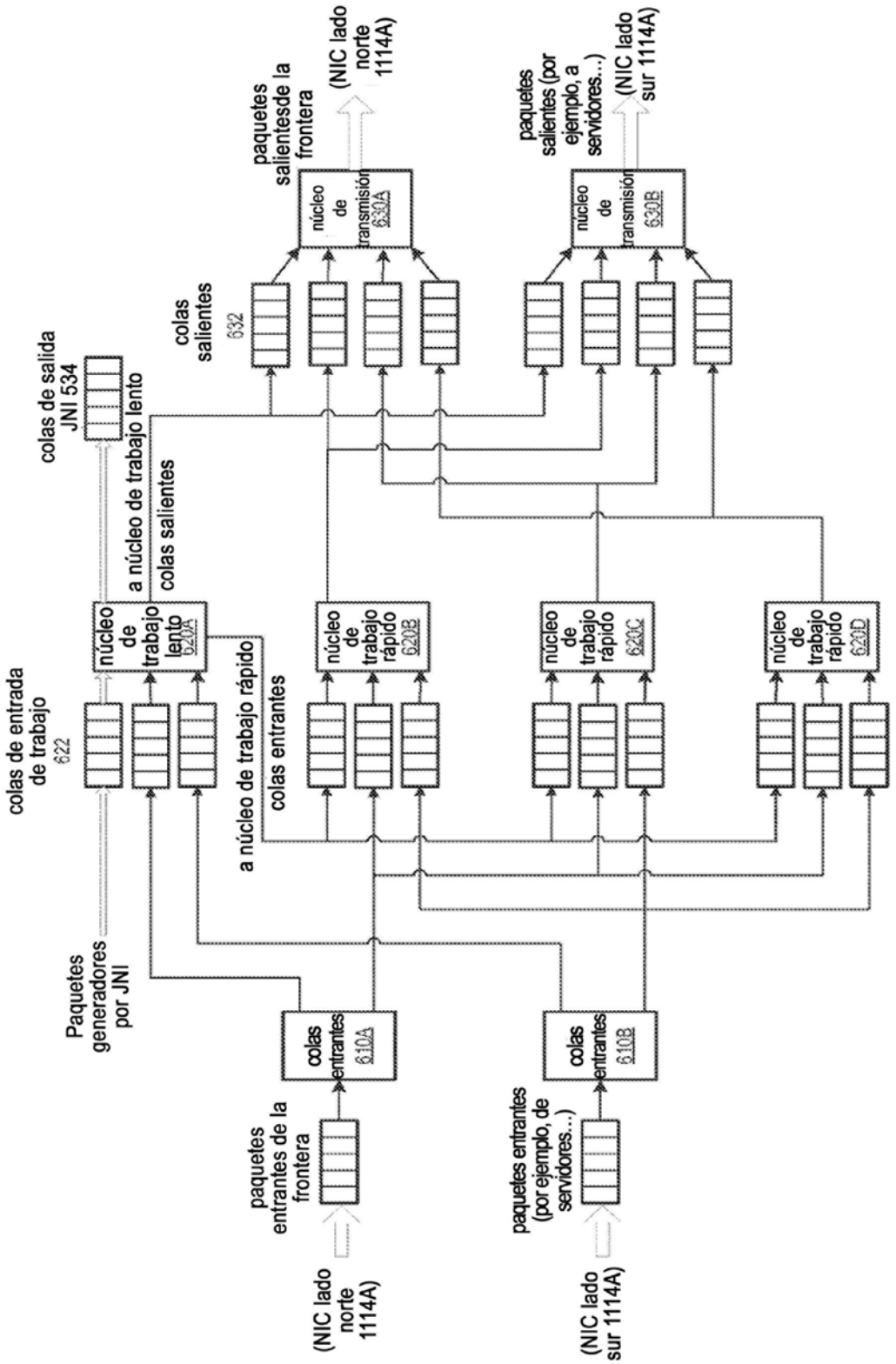
paquetes salientesde la frontera
(NIC lado norte 1114A)
paquetes salientes (por ejemplo, a servidores...)
(NIC lado sur 1114A)
núcleo de transmisión 630A
núcleo de transmisión 630B
colas salientes 632
colas de salida JNI 534
a núcleo de trabajo lento
colas salientes
núcleo de trabajo lento 620A
núcleo de trabajo rápido 620B
núcleo de trabajo rápido 620C
núcleo de trabajo rápido 620D
colas de entrada de trabajo 622
a núcleo de trabajo rápido
colas entrantes
Paquetes generadores por JNI
colas entrantes 610A
colas entrantes 610B
paquetes entrantes de la frontera
paquetes entrantes (por ejemplo, de servidores...)
(NIC lado norte 1114A)
(NIC lado sur 1114A)

FIG. 26

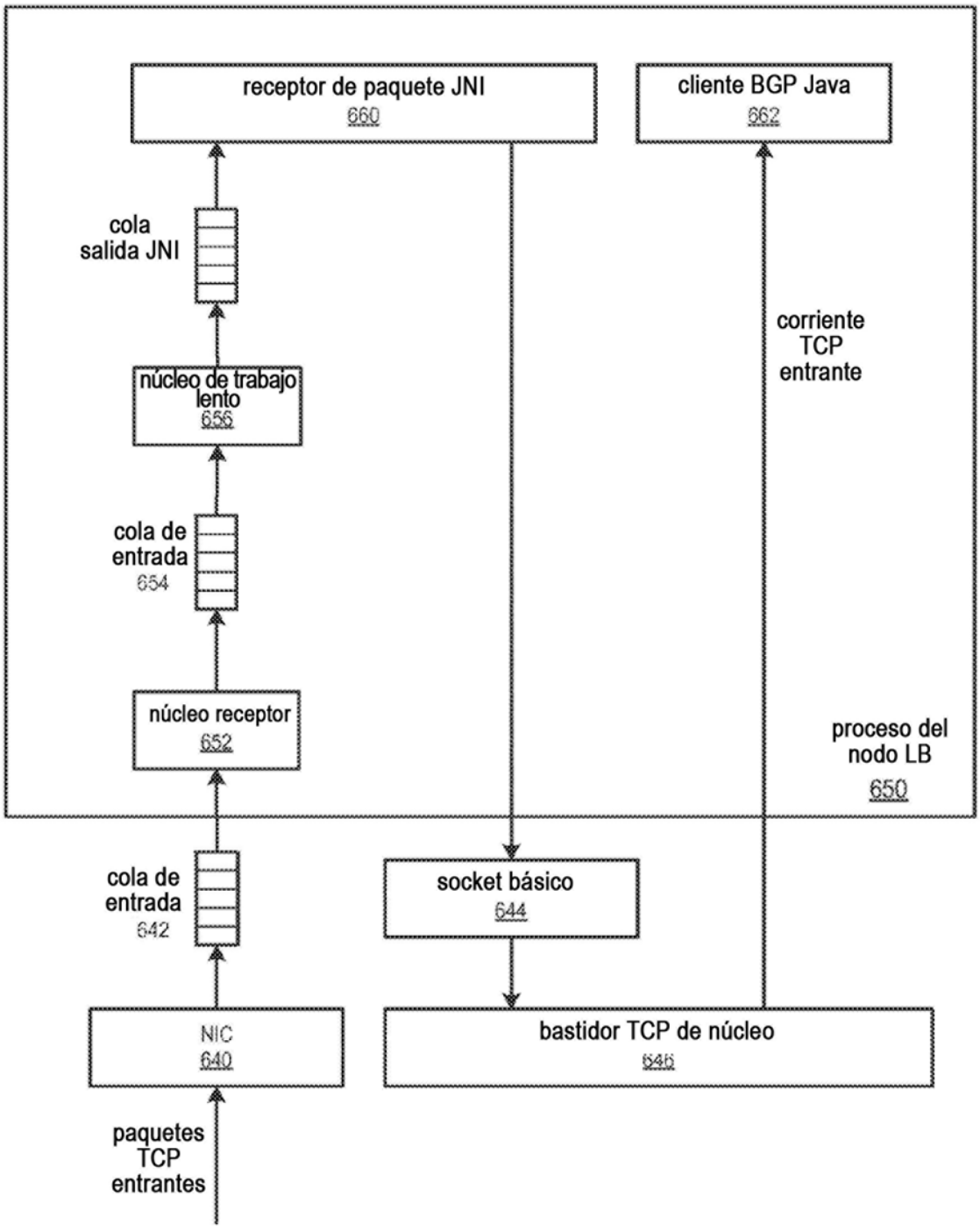
receptor de paquete JNI
660
cliente BGP Java
662
cola
salida JNI
corriente
TCP
entrante
núcleo de trabajo
lento
656
cola de
entrada
654
núcleo receptor
652
proceso del
nodo LB
650
cola de
entrada
642
socket básico
644
NIC
640
bastidor TCP de núcleo
646
paquetes
TCP
entrantes

FIG. 27

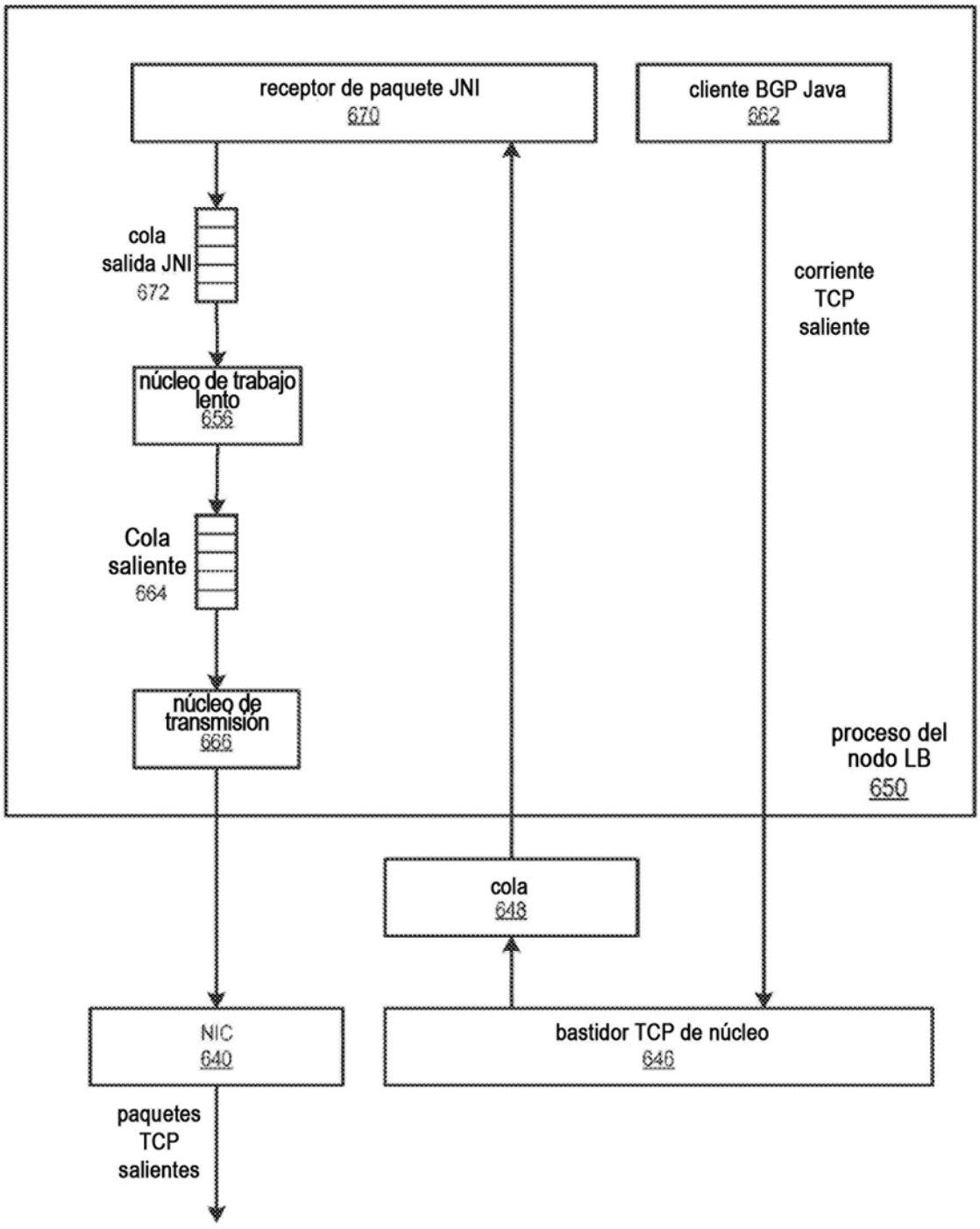
receptor de paquete JNI
670
cliente BGP Java
662
cola
salida JNI
672
corriente
TCP
saliente
núcleo de trabajo
lento
656
Cola
saliente
664
núcleo de
transmisión
666
proceso del
nodo LB
650
cola
648
NIC
640
bastidor TCP de núcleo
646
paquetes
TCP
salientes

FIG. 28

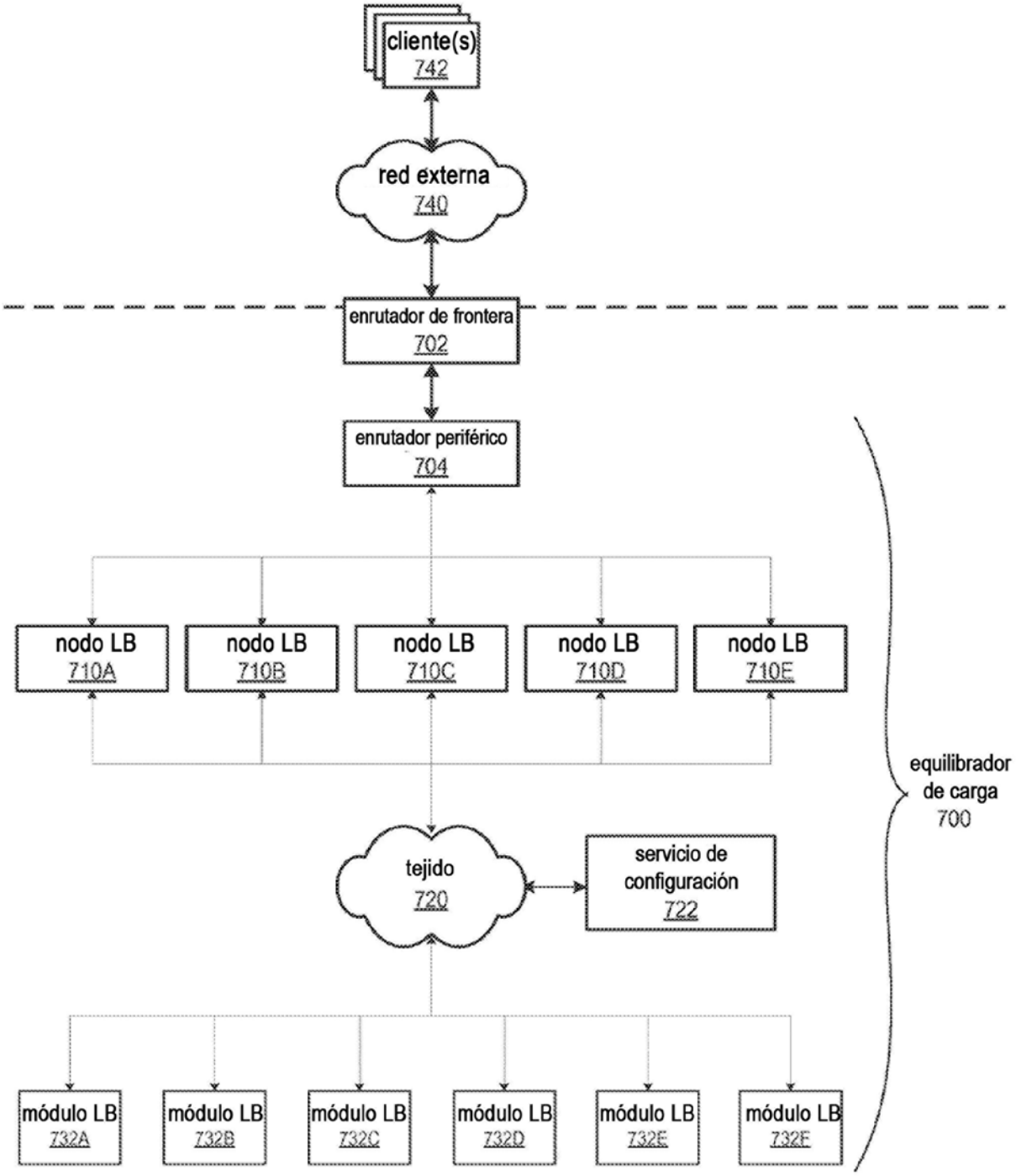
cliente(s)
742
red externa
740
enrutador de frontera
702
enrutador periférico
704
nodo LB
710A
nodo LB
710B
nodo LB
710C
nodo LB
710D
nodo LB
710E
equilibrador
de carga
700
tejido
720
servicio de
configuración
722
módulo LB
732A
módulo LB
732B
módulo LB
732C
módulo LB
732D
módulo LB
732E
módulo LB
732F

FIG. 29

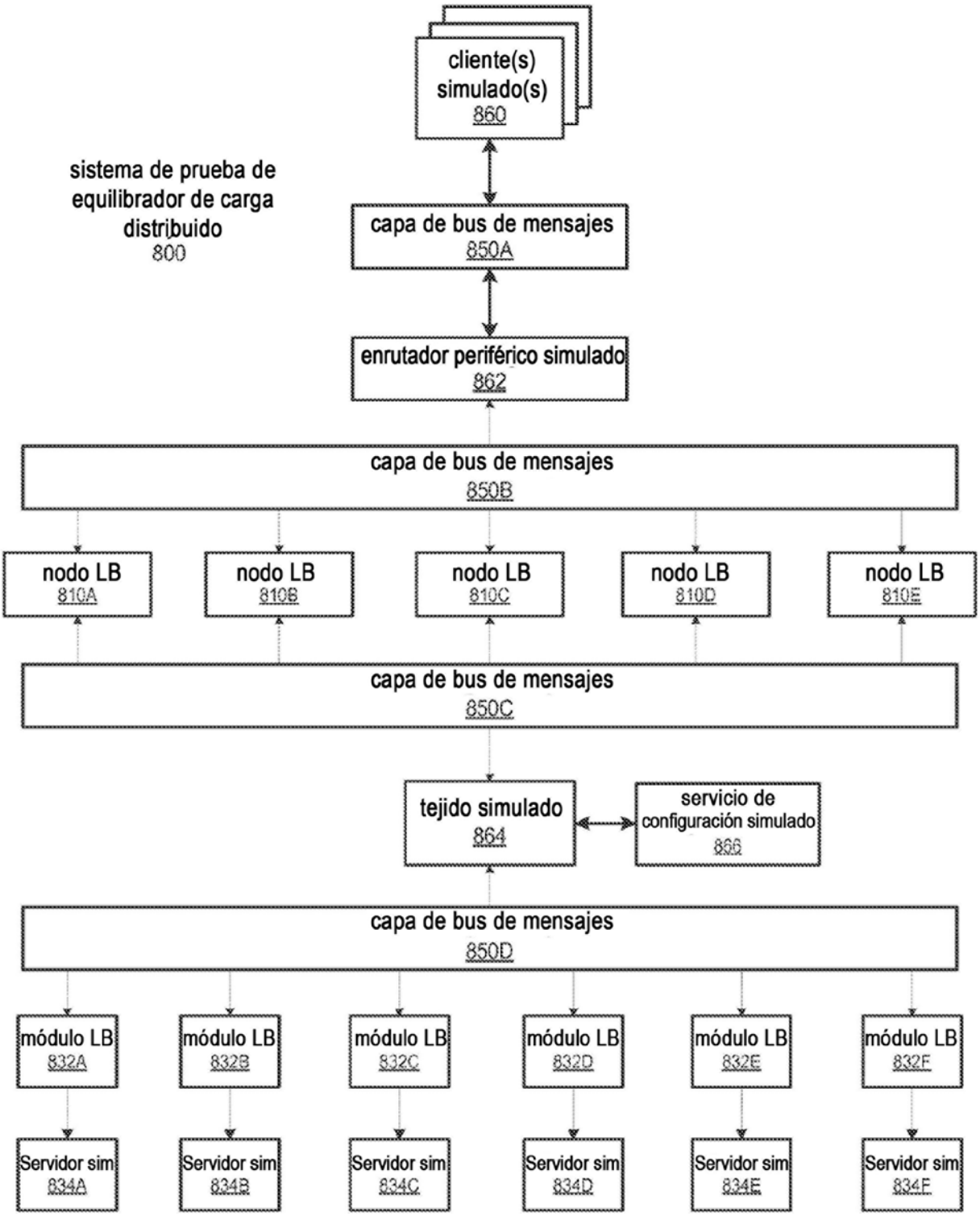
sistema de prueba de equilibrador de carga distribuido
800
cliente(s) simulado(s)
860
capa de bus de mensajes
850A
enrutador periférico simulado
862
capa de bus de mensajes
850B
nodo LB
810A
nodo LB
810B
nodo LB
810C
nodo LB
810D
nodo LB
810E
capa de bus de mensajes
850C
tejido simulado
864
servicio de configuración simulado
866
capa de bus de mensajes
850D
módulo LB
832A
módulo LB
832B
módulo LB
832C
módulo LB
832D
módulo LB
832E
módulo LB
832F
Servidor sim
834A
Servidor sim
834B
Servidor sim
834C
Servidor sim
834D
Servidor sim
834E
Servidor sim
834F

FIG. 30

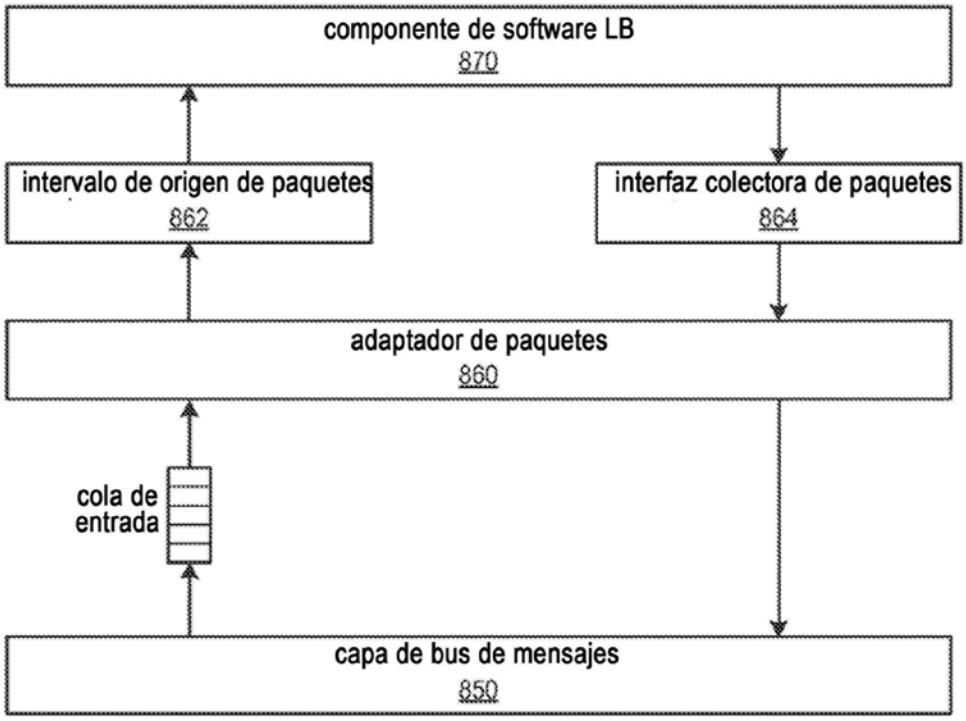
componente de software LB
870
intervalo de origen de paquetes
862
interfaz colectora de paquetes
864
adaptador de paquetes
860
cola de entrada
capa de bus de mensajes
850

FIG. 31

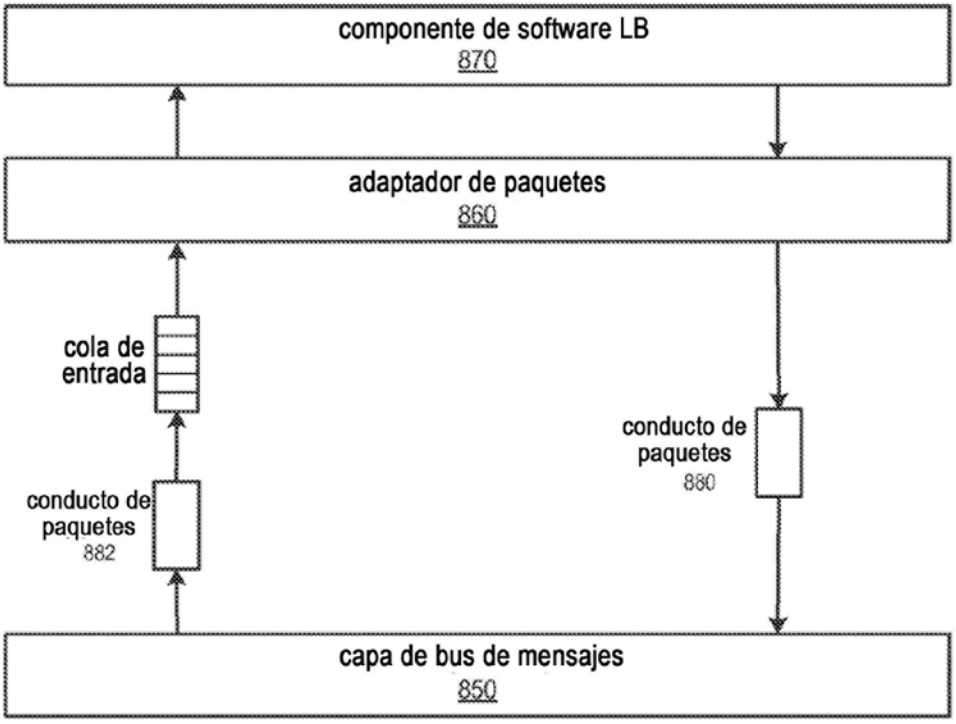
componente de software LB
870
adaptador de paquetes
860
cola de entrada
conducto de paquetes
882
conducto de paquetes
880
capa de bus de mensajes
850

FIG. 32

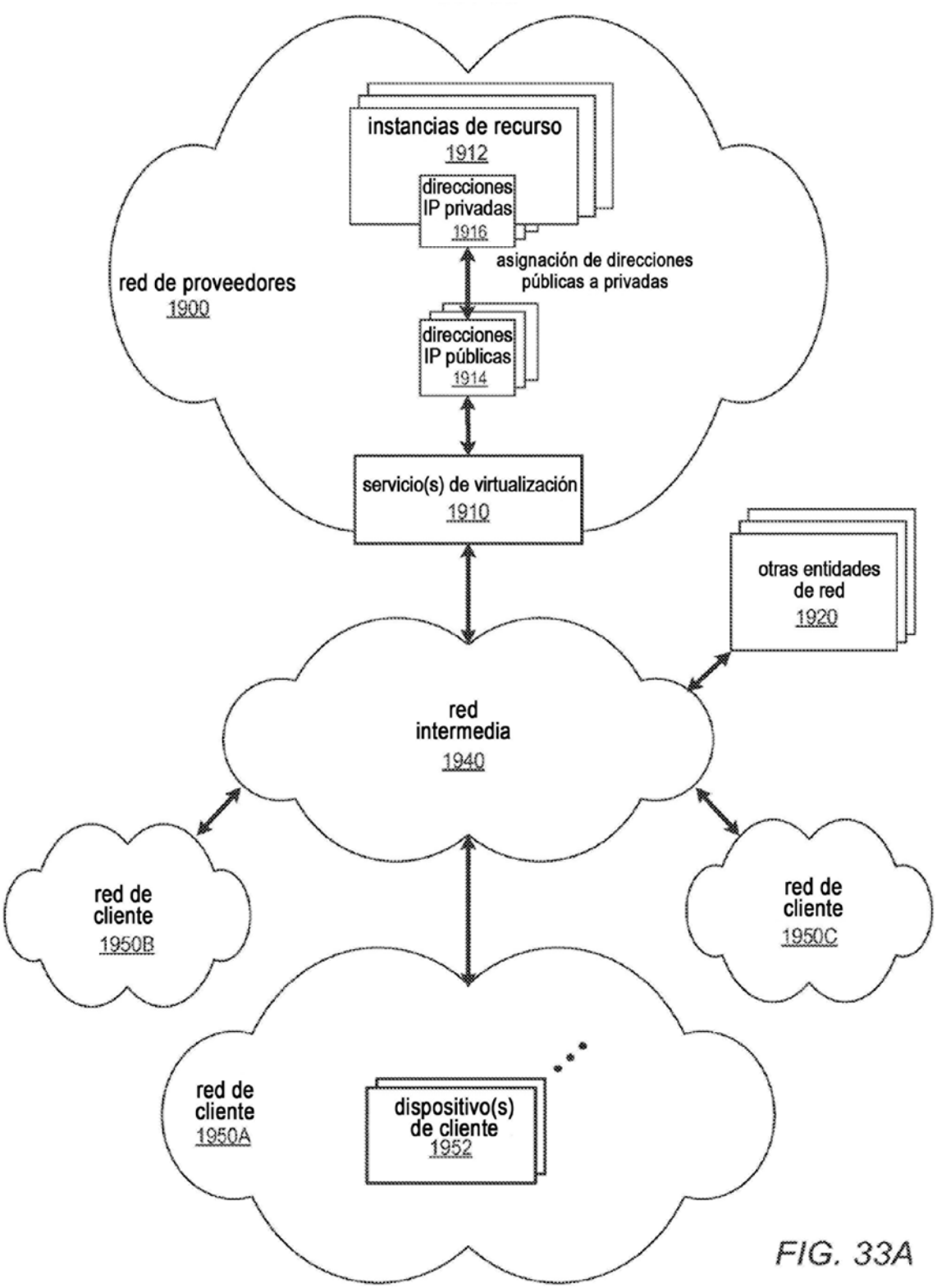
instancias de recurso
1912
direcciones IP privadas
1916
asignación de direcciones públicas a privadas
red de proveedores
1900
direcciones IP públicas
1914
servicio(s) de virtualización
1910
otras entidades de red
1920
red intermedia
1940
red de cliente
1950B
red de cliente
1950C
red de cliente
1950A
dispositivo(s) de cliente
1952

FIG. 33A

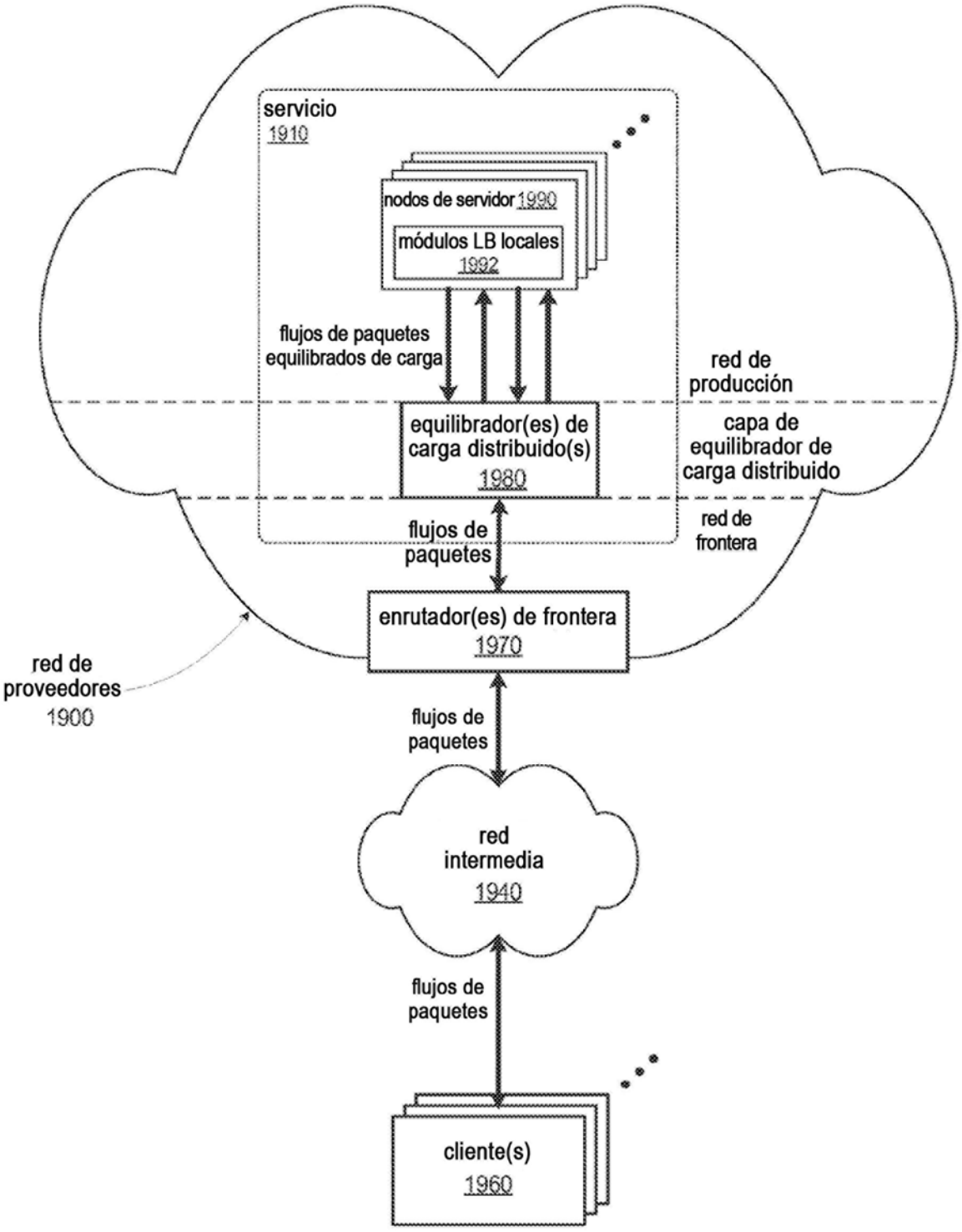
servicio
1910
nodos de servidor 1990
módulos LB locales
1992
flujos de paquetes equilibrados de carga
red de producción
equilibrador(es) de carga distribuido(s)
1980
capa de equilibrador de carga distribuido
red de frontera
flujos de paquetes
enrutador(es) de frontera
1970
red de proveedores
1900
flujos de paquetes
red intermedia
1940
flujos de paquetes
cliente(s)
1960

FIG. 33B

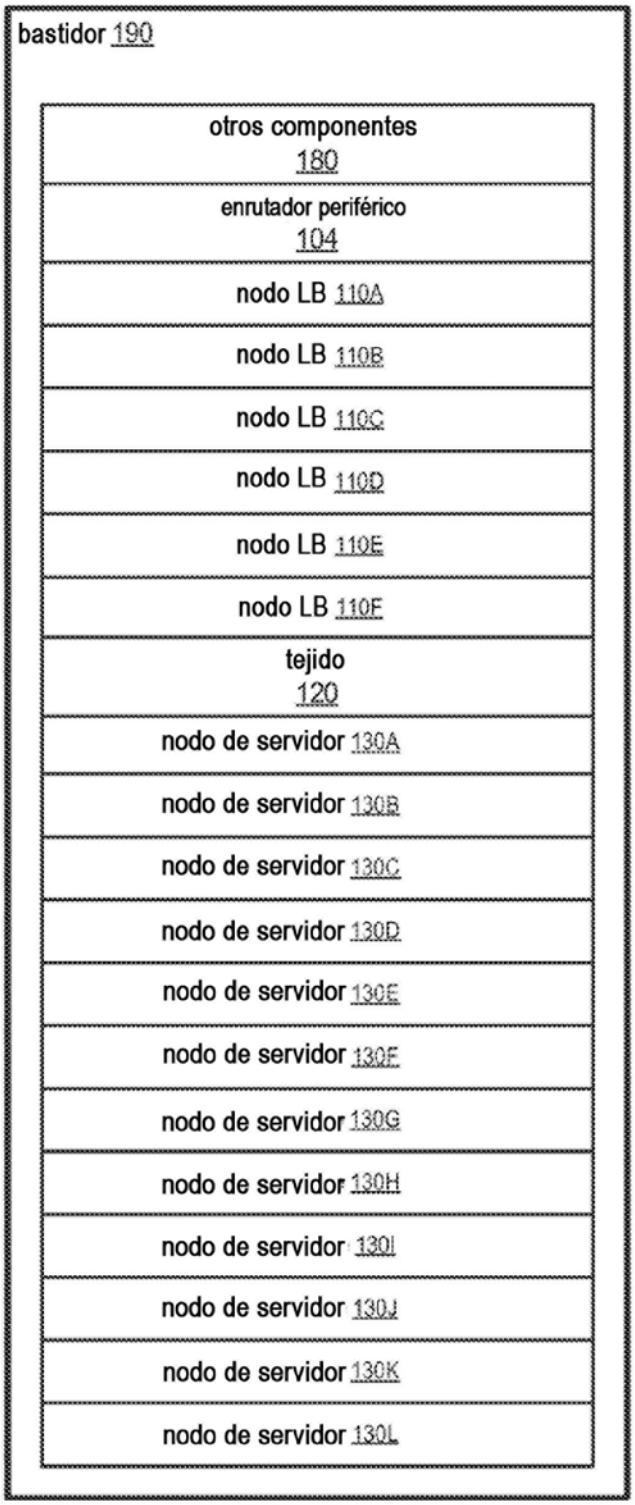

FIG. 34A

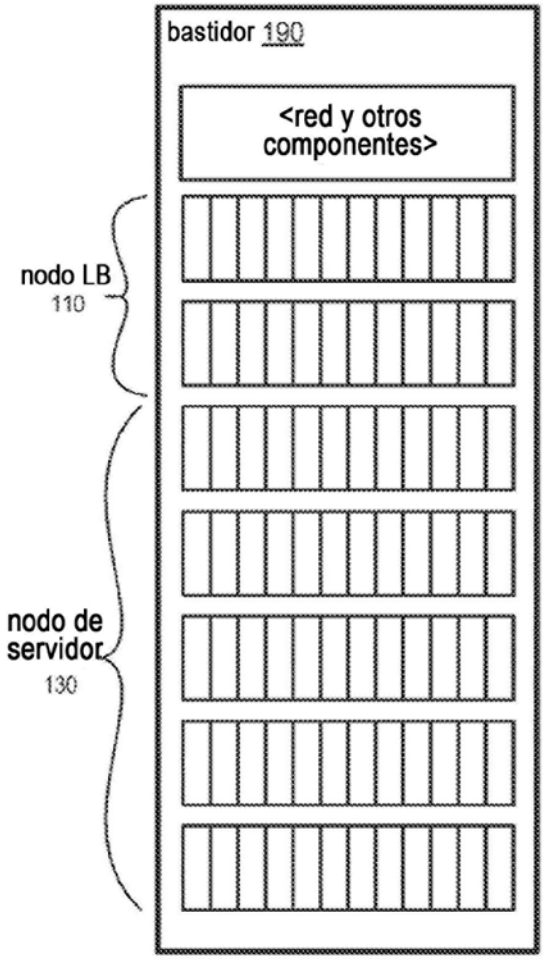

FIG. 34B

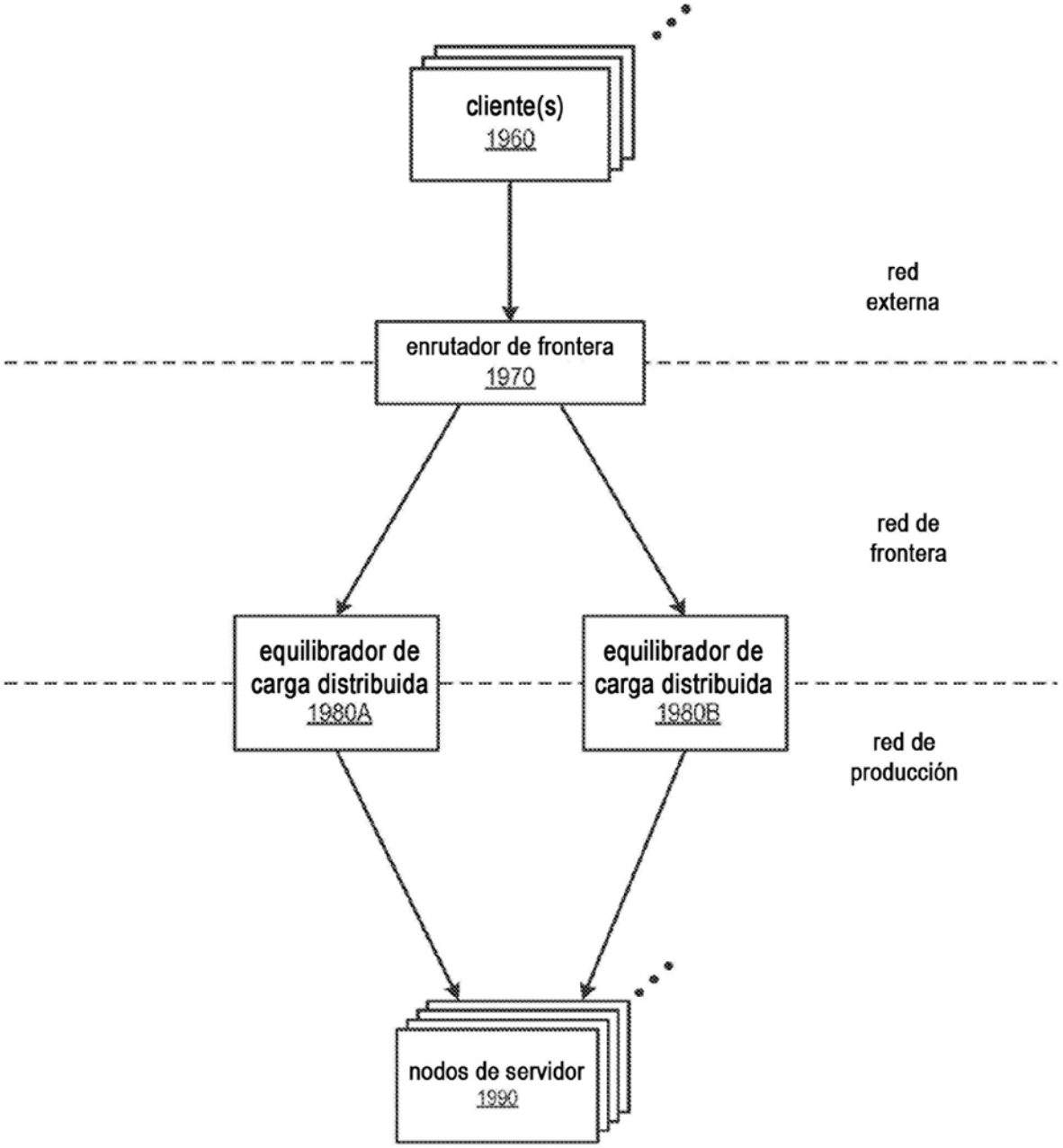
cliente(s)
1960
red
externa
enrutador de frontera
1970
red de
frontera
equilibrador de
carga distribuida
1980A
equilibrador de
carga distribuida
1980B
red de
producción
nodos de servidor
1990

FIG. 35

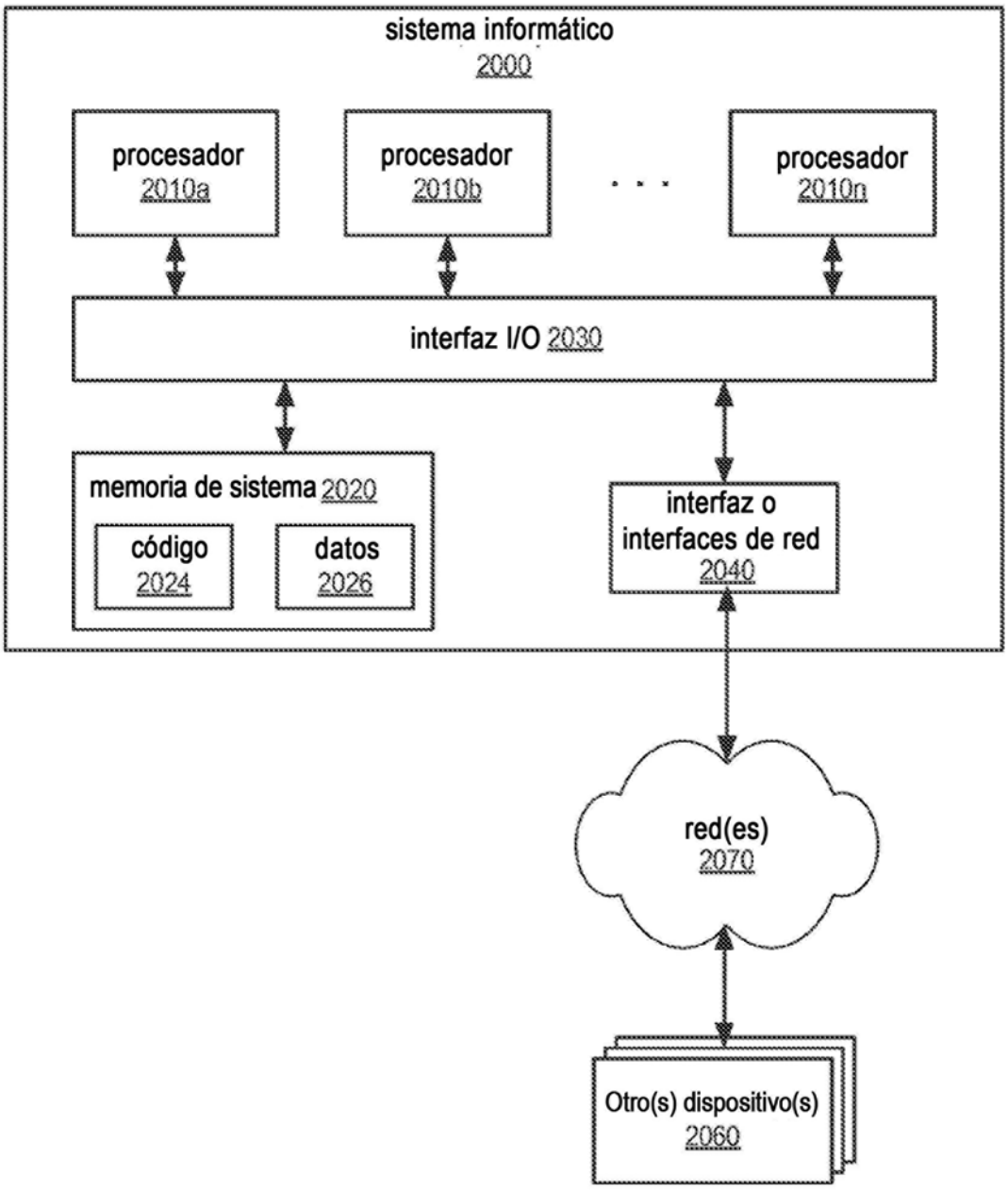
sistema informático
2000
procesador
2010a
procesador
2010b
procesador
2010n
interfaz I/O 2030
memoria de sistema 2020
código
2024
datos
2026
interfaz o
interfaces de red
2040
red(es)
2070
Otro(s) dispositivo(s)
2060

FIG. 36