(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 713 078**

(51) Int. CI.:

H04L 12/28 (2006.01)
H04L 12/701 (2013.01)
H04L 12/931 (2013.01)
H04L 12/46 (2006.01)
H04L 12/24 (2006.01)
H04L 12/751 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **06.08.2012 PCT/US2012/049692**

(87) Fecha y número de publicación internacional: **07.02.2013 WO13020126**

(96) Fecha de presentación y número de la solicitud europea: **06.08.2012 E 12819833 (0)**

(97) Fecha y número de publicación de la concesión europea: **05.12.2018 EP 2740242**

(54) Título: **Sistema y método para implementar y gestionar redes virtuales**

(30) Prioridad:

04.08.2011 US 201161514990 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
17.05.2019

(73) Titular/es:

MIDO HOLDINGS LTD. (100.0%)
c/o Mercuris Avocats Rue du Grand-Chêne 2
1003 Lausanne, CH

(72) Inventor/es:

DUMITRIU, DAN MIHAI;
LENGLET, ROMAIN F.V.;
DE CANDIA, GIUSEPPE y
MANDELSON, JACOB L.

(74) Agente/Representante:

MILTENYI , Peter

DESCRIPCIÓN

Sistema y método para implementar y gestionar redes virtuales

Antecedentes y sumario

La presente divulgación se refiere a la conexión en red, y más particularmente sistemas y métodos que implementan y gestionan redes virtuales.

La aparición de informática basada en la nube ha creado nuevas demandas de proveedores de servicio. A los proveedores de servicio les gustaría dotar a cada cliente de una red virtual, con la capacidad de añadir anfitriones y de cambiar la topología desacoplada de la red física. La virtualización de la red permite que los proveedores de servicio creen topologías de red configurables por el cliente que pueden cambiarse alterando enrutadores virtuales y conmutadores virtuales sin ningún cambio de hardware. Los enrutadores virtuales permiten también la segregación de datos de clientes por seguridad y precios basados en uso. Un hardware dedicado puede proporcionar algunas de estas características pero puede ser caro. Permanece la necesidad de una herramienta que permita que una red virtual se superponga sobre una red existente y que permita cambiar esa topología de la red virtual independientemente de la red subyacente.

El documento US 7.991.859 B1 representa la técnica anterior más próxima y técnicas para proporcionar redes informáticas virtuales gestionadas cuya topología de red lógica configurada puede tener uno o más dispositivos de conexión en redes virtuales, tales como mediante un servicio de red configurable accesible por red, con funcionalidad de conexión en red correspondiente proporcionada para comunicaciones entre múltiples nodos informáticos de una red informática virtual emulando la funcionalidad que se proporcionaría por los dispositivos de conexión en red si estuvieran presentes físicamente.

El documento EP 2139178 A1 se refiere a un método de determinar una trayectoria de ruta en una red superpuesta entre homólogos, y un nodo de red y un producto de programa informático para ejecutar dicho método. La red superpuesta entre homólogos comprende una pluralidad de nodos. Se identifica un hardware físico en el que funciona un primer nodo de dicha pluralidad de nodos.

El objeto de la invención se soluciona mediante las reivindicaciones independientes. En las reivindicaciones dependientes se mencionan realizaciones adicionales.

En el presente documento se dan a conocer un sistema y un método que facilitan el enrutamiento de paquetes usando una red virtual superpuesta en una red física. En diversas realizaciones, la presente divulgación prevé la interconexión flexible de elementos de red en múltiples capas del modelo OSI, incluyendo, L2 (Capa-2, es decir Capa de enlace), L3 (Capa-3, es decir Capa de red) y L4 (Capa-4, es decir Capa de transporte). Los elementos de red pueden estar interconectados con conmutadores L2 virtuales y enrutadores L3. Los paquetes de las redes L2 virtuales pueden transportarse por la red L3 existente usando tunelización, sin requerir ningún cambio a la red L3. Pueden usarse diversos métodos de tunelización, tal como GRE, Ethernet sobre IP, VXLan, MPLS sobre IP o CAPWAP. Los paquetes de protocolo de Internet (IP) enrutados por el enrutador L3 virtual pueden transportarse por la red L3 existente, sin requerir ningún cambio a la red L3 existente.

En una realización, para los elementos a los que se conectan, los conmutadores L2 virtuales y enrutadores L3 virtuales se tratan de conmutadores L2 físicos y enrutadores L3 físicos, aunque no pueden implementarse usando elementos de red L2 y L3 físicos. Puede haber un número arbitrario de elementos de red virtuales (conmutadores o enrutadores), conectados virtualmente cada uno a un número arbitrario de elementos de red. En una configuración, cada conmutador L2 virtual está conectado a un enrutador L3 virtual, que puede conectarse a un número arbitrario de otros enrutadores L3.

Los conmutadores L2 virtuales y los enrutadores L3 virtuales del sistema pueden conectarse a un gran número de elementos de red, independientemente de la separación geográfica. El sistema puede conectar elementos que son o bien físicos o bien virtuales, conectando, por ejemplo, máquinas virtuales emuladas en ordenadores servidores a enrutadores físicos que están conectados a Internet.

Se proporcionan un método y un sistema para crear y gestionar redes virtuales que comprenden una pluralidad de enrutadores y conmutadores virtuales. El método y el sistema también pueden proporcionar servicios de cortafuegos L3/L4, servicios de traducción de dirección de red de origen y/o destino, y equilibrio de carga tal como se describe en más detalle a continuación. En el presente documento se da a conocer un método para enrutar un paquete desde un primer nodo hasta un segundo nodo que comprende recibir un paquete en un primer nodo de una red subyacente; acceder a una tabla de enrutamiento virtual para determinar un siguiente salto para el paquete en una topología de red virtual, donde el siguiente salto es o bien un puerto (lógico) orientado hacia el interior o bien un puerto (materializado) orientado hacia el exterior, y continuar accediendo a tablas de enrutamiento virtuales posteriores en serie hasta que se determine que el siguiente salto sea un puerto orientado hacia el exterior en un segundo nodo de la red; y enviar el paquete por la red subyacente al puerto orientado hacia el exterior del segundo nodo. La etapa de acceder a una tabla de enrutamiento virtual para determinar un siguiente salto para el paquete también puede incluir ejecutar una búsqueda en cada tabla de enrutamiento virtual, donde la tabla de búsqueda contiene los datos del

siguiente salto para el paquete. En una realización, el primer nodo de la red está configurado para acceder a una red externa, y el segundo nodo de la red está configurado para hospedar una máquina virtual. El método también puede incluir aplicar una modificación de preenrutamiento y/o una modificación de posenrutamiento al paquete para al menos un salto en la red virtual. En una realización, el siguiente salto para un paquete se determina a partir de la dirección de origen y/o la dirección de destino. Además, los procesos de preenrutamiento y posenrutamiento pueden utilizar la dirección de origen, el puerto de origen, la dirección de destino y/o el puerto de destino para determinar la traducción o la modificación deseada del paquete. El método también puede comprender almacenar al menos una tabla de enrutamiento virtual en un estado distribuido en una pluralidad de nodos en la red subyacente. En diversas realizaciones, la red subyacente puede incluir una red Ethernet, una red de IP privada, una red de IP pública u otras redes capaces de proporcionar conectividad entre los nodos.

También se da a conocer un método para enrutar paquetes que comprende las etapas de recibir un paquete de un flujo en un primer nodo; acceder a una tabla de flujo y determinar que el paquete no coincide con una regla de flujo existente; comunicar el paquete a un motor de decisión; acceder a una topología de red virtual almacenada en una base de datos compartida accesible por una pluralidad de nodos; crear una regla de flujo para el paquete; y comunicar la regla de flujo a la tabla de flujo. La etapa de crear una regla de flujo puede comprender además determinar una secuencia de enrutamiento para el paquete en la red basándose en una topología virtual establecida por un titular de red.

También se da a conocer un método de seguimiento de conexiones con estado para borrar una entrada de flujo que comprende las etapas de recibir un paquete FIN con un número de secuencia en un nodo perimetral con un conmutador configurable de flujo; identificar una regla de flujo correspondiente al paquete en el conmutador configurable de flujo; identificar la regla de flujo para su borrado y comunicar la regla de flujo identificada a un estado distribuido en una base de datos compartida; y comunicar el paquete basándose en la regla de flujo correspondiente. En realizaciones, el sistema proporciona medios para simular una máquina de estado de conexión TCP y guardar su estado en la base de datos compartida.

En realizaciones, el flujo puede ser un flujo entrante o un flujo saliente de una conexión TCP. El método puede incluir además borrar el flujo identificado tras recibir un paquete de ACK correspondiente al paquete FIN. En una realización, el método comprende también identificar un flujo de sentido opuesto almacenado en el estado distribuido que corresponde al flujo identificado; identificar el flujo de sentido opuesto para su borrado; y borrar el flujo identificado y el flujo de sentido opuesto tras recibir un paquete de ACK correspondiente al paquete FIN.

En otra realización, un método para realizar la traducción de dirección de red de destino comprende las etapas de recibir un primer paquete en un primer nodo, teniendo el primer paquete una dirección de destino; crear una primera regla de flujo correspondiente al primer paquete, donde la primera regla de flujo comprende una agregación de las modificaciones hechas a un paquete que recorre una pluralidad de dispositivos virtuales en la topología de red virtual; aplicar la primera regla de flujo al primer paquete; recibir un segundo paquete en un segundo nodo en respuesta al primer paquete, teniendo el segundo paquete una dirección de origen; crear una segunda regla de flujo correspondiente al segundo paquete; y acceder el primer flujo desde un estado distribuido y aplicar la traducción de dirección de red de destino a la dirección de origen del segundo paquete. El método también puede comprender esperar hasta que la primera regla de flujo se almacene en el estado distribuido antes de reenviar el primer paquete de manera que el segundo paquete no se recibe hasta que se almacena la primera regla de flujo en el estado distribuido. En una realización, el primer paquete y el segundo paquete corresponden a una conexión TCP. En otra realización, el método comprende además aplicar un algoritmo de equilibrio de carga para equilibrar cargas en los recursos de red subyacente.

Breve descripción de los dibujos

La figura 1 ilustra una realización de un método para invalidar un flujo;

la figura 2 ilustra una realización de un método para aprendizaje de MAC;

la figura 3 ilustra una realización de un método para desaprendizaje de MAC;

la figura 4 ilustra una realización de un método para seguimiento de conexión;

la figura 5 ilustra una vista física de un sistema para la gestión de VPN;

la figura 6 ilustra una vista física de otro sistema para la gestión de VPN.

La figura 7 ilustra un ejemplo de un ordenador servidor usado por el sistema para enrutar paquetes hasta y desde una red general tal como Internet hasta un tejido IP de un proveedor de servicios.

La figura 8 ilustra una red física de ejemplo.

La figura 9 ilustra una red virtual de ejemplo que puede superponerse en la red física de la figura 8.

La figura 10 ilustra un proceso que se ejecuta en un conector de borde para enrutar paquetes en una topología

virtual compuesta de enrutadores virtuales.

La figura 11 ilustra un proceso que se ejecuta en un conector de borde para conmutar y enrutar paquetes en una topología virtual compuesta de enrutadores virtuales y conmutadores virtuales.

La figura 12 es una continuación del proceso de la figura 11.

La figura 13 ilustra una realización de una red virtual.

La figura 14 ilustra la red virtual de la figura 13 superpuesta en una red física.

Descripción detallada

Haciendo referencia generalmente a las figuras 1 a 14, las realizaciones del sistema y el método dados a conocer se refieren a sistemas informáticos que implementan y gestionan redes virtuales y pueden proporcionar soluciones de red definidas por software. El sistema y el método dados a conocer proporcionan una capa de abstracción de software para virtualizar una red para mejorar la efectividad de sistemas informáticos en la nube al tiempo que se reduce la complejidad de redes físicas y los costes de mantenimiento asociados.

En diversas realizaciones, se da a conocer un método informático que incluye recibir un paquete que llega a una primera interfaz de red de un primer nodo de una red subyacente. La primera interfaz de red puede implementarse en hardware o software en el primer nodo. Un motor de decisión puede invocarse para determinar cómo se gestionará el paquete. En un aspecto, el paquete y una identificación de la interfaz de red en la que el paquete llegó a la red se comunican al motor de decisión para procesarse. El motor de decisión puede simular cómo recorrerá el paquete la topología de red virtual que incluye cada uno de una pluralidad de dispositivos de red virtuales encontrados por el paquete. Además de simular cómo se pasa un paquete desde un dispositivo al siguiente a través de la red virtual, el motor de decisión también puede simular cómo afecta a cada uno de los dispositivos virtuales el paquete, tal como modificando las cabeceras de protocolo de paquete. Basándose en los resultados de simulación, el sistema puede procesar el paquete aplicando cada una de las modificaciones determinadas o una agregación de las modificaciones de modo que el paquete puede emitirse desde una interfaz de red en uno de los nodos de la red, donde la interfaz de red específica en la que emitir el paquete se determinó durante la simulación por el motor de decisión.

En cada etapa a través de la topología de red virtual, el motor de decisión determina cómo puede gestionarse el paquete por dispositivos sucesivos. En un ejemplo, el motor de decisión puede determinar que un paquete debe soltarse o ignorarse. Soltar un paquete puede producirse cuando se asocia un paquete dado con una comunicación o flujo que el sistema ya no está procesando. En otros ejemplos, un paquete puede soltarse puesto que un dispositivo virtual carece de instrucciones suficientes para gestionar un paquete del tipo recibido. Alternativamente, un dispositivo puede ser incapaz de enrutar de manera satisfactoria un paquete dado al destino especificado. Puede proporcionarse un mensaje de error u otra respuesta para alertar al remitente del paquete de que el paquete no llegó a su destino.

Para muchos paquetes, el motor de decisión determinará que el paquete debe emitirse desde un puerto virtual correspondiente a una segunda interfaz de red. La segunda interfaz de red puede estar en el primer nodo o en un segundo o un nodo diferente de la red subyacente dependiendo del destino del paquete y el mapeo de puerto a anfitrión virtual. Cuando el paquete va a entregarse a una segunda interfaz de red en un segundo nodo, el motor de decisión determina cómo va a procesarse el paquete y luego se entrega el paquete por la red subyacente al segundo nodo que va a emitirse por la segunda interfaz de red.

En múltiples realizaciones, las cabeceras de protocolo de los paquetes pueden modificarse antes de la entrega a la segunda interfaz de red. La modificación de las cabeceras de protocolo puede proporcionar la traducción de dirección de red, tunelización, VPN u otras características tal como se comenta más a fondo a continuación.

En una realización, el método incluye mantener un mapa de identificadores de nodo para direcciones de nodo para nodos en la red subyacente. Pueden usarse identificadores de nodo para distinguir nodos individuales para fines de enrutar paquetes en la red subyacente. Para entregar un paquete desde un primer nodo hasta un segundo nodo, el paquete puede reenviarse como la carga útil de un paquete de protocolo de tunelización (tal como Ethernet+IP+GRE). En realizaciones, el paquete de protocolo de tunelización tiene una clave de túnel que codifica un identificador global de una segunda interfaz de red. Un identificador global puede ser único dentro de la red de manera que cada interfaz de red puede identificarse de manera única. En cambio, un identificador local puede usarse dentro de un nodo, o dentro de un subconjunto de la red para identificar de manera única un puerto o una interfaz dentro de un subconjunto de la red. Cuando el paquete de protocolo de tunelización se recibe en el segundo nodo, la carga útil que contiene el paquete original se extrae junto con la clave de túnel. Entonces, la clave de túnel puede decodificarse para determinar el segundo identificador de interfaz de red virtual y el paquete emitido desde la segunda interfaz de red. De esta manera, el sistema puede utilizar el motor de decisión para determinar cómo ha de gestionarse un paquete cuando recorre el sistema y también para transportar de manera eficiente el paquete una vez que se hace la determinación.

En otras realizaciones, el motor de decisión determina que debe emitirse un paquete desde un conjunto de interfaces de red. Puede ser necesario emitir desde un conjunto de interfaces de red en aplicaciones de multidifusión o difusión. Las interfaces de red desde las que debe emitirse el paquete pueden ser locales para un solo nodo de la red, o pueden dispersarse por dos o más nodos. En cualquier caso, el sistema determinó que el paquete debe emitirse desde un conjunto de interfaces de red correspondiente a un identificador de conjunto de interfaz. Entonces el paquete se procesa entregando el paquete a cada interfaz de red en el conjunto que es local para el primer nodo. El paquete también se reenvía, con modificaciones, desde el primer nodo hasta el segundo nodo, a lo largo de un túnel. El paquete puede reenviarse como la carga útil de un paquete de protocolo de tunelización usando una clave de túnel que codifica el identificador de conjunto de interfaz. Cuando se recibe el paquete de tunelización en el segundo nodo, la clave de túnel puede decodificarse para determinar el identificador de conjunto de interfaz, y el paquete emitido en interfaces de red cualesquiera incluidas en ese conjunto que son locales para el segundo nodo. El conjunto de interfaces de red asociado con unos identificadores de conjunto de interfaz dados pueden almacenarse en la base de datos compartida accesible por cada nodo del sistema. Por tanto si un nodo recibe un identificador de conjunto de interfaz desconocido, el nodo puede acceder a la base de datos compartida para determinar qué interfaces de red están incluidas en el conjunto identificado. Además, un nodo puede almacenar o copiar en memoria caché el mapeo de interfaces de red a identificadores de conjunto de interfaz localmente en el nodo. Sin embargo, cuando el identificador de conjunto de interfaz cambia, los datos copiados en memoria caché localmente se invalidan y el nodo puede acceder a la base de datos compartida para recuperar el mapeo actual o actualizado de interfaces a conjuntos de interfaz. En realizaciones, una interfaz de red virtual puede pertenecer a más de un conjunto de interfaz.

En la aplicación en la que la red subyacente soporta multidifusión (tal como multidifusión de IP), cada identificador de conjunto de interfaz puede mapearse a una dirección de multidifusión, y cada nodo de la red subyacente puede mantener una suscripción de multidifusión para cada uno de los identificadores de conjunto de interfaz a los que pertenece al menos una de las interfaces de red virtual mapeadas a ese nodo. Entonces, pueden multidifundirse paquetes como la carga útil de un paquete de protocolo de tunelización a un segundo o más nodos. Entonces, cada nodo emite el paquete desde cualquier interfaz de red correspondiente a las interfaces en el conjunto que son locales a ese nodo. En diversas realizaciones, los identificadores de conjunto de interfaz están mapeados de manera única a direcciones de multidifusión.

Si la red subyacente no soporta la multidifusión, el motor de decisión determina el conjunto de nodos de red subyacente que tienen interfaces de red locales que pertenecen al conjunto de interfaces de red al que va a enviarse el paquete. Entonces, el paquete se reenvía desde el primer nodo hasta cada nodo en el conjunto de nodos de red subyacente en un paquete de protocolo de tunelización tal como se describió anteriormente. Entonces, cada nodo emitió el paquete a las interfaces de red correspondientes asociadas con el conjunto identificado.

El motor de decisión determina cómo gestionar un paquete dado basándose en una simulación de ese paquete que recorre la topología de red virtual. En muchas aplicaciones, están asociados múltiples paquetes como un flujo y cada paquete en el flujo, donde va a procesarse de la misma manera cada paquete en el flujo, por ejemplo, todos los paquetes de una dirección de una conexión TCP. En realizaciones, tras recibir el primer paquete de flujo, el sistema invoca el motor de decisión para determinar cómo va a gestionarse el paquete de ese flujo. Entonces, el motor de decisión puede almacenar las acciones o reglas para gestionar paquetes posteriores de ese flujo. Las acciones o reglas almacenadas pueden almacenarse en la base de datos compartida de manera que las reglas están disponibles para todos los nodos del sistema para gestionar paquetes de un flujo dado. Alternativamente, las reglas pueden almacenarse localmente.

En realizaciones, la salida del motor de decisión incluye un patrón de cabecera de protocolo de paquete que puede usarse para hacer coincidir otros paquetes para los que la salida del motor de decisión será la misma que para el primer paquete. Dicho de otro modo, el patrón de cabecera de protocolo de paquete puede usarse para identificar paquetes que se tratarán de manera idéntica aplicando las acciones o reglas determinadas para el primer paquete. En realizaciones, la salida del motor de decisión es el resultado de la simulación realizada para determinar cómo ha de procesarse el paquete. Se almacenan el patrón de cabecera de protocolo de paquete y el resultado de la simulación para el primer paquete. Tras recibir un segundo paquete, las cabeceras del segundo paquete se comparan con el patrón de cabecera de protocolo de paquete al tiempo que se ignoran campos que cambian en una base por paquete tal como número de secuencia TCP. Si el segundo paquete coincide con el patrón, entonces el segundo paquete se considera que forma parte del mismo flujo y el resultado almacenado anteriormente de la simulación para el primer paquete se recupera y se aplica al segundo paquete. El resultado almacenado de la simulación puede recuperarse desde la base de datos compartida, una memoria caché local o cualquier otra memoria adecuada para contener las reglas que van a aplicarse a paquetes del flujo. En realizaciones, el motor de decisión puede aplicar al flujo las reglas a los paquetes segundos y posteriores en un flujo. En otras realizaciones, sin embargo, la simulación asociada con el primer paquete para determinar las reglas para el flujo, y la aplicación de esas reglas a paquetes pueden dividirse para mejorar la velocidad o eficiencia del sistema.

La copia en memoria caché o el almacenamiento de resultados de simulación específicos puede determinarse por el sistema. Por ejemplo, el motor de decisión puede sugerir qué resultados de simulación deben copiarse en memoria caché. En este contexto, una sugerencia puede incluir una recomendación de que se almacena un resultado específico sujeto a otra consideración tal como eficiencia o capacidad de almacenamiento disponible. En un ejemplo,

el sistema puede elegir no almacenar los resultados de simulación para un flujo usado con poca frecuencia. También puede informarse al motor de decisión sobre qué desenlaces o resultados de simulación están copiados en memoria caché realmente o qué resultados se han invalidado o expulsado de los resultados copiados en memoria caché. Un resultado copiado en memoria caché puede invalidarse por una variedad de motivos. En un ejemplo, un desenlace puede expulsarse para liberar espacio de almacenamiento para flujos nuevos o usados con más frecuencia. El rendimiento del sistema puede mermar a medida que aumenta el número de reglas de flujo almacenadas, por tanto en algunas realizaciones, el número de reglas de flujo almacenadas puede limitarse para aumentar la eficiencia y la velocidad de funcionamiento.

Con el tiempo, puede cambiar la configuración de la topología de red virtual y/o los dispositivos virtuales. Esto puede provocar que resultados de simulación realizados anteriormente ya no reflejen las reglas apropiadas que van a aplicarse a paquetes posteriores para uno o más flujos establecidos anteriormente. Por tanto, el sistema puede comunicar entre la base de datos compartida, el motor de decisión y otros componentes para detectar y responder a cambios en los resultados de simulación copiados en memoria caché. En un ejemplo, el motor de decisión puede pedir en cualquier momento que se elimine una entrada copiada en memoria caché específica basándose en un cambio en la configuración tal como se refleja en el estado global almacenado en la base de datos compartida. Cuando se elimina una entrada copiada en memoria caché, tras la recepción del siguiente paquete correspondiente al flujo eliminado, el motor de decisión se invocará y la simulación se recalculará para determinar cómo han de procesarse paquetes de ese flujo. Entonces, el resultado de simulación revisado puede almacenarse y aplicarse a paquetes posteriores que coinciden con el patrón de cabecera de protocolo de paquete asociado con el flujo.

El sistema puede determinar el patrón de cabecera de protocolo de paquete para un flujo dado durante la simulación de un paquete. En realizaciones, un paquete incluye una cabecera de protocolo que tiene una pluralidad de campos y el patrón se determina identificando cada uno de los campos que se leen durante la simulación por el motor de decisión. De esta manera, se identifica cada campo leído a medida que se recorre la topología de red virtual, incluyendo un campo leído por la pluralidad de dispositivos de red virtuales. Cada campo que se lee puede incluirse como parte del patrón. En cambio, los campos que no se leen y por tanto no afectan a la gestión del paquete pueden designarse como comodines.

En todavía otra realización, se recibe un primer paquete en un segundo nodo de la red subyacente y el segundo nodo genera un patrón de cabecera de protocolo que puede usarse para identificar otros paquetes que tienen la misma clave de túnel que el primer paquete. Entonces, el segundo nodo usa el patrón de cabecera de protocolo para identificar un segundo paquete que coincide con el patrón de cabecera de protocolo de paquete y emite los segundos (y posteriores) paquetes desde las mismas interfaces de red locales desde las que se emitió el primer paquete. De esta manera, el segundo nodo simplifica el proceso de determinar en qué puertos emitir el segundo paquete y mejora la eficiencia del sistema. En realizaciones, un conmutador programable de flujo en el nodo puede depender del patrón de cabecera de protocolo en combinación con una clave de túnel, o puede depender sólo de la clave de túnel para identificar paquetes posteriores que se tratarán de manera similar a un primer paquete.

En otra realización, al tiempo que se determina la simulación para un primer paquete, el motor de decisión puede pedir que un sistema de base genere uno o más paquetes adicionales desde una interfaz de red o conjunto de interfaces de red. En realizaciones, pueden requerirse paquetes adicionales para determinar el comportamiento de puertos de la red. Al pedir paquetes adicionales, el motor de decisión puede obtener información adicional en partes de la red para ayudar en el proceso de simulación. Cada uno de los paquetes adicionales puede procesarse sustancialmente tal como se describe en el presente documento, o puede dotarse de un tratamiento diferente según sea necesario para que el motor de decisión desarrolle la información necesaria sobre la red.

En otra realización, un método informático incluye mantener una base de datos compartida accesible desde una red subyacente que tiene una pluralidad de nodos. La base de datos compartida almacena una topología de red virtual y configuraciones de dispositivo virtual para una pluralidad de dispositivos de red virtuales. Un paquete de red llega a una primera interfaz de red de un primer nodo de la red subyacente. El método incluye además determinar una acción para procesar el paquete de red basándose en una simulación del recorrido del paquete de la topología de red virtual incluyendo la pluralidad de dispositivos de red virtuales. En realizaciones, la acción es una regla de flujo operable en un conmutador programable de flujo operable para procesar paquetes recibidos en el nodo de la red subyacente.

La simulación del recorrido del paquete de la topología de red virtual puede realizarse por un motor de decisión, tal como el motor de decisión comentado anteriormente. El motor de decisión puede ser operable en cada uno de la pluralidad de nodos para realizar la simulación para paquetes recibidos en cada nodo. Alternativamente, el motor de decisión puede operar en un nodo independiente en comunicación con cada uno de los nodos que recibe paquetes para los que se requiere una simulación.

La topología de red virtual incluye una pluralidad de puertos virtuales correspondientes a la pluralidad de dispositivos de red virtuales. Cada dispositivo de red virtual tiene uno o más puertos virtuales. Unos puertos virtuales pueden ser o bien un puerto orientado hacia el exterior asociado con una interfaz de red de un nodo de la red subyacente, o un puerto orientado hacia el interior asociado con un enlace virtual entre dispositivos de red virtuales. Un enlace virtual representa la conexión lógica de un puerto virtual a otro puerto virtual y también puede denominarse cable virtual.

La base de datos compartida almacena la topología de red virtual y configuraciones de dispositivo virtual que incluyen la configuración de los puertos virtuales. En realizaciones, la base de datos compartida puede incluir una o más de una configuración para cada uno de la pluralidad de puertos virtuales que incluyen una identificación del puerto virtual como una de un puerto orientado hacia el exterior o un puerto orientado hacia el interior, una configuración para cada uno de la pluralidad de dispositivos de red virtuales asociados con la pluralidad de puertos virtuales, un mapeo de identificadores de interfaz de red a identificadores de los nodos de red subyacente, un mapeo bidireccional de puertos exteriores a interfaces de red de nodos de red subyacente correspondientes, y un mapeo de cada puerto orientado hacia el interior de cada dispositivo al puerto orientado hacia el interior homólogo de otro dispositivo conectado mediante un enlace virtual. Tal como se usa en el presente documento, un puerto orientado hacia el interior homólogo es el puerto virtual conectado a un puerto virtual dado por una conexión lógica. Un puerto orientado hacia el interior tiene un solo igual, por tanto cada puerto virtual interior es el igual del puerto virtual interior al que está conectado.

La configuración de puertos virtuales puede configurarse dependiendo de la configuración deseada del sistema, y un usuario de sistema puede definir los puertos virtuales. Un paquete que entra en/sale de un puerto virtual exterior está entrando a/saliendo de la red virtual. En cambio, un paquete que entra en/sale de un puerto virtual interior permanece en la red. De esta manera, un puerto virtual puede caracterizarse como exterior o interior dependiendo de si el paquete entra a/sale de la red virtual cuando pasa a través del puerto.

En una realización, el motor de decisión opera localmente en un primer nodo y se comunica con la base de datos compartida que contiene la topología de red virtual y las configuraciones de dispositivo. La base de datos compartida puede contener una copia maestra o autoritativa de la topología e información de configuración de dispositivo. Para mejorar la eficiencia, al menos una parte de la topología de red virtual y la información de configuración de dispositivo virtual pueden copiarse en memoria caché localmente en nodos individuales. Los datos copiados en memoria caché pueden actualizarse cuando se modifica la base de datos compartida. En una realización, sólo aquellas partes de la topología o configuración de dispositivo usadas por un nodo dado se copian en memoria caché en el nodo. Tras la simulación de una llegada de un paquete a un dispositivo virtual, el sistema puede cargar la configuración del dispositivo virtual desde la base de datos compartida hasta el nodo que realiza la simulación y puede copiar en memoria caché la configuración de dispositivo para un futuro uso en el nodo.

Realizaciones del método informático incluyen también mapear la primera interfaz de red del primer nodo a un puerto virtual correspondiente y recuperar la configuración del puerto virtual y el dispositivo asociado con el puerto virtual desde la base de datos compartida. La acción para procesar el paquete de red se determina entonces basándose en una simulación del dispositivo asociado con el puerto virtual. La acción determinada puede incluir uno o más de modificar un estado interno de un dispositivo de red, soltar el paquete, modificar las cabeceras de protocolo del paquete, emitir el paquete desde uno o más puertos virtuales de un dispositivo de red, y emitir un paquete diferente desde uno o más puertos virtuales del dispositivo de red. En realizaciones, emitir el paquete desde uno o más puertos virtuales del dispositivo de red puede incluir emitir el paquete desde un puerto orientado hacia el exterior o un puerto orientado hacia el interior.

Al determinar cómo procesar un paquete, el motor de decisión puede recorrer múltiples dispositivos virtuales conectados por puertos virtuales orientados hacia el interior. En una realización, el motor de decisión determina un puerto orientado hacia el interior homólogo para un segundo puerto virtual y recupera la configuración del puerto orientado hacia el interior homólogo y el dispositivo de red en el que se ubica el puerto orientado hacia el interior homólogo. Entonces, el motor de decisión puede simular el funcionamiento del dispositivo de red asociado con el puerto orientado hacia el interior homólogo para determinar cómo ha de procesarse el paquete. De esta manera, el motor de decisión puede simular una ruta a través de la topología de red virtual incluyendo cualquier número de dispositivos de red virtuales con el fin de determinar cómo ha de procesarse un paquete o flujo dado.

Si la acción determinada es emitir un paquete desde uno o más puertos virtuales orientados hacia el exterior, el sistema mapea cada puerto virtual orientado hacia el exterior a una interfaz de red correspondiente y un nodo de la red subyacente y luego emitir el paquete desde cada una de las interfaces de red correspondientes.

Los procesos de simulación descritos en el presente documento se repiten hasta que el motor de decisión ha simulado el último dispositivo virtual recorrido por el paquete. El motor de decisión proporciona una acción o resultado de simulación que va a aplicarse al paquete y a paquetes posteriores que coinciden con un patrón de cabecera de protocolo de paquete. La acción o resultado de simulación incluye una modificación agregada del paquete para modificar la cabecera de protocolo del paquete para que coincida la configuración de las cabeceras ya que el paquete se emitirá por el último dispositivo virtual, basándose en todas las modificaciones aplicadas a lo largo del recorrido de la topología de red virtual. De esta manera, el motor de decisión determina a través de la simulación la modificación necesaria al paquete de modo que el paquete puede modificarse y enrutarse de manera eficiente a través de la red.

Tal como se comentó anteriormente, el paquete incluye una cabecera de protocolo que tiene una pluralidad de campos. El sistema determina un patrón de cabecera de protocolo de paquete usado para identificar paquetes para los que una regla de flujo o acción determinada se aplicará basándose en el resultado de simulación. En una realización, el sistema determina el patrón de cabecera de protocolo de paquete identificando cada uno del campo

de la cabecera de protocolo que se leyó durante la simulación de la topología de red virtual y los dispositivos de red virtuales. De esta manera, los campos de la cabecera de protocolo de los que se depende para atravesar la red se identifican de modo que la regla de flujo determinada puede aplicarse a paquetes que deben procesarse de la misma manera. Aquellos campos de la cabecera de protocolo de los que no se depende pueden tratarse como comodines o, de otro modo, excluirse de la consideración en el proceso de hacer coincidir la cabecera de protocolo de paquetes posteriores con el patrón determinado. El patrón de cabecera de protocolo de paquete y el resultado de simulación correspondiente pueden almacenarse en el nodo. En una realización, el patrón y el resultado de simulación correspondiente se almacenan como regla de flujo para su uso en un conmutador configurable de flujo configurado para procesar paquetes posteriores que llegan al nodo. Cuando un paquete llega a un nodo para el que no se ha creado una regla de flujo, el sistema puede invocar el motor de decisión para realizar una simulación para el paquete.

El resultado de simulación producido por el motor de decisión es dependiente de la topología de red virtual y las configuraciones de dispositivo virtual, al menos para aquellos dispositivos virtuales recorridos durante la simulación. Cuando la topología o configuraciones de dispositivo cambian, el resultado de simulación determinado anteriormente y las acciones correspondientes que van a aplicarse a un paquete pueden no ser ya correctos. Para dar cabida a tales cambios, el sistema está configurado para invalidar un patrón de cabecera de protocolo de paquete almacenado y el resultado de simulación almacenado correspondiente tras un cambio en la topología de red virtual o configuraciones de dispositivo virtual. En una realización, el sistema invalida todos los patrones almacenados y los resultados de simulación tras detectar un cambio. En otras realizaciones, sólo se invalidan aquellos resultados almacenados que dependen del dispositivo virtual cambiado. En un ejemplo, el conjunto recorrido de dispositivos virtuales recorridos durante la simulación se determina durante la simulación por el motor de decisión. El conjunto recorrido de dispositivos virtuales se asocia entonces con la cabecera de protocolo de paquete y/o el resultado de simulación. Cuando se detecta un cambio en la configuración de un dispositivo virtual, pueden invalidarse los resultados de simulación almacenados asociados con cualquier conjunto recorrido que contiene el dispositivo virtual cambiado. De esta manera, el sistema determinó de manera eficiente qué reglas de flujo deben invalidarse basándose en un cambio de dispositivo virtual dado. Un método de determinar un conjunto recorrido de dispositivos virtuales y unos flujos de invalidación basándose en un cambio en una configuración de dispositivo virtual se ilustran adicionalmente en la figura 1. En otras realizaciones, pueden invalidarse o expulsarse flujos cuando la memoria caché es un recurso de espacio limitado. Por ejemplo, el sistema puede seguir localmente (en el primer nodo) todos los motores de decisión que están copiados en memoria caché por el sistema subyacente, seguir una vez de “última coincidencia” de una decisión que es la última vez que un paquete coincidió con un patrón de decisión y que las acciones de decisión se aplicaron al paquete. Entonces, el sistema puede consultar la vez de “última coincidencia” de todas las decisiones y expulsar aquellas decisiones que no se han usado en el tiempo más largo. La consulta de la vez de última coincidencia puede realizarse a una frecuencia especificada o puede realizarse según sea necesario para mantener el tamaño de la memoria caché de decisiones almacenadas por debajo de un tamaño especificado. El sistema también puede eliminar decisiones aleatorias que se crearon “recientemente”. La eliminación de decisiones aleatorias creadas recientemente puede ser eficiente cuando una mayoría de decisiones recientes son para flujos de paquete de corta duración (en comparación con decisiones supervivientes más antiguas que tienen un porcentaje comparativamente más alto de flujos de larga duración). Los procesos para invalidar flujos pueden usarse individualmente o en combinación para gestionar la memoria caché de datos almacenados en el nodo dentro de parámetros deseados. El sistema puede ajustar también la tasa de invalidaciones o expulsiones basándose en la tasa de nuevas invocaciones del motor de decisión, que se correlacionan con la adición de nuevas decisiones a la memoria caché almacenada.

El sistema también está configurado para hacer converger con tráfico correcto, de manera eficiente y con alteración mínima, las decisiones copiadas en memoria caché para que concuerden con respecto a configuraciones de dispositivo virtual actualizadas. La convergencia de decisiones copiadas en memoria caché, que puede caracterizarse como invalidaciones o expulsiones basadas en exactitud de flujos anteriormente almacenados. Tal como se usa en el presente documento, una decisión que concuerda, con respecto a una configuración de red virtual inicial más algún cambio, es una a la que puede volver a llegarse mediante una nueva invocación del motor de decisión con el mismo paquete de entrada e interfaz de entrada. En cambio, una decisión que no concuerda es una decisión que no realizará el motor de decisión dadas las mismas entradas debido a la nueva configuración de los dispositivos de red virtuales. En una realización, para un tiempo T hay un periodo P limitado dentro del cual todas las decisiones que se copiaron en memoria caché antes del tiempo T concuerdan con el estado de la configuración de red virtual en o después del tiempo T. Para hacer converger las decisiones, el sistema indexa las decisiones copiadas en memoria caché de manera local mediante los dispositivos que se simularon para esa decisión (estos representan los dispositivos virtuales que recorrió el paquete) y el tiempo en el que se realizó/almacenó en memoria caché la decisión. El sistema recibe entonces actualizaciones locales de la configuración de dispositivo virtual para una actualización de configuración de un primer dispositivo virtual recibida en el tiempo T, espera un tiempo especificado de modo que el número de decisiones realizadas y almacenadas en memoria caché antes del tiempo T ya se ha reducido mediante expulsiones basadas en espacio, y luego interseca el conjunto de decisiones realizadas/almacenadas en memoria caché antes del tiempo T con el conjunto de decisiones que requirieron simular el primer dispositivo virtual. Las decisiones en el conjunto resultante deben validarse entonces volviendo a invocar el motor de decisión con las mismas entradas (y la configuración actual). Para cualquier decisión que haya cambiado, se invalida la decisión antigua y se instala/almacena en memoria caché el nuevo resultado de simulación basado en

la configuración actual actualizada para su uso con paquetes posteriores que coinciden con el flujo.

En otro aspecto, el método y sistema informático dados a conocer en el presente documento incluyen simular uno o más puentes de aprendizaje de MAC, en los que los puertos exteriores de cada puente se mapean a interfaces de uno o más nodos de la red subyacente y en los que cada puerto orientado hacia el interior del puente está conectado a un puerto orientado hacia el interior de un enrutador virtual. El método informático incluye mantener una copia autoritativa de la tabla de aprendizaje de MAC del puente en la base de datos compartida. La tabla de aprendizaje de MAC también puede conocerse como base de datos de filtrado dinámico, mapa de direcciones MAC al puerto a través del cual puede alcanzarse la MAC, en la que una MAC sólo puede alcanzarse a través de uno de los puertos de un puente en un momento cualquiera en una red correctamente configurada. El método incluye además mantener una copia en memoria caché de la tabla de aprendizaje de MAC del puente en cada nodo que tiene una interfaz que mapea a uno de los puertos orientados hacia el exterior del puente, y en cada nodo que simuló un recorrido del paquete de ese puente. La copia en memoria caché de la tabla de aprendizaje de MAC puede actualizarse cuando cambia la copia autoritativa.

En una realización, la invocación del motor de decisión da como resultado la simulación de una trama de Ethernet que lleva a un primer puerto de un puente de Ethernet, y el sistema carga el estado del puente de Ethernet si ninguna invocación previa del motor de decisión lo ha cargado. Una trama de Ethernet entrante puede tener una dirección MAC de destino de unidifusión. En realizaciones, un método incluye además detectar que la MAC de destino es una dirección de unidifusión, y determinar si hay una entrada para esa MAC en la tabla de aprendizaje de MAC. Si la tabla de aprendizaje de MAC incluye una entrada que mapea la MAC a un segundo puerto del puente, el sistema determina que el puente simulado emitirá tal trama a partir del segundo puerto. Si la tabla de aprendizaje de MAC no incluye una entrada para esa MAC, el sistema determina que el puente simulado emitirá la trama a partir de todos sus puertos, excepto aquél en el que llegó.

En otra realización, la trama de Ethernet entrante tiene una MAC de multidifusión o radiodifusión que indica que la trama debe emitirse desde múltiples puertos. Realizaciones del método informático pueden incluir además detectar que la MAC de destino es una dirección de multidifusión o radiodifusión y determinar que el puente simulado emitirá tal trama a partir de todos sus puertos excepto aquél en el que llegó. En aún otra realización, la trama de Ethernet entrante tiene una dirección MAC de origen de unidifusión. Si la tabla de aprendizaje de MAC no tiene ninguna entrada para esta MAC, el sistema añade una entrada que mapea esta MAC al puerto de llegada. Entonces, el sistema inicia un recuento de referencia de tal entrada de tabla de aprendizaje de MAC, local para el nodo en el que se produjo la invocación, en el que el recuento se basa en el número de decisiones copiadas en memoria caché que dieron como resultado una trama con la misma dirección MAC de origen que llega al mismo puerto. El recuento de referencia de tales decisiones copiadas en memoria caché puede resultar útil puesto que el motor de decisión no ve todos los paquetes con la misma MAC de origen que llegan al puerto. Por tanto, cuando el número de tales decisiones copiadas en memoria caché alcanza cero, puede hacerse que caduque (o establecerse para que caduque) la entrada de tabla de aprendizaje de MAC para esta MAC de origen y puerto de llegada. En cada nodo que tiene una copia en memoria caché de la tabla de aprendizaje de MAC del puente (puesto que tiene una interfaz mapeada a un puerto orientado hacia el exterior del puente, o puesto que recientemente simuló el puente) el sistema aprende la actualización de la tabla y expulsa cualquier decisión copiada en memoria caché que se basó en la ausencia de una entrada para esa MAC en la tabla de aprendizaje de MAC, puesto que esos flujos/paquetes ahora pueden suministrarse al puerto de entrada, en lugar de inundarse en todos los puertos del puente.

En otras realizaciones, la tabla de aprendizaje de MAC puede tener ya una entrada para la dirección MAC y el puerto mapeado es el mismo que el puerto de llegada de la trama entrante actual. Entonces, el sistema puede detectar que existe la entrada de tabla de aprendizaje de MAC y no necesita modificarse. El sistema puede incrementar el recuento de referencia local si esta invocación del motor de decisión da como resultado una decisión copiada en memoria caché. Alternativamente, la tabla de aprendizaje de MAC puede tener ya una entrada para la dirección MAC, pero el puerto mapeado puede ser diferente del puerto de llegada de la trama entrante actual. Entonces, el sistema retira la entrada anterior de la tabla de aprendizaje de MAC, y añade una nueva entrada a la tabla de aprendizaje de MAC que asocia la dirección MAC con el puerto de llegada de la trama. En el nodo que es propietario de la interfaz que corresponde al puerto mapeado de la entrada anterior, el sistema toma conocimiento de la retirada de la entrada y expulsa cualquier decisión que realiza el recuento de referencia de esa entrada dado que ahora se basan en información incorrecta. En cada nodo que tiene una copia en memoria caché de la tabla de aprendizaje de MAC del puente (puesto que tiene una interfaz que corresponde a un puerto orientado hacia el exterior del puente, o puesto que recientemente simuló el puente) aprende las actualizaciones de la tabla y expulsa cualquier decisión copiada en memoria caché que se basara en la entrada de tabla de aprendizaje de MAC anterior para esa MAC dado que ahora se basan en información incorrecta. Para ilustrar adicionalmente, en la figura 2 se ilustra un método para aprendizaje de MAC y en la figura 3 se ilustra un método para desaprendizaje de MAC.

En otra realización, se proporciona un método para reducir paquetes inundados en puentes de Ethernet cuando se conocen por adelantado las direcciones MAC que pueden alcanzarse desde cada uno de los puertos de un puente. Por ejemplo, en el caso de una VM invitada adherida a una de las interfaces de red de un nodo que se mapea a uno de los puertos orientados hacia el exterior de un puente virtual, las direcciones MAC pueden conocerse por adelantado de manera que la tabla de aprendizaje de MAC puede poblarse previamente con las entradas de puerto de MAC conocidas.

En otro aspecto, el método informático reduce los paquetes inundados en una red IP interceptando y respondiendo a peticiones de ARP. El método incluye aumentar el estado de un puente con una memoria caché de ARP almacenada en la base de datos compartida. La memoria caché de ARP incluye un mapa de dirección IP de unidifusión a dirección MAC de unidifusión. Como con la tabla de aprendizaje de MAC, la memoria caché de ARP del puente puede poblarse previamente con entradas correspondientes a cualquier enrutador que está conectado al puente a través de puertos orientados hacia el interior. Cada entrada puede determinarse examinando uno de los puertos orientados hacia el interior del puente, recuperando la configuración del puerto homólogo, y extrayendo la dirección IP y MAC del puerto homólogo. El método también puede incluir poblar previamente la memoria caché de ARP del puente con cualquier otra entrada que se conoce por adelantado, tal como en un entorno de gestión en la nube en el que a máquinas virtuales invitadas se les asignan direcciones IP y MAC por usuarios o de manera automática por el sistema. El método también puede incluir reconocer paquetes de IP y extraer la dirección IP de origen y la dirección MAC de origen de la trama de Ethernet de encapsulamiento y por tanto deducir la correspondencia IP-MAC y añadir la entrada apropiada a la memoria caché de ARP.

En aún otra realización, un método para simular uno o más enrutadores IPv4 en el que los puertos orientados hacia el exterior de cada enrutador se mapean a interfaces de uno o más nodos de la red subyacente, y cada puerto orientado hacia el interior de un enrutador está conectado a un puerto orientado hacia el interior o bien de otro enrutador virtual o bien de un puente virtual. El método incluye mantener una copia autoritativa de la memoria caché de ARP del enrutador (mapa de dirección IP de unidifusión a dirección MAC de unidifusión) en la base de datos compartida, cargar previamente la memoria caché de ARP con los pares de direcciones (IPv4, MAC) de los puertos homólogos de todos los puertos orientados hacia el interior del enrutador, mantener una copia autoritativa de la tabla de reenvío del enrutador (conjunto de reglas/rutas que determinan qué puerto de enrutador debe emitir un paquete basándose en escoger la regla con el prefijo de destino de IPv4 coincidente más preciso y un prefijo de origen coincidente) en la base de datos compartida, mantener una copia en memoria caché de la memoria caché de ARP del enrutador y tabla de reenvío en cada nodo que tiene una interfaz que mapea a uno de los puertos orientados hacia el exterior del enrutador, y en cada nodo que recientemente simuló un recorrido del paquete de ese enrutador. Un enrutador se ha simulado recientemente en un nodo si ese nodo tiene al menos una decisión de motor copiada en memoria caché que requiere simular el enrutador. El método también puede incluir actualizar la copia en memoria caché de la memoria caché de ARP y la tabla de reenvío cuando cambia la copia autoritativa en la base de datos compartida. Tras la simulación de la llegada de un paquete en un enrutador de IPv4, el estado del enrutador puede cargarse en este nodo si no se ha cargado ya ninguna invocación de motor de decisión anterior.

El motor de decisión también puede simular un paquete de IPv4 que llega a un primer puerto de un enrutador de IPv4. La dirección de destino del paquete de IPv4 entrante puede ser igual a uno de los puertos del enrutador. Entonces, el sistema detectará que el paquete va dirigido a uno de los puertos del enrutador, determinará que el enrutador soltará el paquete si no se reconoce o no se gestiona su protocolo, detectará si el paquete es una petición de ping (eco de ICMP) y en ese caso generará un paquete de respuesta de ping desde el puerto del enrutador hasta la dirección de origen de IPv4 del primer paquete, e invocará la lógica de simulación del motor de decisión para determinar la trayectoria que seguirá la respuesta de ping desde el puerto a través de la red virtual. Si la lógica de simulación determina que la respuesta de ping saldrá de la red virtual en un puerto orientado hacia el exterior específico (de cualquier dispositivo), entonces el sistema mapeará ese puerto a su interfaz correspondiente y nodo de red subyacente y pedirá que el sistema de llamada emita el paquete de respuesta de ping desde esa interfaz. Alternativamente, la dirección de destino del paquete de IPv4 entrante puede no ser una de las direcciones de los puertos del enrutador, en cuyo caso el sistema consulta la tabla de reenvío del enrutador para determinar la ruta más coincidente dadas las direcciones de IPv4 de origen y destino del paquete. Cuando no se encuentra ninguna ruta coincidente, el sistema determina que tras recibir el primer paquete el enrutador en cuestión soltará el paquete y responderá con un error de ICMP, tal como ruta no alcanzable, e invoca la lógica de simulación del motor de decisión para determinar la trayectoria que seguirá el error de ICMP desde el puerto hasta la red virtual. Si la lógica de simulación determina que el error de ICMP saldrá de la red virtual en un puerto orientado hacia el exterior específico, entonces el sistema mapea ese puerto a su interfaz correspondiente y nodo de red subyacente y pide que el sistema de llamada emita el paquete de error de ICMP desde esa interfaz. De una manera similar, puede generarse un paquete de error de ICMP cuando la ruta coincidente especifica que el destino está prohibido de manera administrativa.

Cuando la simulación determina que la ruta más coincidente especifica que el paquete se reenvíe a través de puerto de enrutador (por ejemplo, puerto de siguiente salto), el sistema puede cambiar la dirección Ethernet de origen del paquete por la MAC del puerto de siguiente salto. Si la pasarela de siguiente salto de la ruta es nula (lo que significa que la dirección de destino está en la misma subred L3 que el puerto de siguiente salto), el sistema consulta la memoria caché de ARP local para determinar la MAC correspondiente al destino de IPv4 del paquete y cambia la dirección Ethernet de destino del paquete por la MAC. Si la pasarela de siguiente salto de la ruta no es nula (lo que significa que el paquete debe reenviarse para pasar a través de al menos un enrutador más antes de alcanzar su destino), el sistema consulta la memoria caché de ARP local para determinar la MAC correspondiente a la dirección IPv4 de la pasarela y cambia la dirección Ethernet de destino del paquete por esa MAC. El sistema puede determinar además que tras recibir el primer paquete (que puede haberse modificado tras la simulación de dispositivos virtuales anteriormente recorridos), el enrutador en cuestión modificará adicionalmente el paquete tal como se describe y lo emitirá a partir del puerto de siguiente salto.

Cuando la memoria caché de ARP no contiene una entrada para la dirección IPv4 consultada (tal como cuando el puerto de siguiente salto es un puerto orientado hacia el exterior), el sistema puede implementar un método que incluye generar un paquete de petición de ARP para la dirección IPv4 deseada, y añadir un par (IPv4, MAC nula), indicado con un momento de último envío establecido al momento actual, a la base de datos compartida para indicar cuándo se envió la última petición de ARP para esa IPv4. El método puede incluir además mapear el puerto de siguiente salto orientado hacia el exterior a su interfaz correspondiente y nodo de red subyacente, pedir que el sistema de llamada emita una petición de ARP desde esa interfaz, y repetir periódicamente la petición de que se emita el paquete de petición de ARP desde esa interfaz. En realizaciones, la petición de ARP se suministrará por el sistema de base como cualquier otro paquete que entró en la red virtual, y por tanto posiblemente a lo largo de un túnel hasta un nodo diferente. El método puede continuar hasta que se agota un tiempo asignado, y luego generar un mensaje de error de ruta ICMP inalcanzable en respuesta al primer paquete. Alternativamente, el método continúa hasta que se recibe una actualización para la copia local de la memoria caché de ARP que incluye una entrada (IPv4, MAC) para la dirección IPv4 deseada, y luego incluye cambiar la dirección Ethernet de destino del paquete por la MAC de esa entrada, y determinar que el enrutador simulado modificará el paquete tal como se describe y lo emitirá desde el puerto de siguiente salto. En realizaciones, la respuesta de ARP se recibirá en un nodo diferente del que procesa el primer paquete si la petición de ARP se emitió a partir de un puerto que se mapea a una interfaz en un nodo diferente. De esta manera, el motor de decisión puede aprender la entrada de ARP a través de la memoria caché de ARP en vez de recibiendo directamente la respuesta de ARP. Cuando se agota un tiempo asignado y no se encuentra la entrada de memoria caché de ARP, el sistema puede responder con un error de ruta de ICMP inalcanzable tal como se describió anteriormente.

En otra realización, una invocación del motor de decisión da como resultado la simulación de un paquete de petición de ARP que llega a un primer puerto de un enrutador IPv4 y en la que la dirección de protocolo objetivo del ARP ("TPA") es la dirección IPv4 del puerto de llegada/primer puerto. Entonces, el sistema puede generar un paquete de respuesta de ARP con una dirección de hardware de origen ("SHA") establecida a la dirección MAC del puerto de llegada, invocar la lógica de simulación del motor de decisión para determinar la trayectoria que seguirá la respuesta de ARP desde el puerto de llegada del primer paquete a través de la red virtual. La respuesta de ARP puede emitirse de una manera similar a las respuestas anteriormente comentadas. Alternativamente, la simulación de motor de decisión puede determinar que el enrutador simulado soltará el paquete de petición de ARP.

En otra realización, una invocación del motor de decisión da como resultado la simulación de un paquete de respuesta de ARP que llega a un primer puerto de un enrutador IPv4. El sistema puede detectar si la respuesta de ARP es en respuesta a una petición de ARP que se generó por el enrutador. En una realización, el sistema comprueba que hay una entrada (IPv4, MAC) en la memoria caché de ARP, indicada con un momento de último envío reciente, aunque la propia MAC sea nula. Si no hay ninguna entrada, el sistema determina que el enrutador soltará una petición de ARP no solicitada de este tipo con el fin de bloquear ataques de denegación de servicio. Alternativamente, el sistema extrae la dirección de hardware de origen, una dirección MAC, y la dirección de protocolo de origen, una dirección IPv4, a partir de la respuesta de ARP y actualiza la memoria caché de ARP. La memoria caché de ARP puede actualizarse localmente y en la base de datos compartida con la entrada (IPv4, MAC).

En otro aspecto, el sistema informático dado a conocer en el presente documento está configurado para realizar un método que incluye simular filtros de entrada y salida de un dispositivo virtual, en el que los filtros incluyen reglas de filtrado individuales que están organizadas en listas que pueden hacer referencia unas a otras mediante reglas de salto. El método también puede incluir especificar una condición que puede leer y aplicar lógica a cualquiera de o todos los campos de las cabeceras de protocolo de red L2-L4 de un paquete, y especificar una acción que debe ejecutarse (por ejemplo soltar, aceptar para procesamiento adicional) cuando un paquete coincide con la condición.

En una realización, el método comprende además mantener las reglas de filtrado para el filtro de entrada/salida de cada dispositivo en la base de datos compartida, mantener una copia local de las reglas de filtrado de un dispositivo en cualquier nodo que ha simulado recientemente el dispositivo, actualizar la copia local de las reglas de filtrado de un dispositivo cuando la copia autoritativa de las reglas cambia en la base de datos compartida, y/o volver a validar decisiones de reenvío de flujo copiadas en memoria caché de manera local que requirieron simulación de un dispositivo, cuando se modifican los filtros de ese dispositivo. El método también puede incluir simular reglas de filtrado que coinciden en cuanto al estado de conexión según el flujo, en el que el estado de conexión según el flujo se sigue independientemente por cada dispositivo simulado, y en el que el conjunto de valores de estado de conexión depende del protocolo de transporte (L4) del paquete. En una realización, el sistema está configurado para realizar un método que incluye tener espacio dedicado en la base de datos central para almacenar el estado de conexión según el dispositivo y el flujo, tras comenzar la simulación de un dispositivo, consultar la base de datos central usando la firma de flujo del paquete para recuperar el estado de conexión. En una realización, la firma de flujo se calcula adjuntando estos campos en este orden: el ID de dispositivo del dispositivo simulado, el campo de origen (por ejemplo, IP) de la cabecera L3 del paquete, el campo de destino L3, el tipo de protocolo L4 (por ejemplo TCP), el campo de origen de la cabecera L4, el campo de destino de cabecera L4. Si no se encuentra ningún estado de conexión en la base de datos central, entonces el paquete constituye un nuevo flujo cuyo estado de conexión es de manera implícita el valor "inicio" del conjunto de estados para el protocolo de red de este paquete. El método también puede exponer el valor de estado de conexión para su coincidencia mediante las reglas de filtrado de este dispositivo. Antes de terminar la simulación del dispositivo, y si la simulación determina que el dispositivo reenviará

el paquete, se establece el estado de conexión para la firma de flujo de este paquete y la firma de flujo de retorno de este paquete según las reglas de transición para el conjunto de valores de estado de conexión para el protocolo de red de este paquete. En una realización similar a la anterior, la firma de flujo de retorno del paquete se calculará adjuntando estos valores en este orden: el ID de dispositivo del dispositivo simulado, el campo de destino de la cabecera L3 del paquete, campo de origen L3, tipo de protocolo L4, campo de destino de cabecera L4, y campo de origen de cabecera L4. La firma de flujo directo y la firma de flujo de retorno también pueden definirse usando campos adicionales que pueden ser útiles en una aplicación dada. Cuando la decisión copiada en memoria caché para el flujo directo caduca, el sistema puede planificar la retirada tanto del estado de conexión asociado con ese flujo como del flujo de retorno. Para ilustrar adicionalmente, en la figura 4 se ilustra una realización de un método para seguir conexiones.

En una realización, el método incluye además simular reglas de filtrado que coinciden con el estado de conexión del flujo del paquete, en el que el estado de conexión según el flujo se comparte entre todos los dispositivos simulados, y en el que el conjunto de estados de conexión depende del protocolo de transporte (L4) del paquete. De esta manera, el estado de conexión se considera una propiedad del flujo, independiente de la trayectoria seguida a través de la red de dispositivos. El resultado es que todos los dispositivos simulados en una única llamada al motor de decisión estarán de acuerdo en cuanto a los estados de conexión del flujo de retorno y del flujo del paquete; y dos dispositivos cualesquiera simulados en dos llamadas diferentes al motor de decisión estarán de acuerdo en cuanto a esos estados de conexión si al menos uno de lo siguiente es cierto: el flujo de retorno entra en la red virtual en el mismo dispositivo desde el que se emite el flujo directo; los paquetes de flujo de retorno tienen direcciones L3 públicas, es decir globalmente únicas. En realizaciones, la base de datos compartida incluye espacio dedicado para almacenar el estado de conexión según el flujo. Tras comenzarse la simulación del recorrido del paquete de la red virtual, puede consultarse la base de datos compartida usando la firma de flujo del paquete. La firma de flujo puede depender del ID de dispositivo del primer dispositivo simulado si al menos uno del origen/destino L3 no es público, es decir globalmente único. La firma de flujo también puede depender del campo de origen (por ejemplo, IP) de la cabecera L3 del paquete, campo de destino L3, tipo de protocolo L4 (por ejemplo TCP), campo de origen de la cabecera L4, y campo de destino de cabecera L4. Si no se encuentra ningún estado de conexión en la base de datos central, entonces el paquete constituye un nuevo flujo cuyo estado de conexión es de manera implícita el valor de inicio del conjunto de estados para este el protocolo de red de este paquete. Entonces puede exponerse el valor de estado de conexión para su coincidencia por las reglas de filtrado de cualquier dispositivo simulado. Antes de terminar la simulación del recorrido del paquete de la red virtual, y si la simulación determina que el paquete se emitirá finalmente desde algún puerto virtual, se establece el estado de conexión para la firma de flujo de este paquete y la firma de flujo de retorno de este paquete según las reglas de transición para el conjunto de valores de estado de conexión para el protocolo de red de este paquete. El estado de conexión se escribe en la base de datos compartida antes de tunelar/reenviar el paquete (y por tanto antes de devolver la decisión) con el fin de evitar una condición de carrera en la que se simula un paquete procedente del flujo de retorno y desencadena una consulta del estado de conexión antes de haberse completado la escritura del estado de conexión en la base de datos compartida. La firma de flujo de retorno del paquete puede calcularse de una manera similar. Tal como se indicó anteriormente, cuando caduca la decisión copiada en memoria caché para el flujo directo, se planifica la retirada tanto del estado de conexión asociado con ese flujo como del flujo de retorno.

En otro aspecto, el método incluye reducir el tiempo de simulación evitando consultar o escribir un estado de conexión cuando este estado no se usará por la simulación del paquete o por la simulación de un paquete de retorno, retrasar la consulta del estado de conexión para el flujo del paquete hasta que una regla de filtrado en algún dispositivo simulado necesite leer tal estado, determinar si la trayectoria probable para el paquete de retorno incluirá simular una regla de filtrado que necesite leer el estado de conexión del flujo de retorno, y en caso negativo omitir la escritura del estado de conexión de flujo tanto directo como de retorno en la base de datos compartida. En realizaciones, el estado de conexión se mantiene en la base de datos compartida de modo que si cualquier paquete del mismo flujo llega a una interfaz de un segundo nodo en un momento posterior, el motor de decisión del segundo nodo llegará a la misma decisión sobre cómo tratar ese flujo (en ausencia de cambios en la configuración de red virtual). Esto es necesario para conservar la integridad de un flujo cuando sus paquetes llegan a las interfaces de más de un nodo de la red subyacente debido a decisiones de enrutamiento externas. Esas decisiones de enrutamiento pueden estar relacionadas o no con la inalcanzabilidad percibida o real de la interfaz de un nodo.

En todavía otras realizaciones, el sistema está configurado para realizar una simulación de traducción de dirección de red (es decir L3) y de transporte (es decir L4) para un dispositivo virtual. De manera similar el sistema está configurado para simular traducciones inversas de las direcciones de red y protocolo de un dispositivo virtual. Estos procesos pueden denominase de manera colectiva “NAT”. En diversas realizaciones, las reglas de NAT individuales pueden seguir o preceder o estar intercaladas con reglas de filtrado, especificar una condición que puede leerse y aplicar lógica a cualquiera de o todos campos de las cabeceras de protocolo de red L2-L4 de un paquete, especificar cómo deben traducirse o traducirse de manera inversa los campos L3 y L4 cuando un paquete coincide con la condición, y/o especificar una acción que va a ejecutarse cuando se ha producido la traducción (por ejemplo aceptar para procesamiento adicional por parte del dispositivo, continuar procesando en el conjunto de reglas). El método puede incluir además mantener las reglas de traducción para el filtro de entrada/salida de cada dispositivo en la base de datos compartida, mantener una copia local de las reglas de traducción de un dispositivo en cualquier nodo que ha simulado recientemente el dispositivo, y actualizar la copia local de las reglas de traducción de un dispositivo

cuando la copia autoritativa de las reglas cambia en la base de datos compartida. Durante la simulación del dispositivo, si el procesamiento de paquete alcanza la regla de NAT, el método incluye determinar si el paquete (posiblemente ya modificado mediante dispositivos o reglas anteriores) satisface la condición de la regla y, en caso afirmativo, modificar el paquete según la traducción o traducción inversa especificada por la regla. Decisiones de reenvío de flujo copiadas en memoria caché de manera local que requieren simulación de un dispositivo pueden entonces validarse de nuevo cuando se modifican las traducciones de ese dispositivo.

En otro aspecto, el sistema implementa un dispositivo virtual físicamente distribuido que soporta NAT de destino con estado, en el que algunas reglas de NAT permiten una elección de objetivo de traducción para direcciones de destino L3 y L4 y especifican una política para realizar la elección entre objetivos de traducción. En realizaciones, el sistema puede almacenar la elección de traducción para cada flujo directo en la base de datos compartida, usando como claves las firmas de flujo tanto directo como de retorno. La firma de flujo directo puede estar compuesta por estos valores en este orden: el ID del dispositivo virtual, la dirección de origen L3 del paquete, dirección de destino L3, número de protocolo L4, dirección de origen L4 y dirección de destino L4. La firma de flujo de retorno puede estar compuesta por estos valores en este orden: el ID del dispositivo virtual, la dirección L3 elegida por la traducción, la dirección de origen L3 del paquete, número de protocolo L4, la dirección L4 elegida por la traducción, la dirección de origen L4 del paquete. La traducción almacenada puede codificar las direcciones de destino L3 y L4 originales del paquete así como las direcciones de destino L3 y L4 elegidas para la traducción. El método incluye además durante la simulación de un dispositivo, si el procesamiento de paquete alcanza una regla de NAT de este tipo (que permite una elección de direcciones de destino) y satisface la condición de la regla, componer una clave tal como se describió anteriormente para la firma de flujo directo y consultar la base de datos compartida para determinar si ya se ha almacenado una traducción (y por tanto ya se ha realizado la elección de direcciones traducidas) mediante una ejecución de motor de decisión anterior (en el nodo de red subyacente local o alguno remoto). Si se encuentra una traducción almacenada de este tipo en la base de datos compartida, entonces se modifican las direcciones de destino L3 y L4 del paquete para dar las direcciones L3 y L4 elegidas, y luego se continúa la simulación. Si no se encuentra una traducción almacenada de este tipo en la base de datos compartida, entonces se realiza una elección según la política especificada, se modifican las direcciones de destino L3 y L4 del paquete según esa elección, se almacena la elección de traducción en la base de datos compartida tal como se describió anteriormente, y luego se continúa la simulación. Durante la simulación de un dispositivo, si el procesamiento de paquete alcanza una regla de traducción inversa que especifica invertir una elección, y el paquete satisface la condición de la regla, entonces se supone que el paquete es un paquete de retorno de un flujo directo traducido, se compone la clave que corresponde a la firma de flujo de retorno, y se consulta la base de datos compartida para determinar si se ha almacenado una traducción para ese flujo de retorno. La firma de flujo de retorno puede estar compuesta por estos valores en este orden: el ID del dispositivo virtual, la dirección de origen L3 del paquete, dirección de destino L3, número de protocolo L4, dirección de destino L4 y dirección de origen L4. Si se encuentra una traducción almacenada de este tipo en la base de datos, entonces se aplica en sentido inverso a este paquete modificando las direcciones de origen L3 y L4 del paquete para dar las direcciones L3 y L4 originales de la traducción almacenada, y luego se continúa la simulación. Si no se encuentra una traducción almacenada de este tipo en la base de datos compartida, entonces la suposición de que el paquete es un paquete de retorno de un flujo directo traducido es incorrecta, por tanto no se necesita aplicar ninguna traducción inversa, y por tanto se continúa la simulación como si no se hubiera satisfecho la condición de la regla inversa. De esta manera, el sistema y método permiten almacenar las traducciones en una base de datos compartida y abordar condiciones de carrera de manera que el dispositivo virtual se comporta correctamente y no puede distinguirse de un dispositivo de hardware que funciona correctamente, pero con una disponibilidad aumentada del dispositivo virtual en comparación con el dispositivo de hardware.

En otro aspecto, algunas reglas de NAT permiten una elección de objetivos de traducción para las direcciones de origen L3 y L4 y especifican una política para realizar esa elección. En realizaciones, el sistema puede almacenar la elección de traducción para cada flujo directo en la base de datos compartida, usando como claves las firmas de flujo tanto directo como de retorno. La traducción almacenada codifica las direcciones de origen L3 y L4 originales del paquete así como las direcciones de origen L3 y L4 elegidas para la traducción. Durante la simulación de un dispositivo, si el procesamiento de paquete alcanza una regla de NAT de este tipo (que permite una elección de direcciones de origen) y satisface la condición de la regla, se compone una clave tal como se describió anteriormente para la firma de flujo directo y se consulta la base de datos compartida para determinar si ya se ha almacenado una traducción (y por tanto ya se ha realizado la elección de direcciones traducidas) mediante una ejecución de motor de decisión anterior (en el nodo de red subyacente local o alguno remoto). Si se encuentra una traducción almacenada de este tipo en la base de datos compartida, entonces las direcciones de origen L3 y L4 del paquete se modifican para dar las direcciones L3 y L4 elegidas, y luego se continúa la simulación. Si no se encuentra una traducción almacenada de este tipo en la base de datos compartida, entonces se realiza una elección según la política especificada, se construye la firma de flujo de retorno según esa elección y se consulta la base de datos para garantizar que no hay ninguna traducción almacenada mediante esa clave, se repite la elección y comprobación de base de datos hasta que la base de datos no devuelve ninguna coincidencia para la clave, después se modifican los campos de origen L3 y L4 del paquete según la elección final, se almacena la elección de traducción en la base de datos compartida tal como se describió anteriormente, y luego se continúa la simulación. La comprobación para detectar la clave de flujo de retorno en la base de datos puede usarse para determinar la exactitud y para evitar ambigüedad en el enrutamiento de flujos de retorno. Durante la simulación de un dispositivo,

si el procesamiento de paquete alcanza una regla de traducción inversa que especifica invertir una elección, y el paquete satisface la condición de la regla, entonces se supone que el paquete es un paquete de retorno de un flujo directo traducido, se compone una clave que corresponde a la firma de flujo de retorno, y se consulta la base de datos compartida para determinar si se ha almacenado una traducción para ese flujo de retorno. Si se encuentra una traducción almacenada de este tipo en la base de datos, entonces se aplica la traducción almacenada en sentido inverso a este paquete modificando las direcciones de destino L3 y L4 del paquete para dar las direcciones L3 y L4 originales de la traducción almacenada, y luego se continúa la simulación. Si no se encuentra una traducción almacenada en la base de datos compartida, entonces la suposición de que el paquete es un paquete de retorno de un flujo directo traducido es incorrecta, por tanto no se necesita aplicar ninguna traducción inversa, y por tanto la simulación puede continuarse como si no se hubiera satisfecho la condición de la regla inversa.

En aún otro aspecto, el número de intentos que intentos que seleccionan traducciones de dirección L3 y L4 que ya están en la base de datos puede reducirse segmentando los intervalos de direcciones L3 y L4 en bloques que pueden reservarse mediante nodos individuales. Cuando se eligen las direcciones L3 y L4 para la traducción, un nodo comprueba localmente si hay combinaciones de direcciones sin usar en su propio bloque, de lo contrario reserva un nuevo bloque. Con frecuencia, esto da como resultado una comunicación de ida y vuelta en la base de datos. Si el nodo no puede reservar un nuevo bloque y no tiene ninguna combinación de direcciones L3 y L4 sin usar disponible para una nueva traducción, entonces intenta usar una combinación de direcciones L3 y L4 aleatoria dentro de las limitaciones especificadas por la regla.

En realizaciones, los protocolos de enrutamiento funcionan a nivel global ya que están diseñados y estudiados en cuanto a sus efectos globales sobre el establecimiento y mantenimiento de conectividad y estabilidad de red. Sin embargo, cualquier enrutador individual sólo necesita mantener una discusión de protocolo de enrutador con sus homólogos inmediatos. Una organización puede hacer funcionar sesiones de protocolo de enrutamiento con sus redes vecinas por una variedad de motivos. Como ejemplos, puede aconsejar a vecinos sobre la mejor trayectoria para entrar en su red para bloques de direcciones específicos, y puede ajustar sus propias decisiones de reenvío basándose en condiciones de red externa. Estas sesiones de enrutamiento son inherentemente con estado, tanto puesto que la discusión puede realizarse a lo largo de una conexión (tal como TCP) en contraposición a un protocolo sin conexión (tal como UDP) como puesto que el objetivo es intercambiar el estado que luego usan los enrutadores para decidir si reenviar paquetes. En realizaciones, el sistema usa un modelo de aplicación para implementar protocolos de enrutamiento. En un modelo de aplicación, se proporciona aislamiento L2 para el conjunto de redes virtuales que se ejecutan en la capa base. En el caso de los protocolos de enrutamiento, el modelo de aplicación puede ser beneficioso puesto que el protocolo de enrutamiento es una parte específica de lógica y una en la que el propio dispositivo virtual es el origen y destino de tráfico. En una realización, en lugar de poner un enrutador L3 completo en una aplicación, sólo se pone en una aplicación el protocolo de enrutamiento entre un puerto virtual y algún homólogo externo. De esta manera, el sistema es más tolerante a fallos ya que aunque la aplicación puede ser un único punto de fallo, no importa, los protocolos de enrutamiento tienen esto en cuenta permitiendo múltiples sesiones entre homólogos a lo largo de múltiples puertos.

El sistema puede estar configurado además para soportar protocolos de enrutamiento (por ejemplo BGP, iBGP, OSPF) en enrutadores IPv4 virtuales implementando un método que incluye almacenar parámetros de configuración para un protocolo de enrutamiento deseado y sesión por homólogos como parte de la configuración del puerto virtual a través del cual el enrutador establecerá la sesión con su homólogo. Esta información puede almacenarse en la base de datos compartida. El método también puede incluir almacenar las rutas notificadas deseadas con la configuración de sesión de protocolo de enrutamiento en la base de datos compartida, cuando un nodo de red subyacente tiene una interfaz pública que se mapea a un puerto virtual con configuración para una sesión de protocolo de enrutamiento, el nodo lanza localmente un programa residente de protocolo de enrutamiento en un contenedor (por ejemplo en una VM). El contenedor obtiene una interfaz “privada” en el anfitrión, y los nodos establecen decisiones de reenvío que permiten que paquetes procedentes de esa sesión de protocolo de enrutamiento, y opcionalmente algunos otros flujos tales como ARP e ICMP, fluyan entre el contenedor y el homólogo evitando el motor de decisión. El método también puede incluir configuración de sesión de protocolo de enrutamiento en un puerto que implica que los paquetes del homólogo llegarán a la interfaz de nodo subyacente correspondiente. De manera similar, paquetes de sesión procedentes del puerto de enrutador virtual hacia el homólogo deben emitirse a través de la interfaz de nodo subyacente correspondiente. Sin embargo, el tráfico de red desde el homólogo hasta la red virtual también llegará a la misma interfaz, y el tráfico desde la red virtual que la tabla de reenvío del enrutador virtual indica que debe pasar a través del homólogo debe emitirse por la misma interfaz. El primer paquete de cada flujo regular (flujo de protocolo no de enrutamiento) dará como resultado una llamada al motor de decisión. En vez de eso, los paquetes de los flujos de protocolo de enrutamiento evitan el motor de decisión. Los que llegan a la interfaz pública se emiten directamente desde la interfaz privada y viceversa. El nodo también sondea el contenedor tanto para forzar las rutas notificadas del puerto virtual para esa sesión de protocolo de enrutamiento como para ver las rutas aprendidas por el programa residente de protocolo de enrutamiento que se ejecuta en el contenedor (es decir, las rutas notificadas por el homólogo). El nodo procesa (por ejemplo agrega) las rutas notificadas por el homólogo y las añade a la tabla de reenvío del enrutador virtual tras establecer su puerto de salida al ID del puerto virtual que tiene la configuración de sesión de protocolo de enrutamiento. Si el contenedor o sesión presenta un fallo, el nodo retira todas de tales rutas que puede haber añadido él mismo a la tabla de reenvío. El resultado es que el homólogo percibe el puerto del enrutador como si

enviara y recibiera tanto tráfico regular (anfitrión final) como tráfico de sesión de protocolo de enrutamiento. Dado que el tráfico de sesión de protocolo de enrutamiento se configura según el puerto, un enrutador IPv4 virtual puede tener más de un puerto con una sesión de protocolo de enrutamiento configurada con uno o más homólogos. Mapeando esos puertos virtuales a interfaces en diferentes nodos de red subyacente, el enrutador virtual no es un único punto de fallo como un enrutador físico. Esto mejora adicionalmente la tolerancia a fallos del sistema en comparación con sistemas anteriormente disponibles.

En aún otro aspecto, el sistema proporciona métodos para implementar o imitar una red privada virtual ("VPN"). En una realización, se proporciona un método para enlazar un dispositivo virtual a una red remota en una parte diferente de Internet que permite que el dispositivo virtual intercambie paquetes con la red remota como si estuviera físicamente conectado, y su enlace fuera seguro y privado. De esta manera, los elementos ajenos no pueden ver el tráfico de enlace ni inyectar tráfico en el enlace. El método puede incluir permitir configuraciones de puertos de dispositivo L2 y L3 virtual indicadas con un identificador de un objeto de configuración de VPN, almacenadas en la base de datos compartida. Pueden asignarse configuraciones de VPN a nodos de red subyacente específicos, o nodos de red subyacente pueden competir por conseguir un enganche en una configuración de VPN en la que la adquisición de un enganche en una configuración VPN indica que el propietario del enganche es responsable de gestionar el enlace de VPN correspondiente. En este último caso, el fallo de un nodo da como resultado la pérdida del enganche por ese nodo y por tanto la desconexión del enlace de VPN en ese nodo si el nodo todavía sigue activo. Por tanto, la adquisición del enlace de VPN por otro nodo es posible. Una configuración de VPN puede incluir un identificador de puerto privado. El puerto privado identifica un puerto virtual en el dispositivo que debe enlazarse a la red remota. El nodo de red subyacente al que se asigna la VPN localmente crea una interfaz de red lógica y la mapea al identificador de puerto privado. Después lanza un programa residente de gestión de VPN (por ejemplo OpenVPN) dentro de un contenedor y enlaza el contenedor con la interfaz recién creada. Por tanto, el tráfico emitido por el nodo a través de esa interfaz (es decir, emitido por la red virtual desde el puerto privado) llega al programa residente de gestión de VPN que a su vez lo cifra y lo reenvía al sitio remoto. Cuando la configuración de VPN especifica que el programa residente de gestión de VPN dentro del contenedor debe reenviar tráfico cifrado al sitio remoto a través de la propia conexión en red del nodo de red subyacente, el programa residente de gestión de VPN en el contenedor debe actuar por tanto como cliente de VPN (puesto que el nodo de red subyacente puede no tener una dirección IPv4 pública). Por tanto, la configuración de VPN especifica la dirección IP pública del programa residente de gestión de VPN remoto a la que debe conectarse el programa residente local. En otro aspecto, todos los nodos de red subyacente pueden no tener acceso directo a Internet y por tanto el tráfico de VPN cifrado debe volver a entrar en la red virtual para tunelarse hasta un dispositivo virtual que tiene un enlace ascendente conectado a Internet (por ejemplo, un enrutador de borde L3 con un puerto habilitado para BGP). En una realización, la configuración de VPN especifica un puerto público, que identifica un puerto virtual en un dispositivo virtual que puede reenviar paquetes (directa o indirectamente) a Internet. La configuración de VPN también especifica si el programa residente de VPN local debe actuar como servidor en una dirección IPv4 y puerto TCP (o UDP) específicos, o cliente que se conecta a una dirección IPv4 y puerto TCP o UDP remotos. El nodo al que se asigna la VPN crea una interfaz de red lógica local, la mapea al puerto virtual público, y la conecta al contenedor de programa residente de gestión de VPN. El programa residente de VPN está configurado para enviar su tráfico cifrado/tunelado desde esa interfaz, y recibirá tráfico cifrado desde el sitio remoto de esa interfaz.

En aún otro aspecto, el sistema proporciona capacidades de DHCP en la red virtual y puede configurar anfitriones (físicos o virtuales) que tienen acceso a la red virtual. De esta manera, no se necesita ningún servidor de DHCP individual, ni tampoco se necesita simular uno en un dominio L2 individual. La configuración de DHCP puede extraerse a partir de dominios L2 y definirse simplemente como un recurso que puede asociarse con un puerto virtual. Cuando un mensaje de petición o descubrimiento de DHCP llega a un puerto virtual (es decir llega a una interfaz correspondiente al puerto exterior de un dispositivo virtual), el motor de decisión de simulación de red del sistema comprueba la configuración del puerto para ver si hay una configuración de DHCP asociada. Si es así, el motor de decisión usa la configuración de DHCP asociada para construir respuestas (ofertas y respuestas de DHCP, respectivamente) a esos mensajes y le indica al nodo que emita esos paquetes desde la interfaz en la que llegó la petición. Alternativamente, el motor de decisión simula el recorrido del paquete de la red como para cualquier otro paquete de red que llega a un puerto virtual. Con respecto a esto, DHCP es otro protocolo transportado por UDP, que a su vez es un L4 que se ejecuta en IP. Este enfoque permite diseñar recursos de DHCP independientemente de la topología de red, y más específicamente, independientemente de dominios L2. Por tanto, los recursos de DHCP pueden compartirse a través de conjuntos arbitrarios de puertos según las necesidades del usuario. En realizaciones, el sistema almacena recursos de DHCP en la base de datos compartida.

En una realización, se proporciona un recurso de DHCP. El recurso de DHCP incluye configuraciones de DHCP definidas por un conjunto de opciones con valores correspondientes. El recurso de DHCP también incluye un conjunto de direcciones IP dinámicas, y posiblemente un conjunto de asignaciones estáticas de direcciones MAC a direcciones IPv4. Los componentes del recurso de DHCP pueden agruparse y asociarse con cualquier puerto exterior de dispositivo virtual. El sistema puede usar un recurso de DHCP en un método que incluye almacenar las definiciones de recurso de DHCP en la base de datos compartida, almacenar el mapeo de puerto virtual exterior a recurso de DHCP en la base de datos compartida, usar el motor de decisión para identificar paquetes de DHCP que llegan a un puerto virtual, y determinar si el puerto virtual se mapea a un recurso de DHCP. Si el puerto virtual no se mapea a un recurso de DHCP, se usan los métodos anteriormente descritos para decidir cómo debe gestionarse el

paquete. Si el puerto virtual se mapea a un recurso de DHCP, se usa la definición de recurso de DHCP para construir la respuesta lógica al paquete según el protocolo de DHCP y según la dirección MAC del remitente. Cuando el remitente está pidiendo una dirección IPv4, el sistema comprueba adicionalmente si existe una asignación estática para la dirección MAC del remitente y devuelve esa dirección IPv4 como la dirección IP ofrecida. Cuando el remitente está pidiendo una dirección IPv4 y el recurso de DHCP no contiene ninguna asignación de dirección IPv4 estática, el sistema comprueba si el recurso define un conjunto de direcciones IPv4 asignadas de manera dinámica. Si es así, y si hay direcciones no reservadas en el conjunto, se reserva una de las direcciones en nombre del cliente (identificada por dirección MAC), y se construye el mensaje de respuesta de DHCP que debe emitirse a través del puerto exterior que recibió la petición. Las reservas de direcciones IPv4 a partir de un conjunto asignable de manera dinámica definido en un recurso de DHCP pueden almacenarse en la base de datos compartida para prevenir colisiones o reutilización. La reserva incluye un alquiler que puede renovarse mediante una petición del cliente. Cuando se renueva un alquiler, el tiempo de caducidad del alquiler puede actualizarse por el motor de decisión para mantener el alquiler durante un periodo de tiempo definido.

En otra realización, el sistema implementa transferencia de estado representacional (también denominada REST API). La REST API puede usarse por el sistema y titulares del sistema para inspeccionar, monitorizar y modificar la red virtual, incluyendo la topología de red virtual. En realizaciones, la REST API proporciona control de acceso basado en funciones y es consciente de quién es propietario de cada parte de la topología virtual. La REST API también puede ser consciente de las funciones y capacidades de uno o más titulares. En un ejemplo, un titular puede crear su propio enrutador y conmutador virtual, y gestionar todos los aspectos usando la REST API. En algunos casos, tales como en nubes lasS, puede haber un titular, tal como un titular de proveedor de servicios, que tiene un conjunto de direcciones IP globales que puede alquilar a otros titulares. En tales sistemas, el titular de proveedor de servicios puede crear un puerto orientado hacia el interior y dar a otro titular la capacidad de enlazarse con ese puerto tal como se describió anteriormente.

Para fines de ilustración, en las figuras 13 y 14 se representan realizaciones de sistemas configurados para implementar uno o más de los métodos dados a conocer en el presente documento. Haciendo referencia a la figura 5, se ilustra una vista física de un sistema que está configurado para usarse con una aplicación de VPN. Un sitio remoto 50 que tiene un servidor de VPN 51 se comunica por Internet 52 con una red subyacente 53. En una realización, el servidor de VPN 51 puede ser un servidor de OpenVPN. La red subyacente puede ser una red de IP privada. Un anfitrión 54 puede ser un nodo conectado a la red subyacente e incluye una interfaz de red 55. La interfaz de red 55 está conectada a un puerto de túnel 56. El puerto de túnel 56 puede usar tunelización de GRE u otros métodos de tunelización tal como se comentó anteriormente. La interfaz de red 55 también puede comunicarse con un cliente de VPN 57 en un contenedor a través de un túnel cifrado 62. El cliente de VPN puede ser un cliente de OpenVPN. El cliente de VPN 57 en el contenedor se comunica con un conmutador programable de flujo 58 a través de tráfico de red virtual 59. El conmutador controlable de flujo 58 también se comunica con un motor de decisión 60 que se comunica con una base de datos compartida 61. Aplicando uno o más de los métodos dados a conocer en el presente documento, el sistema proporciona un programa residente de gestión de VPN que usa la red del anfitrión para alcanzar el servidor de VPN en el sitio remoto.

Haciendo referencia a la figura 6, se ilustra una vista física de otra realización de un sistema para su uso con una aplicación de VPN. Un sitio remoto 70 que tiene un servidor de VPN 71 se comunica por Internet 72 con una interfaz de red 74 de un primer anfitrión 73. El primer anfitrión 73 incluye un motor de decisión 78 que se comunica con un conmutador programable de flujo 76. El conmutador programable de flujo 76 se comunica con una interfaz de red 75 a través de un puerto de túnel 77. Una interfaz de red del primer anfitrión 73 está conectada a una red subyacente 79. La red subyacente también está conectada a una interfaz de red 81 de un segundo anfitrión 80. En una realización, la red subyacente 79 es una red privada que está aislada del Internet público. El segundo anfitrión 80 incluye además un conmutador configurable de flujo 83 que se comunica con la interfaz de red 81 a través de un puerto de túnel 82. El conmutador programable de flujo 83 también se comunica con un motor de decisión 84 y un cliente de VPN 85 en un contenedor. El motor de decisión 84 también se comunica con la base de datos compartida 86 de manera que la base de datos compartida proporciona información de estado distribuido para el sistema. Aplicando uno o más de los métodos dados a conocer en el presente documento, el sistema proporciona un programa residente de gestión de VPN en un entorno informático en la nube usando el enlace ascendente de la red virtual para alcanzar el servidor de VPN en el sitio remoto.

En todavía otras realizaciones, en el presente documento se dan a conocer sistemas y métodos para facilitar el enrutamiento de paquetes usando una red virtual superpuesta en una red subyacente. En realizaciones, la red subyacente es una red física, sin embargo, en otras realizaciones, la red subyacente puede ser una red virtual o lógica. Por motivos de claridad, la red subyacente puede describirse en cuanto a una red física, sin embargo, una o más redes virtuales pueden disponerse en capas unas sobre otras, proporcionando cada una la red subyacente para la siguiente red virtual superpuesta.

Un sistema de la presente divulgación puede incluir una red que interconecta una pluralidad de nodos. Los nodos de la red pueden corresponder a componentes físicos tales como servidores, enrutadores u otros dispositivos informáticos en comunicación con la red. Cada dispositivo puede soportar uno o más nodos. En otra realización, los nodos pueden representar dispositivos lógicos o virtuales. La red puede ser una red privada mantenida por un proveedor de servicios, en la que el proveedor de servicios vende, alquila o proporciona de otro modo capacidades

de red a una pluralidad de titulares. La red puede tener uno o más nodos, tales como nodos perimetrales, que proporcionan conectividad a una red pública. En un ejemplo, la red incluye una pluralidad de nodos orientados hacia Internet que proporcionan múltiples trayectorias de comunicación de entrada/salida entre Internet y la red. Los nodos orientados hacia Internet pueden ser enrutadores conectados a Internet. En otro ejemplo, la red incluye una pluralidad de nodos configurados para albergar máquinas virtuales de titular. Los nodos que albergan máquinas virtuales de titular pueden ser servidores anfitriones u otros dispositivos con los recursos necesarios para hacer funcionar una o más máquinas virtuales de titular. En algunas implementaciones, un nodo puede albergar múltiples máquinas virtuales de un único titular. En otra realización, un nodo puede albergar múltiples máquinas virtuales propiedad de diferentes titulares. En aún otra realización, un nodo puede funcionar tanto para albergar una máquina virtual de titular como para proporcionar conectividad a Internet a la red.

En diversas realizaciones, se da a conocer un método para enrutar un paquete desde un primer nodo hasta un segundo nodo. El método incluye recibir un paquete en un primer nodo de la red. El método incluye además invocar un motor de decisión para simular cómo recorrerá el paquete una red virtual. La simulación puede incluir acceder a una tabla de enrutamiento virtual para determinar un siguiente salto para el paquete, en la que el siguiente salto es o bien un puerto orientado hacia el interior (también denominado puerto lógico) o bien un puerto orientado hacia el exterior (también denominado puerto materializado), y continuar accediendo a tablas de enrutamiento virtuales posteriores en serie hasta que se determina que el siguiente salto es un puerto orientado hacia el exterior en un segundo nodo de la red. Tras haber determinado el motor de decisión cómo procesar el paquete, el paquete puede enviarse por la red subyacente al puerto orientado hacia el exterior del segundo nodo. En realizaciones, la red subyacente puede ser una red de Ethernet, una red de IP privada o pública, u otra red que proporciona conectividad entre la pluralidad de nodos.

En una realización, cada nodo de la red contiene un conector de borde. Cada conector de borde contiene una implementación de un conmutador configurable de flujo y motor de decisión que se ejecutan en el mismo nodo o anfitrión físico. En una realización, un conmutador configurable de flujo puede comprender software tal como Open vSwitch. El motor de decisión puede simular uno o más conmutadores L2 virtuales y enrutadores L3 virtuales. Un conector de borde puede tener interfaces físicas, interfaces virtuales o ambas. Las interfaces virtuales son interfaces tales como, por ejemplo, interfaces de escucha o interfaces virtuales a nivel de núcleo. Las interfaces físicas son, por ejemplo, una tarjeta de interfaz de red física (NIC).

Un conmutador configurable de flujo es un componente de software que aplica una lista de acciones a todos los paquetes que coinciden con una regla de flujo. Asociada con una lista de acciones hay una coincidencia de flujo que especifica qué paquetes coinciden con el flujo. En algunas realizaciones, la coincidencia de flujo puede especificarse mediante un patrón de cabecera de protocolo de paquete. La coincidencia de flujo puede basarse en una o más porciones de los datos de paquete, incluyendo, por ejemplo, los puertos de origen y destino, direcciones de origen y destino, dirección MAC. La coincidencia de flujo también puede basarse en combinaciones de datos de paquete o subconjuntos de datos de paquete, tales como una porción de las direcciones de origen o destino. Una regla de flujo puede comprender al menos una coincidencia de flujo y una lista de acciones, y puede denominarse “flujo”. Dos flujos (uno entrante, uno saliente) forman una conexión. Generalmente, dos flujos, un flujo entrante y un flujo saliente, forman una conexión para comunicaciones entre un cliente fuera de la red y una máquina virtual del titular u otro servicio proporcionado dentro de la red. Cada flujo representado por una o más reglas de flujo puede almacenarse en un estado distribuido mantenido en una base de datos compartida. En una realización, cada flujo se almacena en un estado distribuido mantenido en un nodo de la red accesible por todos los demás nodos que requieren acceso al estado distribuido. Los flujos almacenados pueden indexarse mediante su coincidencia de flujo o mediante otros criterios asociados con las reglas de flujo.

En una realización, puede mantenerse una tabla de flujo que copia en memoria caché las decisiones de enrutamiento realizadas para el primer paquete en una dirección de una conexión. La tabla de flujo se mantiene dentro del conmutador configurable de flujo. La red puede tener múltiples puntos de acceso posibles a la red externa, y las conexiones no necesitan usar la misma ruta virtual entrante como saliente. Permitir diferentes rutas entrantes y salientes puede mejorar la tolerancia a fallos del sistema en caso de interrupciones en determinadas porciones de la red. Permitir diferentes rutas entrantes y salientes también puede permitir un uso mejorado de recursos de red equilibrando cargas entre diferentes trayectorias en la red.

La red también puede comprender elementos de reenvío que enrutan y conmutan paquetes entre los nodos de la red. Los elementos de reenvío pueden ser conmutadores L2, enrutadores L3 o combinaciones de conmutadores L2 y enrutadores L3. Los elementos de reenvío pueden ser o bien físicos o bien virtuales, y la red puede incluir combinaciones de elementos de reenvío físicos y virtuales. Un elemento de reenvío físico es un componente de hardware, mientras que un elemento de reenvío virtual puede implementarse en software. En una realización, un elemento de reenvío virtual se implementa usando tablas. Por ejemplo, el motor de decisión puede usarse para simular el enrutamiento y la conmutación de paquetes según una topología virtual establecida para la red.

En la red, pueden conectarse enrutadores virtuales a otros enrutadores virtuales para construir una topología de red virtual que puede ilustrarse mediante un gráfico de red virtual. Cada enrutador virtual puede tener una pluralidad de puertos virtuales, en el que cada puerto virtual es o bien un puerto orientado hacia el interior (lógico) o bien un puerto orientado hacia el exterior (materializado). Por ejemplo, cada enrutador virtual puede incluir una tabla de

enrutamiento virtual, y los puertos orientados hacia el interior pueden identificarse realizando una consulta en la tabla de enrutamiento virtual para determinar el siguiente salto para un paquete que está enrutándose por el enrutador virtual. Cada consulta puede conducir a un puerto orientado hacia el interior homólogo de otro enrutador virtual o un puerto orientado hacia el exterior, permitiendo que el motor de decisión simule el recorrido de una topología virtual que tiene múltiples enrutadores virtuales. En una realización, un puerto orientado hacia el exterior puede corresponder a un puerto en un conmutador configurable de flujo, tal como un puerto de túnel. En algunas realizaciones, un puerto orientado hacia el exterior puede corresponder a la ubicación de un nodo que proporciona conectividad a Internet. En otra realización, un puerto orientado hacia el exterior puede corresponder a la ubicación de una máquina virtual que funciona dentro de la red. Para puertos orientados tanto hacia el interior como hacia el exterior, la configuración estática del puerto virtual en el árbol de configuración compartido contiene explícitamente el tipo de puerto (es decir, orientado hacia el interior o hacia el exterior) y, en el caso de puertos orientados hacia el interior, el identificador universalmente único ("port_uuid") del otro extremo del enlace virtual (es decir el puerto orientado hacia el interior homólogo). Adicionalmente, los enrutadores virtuales pueden tener sus propias direcciones IP. Adicionalmente, cada enrutador virtual puede soportar protocolos tales como protocolo de pasarela de frontera ("BGP") y/o protocolo de pasarela interna ("IGP").

En otra realización, los conectores de borde pueden tener puertos de túnel que no son puertos de los enrutadores virtuales. El puerto de túnel puede usarse para conectar un conector de borde a otro conector de borde a través de la red. Por ejemplo, el conmutador configurable de flujo de un conector de borde puede conectarse al conmutador configurable de flujo de otro conector de borde mediante un puerto de túnel. En una realización, un paquete puede llegar a un conector de borde destinado a una máquina virtual en otro conector de borde. Cuando un paquete está destinado a un puerto orientado hacia el exterior en otro conector de borde, se envía a ese conector de borde a través de un túnel. Puede mantenerse una tabla en un estado distribuido que mapea puertos a conectores de borde y una tabla que mapea conectores de borde a túneles. Por tanto un conector de borde puede determinar a través de qué túnel enviar un paquete basándose en un puerto seleccionado (no local). En otra realización, el mapeo de puertos orientados hacia el exterior a conectores de borde y de conectores de borde a túneles puede mantenerse en un nodo independiente, y los conectores de borde pueden comunicarse con el nodo independiente para determinar el túnel apropiado para un paquete.

En una realización, los conectores de borde en cada nodo tienen acceso a un estado distribuido, que puede almacenarse en una base de datos compartida. El estado distribuido se mantiene y comparte por los conectores de borde. El estado distribuido puede contener, por ejemplo, el árbol de configuración y otros datos referentes a la topología de red virtual y/o física. En una realización, puede implementarse un estado distribuido usando Zookeeper y memcache. En otra realización, parte del estado distribuido es un árbol de configuración, pero se contemplan otras estructuras tales como tablas de elección arbitraria y árboles n-arios. Los conectores de borde pueden acceder al árbol de configuración y a otros datos compartidos según se necesite, tal como mediante el motor de decisión.

El término "cliente" se usa en el presente documento para indicar un cliente de red externa, tal como un navegador web, que está intentando alcanzar un servidor albergado dentro del sistema, por ejemplo, para acceder a los servicios de una máquina virtual. El término "titular" se usa para indicar un cliente del proveedor de servicios. Un titular puede tener una o más máquinas virtuales u otros servicios que funcionan en máquinas físicas dentro del sistema, y puede querer establecer dinámicamente reglas de equilibrio de carga o traducción de dirección de red ("NAT") entre estas máquinas virtuales y los clientes.

Haciendo ahora referencia a la figura 7, se ilustra un servidor 101 con dos tarjetas de interfaz de red, NIC A 111 y NIC B 112. Para fines de ilustración, algunos nodos pueden designarse como nodos perimetrales orientados hacia clientes de Internet, y que proporcionan conectividad a Internet a la red. Otros nodos pueden designarse como nodos anfitriones configurados para albergar máquinas virtuales de titular u otros servicios dentro de la red. Para fines de ilustración, los nodos perimetrales y nodos anfitriones pueden mostrarse con arquitecturas simétricas, sin embargo, en diversas realizaciones, puede usarse una variedad de arquitecturas para los diversos nodos en el sistema. Aunque se ilustra en cuanto a nodos perimetrales orientados hacia Internet y nodos que albergan máquinas virtuales, el sistema también puede contener nodos intermedios incluyendo dispositivos de almacenamiento de datos y servidores de soporte deseados para facilitar el funcionamiento de la red. Tal como se muestra en la figura 7, NIC A 111 tiene una conexión a Internet 151 y NIC B 112 tiene una conexión al tejido de proveedor interno 152 (la red subyacente). El tejido de proveedor interno 152 puede ser una red de IP privada u otra red que proporciona conectividad IP entre los nodos.

El sistema incluye un componente de software que implementa muchas de las características de la red virtual superpuesta en la red física. Para ilustrar el funcionamiento de los componentes de software, las acciones tras la recepción de un paquete se describen para operaciones seleccionadas.

En una realización, se recibe un paquete SYN para establecer una conexión TCP. El paquete SYN se recibe desde Internet 151 en NIC A 111. Se reciben paquetes por el conector de borde en el conmutador configurable de flujo 161 para su conmutación. El conmutador configurable de flujo 161 intenta identificar una regla de flujo haciendo coincidir datos asociados con el paquete con las reglas de flujo almacenadas en la tabla de flujo 162. Los datos coincidentes pueden incluir, por ejemplo, puertos de origen y destino, direcciones de red, direcciones MAC u otros datos asociados con el paquete. El paquete SYN es normalmente el primer paquete en un flujo y por tanto el conmutador

configurable de flujo 161 no encuentra una entrada correspondiente al primer paquete en la tabla de flujo 162. Tras no encontrar una entrada correspondiente en la tabla de flujo, el conmutador configurable de flujo 161 realiza una llamada de función a un motor de decisión 165 y comunica el paquete al motor de decisión. El paquete puede llegar a un puerto del conmutador configurable de flujo, y el conmutador configurable de flujo puede comunicar el ID de puerto entrante al motor de decisión con el paquete. Aunque la función del conmutador programable de flujo y el motor de decisión se describen por separado por motivos de claridad, resultará evidente que los componentes de software pueden integrarse según se desee. Alternativamente, cada componente puede dividirse o combinarse con otros componentes siempre que se mantengan las funciones del componente. En una realización, el motor de decisión se comunica con el conmutador configurable de flujo 161 a través del protocolo OpenFlow y traduce el ID de puerto entrante del conmutador configurable de flujo en un ID de puerto virtual ("vport"). Alternativamente, este mapeo puede basarse en dirección MAC o credenciales 802.1x en lugar del ID de puerto entrante. El resto del enrutamiento del paquete puede depender de su información L3. El motor de decisión 165 tiene la lógica para simular la ruta del paquete a través de la topología de red virtual. En una realización, sólo el primer paquete de una conexión provocará una llamada al motor de decisión, puesto que, una vez creado el flujo en la tabla de flujo 162 ese flujo puede aplicarse a paquetes posteriores del mismo flujo.

Para crear una regla de flujo asociada con un nuevo flujo, en una realización el motor de decisión construye una lista de acciones que indica cómo procesar y reenviar el paquete y la inserta como una regla de flujo en la tabla de flujo. En paquetes posteriores que coinciden con los criterios para ese flujo se aplica la lista de acciones, que puede incluir enrutar el paquete a un puerto dado. Si el paquete estaba destinado a otro conector de borde que se ejecuta en otro servidor, puede enrutarse al otro conector de borde a través de un puerto de túnel. Los puertos de túnel pueden conectar conectores de borde o nodos en la red subyacente y se usan para reenviar paquetes entre conectores de borde. En vez de eso, cuando un paquete está destinado a un puerto virtual en otro conector de borde, se envía a ese conector de borde a través de un túnel. El protocolo de túnel es, en una realización, GRE+IP. Este protocolo de tunelización permite que un conmutador configurable de flujo 161 en un servidor 101 se comunique a través del tejido de proveedor interno 152 con otro conmutador configurable de flujo (no representado) en otro servidor (no representado). La figura 8 ilustra la interconexión física de una pluralidad de conectores de borde 203, 204, 205 en una pluralidad de anfitriones respectivos 210, 221, 222 conectados mediante el tejido de red L3 interno del proveedor 202. Con el fin de ilustrar, las máquinas virtuales 211 y 212 funcionan en el anfitrión 221, mientras que las máquinas virtuales 213 y 214 funcionan en el anfitrión 222. Una consola de gestión 206 también puede estar conectada al tejido de red interno 202, que forma la red subyacente. La figura 9 ilustra la topología virtual que está superpuesta en esta red física. El protocolo de tunelización permite que el tejido enrute los paquetes entre conmutadores configurables de flujo sin modificación del hardware en el tejido de proveedor 152. Puesto que los paquetes reales se desplazan por IP (L3) en contraposición a Ethernet (L2), la red puede ajustarse a escala y puede no estar limitada por limitaciones de distancia aplicables a comunicaciones por Ethernet. Los puntos finales de los túneles son puertos en los conmutadores configurables de flujo de los conectores de borde, pero los puertos de túnel se tratan de manera diferente a puertos orientados hacia el exterior. La porción de IP de la cabecera de paquete de túnel permite que el paquete obtenga el anfitrión correcto, y luego la porción de GRE de la cabecera sirve para llevar el paquete al puerto de túnel correcto. Aún otra clave en la cabecera sirve para identificar el puerto orientado hacia el exterior de destino, de modo que el conector de borde de recepción puede enrutar el paquete al puerto local correcto.

Haciendo ahora referencia a la figura 8, se ilustra una red que comprende tres conectores de borde 203, 204 y 205, en la que cada conector de borde reside en un anfitrión. Continuando con el ejemplo anterior, se supone que un paquete se recibió en el conector de borde 203 en una tarjeta de interfaz de red física (NIC) desde Internet 151 a través del enrutador conectado a Internet 201 y que el paquete está destinado a la máquina virtual 211. Se recuerda que el paquete es el primer paquete de un flujo, de modo que no hay ninguna regla de flujo correspondiente al paquete en la tabla de flujo. Puesto que no hay ninguna entrada de flujo correspondiente en la tabla de flujo, se invoca el motor de decisión. El motor de decisión determina un puerto virtual (vport) basándose en el puerto en el que se recibió el paquete por el conmutador configurable de flujo, y posiblemente en la dirección MAC y credenciales 802.1x. El vport en este caso es un puerto orientado hacia el exterior (materializado) correspondiente a una NIC y un puerto en el conmutador configurable de flujo. El motor de decisión usa el vport para determinar a qué enrutador virtual o conmutador virtual está conectado el puerto. Tal como se comentó anteriormente, un enrutador virtual puede implementarse mediante una tabla accesible para el motor de decisión y mantenida en un estado distribuido. Una vez que el motor de decisión determina qué enrutador virtual está conectado al puerto orientado hacia el exterior, el motor de decisión selecciona una ruta coincidente identificando la dirección IP de destino en la tabla de enrutadores virtuales correspondiente. En una realización, el motor de decisión puede seleccionar una ruta a partir de varias rutas, o varias rutas de igual coste, usando un algoritmo de equilibrio de carga.

En otra realización, cuando el motor de decisión accede a una tabla de enrutadores virtuales para consultar una dirección IP, pueden aplicarse procesos de preenrutamiento y posenrutamiento. El proceso de preenrutamiento puede alterar el paquete, incluyendo las direcciones IP de origen y destino y puertos de origen y destino, para realizar la traducción de dirección de red ("NAT"). El método de enrutamiento puede comprender extraer las direcciones IP de origen y destino, consultar las direcciones IP en una tabla de enrutamiento virtual correspondiente a un enrutador virtual, seleccionar un destino (si se encuentra más de una ruta) y reenviar el paquete al puerto correspondiente a la entrada de ruta. El reenvío del paquete depende de si el siguiente salto de la ruta coincidente

es un puerto orientado hacia el interior (lógico) o un puerto orientado hacia el exterior (materializado). Dado que pueden implementarse enrutadores virtuales como tablas, el enrutamiento entre dos enrutadores virtuales comprende una consulta en tablas de enrutadores virtuales sucesivas. En una realización, se mantiene una tabla de enrutamiento global para cada enrutador L3 virtual. La tabla de enrutamiento global puede almacenarse en un estado distribuido en la base de datos compartida. Alternativamente, la tabla de enrutamiento global puede almacenarse en un conector de borde seleccionado. En otra realización, la tabla de enrutamiento global se mantiene en cada conector de borde y los conectores de borde actúan conjuntamente para mantener y actualizar la tabla de enrutamiento global en cada uno de los otros conectores de borde en la red.

Haciendo ahora referencia a la figura 9, se ilustra una topología virtual que puede superponerse en una red subyacente, tal como la red física de la figura 8. En un ejemplo, un paquete puede llegar a un puerto orientado hacia el exterior asociado con el enrutador L3 virtual 301 y su dirección IP de destino es la dirección IP de la VM 211. El motor de decisión puede usar el vport en el que llegó el paquete para determinar a qué enrutador virtual está conectado el vport, en este caso el enrutador L3 virtual 301. En una realización, el enrutador L3 virtual 301 puede ser un enrutador de proveedor, creado y administrado por el proveedor de servicios que hace funcionar la red. El motor de decisión puede usar entonces la dirección IP asociada con el paquete para determinar un puerto de salida para el paquete. Si el puerto de salida es un puerto orientado hacia el exterior local, entonces se establece un flujo en el conmutador configurable de flujo y se enruta el paquete hasta el puerto orientado hacia el exterior local. Si el puerto orientado hacia el exterior no es local, el paquete se enruta fuera de un puerto de túnel según una tabla de vport a anfitrión y una tabla de anfitrión a puerto de túnel. Si el puerto es un puerto orientado hacia el interior de otro enrutador o conmutador, entonces se repite el mismo proceso de consulta hasta que se identifica un puerto orientado hacia el exterior. Para continuar con la figura 9, la consulta en la tabla del enrutador virtual 301 puede devolver un puerto orientado hacia el interior correspondiente al enrutador virtual 302. Tras, o en combinación con, la consulta, pueden aplicarse procesos de posenrutamiento al paquete según se desee. Cuando la consulta devuelve un puerto orientado hacia el interior correspondiente a otro enrutador virtual, en este caso el enrutador L3 virtual 302, el motor de decisión puede repetir el mismo proceso para el enrutador virtual 302. El enrutador virtual 302 puede ser, por ejemplo, un enrutador virtual creado por un titular para enrutar tráfico entre las máquinas virtuales del titular, las máquinas virtuales 211 y 212. Las máquinas virtuales del titular pueden estar en el mismo anfitrión, o pueden estar ubicadas en anfitriones diferentes dentro de la red. Un titular puede alquilar recursos de red del proveedor de servicios para hacer funcionar cualquier número de máquinas virtuales u otros servicios dentro de la capacidad de la red sujeta a reglas establecidas por el proveedor de servicios. El motor de decisión realiza una simulación que puede incluir cualquier preenrutamiento asociado con el enrutador L3 virtual 302, consultar la dirección IP en la tabla de enrutamiento virtual para determinar un siguiente salto, y cualquier posenrutamiento. En este ejemplo, el siguiente salto es la máquina virtual 211, que está albergada en un conector de borde diferente del conector de borde 203. La tabla de enrutadores virtuales para el enrutador virtual 302 proporciona un vport correspondiente a la VM 211 según se configura por el titular o el proveedor de servicios. En una realización, el proveedor de servicios puede mover máquinas virtuales de titular entre diferentes nodos en la red para gestionar el uso de equipos o mantener operaciones durante el mantenimiento o la reparación de componentes físicos en la red. Entonces, el motor de decisión consulta la ubicación física del vport de salida en un diccionario de ubicaciones de puerto mantenido en el estado distribuido. Puesto que todos los paquetes reenviados por los conmutadores son paquetes L2, hay un espacio en los paquetes L2 para direcciones MAC. Sin embargo, puesto que hay puertos de túnel entre dos conmutadores configurables de flujo, las direcciones MAC pueden no ser necesarias para determinadas aplicaciones. Más específicamente, en determinadas realizaciones, no hay necesidad de reenviar las direcciones MAC reales puesto que el conector de borde de salida puede construir la dirección MAC basándose en su propia información local, usando ARP para determinar la MAC de siguiente salto. En vez de eso, se codifica el vport del destino (en este caso la VM 211) en el espacio para la dirección MAC. Después se envuelve el paquete en GRE+IP con la dirección IP del nodo perimetral como destino. Ahora el paquete está listo para enrutarse a través de la red L3. Haciendo referencia a la figura 7, una lista de acciones que contiene cualquier pre y posenrutamiento y el destino de enrutamiento puede instalarse en la tabla de flujo 162 para hacer coincidir todos los paquetes futuros de este flujo y el paquete puede enviarse hacia fuera a través del protocolo de tunelización y a través del enrutador de sistema operativo 113 y luego a la NIC B 112. Tras salir de la NIC B 112, el paquete se enruta por el tejido de proveedor interno 152 como se enrutaría cualquier otro paquete de IP, con la dirección IP de destino del conector de borde 204.

Cuando se recibe el paquete por el conector de borde 204, se recibe en un túnel correspondiente a un puerto de túnel del conmutador configurable de flujo. Puesto que el paquete se recibe en un puerto de túnel, el conector de borde puede tratar este paquete de manera diferente a un paquete que entra en un puerto orientado hacia el exterior. El paquete es de nuevo el primer paquete recibido en este flujo y el conector de borde 204 invocará el motor de decisión. En una realización, la clave de túnel codifica el id de vport de destino. El motor de decisión puede usar el id de vport para determinar una dirección MAC y el número de puerto local de la máquina virtual 211. En algunos casos, el motor de decisión puede iniciar una petición de ARP para determinar la dirección MAC de la VM 211. Alternativamente, la dirección MAC puede copiarse en memoria caché en una tabla de ARP. Se mantiene una tabla de ARP (de IP a MAC) por cada puerto de un enrutador virtual. La tabla de ARP puede compartirse en estado distribuido almacenado en una base de datos compartida. Tras haber determinado el motor de decisión el vport de la VM 211, el sistema puede instalar un flujo en la tabla de flujo para enrutar futuros paquetes de este flujo. Entonces puede enrutarse el paquete al puerto del conmutador configurable de flujo correspondiente a la VM 211. Aunque la VM 211 es una máquina virtual local que se ejecuta en el anfitrión 221, que también alberga el conector de borde

205, el motor de decisión todavía puede usar la dirección IP de destino para encontrar la dirección MAC de destino. De esta manera, el sistema extrae si la VM es local o un puerto convencional a otro enrutador o conmutador demostrando adicionalmente la flexibilidad del sistema.

Una vez que se han establecido los flujos, los paquetes entrantes posteriores en la misma conexión coincidirán con los flujos en la tabla de flujos de los conectores de borde 203 y 204 y se modificarán y reenviarán por los conmutadores configurables de flujo en esas máquinas sin invocar un motor de decisión. Este proceso establece el flujo entrante de una conexión a través del sistema al destino deseado.

Cuando la VM 211 responde en la misma conexión, el primer paquete que envía desencadenará el sistema para establecer un flujo correspondiente en el sentido opuesto. Cuando se establece un nuevo flujo, el motor de decisión puede acceder al estado distribuido para determinar si se estableció anteriormente un flujo en el sentido opuesto. Este estado distribuido facilita la implementación de otros procesos, tales como NAT y también permite que el sistema limpie conexiones terminadas tal como se describe adicionalmente a continuación. En otras realizaciones, máquinas virtuales albergadas en componentes físicos diferentes pueden conectarse al mismo enrutador virtual.

Haciendo ahora referencia a la figura 10, se ilustra un resumen de alto nivel de un proceso que se ejecuta en un conector de borde que realiza una realización de método descrita anteriormente. En una realización, un conector de borde se ejecuta en una máquina física con al menos una CPU que recibe paquetes de una red externa, tal como Internet, en el que los paquetes van dirigidos a una dirección IP asociada con un titular del sistema. Las direcciones IP de titular pueden asignarse a máquinas virtuales de titular que se ejecutan en uno o más anfitriones dentro del sistema. En una realización, la dirección IP asociada con una máquina virtual de titular puede permanecer constante aunque el titular o proveedor de servicios reubique la máquina virtual a un anfitrión diferente dentro del sistema. En una realización, el sistema permite que múltiples titulares compartan el enlace ascendente de un proveedor de servicios, permitiendo enrutar múltiples direcciones IP en un enlace ascendente a diferentes conectores de borde y diferentes máquinas virtuales de titular. Cuando el conector de borde recibe un paquete en la etapa 410, extrae una pluralidad de datos en la etapa 412, incluyendo, pero sin limitarse a, direcciones de origen y destino y puertos de origen y destino. Tras extraer los datos, el conector de borde consulta la pluralidad de datos en una tabla de flujo (etapa 414) y determina si ya se ha establecido el flujo (etapa 416). Un ejemplo de un flujo sería un sentido de una conexión TCP, combinándose un flujo entrante y saliente para formar una única conexión TCP. Si el flujo ya existe, se aplica la lista de acciones de flujo al paquete y se reenvía el paquete al puerto del conmutador configurable de flujo indicado por la lista de acciones de flujo en la etapa 418.

Si el flujo no existe, este es el primer paquete en el flujo recibido por el nodo, y el conector de borde debe determinar en la etapa 417 a qué puerto virtual llegó el paquete, basándose, por ejemplo, en la dirección MAC, direcciones de origen y destino, o puertos de origen y destino. Una vez que el conector de borde determina el ID de puerto virtual, el conector de borde puede determinar a qué elemento de reenvío virtual está conectado ese puerto. En la realización de la figura 10, el elemento de reenvío virtual es un enrutador virtual, pero pueden usarse otros elementos de reenvío virtuales, tales como conmutadores virtuales, según sea necesario en el sistema tal como se comenta a continuación. Una vez que el conector de borde determina el VFE, el conector de borde realiza otra consulta en la etapa 420. La consulta se realiza consultando la dirección IP de destino en una serie de elementos de reenvío virtuales. Los elementos de reenvío virtuales pueden comprender cualquier combinación de enrutadores virtuales y conmutadores virtuales que incluyen tablas de enrutamiento virtuales para determinar la trayectoria apropiada para el paquete que está reenviándose. En la realización mostrada, en la etapa 420, el motor de decisión determina el destino del paquete en un primer elemento de reenvío virtual. El primer elemento de reenvío virtual puede ser un enrutador virtual, en cuyo caso el destino devuelto puede ser o bien un puerto orientado hacia el exterior o bien un puerto orientado hacia el interior. Tal como se indicó anteriormente, un puerto orientado hacia el interior está emparejado con otro puerto orientado hacia el interior de un segundo enrutador virtual, y el segundo enrutador virtual tiene otra tabla de enrutamiento. Si se devuelve un puerto orientado hacia el interior, el motor de decisión consulta la dirección de destino en la tabla de enrutamiento del segundo enrutador virtual en la etapa 420 y continúa hasta que se devuelve un puerto orientado hacia el exterior. En una realización, cada titular puede tener un único enrutador virtual configurado para enrutar todos los paquetes gestionados por ese titular. En otras realizaciones, algunos titulares pueden tener una pluralidad de enrutadores virtuales, conmutadores virtuales u otros elementos de reenvío virtuales que definen la porción del titular de la topología de red virtual. El motor de decisión también construye una serie de acciones que van a realizarse en el paquete a partir de cada tabla de enrutamiento virtual. Cada etapa de enrutamiento también puede tener procesos preenrutamiento o posenrutamiento que se añaden a la lista de acciones y se incorporan en la regla de flujo que va a aplicarse a los paquetes que coinciden con el flujo.

Una vez que se ha devuelto un puerto orientado hacia el exterior (etapa 424), el conector de borde determina si el puerto orientado hacia el exterior es local (etapa 426). Si el puerto es local, se añade una acción a la lista de acciones (etapa 430) para enrutar el paquete al puerto orientado hacia el exterior local. En diversas realizaciones, el puerto orientado hacia el exterior local puede ser una tarjeta de interfaz de red o máquina virtual. Después se añade la regla de flujo a la tabla de flujo (etapa 430) y se aplica al paquete (etapa 418). Si el puerto orientado hacia el exterior no es local, entonces el puerto está en un conector de borde diferente. En una realización, los conectores de borde pueden conectarse mediante puertos de túnel, tales como puertos de túnel GRE_IP. En la etapa 432, el conector de borde accede a una tabla de puerto virtual a túnel para intentar mapear el puerto orientado hacia el exterior a un puerto de túnel. Si no hay ninguna entrada correspondiente en el mapeo de tabla de puerto virtual a

túnel, se añade una acción para soltar el paquete y enviar un paquete de ICMP a la lista de acciones (etapa 434), se añade la regla de flujo a la tabla de flujo (etapa 430) y se aplica la regla de flujo al paquete (etapa 418).

Si el puerto orientado hacia el exterior está en la tabla de puerto orientado hacia el exterior a túnel, entonces se añade una acción para emitir el paquete a ese túnel a la lista de acciones (etapa 436), se añade el flujo a la tabla de flujo (etapa 430), y se aplica la lista de acciones al paquete (etapa 418).

En una realización, el sistema instala la lista de acciones y regla de flujo en la trayectoria de datos de conmutador configurable de flujo, y el conmutador configurable de flujo aplica la lista de acciones a cualquier paquete posterior que coincide con la regla de flujo, tal como se muestra en la etapa 416. Tal como se describió anteriormente, parte de la lista de acciones incluye a qué puerto del conmutador configurable de flujo debe enviarse el paquete. El conector de borde consulta el puerto en una tabla de puerto a IP de anfitrión, y envía el paquete a la dirección IP. Después almacena la lista de acciones en una tabla de flujo. En todos los paquetes posteriores que tienen una pluralidad de datos coincidentes se aplicará el mismo conjunto de acciones, dando como resultado que se enruten a la misma dirección IP.

Durante el proceso de identificar la dirección de destino en los enrutadores virtuales, el flujo puede ser no enrutable, incluirse en un agujero negro o coincidir con una ruta de rechazo, punto en el cual se suelta el paquete, o se devuelven paquetes de ICMP. En esta realización, puede crease un flujo para soltar todos los paquetes que coinciden con la regla del flujo. De esta manera, el sistema puede configurarse para gestionar paquetes no enrutables o examinar selectivamente datos no deseados según reglas establecidas por el proveedor de servicios o titular.

En aún otra realización, un conector de borde que alberga VM de titular puede tener múltiples direcciones IP y múltiples NIC conectadas a la red de tejido interno. En tal caso, los conectores de borde orientados a Internet pueden seleccionar una de múltiples trayectorias al conector de borde que alberga VM. Además, un conector de borde que alberga VM con múltiples direcciones IP puede tener un único ID para identificar el conector de borde, y el motor de decisión que enruta flujos puede seleccionar una de las direcciones IP del conector de borde que alberga VM, por ejemplo usando un algoritmo de equilibrio de carga o de manera aleatoria.

Otra realización del sistema puede usar identificadores para los nodos perimetrales distintos de direcciones IP. Por ejemplo, el tejido de red puede basarse en circuito, tal como conmutación de etiquetas multiprotocolo (“MPLS”) u otro controlador de OpenFlow personalizado con circuitos dedicados entre conectores de borde. En esta realización, los circuitos pueden sustituir a los túneles de GRE entre conmutadores configurables de flujo en nodos perimetrales.

En otra realización, el sistema proporciona etapas de preenrutamiento y posenrutamiento antes y después de la etapa de enrutamiento. El preenrutamiento y posenrutamiento pueden usarse para implementar la traducción de dirección de red (NAT), equilibrado de carga u otras características L3/L4. En una realización, la etapa de preenrutamiento puede cambiar el destino del flujo (como, por ejemplo, en la traducción de dirección de red) y el enrutamiento saliente puede cambiar el origen del flujo (de nuevo, como ejemplo, la traducción de dirección de red). Para coordinar los mapeos realizados por los flujos directo e inverso que componen una única conexión, tal como en una conexión TCP, pueden almacenarse traducciones de conexión en el estado distribuido con un gran tiempo asignado. Estas traducciones también pueden limpiarse de manera proactiva cuando el seguimiento de conexión detecta una conexión cerrada de manera limpia.

En una realización del presente sistema, se implementa NAT usando transformaciones de preenrutamiento y posenrutamiento. En la configuración de flujo, la etapa de preenrutamiento de NAT determina si se estableció anteriormente un flujo en el sentido opuesto (entrante frente a saliente) en el estado distribuido y, si es así, el mapa anteriormente creado se invierte para el nuevo flujo. Puesto que las reglas de flujo se almacenan en el sistema de estado distribuido accesible por todos los nodos, tras la creación de un nuevo flujo en un nodo diferente, es posible determinar si se creó anteriormente un flujo de sentido opuesto. Si el flujo de sentido opuesto no está en el estado distribuido, el motor de decisión crea una nueva traducción y almacena su mapa de traducción en el estado distribuido asociado con el nuevo flujo. Para el flujo entrante, o alternativamente el primer flujo establecido, puede aplicarse la traducción de dirección a la dirección de destino antes de la etapa de enrutamiento y la etapa de enrutamiento puede enrutar basándose en la dirección IP traducida. En el flujo saliente, o alternativamente el segundo flujo en la conexión, puede realizarse NAT tras el enrutamiento, para traducir la dirección de origen para que sea la dirección IP externa no privada. La información de traducción puede almacenarse en el estado distribuido asociado con la regla de flujo antes de reenviar el paquete inicial del flujo de manera que la información de traducción es accesible cuando se recibe el paquete inicial del flujo inverso en el conector de borde correspondiente.

En otra realización, la traducción de dirección de red de destino (DNAT) puede usarse para traducir de una dirección IP públicamente disponible a una dirección de red privada para exponer servicios albergados en una red privada a Internet general. En algunas realizaciones, puede proporcionarse una zona desmilitarizada (DMZ) entre Internet general y la red privada. En un proceso de DNAT, la dirección de destino puede traducirse durante una etapa de preenrutamiento para el flujo entrante, y en el flujo saliente correspondiente, la dirección de origen puede traducirse durante una etapa de posenrutamiento. En una implementación, la dirección de destino puede ser uno de varios servidores posibles, y el servidor de destino puede seleccionarse mediante un algoritmo de equilibrado de carga, tal

como un algoritmo aleatorio o un algoritmo de rotación.

En la traducción de dirección de red de origen (SNAT) múltiples clientes en la LAN privada pueden compartir la misma dirección IP pública. Una dirección de origen asociada con conexiones salientes, tales como conexiones desde máquinas virtuales de titular hasta una red externa, puede traducirse a la misma dirección IP en el flujo saliente. En el flujo entrante correspondiente, una dirección IP de destino de paquete puede traducirse a la dirección IP privada correspondiente basándose, por ejemplo, en el número de puerto y la dirección IP de origen.

En otra realización, el sistema puede configurarse para proporcionar simulación de ARP para redes de área local privadas. El sistema puede permitir que anfitriones de red individuales tales como un invitado de máquina virtual se conecten a un puerto de enrutador virtual sin consumir direcciones de pasarela/radiodifusión suplantando a otros anfitriones individuales cuando el anfitrión implementa ARP para los mismos. En un diseño basado en Ethernet tradicional, esto consumiría al menos un intervalo de /30 direcciones, incluyendo la dirección del invitado, la dirección de pasarela, y la dirección de radiodifusión, más una dirección sin usar.

Como método de reducir el número de direcciones IP consumidas, cada puerto del enrutador puede configurarse con una dirección MAC y un prefijo de red (nw_prefix) y un indicador que indica si hay un anfitrión individual conectado al puerto. La dirección de pasarela usada puede ser la primera dirección en el intervalo de nw_prefix. Si el indicador de anfitrión individual no está establecido, el enrutador puede gestionar tráfico hacia y desde el puerto según sus reglas de funcionamiento convencionales. Si el indicador de anfitrión individual está establecido, la porción de dirección del nw_prefix especifica la dirección del anfitrión de red individual de ese puerto. Los puertos aguas abajo del enrutador pueden configurarse de manera que comprenden nw_prefixes no solapantes con el indicador de anfitrión individual no establecido y puertos con el indicador de anfitrión individual establecido, que pueden compartir intervalos de dirección idénticos especificados por sus nw_prefixes. En muchas realizaciones, los intervalos de dirección usados por los puertos de anfitrión individual y no de anfitrión individual no se solaparán.

Si se envía un paquete de IP entre puertos con el indicador de anfitrión individual establecido, el enrutador puede reenviar el paquete de IP sin comprobar o reducir el tiempo de vida (“TTL”), emulando un conmutador L2. Si se recibe una petición de ARP desde un puerto con el indicador de anfitrión individual establecido para la dirección asociada con otro puerto de anfitrión individual, el enrutador responde al ARP, suplantando al objetivo. El resultado es que un anfitrión individual que desea enviar tráfico a un anfitrión fuera de lo que considera su segmento local implementará ARP para la dirección de pasarela, y el comportamiento normal del enrutador devolverá la MAC de su puerto y entonces el anfitrión enviará sus paquetes de IP. Un anfitrión individual que desea enviar tráfico a un anfitrión que considera parte de su segmento local implementará ARP para ese anfitrión directamente. El enrutador responderá a ese ARP si tiene un puerto de anfitrión individual para esa dirección, en cuyo caso el anfitrión enviará entonces sus paquetes de IP. El comportamiento de anfitriones que no están en un puerto con indicador de anfitrión individual puede permanecer inalterado.

En otra realización del sistema, puede usarse el seguimiento de conexiones con estado para realizar el seguimiento del ciclo de vida de conexiones, de manera que datos asociados con esas conexiones pueden limpiarse tras determinados acontecimientos, tales como terminación de la conexión. Los datos que van a limpiarse pueden incluir diversos datos de estado de conexión datos, incluyendo datos almacenados en el estado distribuido, tales como mapeos de NAT y LB con estado, cuando la conexión se apaga de manera limpia. Si una conexión no se apaga de manera limpia, por ejemplo si uno u otro lado presenta un fallo grave o se desconecta, entonces el estado de conexión puede caducarse tras un tiempo asignado configurable grande. Las conexiones pueden ser conexiones TCP, y están compuestas por dos flujos, un flujo directo y un flujo de retorno. En el caso de TCP, el sistema puede simular la máquina de estado de conexión TCP con el fin de determinar el estado de conexión.

En aún otra realización, el sistema proporciona el flujo de retorno de una conexión que va a gestionarse por un nodo diferente del flujo directo de la conexión. Una conexión de este tipo puede denominarse flujo dividido, caracterizado porque el flujo directo e inverso se gestiona por un motor de decisión diferente. En una realización, el sistema soporta flujos divididos haciendo que el motor de decisión que ve los flujos directo e inverso comunique el cierre de sus lados respectivos. Por ejemplo, el motor de decisión que gestiona la FIN del flujo directo puede notificar al motor de decisión que gestiona el flujo de retorno que instale una acción que coincida con el ACK de la FIN, o viceversa. Los motores de decisión actúan conjuntamente de manera que pueden identificar cuando ambos lados de una conexión se han cerrado y pueden limpiar los datos asociados con la conexión cerrada. Esta comunicación entre los motores de decisión puede producirse a través del estado compartido en el sistema de estado distribuido. Adicionalmente, el sistema de estado distribuido puede identificar determinadas condiciones, tales como el cierre de ambos lados de una conexión, y puede comunicar notificaciones al motor de decisión que gestiona cada uno de los flujos de la comunicación.

En otra realización, cuando un nodo perimetral o conector de borde gestiona la configuración de un flujo, o bien directo o bien inverso, que forma parte de una conexión que debe seguirse (basándose en si es una conexión TCP, y si se necesita un seguimiento con estado, por ejemplo si la conexión está sometiéndose a NAT), el conector de borde añadirá una acción que comprueba el bit de FIN de TCP y emite el paquete FIN. Tras recibir un paquete FIN, el motor de decisión que gestiona el flujo inverso puede instalar una acción que comprueba el ACK de la FIN. Cuando el sistema observa el ACK de la FIN, se considera que la conexión está semiabierta, de manera que no se

espera ningún dato salvo los ACK. Si se reciben datos mediante una conexión semiabierta, el sistema puede generar un error que indica que el sistema experimentó una condición inesperada.

Cuando un motor de decisión recibe un nuevo flujo, instalará una regla que comprueba los indicadores RST y FIN de TCP. Si el sistema recibe un paquete RST, modifica la regla de flujo para que la conexión tenga un tiempo asignado corto, ya que la conexión va a terminarse una vez que el homólogo reciba el paquete RST. Si el sistema recibe un paquete FIN, inserta en la lista de acciones del flujo de retorno una acción que hace coincidir el número de secuencia de reconocimiento que es el número de secuencia del paquete FIN. Si el sistema obtiene un paquete que reconoce una FIN, marca ese lado de la conexión como cerrado. Si ambos lados están cerrados, modifica la regla de flujo para que la conexión tenga un tiempo asignado corto. En algunos casos, el ACK de la FIN puede soltarse, en cuyo caso el lado que se cierra retransmitirá el paquete FIN con el mismo número de secuencia. Cuando las reglas de flujo caducan, el sistema identifica que se cierra la conexión y puede limpiar datos de estado adicionales tales como seguimiento de NAT.

En otra realización del sistema y método dados a conocer en el presente documento, se proporciona un conmutador virtual como elemento de reenvío virtual adicional. El sistema puede transmitir paquetes L2 entre los puertos de los conectores de borde que forman parte de cada conmutador L2 virtual. De esta manera el sistema puede simular el funcionamiento de un conmutador L2 físico que transmite paquetes entre NIC conectadas al conmutador físico. El sistema también puede transmitir paquetes L3 de paquete tal como se describió anteriormente usando enrutadores virtuales. Cuando se configura un flujo, se identifica el UUID de vport entrante a partir del mapeo de una dirección MAC o puerto de entrada. Basándose en este UUID de vport, se determina el dispositivo virtual al que pertenece el vport. Basándose en el tipo de dispositivo virtual (conmutador o enrutador), el paquete o bien se enruta (tal como se describió anteriormente) o bien se conmuta. Es decir, si el paquete es un paquete L3, se gestiona según el proceso de enrutador virtual descrito anteriormente. Alternativamente, el paquete es un paquete L2 y se procesa mediante un conmutador virtual, tal como se ilustra en las figuras 5 y 6. El proceso ilustrado en las figuras 5 y 6 es sustancialmente similar al proceso ilustrado en la figura 10. Tras haberse determinado el VFE en la etapa 417, el conector de borde determina si el VFE es un enrutador virtual o un conmutador virtual. Si el VFE es un enrutador virtual, el procesamiento continúa tal como se describe con respecto a la figura 10. Si el VFE es un conmutador virtual, el procesamiento continúa en el punto A (520), conectado al punto A (520) en la figura 12. Tal como se ilustra en la figura 12, si el VFE es un conmutador virtual, entonces el conector de borde determina si la dirección MAC de destino es una dirección de radiodifusión o una dirección MAC de unidifusión (etapa 610). Si la dirección MAC es una dirección de radiodifusión, entonces se envía el paquete a cada puerto conectado al conmutador virtual (etapa 620). Basándose en cada paquete, esta etapa puede ser idéntica al proceso de la figura 10 comenzando con la etapa 426. Para cada puerto orientado hacia el exterior que es un miembro del VFE, el paquete se envía o bien al vport local o bien al puerto de túnel correspondiente a ese puerto orientado hacia el exterior.

Si el paquete no es un paquete de radiodifusión (por ejemplo un paquete de unidifusión), entonces se determina la MAC de destino, por ejemplo, consultando la MAC de destino en una tabla de MAC a vport (etapa 630). Si no hay una entrada correspondiente (sometido a prueba en la etapa 640), entonces se añade una acción de soltar a la lista de acciones (etapa 650). Entonces el procesamiento continúa en el punto B en la figura 11, en el que se añade la regla a la tabla de flujo (430) y se aplica la acción al paquete (418).

Si hay un vport correspondiente en la tabla de MAC a vport de la etapa 640, entonces el procesamiento continúa en el punto C en la figura 11, el procesamiento continúa en la etapa 426, tal como se describió anteriormente.

Haciendo ahora referencia a las figuras 7 y 8, se ilustra otra realización del sistema y método dados a conocer en el presente documento. Tal como se muestra en la figura 13, una topología de red virtual incluye un enrutador virtual de proveedor 900 que tiene múltiples conexiones 901 a una red externa, tal como Internet general 902. En esta configuración la red virtual está dotada de múltiples trayectorias de comunicación a la red externa permitiendo flexibilidad y redundancia en el sistema. El enrutador virtual de proveedor 900 puede tener una pluralidad de puertos orientados hacia el exterior correspondientes a una pluralidad de nodos perimetrales, en el que un nodo perimetral es un componente físico que proporciona acceso a la red externa. En una realización, un nodo perimetral puede ser un servidor o enrutador orientado hacia Internet. La topología de red virtual también puede comprender una pluralidad de enrutadores virtuales de titular. En una configuración, cada enrutador virtual de titular puede estar asociado con un centro de datos virtual de titular. Tal como se muestra en la figura 13, un centro de datos virtual de primer titular 903 puede incluir un enrutador virtual de primer titular 904 en comunicación con una pluralidad de máquinas virtuales de primer titular 905. Las máquinas virtuales de primer titular 905 también pueden comunicarse con un conmutador virtual de titular 906, que puede ser un conmutador de Ethernet virtual tal como se ilustra. Las máquinas virtuales de primer titular 905 pueden residir en uno o más de un servidor o nodo anfitrión en la red.

Tal como también se muestra en la figura 13, la topología de red virtual puede tener un centro de datos virtual de segundo titular 907, que incluye un enrutador virtual de segundo titular 910 en comunicación con el enrutador virtual de proveedor 900 y una pluralidad de máquinas virtuales de segundo titular 909. La pluralidad de máquinas virtuales de segundo titular 909 también pueden comunicarse con un conmutador virtual de segundo titular 908, que puede ser un conmutador de Ethernet virtual tal como se ilustra.

Los enrutadores virtuales también pueden realizar funciones adicionales tales como equilibrado de carga, DHCP y/o

traducción de dirección de red según desee cada titular. Aunque sólo se ilustra un enrutador virtual para cada titular, en otras realizaciones, un titular puede emplear una pluralidad de enrutadores virtuales creando una topología de red virtual específica de titular. Una topología de red virtual específica de titular puede proporcionar organización de máquinas virtuales de titular en disposiciones deseadas o proporcionar aislamiento entre máquinas virtuales controladas por el mismo titular, tal como cuando un titular está usando la red para albergar múltiples procesos empresariales o funciones diferenciadas.

En otra realización, un enrutador virtual de titular puede proporcionar acceso seguro a una oficina de titular remota u otra ubicación. Tal como se ilustra, el enrutador virtual de segundo titular 910 proporciona una conexión al enrutador de VPN de segundo titular 911 y la red de oficina de segundo titular 912 en la oficina de segundo titular 913. De esta manera, cada titular puede definir la configuración de su centro de datos virtual. Por tanto, un proveedor de servicios que usa el sistema y método dados a conocer en el presente documento puede proporcionar muchas soluciones personalizables por el titular en una red física.

Haciendo ahora referencia a la figura 14, la topología de red virtual ilustrada en la figura 13 se muestra superpuesta en una red física. La red física puede comprender una pluralidad de nodos perimetrales 920 configurados para acceder a una red externa, tal como Internet 902. La red física también puede incluir una pluralidad de nodos anfitriones 921 configurados para albergar máquinas virtuales. La red 922 puede interconectar la pluralidad de nodos perimetrales 920 y la pluralidad de nodos anfitriones 921 y estar adaptada para transportar paquetes de datos a través del sistema. En una realización, la red puede ser una red de IP privada. Los nodos perimetrales 920 y los nodos anfitriones 921 pueden tener arquitecturas simétricas. En una realización, los nodos perimetrales 920 y los nodos anfitriones 921 son servidores de uso general configurados para funcionar en un sistema informático en la nube. En otra realización, los nodos perimetrales 920 son enrutadores orientados hacia Internet dedicados. En aún otra realización, un servidor u otro dispositivo informático puede funcionar como nodo perimetral y como nodo anfitrión en la misma red. El sistema también incluye un sistema de estado distribuido en comunicación con cada uno de los nodos perimetrales y cada uno de los nodos anfitriones a través de la red. El sistema de estado distribuido puede almacenar datos asociados con la topología de red virtual y puede almacenarse en una base de datos compartida. El sistema puede incluir un componente de software que funciona en cada uno de los nodos y que implementa la topología de red virtual que incluye el enrutador virtual de proveedor y cada uno de los enrutadores virtuales de titular. A medida que se configuran nuevas rutas, el componente de software que funciona en cada uno de los nodos puede comunicarse con el sistema de estado distribuido de manera que el estado distribuido mantiene un mapeo exhaustivo de la topología de red virtual y reglas de flujo para el sistema. En otros ejemplos, el sistema de estado distribuido puede subdividirse de manera que se mantienen múltiples estados distribuidos para porciones seleccionadas de la red virtual.

Tal como se ilustra en la figura 14, la topología de red virtual está superpuesta en la red física. El enrutador virtual de proveedor puede tener un puerto orientado hacia el exterior asociado con cada uno de los nodos perimetrales 920. Los puertos orientados hacia el exterior del enrutador virtual de proveedor 900 pueden mapearse a uno o más puntos de acceso para proveedores de servicio de Internet y proporcionar múltiples conexiones entre el sistema y una red externa, tal como Internet. El enrutador virtual de proveedor 900 también puede tener puertos interiores que definen enlaces virtuales a puertos orientados hacia el interior homólogos correspondientes de enrutadores virtuales de titular. Tal como se ilustra, puede seleccionarse el rendimiento de cada enlace virtual en el sistema. Por ejemplo, el proveedor de servicios puede proporcionar un enlace virtual de 50 Mbps al enrutador virtual de primer titular 904, pero proporcionar un enlace virtual de 10 Mbps al enrutador virtual de segundo titular 910. Como los enlaces virtuales son configurables, si el segundo titular desea adquirir un mayor rendimiento para su centro de datos virtual, el proveedor de servicios puede modificar el rendimiento disponible sin modificar el hardware.

En la realización ilustrada, cada nodo anfitrión 920 alberga una máquina virtual asociada con el primer titular y una máquina virtual asociada con el segundo titular. Usando la topología de red virtual, el proveedor de servicios puede reasignar máquinas virtuales de titular entre nodos anfitriones disponibles sin reconfigurar el hardware de la red física. La topología de red virtual almacenada en el sistema de estado distribuido permite reconfigurar dinámicamente el sistema.

En otra realización, cada uno de la pluralidad de enrutadores virtuales de titular puede configurarse para exponer al menos una dirección IP pública y puede configurarse para acceder a una red externa a través de uno o más de la pluralidad de nodos perimetrales. Permitiendo que cada centro de datos virtual de titular acceda a la red externa a través de una pluralidad de nodos perimetrales, es menos probable que el fallo de un único nodo perimetral interrumpa la disponibilidad de los servicios del titular que funcionan en la red.

Tal como se usan en el presente documento, los términos “caché”, “copiar en memoria caché” u otras variaciones se refieren a todas las formas de almacenamiento de datos temporal independientemente de si los datos se almacenan en memoria explícitamente designada como caché.

REIVINDICACIONES

1. Un sistema informático que comprende:

una pluralidad de nodos (54, 73, 80, 210, 221, 222) interconectados mediante una red subyacente (53, 79, 202), en el que cada nodo incluye una o más interfaces de red (55, 74, 75, 81), estando conectada la una o más interfaces de cada nodo a la red subyacente,

una pluralidad de dispositivos virtuales (211, 212, 213, 214) que se simulan en al menos uno de la pluralidad de nodos, teniendo cada dispositivo virtual una pluralidad de puertos virtuales, en el que cada puerto virtual corresponde a uno de un puerto orientado hacia el exterior o un puerto orientado hacia el interior, correspondiendo cada puerto orientado hacia el exterior a una de las interfaces de red de los nodos de la red subyacente, teniendo cada puerto orientado hacia el interior un enlace virtual con otro puerto orientado hacia el interior de los dispositivos de red virtuales y

una base de datos compartida (61, 86) que almacena una pluralidad de reglas de flujo y una topología de red virtual que incluye una configuración de los puertos virtuales y los dispositivos virtuales, incluyendo la configuración, para cada uno de los puertos virtuales, una identificación del puerto virtual como una de un puerto orientado hacia el exterior o un puerto orientado hacia el interior, siendo la base de datos compartida accesible por la pluralidad de nodos,

estando caracterizado el sistema informático por que comprende además

un motor de decisión (78, 84, 165) operable para:

simular un recorrido de un paquete de red de la topología de red virtual desde un puerto orientado hacia el exterior de entrada de un primer dispositivo virtual hasta un puerto orientado hacia el exterior de salida de un último dispositivo virtual, en el que el puerto orientado hacia el exterior de salida del último dispositivo virtual corresponde a una interfaz de red en la que el paquete de red se emitirá desde la red subyacente,

determinar un patrón de cabecera de paquete identificando cada campo de la cabecera de protocolo del paquete que se lee durante la simulación del recorrido, en el que el patrón de cabecera de protocolo de paquete incluye un comodín para cualquier campo de la cabecera de protocolo del paquete que no se lee durante la simulación del recorrido,

determinar una modificación de cabecera de protocolo que va a aplicarse al paquete de red basándose en una configuración de cada dispositivo virtual recorrido por el paquete durante la simulación del recorrido,

comunicar el patrón de cabecera de protocolo de paquete y la modificación de cabecera de protocolo determinada a la base de datos compartida,

almacenar el patrón de cabecera de protocolo de paquete y la modificación de cabecera de protocolo determinada como una regla de flujo en la base de datos compartida;

tras recibir un paquete posterior, seleccionar una regla de flujo desde la base de datos compartida haciendo coincidir una cabecera del paquete posterior con el patrón de cabecera de protocolo de paquete almacenado, y

aplicar la modificación de cabecera de protocolo determinada de la regla de flujo seleccionada a paquetes posteriores que coinciden con el patrón de cabecera de protocolo de paquete de manera que el paquete se entrega al segundo nodo de la red subyacente y se emite desde la segunda interfaz de red,

estando determinada la simulación del recorrido de la topología de red virtual en parte consultando la base de datos compartida usando un campo de la cabecera de protocolo del paquete de red.

2. El sistema informático según la reivindicación 1, en el que el motor de decisión es operable adicionalmente para simular un puente de aprendizaje de MAC que tiene una tabla de aprendizaje de MAC y actualizar la tabla de aprendizaje de MAC del puente de aprendizaje de MAC en la base de datos compartida.

3. El sistema informático según la reivindicación 1 ó 2, en el que el motor de decisión es operable adicionalmente para identificar paquetes de IP que están encapsulados en una trama de Ethernet de encapsulamiento y extraer una dirección IP de origen y una dirección MAC de origen de la trama de Ethernet de encapsulamiento, y actualizar una memoria caché de ARP de un puente virtual para asociar la dirección IP y la dirección MAC identificadas.

4. El sistema informático según una de las reivindicaciones 1 a 3, en el que el motor de decisión está configurado adicionalmente para simular al menos uno de un filtro de entrada y un filtro de salida de al menos un dispositivo virtual, en el que cada filtro está adaptado para someter a prueba los paquetes para una condición especificada e incluye reglas de filtrado que van a aplicarse a paquetes que coinciden con la

condición especificada.

5. El sistema informático según una de las reivindicaciones 1 a 4, en el que el motor de decisión está configurado adicionalmente para simular la aplicación de una traducción de dirección de red a paquetes recibidos por al menos uno de los dispositivos virtuales y almacenar la traducción de dirección de red para paquetes de tanto un flujo directo como un flujo inverso en la base de datos compartida.

6. El sistema informático según una de las reivindicaciones 1 a 5, en el que el motor de decisión está configurado además para asignar dinámicamente una dirección IP desde un recurso de DHCP hasta una dirección MAC de un remitente en respuesta a una petición de dicho remitente recibida en un puerto orientado hacia el exterior y almacenar la dirección IP asignada en la base de datos compartida.

7. Un método informático que comprende:

recibir un paquete de red que llega en una primera interfaz de red de un primer nodo de una red subyacente (53, 79, 202), comprendiendo la red subyacente una pluralidad de nodos interconectados (54, 73, 80, 210, 221, 222);

comunicar al menos el paquete y un identificador de la primera interfaz de red a un motor de decisión;

determinar cómo debe procesarse el paquete basándose en una simulación por el motor de decisión (78, 84, 165) de un recorrido de una topología de red virtual que incluye una pluralidad de dispositivos de red virtuales, en el que el motor de decisión se comunica con una base de datos compartida (61, 86) accesible desde la pluralidad de nodos interconectados, almacenando la base de datos compartida una pluralidad de reglas de flujo y la topología de red virtual y configuraciones de dispositivo virtual para la pluralidad de dispositivos de red virtuales, teniendo cada dispositivo virtual una pluralidad de puertos virtuales, en el que cada puerto virtual corresponde a uno de un puerto orientado hacia el exterior o un puerto orientado hacia el interior,

estando caracterizado el método informático por que

cada puerto orientado hacia el exterior corresponde a una de las interfaces de red de los nodos de la red subyacente,

por que cada puerto orientado hacia el interior tiene un enlace virtual con otro puerto orientado hacia el interior de los dispositivos de red virtuales;

por que la determinación de cómo debe procesarse el paquete incluye determinar que el paquete debe emitirse desde una segunda interfaz de red de un segundo nodo de la red subyacente, y determinar cómo las cabeceras de protocolo de los paquetes deben modificarse antes de la entrega al segundo nodo de la red subyacente,

y por que el método informático comprende además:

crear un patrón de cabecera de protocolo de paquete identificando cada campo de la cabecera de paquete que se lee durante la simulación del recorrido, en el que el patrón de cabecera de protocolo de paquete incluye un comodín para cualquier campo de la cabecera de protocolo del paquete que no se lee durante la simulación del recorrido,

determinar una modificación de cabecera de protocolo que va a aplicarse al paquete de red basándose en una configuración de cada dispositivo virtual recorrido por el paquete durante la simulación del recorrido, y

comunicar el patrón de cabecera de protocolo de paquete y la modificación de cabecera de protocolo determinada a la base de datos compartida,

almacenar el patrón de cabecera de protocolo de paquete y la modificación de cabecera de protocolo determinada como una regla de flujo en la base de datos compartida; y

procesar el paquete basándose en la simulación, en el que procesar el paquete incluye entregar el paquete al segundo nodo de la red subyacente y emitir el paquete desde la segunda interfaz de red;

tras recibir paquete posterior, seleccionar una regla de flujo desde la base de datos compartida haciendo coincidir una cabecera del paquete posterior con el patrón de cabecera de protocolo de paquete almacenado, y

aplicar la modificación de cabecera de protocolo determinada de la regla de flujo seleccionada al paquete posterior que coincide con el patrón de cabecera de protocolo de paquete de manera que el paquete se entrega al segundo nodo de la red subyacente y se emite desde la segunda interfaz de red.

8. El método informático según la reivindicación 7, en el que cada nodo de la red subyacente está configurado para hacer funcionar el motor de decisión para realizar la simulación para determinar cómo debe procesarse un paquete que llega al nodo.

9. El método informático según la reivindicación 7 u 8, en el que la determinación de cómo debe procesarse el paquete incluye determinar un patrón de cabecera de protocolo de paquete a partir de un resultado de la simulación por el motor de decisión para el paquete, comprendiendo además el método:

almacenar el patrón de cabecera de protocolo de paquete y el resultado de la simulación para el paquete,

recibir un segundo paquete en el primer nodo de la red subyacente,

comparar el segundo paquete con el patrón de cabecera de protocolo de paquete,

si el segundo paquete coincide con el patrón de cabecera de protocolo de paquete, determinar cómo debe procesarse el segundo paquete recuperando el resultado almacenado de la simulación para el paquete, y

procesar el segundo paquete basándose en el resultado recuperado de la simulación por el motor de decisión para el paquete.

10. El método informático según una de las reivindicaciones 7 a 9

en el que la determinación de cómo debe procesarse el paquete incluye determinar que el paquete debe emitirse desde una segunda interfaz de red del primer nodo de la red subyacente, y

en el que el procesamiento del paquete incluye emitir el paquete desde la segunda interfaz de red.

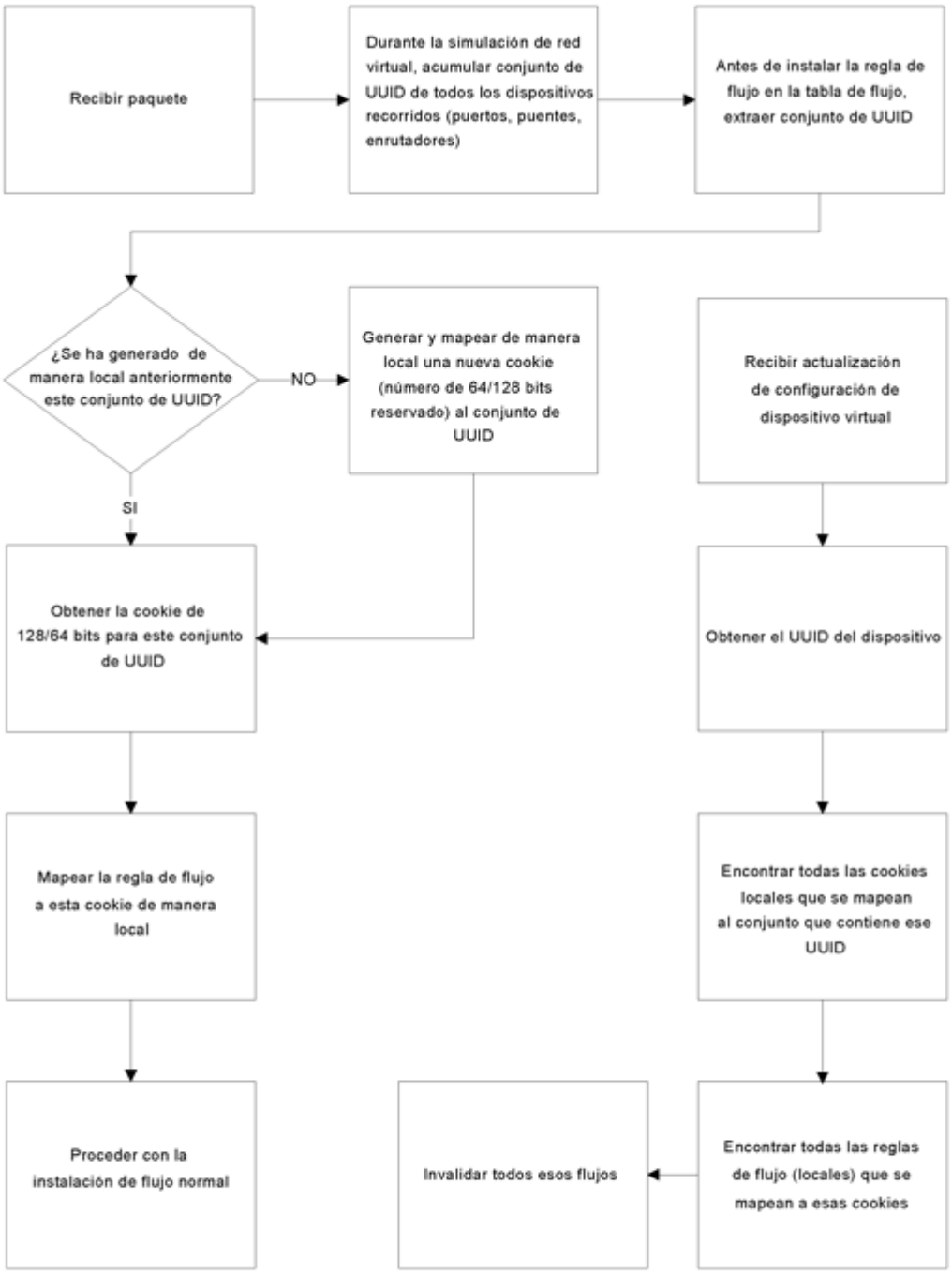
Recibir paquete
Durante la simulación de red virtual, acumular conjunto de UUID de todos los dispositivos recorridos (puertos, puentes, enrutadores)
Antes de instalar la regla de flujo en la tabla de flujo, extraer conjunto de UUID
¿Se ha generado de manera local anteriormente este conjunto de UUID?
NO
Generar y mapear de manera local una nueva cookie (número de 64/128 bits reservado) al conjunto de UUID
Recibir actualización de configuración de dispositivo virtual
SI
Obtener la cookie de 128/64 bits para este conjunto de UUID
Obtener el UUID del dispositivo
Mapear la regla de flujo a esta cookie de manera local
Encontrar todas las cookies locales que se mapean al conjunto que contiene ese UUID
Proceder con la instalación de flujo normal
Invalidar todos esos flujos
Encontrar todas las reglas de flujo (locales) que se mapean a esas cookies

FIG. 1

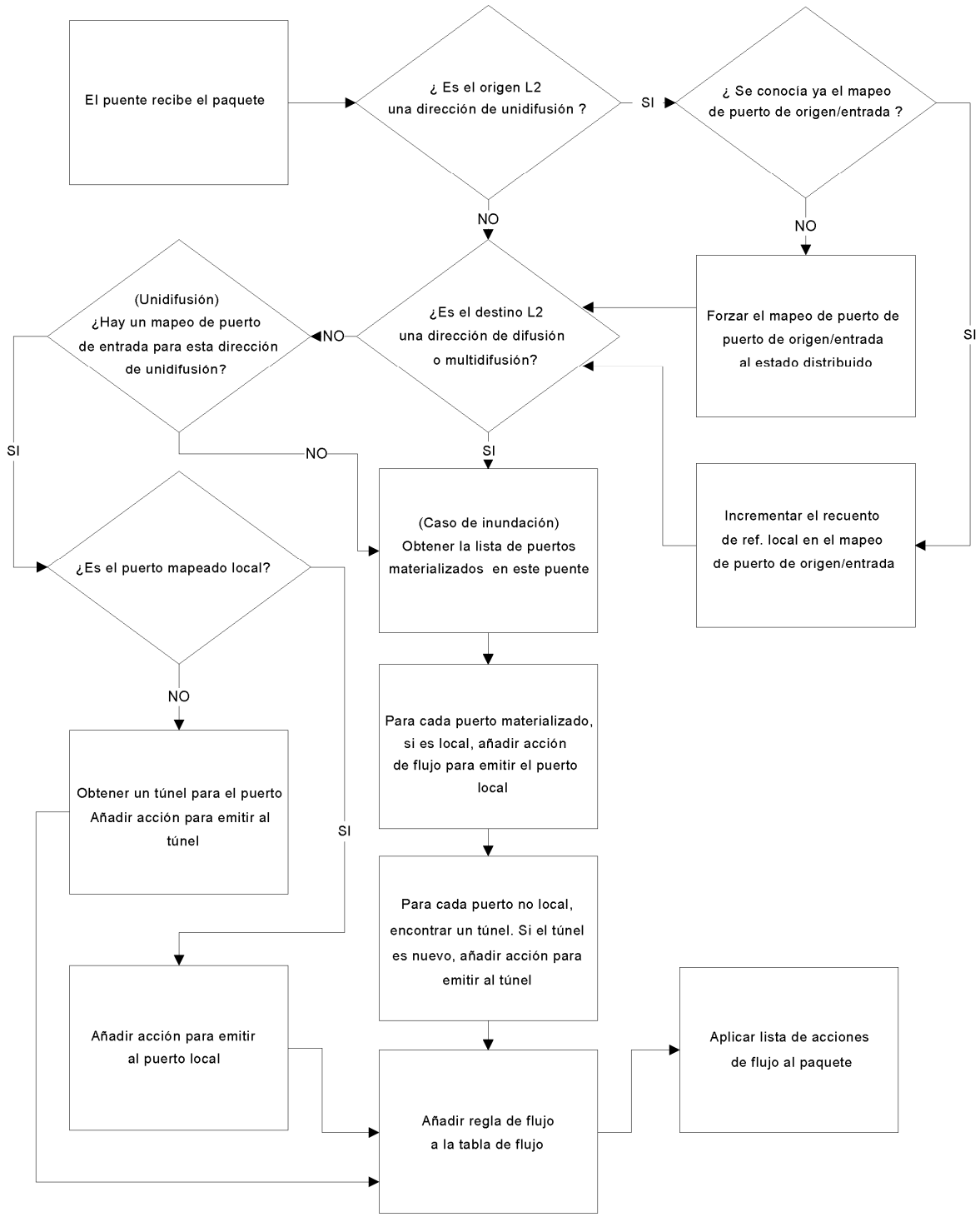
El puente recibe el paquete
¿ Es el origen L2 una dirección de unidifusión ?
SI
¿ Se conocía ya el mapeo de puerto de origen/entrada ?
NO
SI
NO
Forzar el mapeo de puerto de puerto de origen/entrada al estado distribuido
¿Es el destino L2 una dirección de difusión o multidifusión?
NO
(Unidifusión) ¿Hay un mapeo de puerto de entrada para esta dirección de unidifusión?
SI
NO
SI
Incrementar el recuento de ref. local en el mapeo de puerto de origen/entrada
(Caso de inundación) Obtener la lista de puertos materializados en este puente
¿Es el puerto mapeado local?
NO
Para cada puerto materializado, si es local, añadir acción de flujo para emitir el puerto local
Obtener un túnel para el puerto Añadir acción para emitir al túnel
SI
Para cada puerto no local, encontrar un túnel. Si el túnel es nuevo, añadir acción para emitir al túnel
Añadir acción para emitir al puerto local
Añadir regla de flujo a la tabla de flujo
Aplicar lista de acciones de flujo al paquete

FIG. 2


El flujo caduca
¿ Es el origen L2 una dirección de unidifusión ?
NO
Terminado. No hacer nada
SI
Reducir el recuento de ref. local en el mapeo de dirección/puerto de entrada
NO
¿Es el recuento de ref.cero?
SI
Retirar el mapeo de dirección/puerto de entrada del estado distribuido

FIG. 3

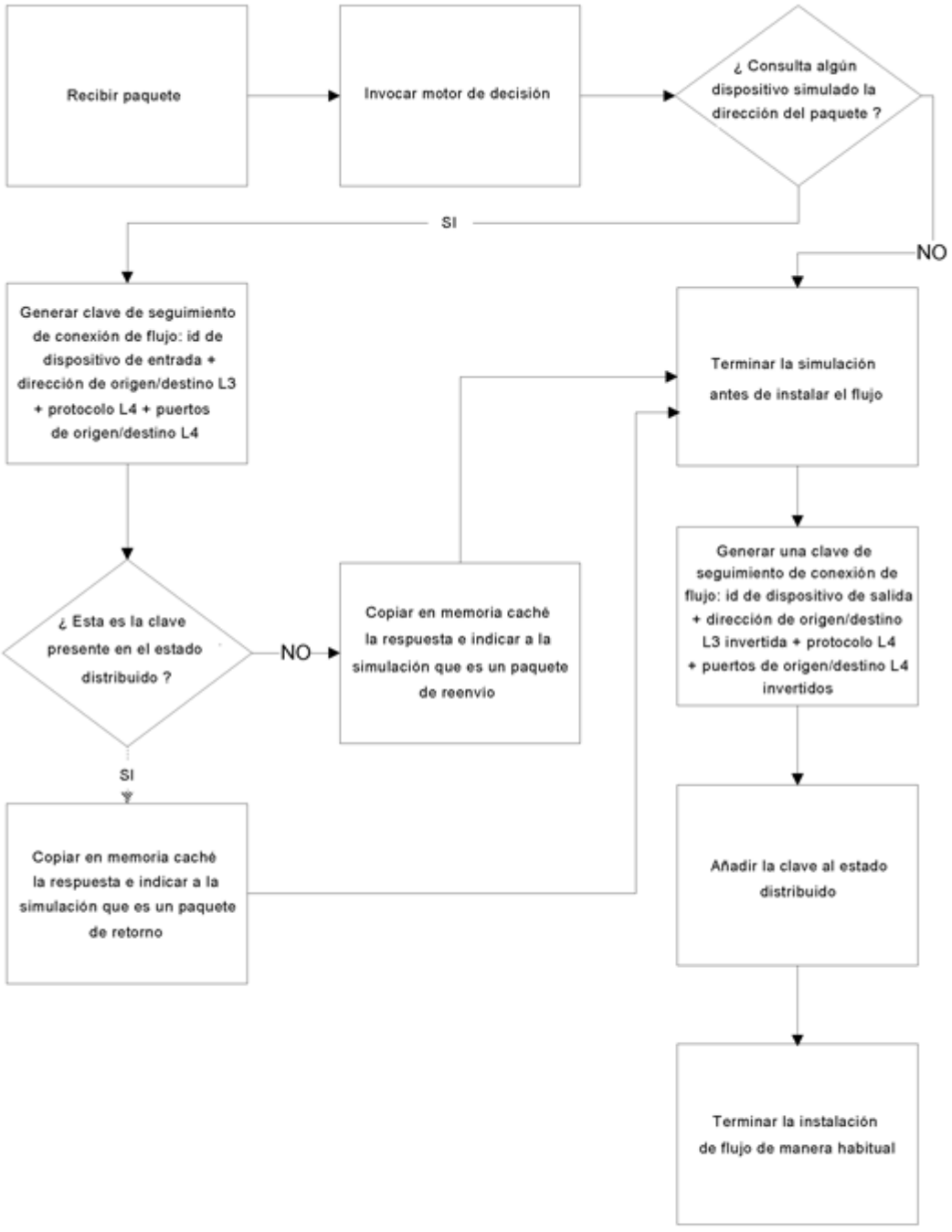
Recibir paquete
Invocar motor de decisión
¿ Consulta algún dispositivo simulado la dirección del paquete ?
SI
NO
Generar clave de seguimiento de conexión de flujo: id de dispositivo de entrada + dirección de origen/destino L3 + protocolo L4 + puertos de origen/destino L4
Terminar la simulación antes de instalar el flujo
¿ Esta es la clave presente en el estado distribuido ?
NO
Copiar en memoria caché la respuesta e indicar a la simulación que es un paquete de reenvio
Generar una clave de seguimiento de conexión de flujo: id de dispositivo de salida + dirección de origen/destino L3 invertida + protocolo L4 + puertos de origen/destino L4 invertidos
SI
Copiar en memoria caché la respuesta e indicar a la simulación que es un paquete de retorno
Añadir la clave al estado distribuido
Terminar la instalación de flujo de manera habitual

FIG. 4

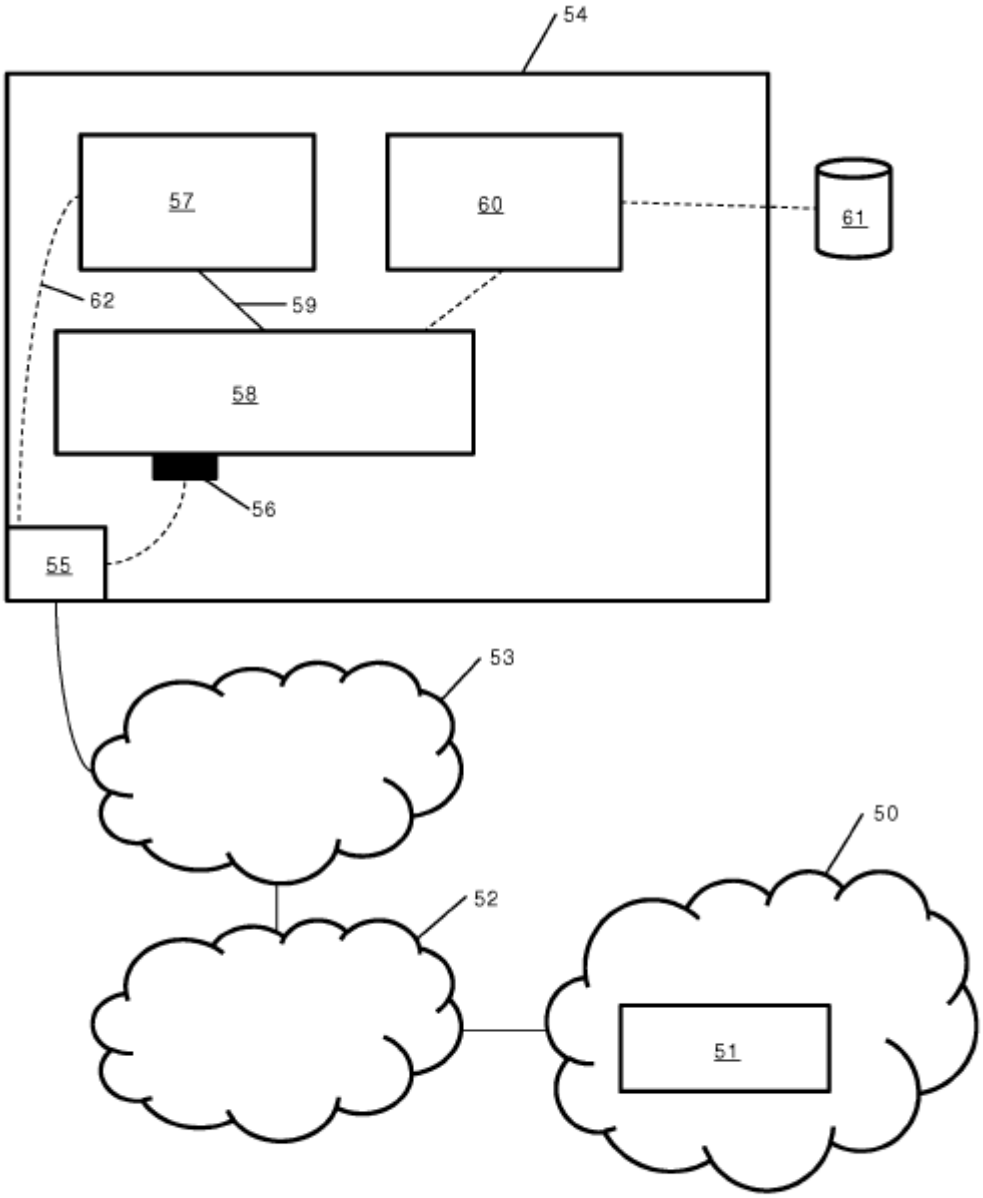

FIG. 5

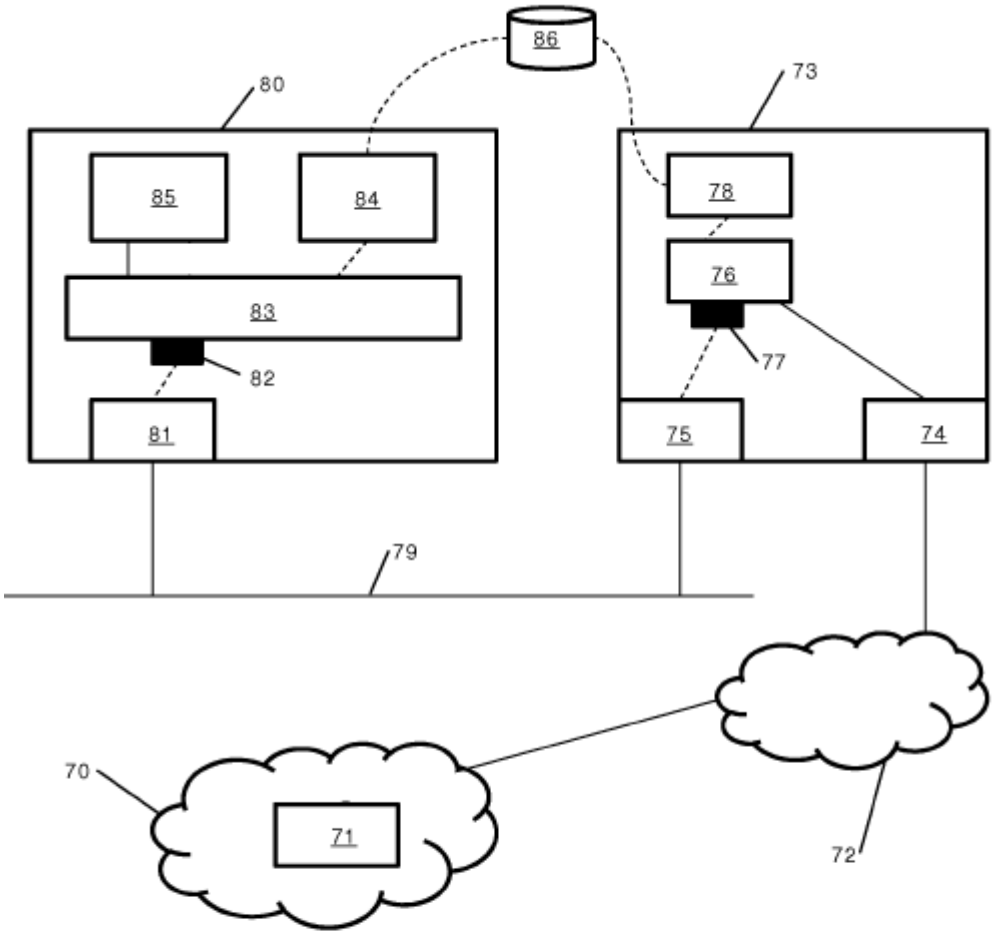

FIG. 6

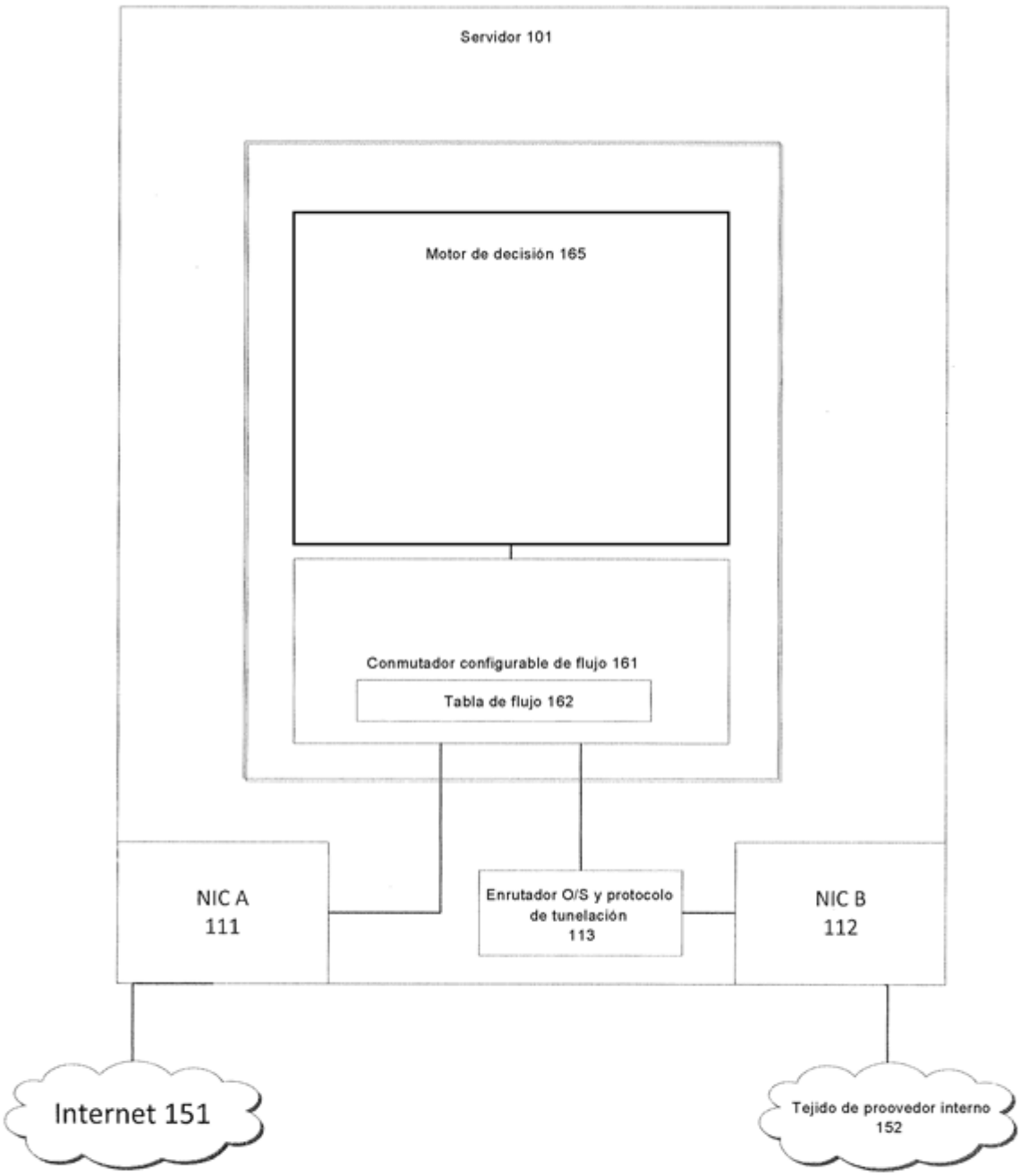
Servidor 101
Motor de decisión 165
Conmutador configurable de flujo 161
Tabla de flujo 162
NIC A
111
Enrutador O/S y protocolo
de tunelación
113
NIC B
112
Internet 151
Tejido de proovedor interno
152

FIG. 7

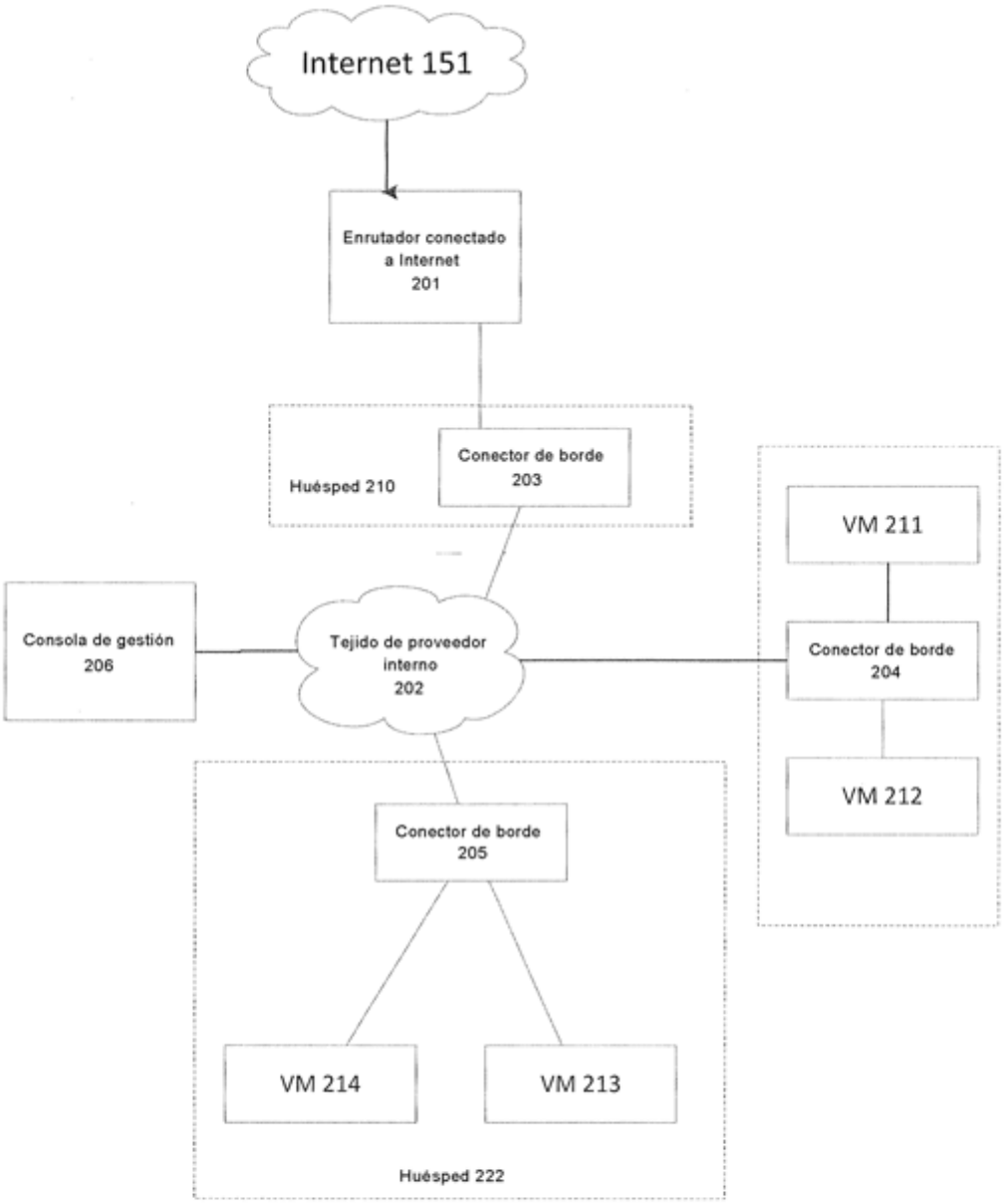
Internet 151
Enrutador conectado a Internet 201
Huésped 210
Conector de borde 203
VM 211
Consola de gestión 206
Tejido de proveedor interno 202
Conector de borde 204
VM 212
Conector de borde 205
VM 214
VM 213
Huésped 222

FIG. 8

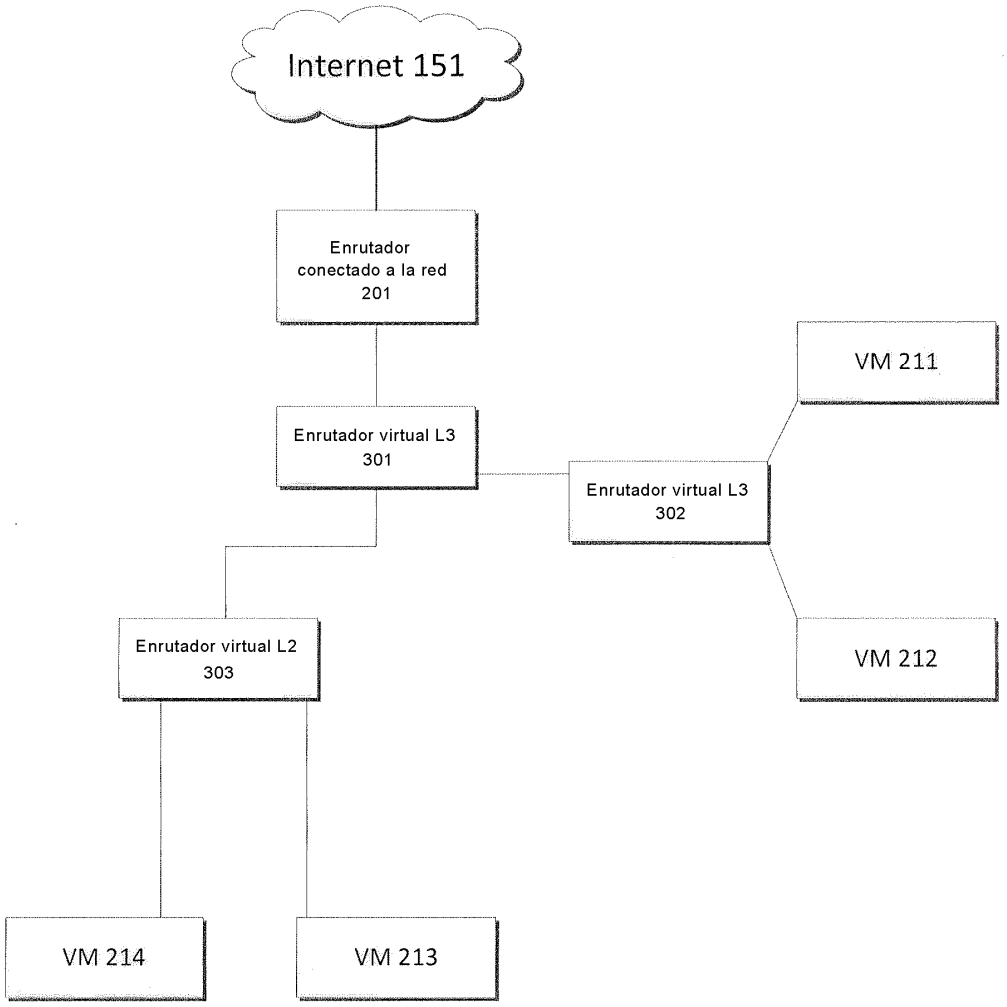
Internet 151
Enrutador conectado a la red 201
Enrutador virtual L3 301
Enrutador virtual L3 302
VM 211
Enrutador virtual L2 303
VM 212
VM 214
VM 213

FIG. 9

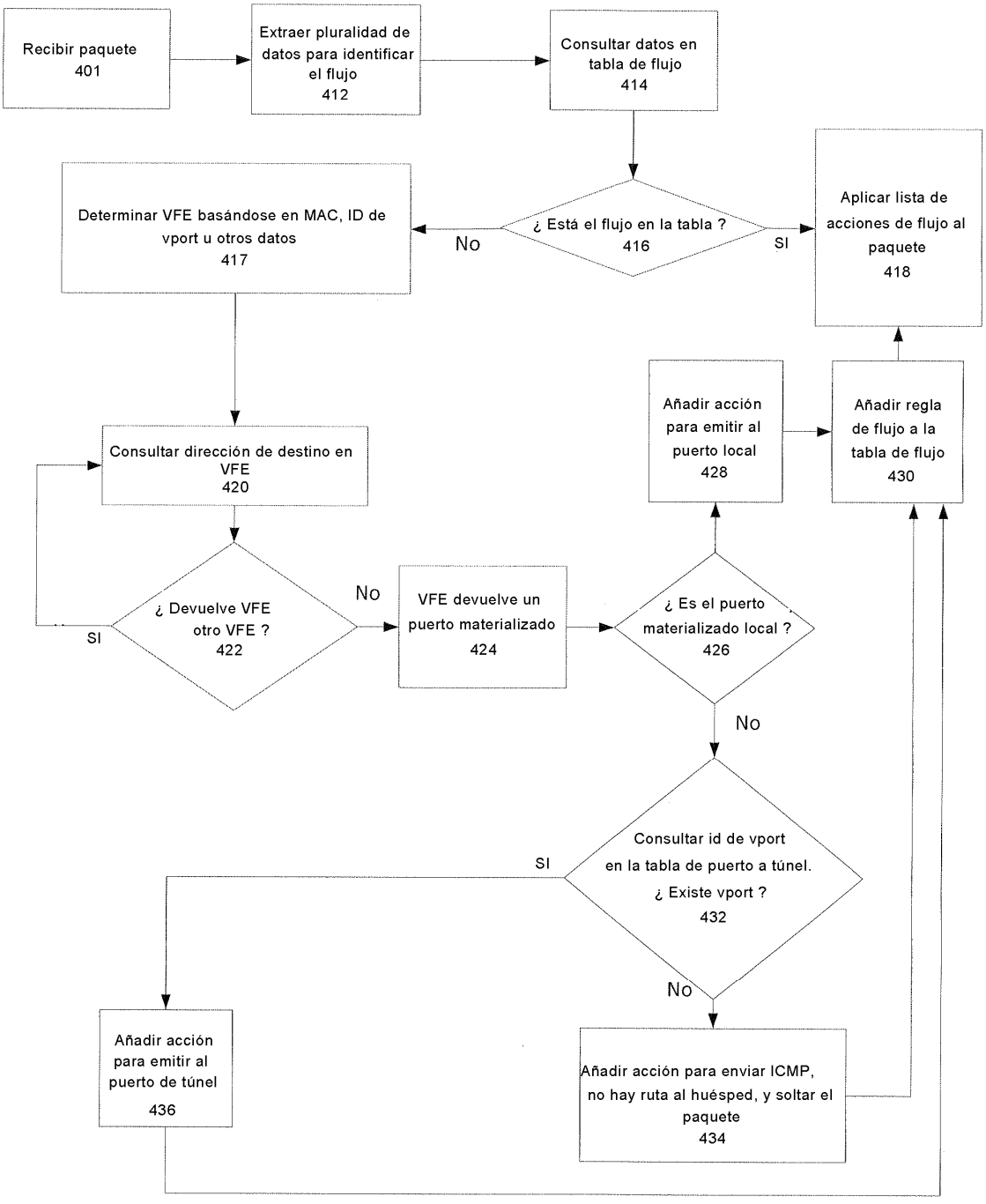
Recibir paquete
401
Extraer pluralidad de datos para identificar el flujo
412
Consultar datos en tabla de flujo
414
¿ Está el flujo en la tabla ?
416
No
Sí
Aplicar lista de acciones de flujo al paquete
418
Determinar VFE basándose en MAC, ID de vport u otros datos
417
Consultar dirección de destino en VFE
420
¿ Devuelve VFE otro VFE ?
422
No
Sí
VFE devuelve un puerto materializado
424
¿ Es el puerto materializado local ?
426
Añadir acción para emitir al puerto local
428
Añadir regla de flujo a la tabla de flujo
430
No
Consultar id de vport en la tabla de puerto a túnel. ¿ Existe vport ?
432
Sí
No
Añadir acción para emitir al puerto de túnel
436
Añadir acción para enviar ICMP, no hay ruta al huésped, y soltar el paquete
434

FIG. 10

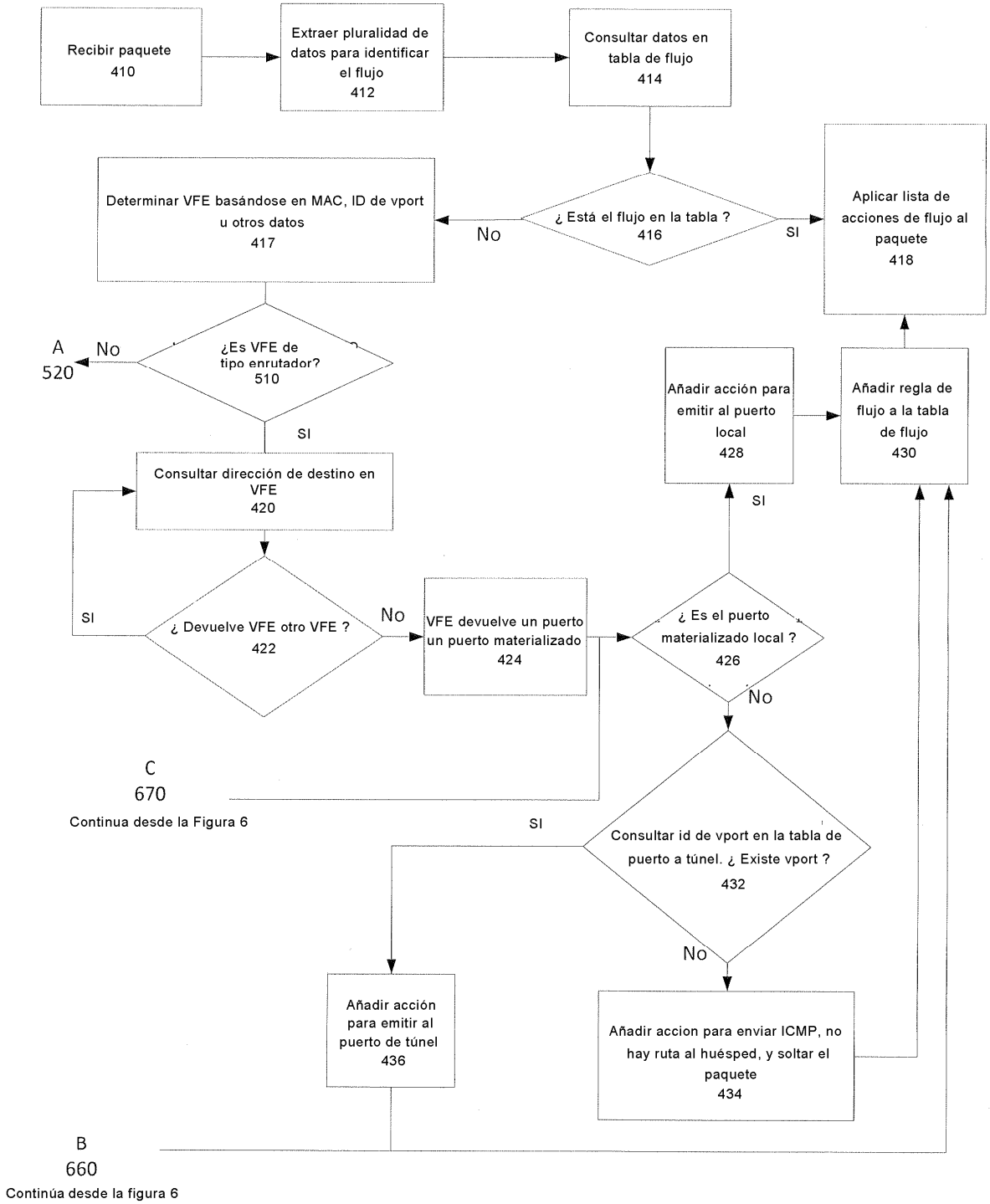
Recibir paquete
410
Extraer pluralidad de datos para identificar el flujo
412
Consultar datos en tabla de flujo
414
¿ Está el flujo en la tabla ?
416
No
Sí
Aplicar lista de acciones de flujo al paquete
418
Determinar VFE basándose en MAC, ID de vport u otros datos
417
A
520
No
¿Es VFE de tipo enrutador?
510
Sí
Consultar dirección de destino en VFE
420
Sí
¿ Devuelve VFE otro VFE ?
422
No
VFE devuelve un puerto un puerto materializado
424
¿ Es el puerto materializado local ?
426
Sí
Añadir acción para emitir al puerto local
428
Añadir regla de flujo a la tabla de flujo
430
No
C
670
Continua desde la Figura 6
Sí
Consultar id de vport en la tabla de puerto a túnel. ¿ Existe vport ?
432
No
Añadir acción para emitir al puerto de túnel
436
Añadir accion para enviar ICMP, no hay ruta al huésped, y soltar el paquete
434
B
660
Continúa desde la figura 6

FIG. 11

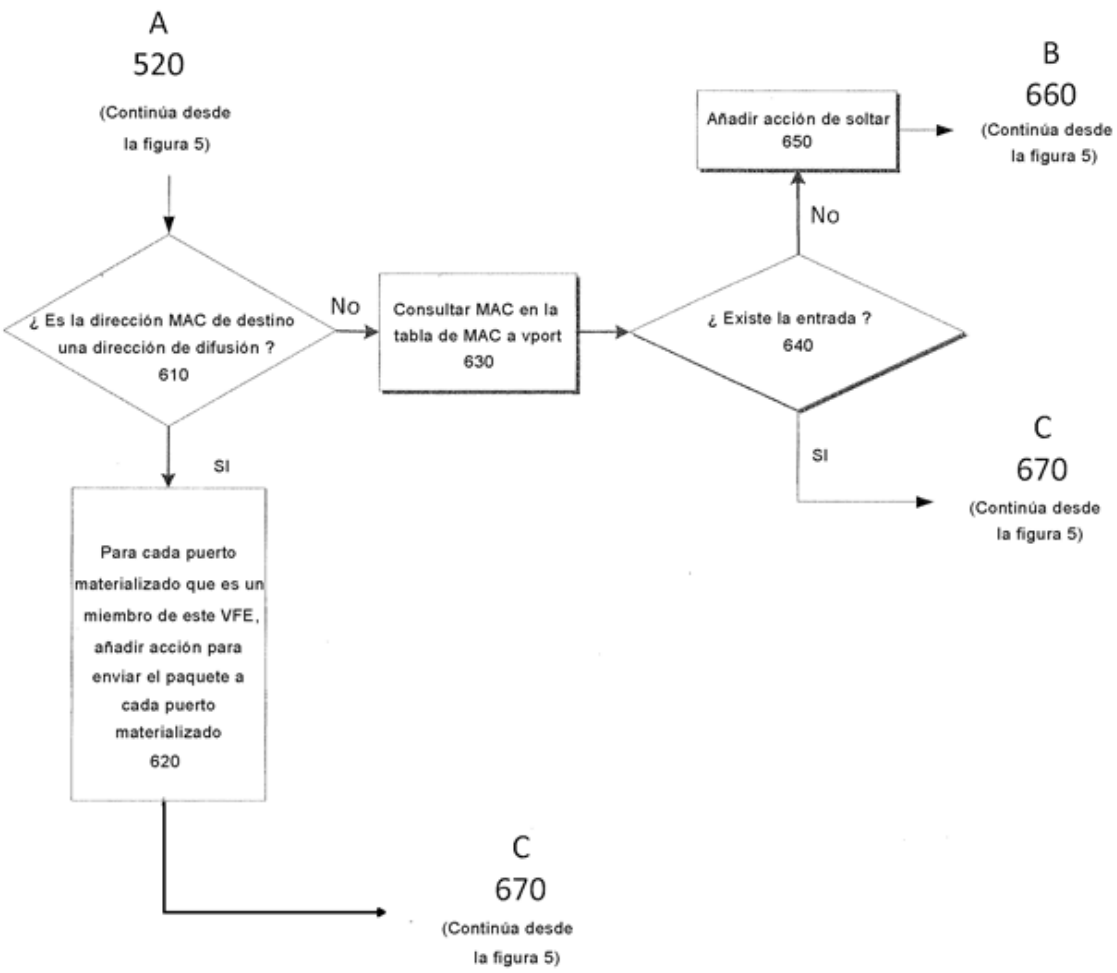
A
520
(Continúa desde la figura 5)
¿ Es la dirección MAC de destino una dirección de difusión ?
610
No
Consultar MAC en la tabla de MAC a vport
630
¿ Existe la entrada ?
640
No
Añadir acción de soltar
650
B
660
(Continúa desde la figura 5)
SI
C
670
(Continúa desde la figura 5)
SI
Para cada puerto materializado que es un miembro de este VFE, añadir acción para enviar el paquete a cada puerto materializado
620
C
670
(Continúa desde la figura 5)

FIG. 12

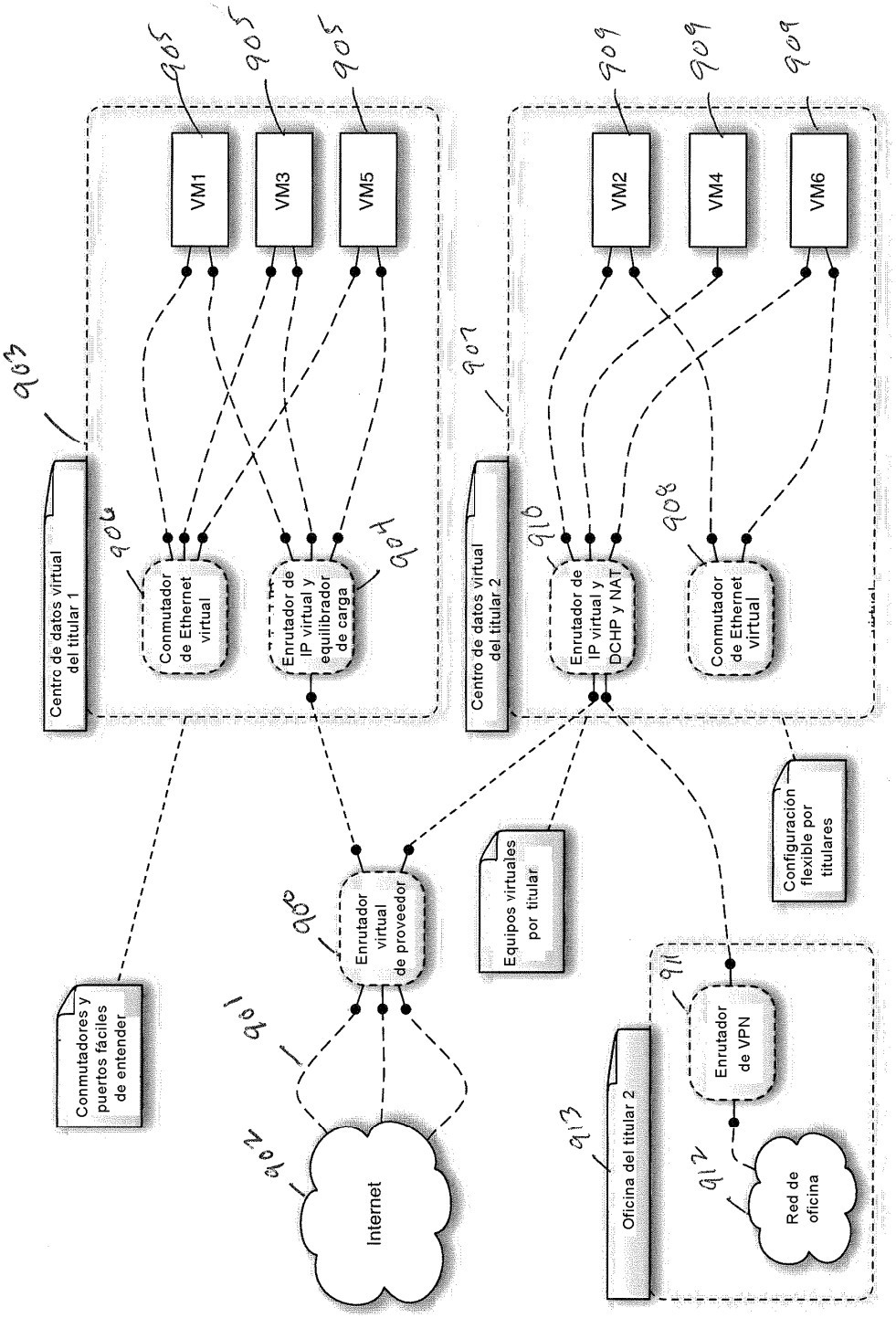
903
905
905
905
VM1
VM3
VM5
907
909
909
909
VM2
VM4
VM6
Centro de datos virtual del titular 1
906
Conmutador de Ethernet virtual
904
Enrutador de IP virtual y equilibrador de carga
Centro de datos virtual del titular 2
910
Enrutador de IP virtual y DCHP y NAT
908
Conmutador de Ethernet virtual
Conmutadores y puertos fáciles de entender
902
Enrutador virtual de proveedor
Equipos virtuales por titular
Configuración flexible por titulares
901
Internet
911
Enrutador de VPN
913
Oficina del titular 2
912
Red de oficina

FIG. 13

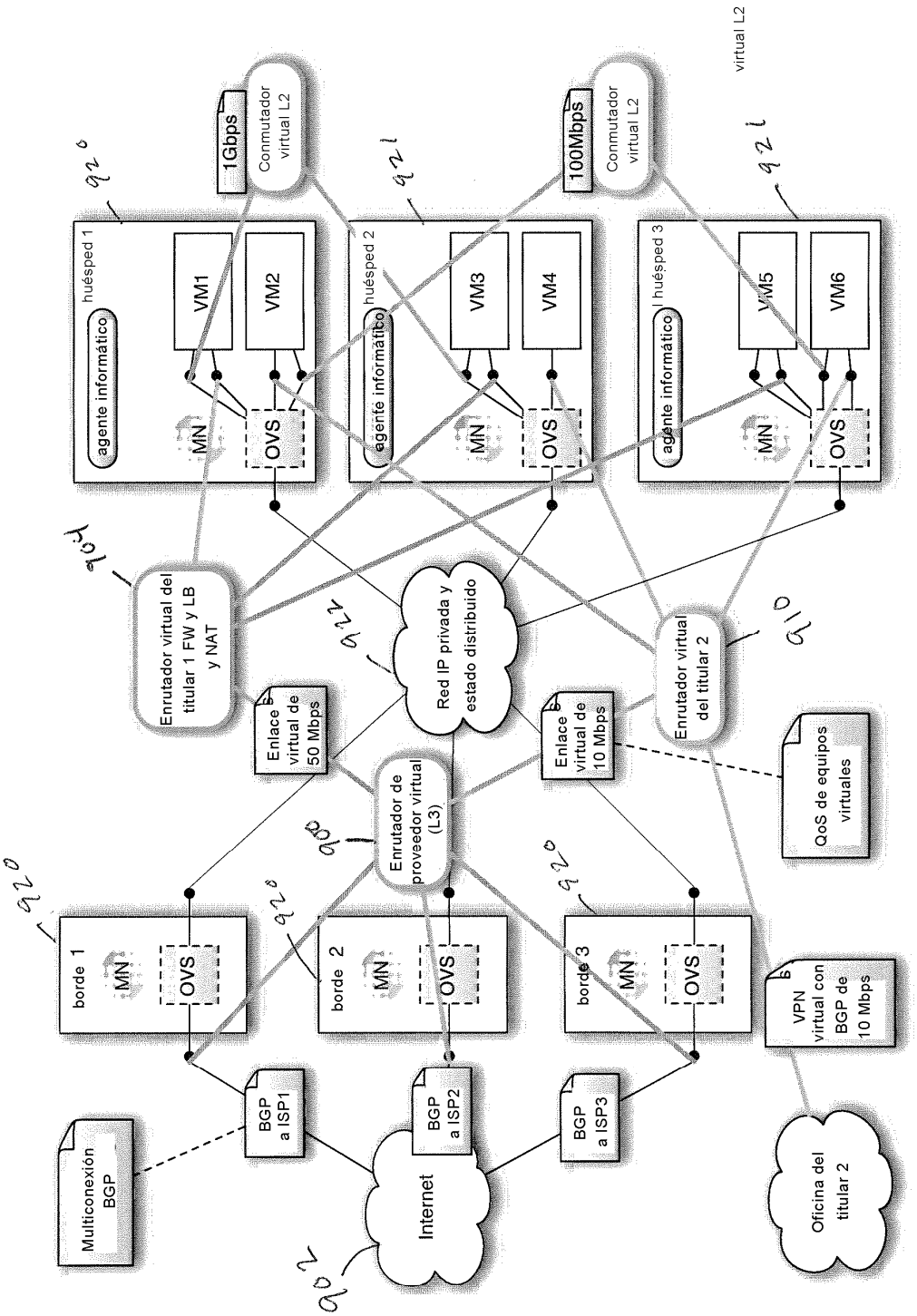
huésped 1
agente informático
VM1
VM2
MN
OVS
1Gbps
Conmutador virtual L2
huésped 2
agente informático
VM3
VM4
MN
OVS
100Mbps
Conmutador virtual L2
huésped 3
agente informático
VM5
VM6
MN
OVS
Conmutador virtual L2
Enrutador virtual del titular 1 FW y LB y NAT
Red IP privada y estado distribuido
Enrutador virtual del titular 2
Enlace virtual de 50 Mbps
Enlace virtual de 10 Mbps
Enrutador de proveedor virtual (L3)
QoS de equipos virtuales
borde 1
MN
OVS
borde 2
MN
OVS
borde 3
MN
OVS
Multiconexión BGP
BGP a ISP1
BGP a ISP2
BGP a ISP3
Internet
VPN virtual con BGP de 10 Mbps
Oficina del titular 2

FIG. 14