

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 709 977**

51 Int. Cl.:

**H04L 12/751** (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **24.12.2014 PCT/EP2014/079318**

87 Fecha y número de publicación internacional: **15.10.2015 WO15154834**

96 Fecha de presentación y número de la solicitud europea: **24.12.2014 E 14820893 (7)**

97 Fecha y número de publicación de la concesión europea: **14.11.2018 EP 3117571**

54 Título: **Bucles de ejecución**

30 Prioridad:

**11.04.2014 GB 201406568**

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

**22.04.2019**

73 Titular/es:

**ENTUITY LIMITED (100.0%)  
9a Devonshire Square  
London EC2M 4YL, GB**

72 Inventor/es:

**ROPER, JEFFREY JOHN**

74 Agente/Representante:

**UNGRÍA LÓPEZ, Javier**

ES 2 709 977 T3

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

## DESCRIPCIÓN

## Bucles de ejecución

5 La presente invención se refiere a la ejecución de bucles de programa de ordenador, particularmente, pero no exclusivamente en el contexto de la identificación de una ruta en una red de dispositivos interconectados. Las redes de ordenadores forman la base para la infraestructura IT (tecnología de la información) en una amplia variedad de contextos. Tales redes de ordenadores incluyen dispositivos interconectados de varios tipos. La finalidad de la red es soportar el flujo de mensajes entre dichos dispositivos con el fin de suministrar información, aplicaciones y servicios, etc., por la red. Se dispone de varias técnicas para gestionar una red. En este contexto, 10 gestionar una red incluye supervisar la red para identificar puntos de fallo y otras zonas problemáticas, tal como puntos calientes, y proporcionar información a administradores y usuarios de la red para poder resolver los problemas. Hay varias herramientas disponibles para proporcionar una topología de red. La topología de una red identifica como están conectados física o lógicamente uno a otro los dispositivos en la red. Así, cualquier dispositivo concreto único puede tener una o varias conexiones a un dispositivo contiguo. Hay disponibles herramientas informatizadas que "descubren" una red, y crean topologías de red que definen la interconexión de los dispositivos en la red, y la naturaleza de dichos dispositivos. 15 Un ejemplo de este tipo se muestra en el documento US 2014/036729 A1. La determinación de la topología de red se puede hacer de muchas formas. Las técnicas que pueden ser utilizadas por separado o en combinación obteniendo una buena representación de la conectividad de red incluyen, por ejemplo:

- *Protocolo de descubrimiento de Cisco (CDP)*
- *Protocolo de descubrimiento de capa de enlace (LLDP)*
- *Protocolo de gestión de red SynOptics (SoNMP)*
- 25 • *Protocolo de árbol de expansión (STP)*
- *IP Traceroute*
- *Descubrimiento de vecinos IPv6*
- *Adiciones/modificaciones/borrados de usuario*

30 Conocer la topología de una red es sumamente útil, pero no proporciona una solución a todos los problemas que pueden producirse. Las redes se utilizan cada vez más para proporcionar la infraestructura para soportar la distribución de aplicaciones y servicios entre posiciones geográficas remotas, o por distancias largas o en redes sumamente complejas con gran número de dispositivos interconectados. Los administradores de red y usuarios están cada vez más interesados en conocer no necesariamente todos los detalles de la red, sino en entender el funcionamiento del suministro de aplicaciones y servicios por una red. Así, la denominada supervisión de "extremo a extremo" cada vez es más popular. Con supervisión de "extremo a extremo", el rendimiento de las aplicaciones que implican flujo de mensajes desde un dispositivo fuente a un dispositivo destino es supervisado cuando son distribuidas entre dichos dispositivos fuente y destino. Los parámetros de rendimiento pueden ser usados para 35 estimar o averiguar posibles fallos en la red, aunque no proporcionan información específica acerca de la posición de los fallos y por lo tanto no apuntan directamente a una solución. 40

A menudo, un dispositivo fuente es un servidor que proporciona un servicio concreto, y el dispositivo destino es un terminal de cliente que está conectado al servidor mediante la red y que requiere el uso de dicho servicio. El término "dispositivo" usado aquí pretende cubrir cualesquiera dispositivos que puedan estar conectados en una red. El 45 término "servidor" se usa para denotar un dispositivo que es responsable de distribuir un servicio o aplicación, y el término "cliente" se usa para denotar un dispositivo (basado en usuario u otra máquina dependiente o servidor) que depende de dicha aplicación o servicio.

Una dificultad significativa al averiguar dónde podría estar un problema cuando se puede ver que se está deteriorando el rendimiento de una aplicación, es una falta de comprensión de la ruta a través de la red que podría haber tomado el flujo de mensajes para dicha aplicación. Las redes dependen de muchos tipos de dispositivos de red (por ejemplo, enrutadores, conmutadores, cortafuegos, equilibradores de carga, etc.) para conectar sus dispositivos de punto final, de tal manera que es sumamente difícil decir con respecto a cualquier punto final fuente dado como el mensaje de dicho punto final será dirigido a través de la red a un punto final de destino dado. La 50 complejidad de tal determinación de ruta es exacerbada por el uso de múltiples rutas alternas, rutas redundantes, equilibrio de carga, etc. 55

Se ha intentado predecir como un paquete concreto será enrutado a través de una red. Tales predicciones se basan en un modelo complejo de la topología de red con juntamente con indicaciones en base a dispositivo sobre cómo se comportará un dispositivo concreto en la red. Los dispositivos de red pueden ser altamente sofisticados, y se ha desarrollado gran número de algoritmos complejos para determinar una estrategia de enrutamiento en cualquier dispositivo concreto. Además, dicha estrategia de enrutamiento puede depender del tráfico y otras consideraciones medioambientales que afectan a la red (tal como fallo de otros dispositivos, etc.). Los algoritmos complejos que pueden ser aplicados por un dispositivo para determinar una estrategia de enrutamiento pueden incluir, por ejemplo: 60 65

- Tecnología de interfaz de entrada y de interfaz de entrada
- Cabeceras de paquete (L2, L3, MPLS, ATM, etc.)
- Rutas estáticas y conectadas directamente
- 5   • Tablas de enrutamiento compartidas (conocimiento pleno de BGP, OSPF, RIP, EIGRP, etc. - vecinos activos, estados de enlace, costos de ruta, pesos de ruta, etc.).
- Tablas de enrutamiento MAC aprendidas
- Listas de control de acceso
- Tecnologías de superposición de red (por ejemplo, MPLS, 802, 1q VLAN), etc.
- Tecnologías de prevención de bucle - por ejemplo, PVSTP
- 10   • Protocolos de tunelización (MPLS, IPSEC, SSL, GRE)
- Carga equilibrada/enlaces redundantes
- Puertas de enlace por defecto

15 Sin embargo, aunque en principio es posible predecir donde se enviara un paquete dado a continuación en un dispositivo concreto, esto requiere una amplia cantidad de datos cuya recogida es lenta, y que pueden quedar atrasados en segundos debido a la naturaleza en tiempo real de la operación de los dispositivos de enrutamiento. Además, la mera adquisición de estos datos puede imponer una carga significativa tanto a los dispositivos de red como a las redes.

20 Además, los modelos requeridos para simular modernos dispositivos de enrutamiento son sumamente complejos y, sin el modelo completo, su comportamiento no se puede prever correctamente. Para mantener completo el modelo, es necesario que tales soluciones sean actualizadas regular y rápidamente incluyendo desarrollos en las tecnologías de enrutamiento.

## 25 **Sumario**

En el nuevo enfoque aquí detallado, los dispositivos de red son consultados sobre qué harían con un paquete hipotético (en contraposición a la consulta de especificidades de protocolo de enrutamiento). No hay que conocer ni mantener los protocolos de enrutamiento que alimentan las tablas de envío y enrutamiento de los dispositivos.

30 Los inventores han desarrollado un proceso de ejecución de bucle para permitir la implementación eficiente de un proceso de identificación de ruta. Sin embargo, el proceso de ejecución del bucle tiene una aplicabilidad más amplia.

35 Es aplicable a la resolución de problemas donde el problema tiene varias etapas y cada etapa puede resolverse mediante una gama de técnicas donde las técnicas están ordenadas en términos de precisión/conveniencia y no se sabe a priori qué técnica debe ser aplicada en cada etapa.

40 Según un aspecto de la presente invención, se proporciona un método para ejecutar un programa informático de bucle que comprende: recibir en una unidad de ejecución un conjunto de variables de estado que definen un estado de entrada, donde una de las variables de estado define una secuencia de opciones de bucle; registrar el estado de entrada en una unidad de almacenamiento; en la unidad de ejecución, ejecutar una primera opción de bucle en la secuencia ordenada de opciones de bucle en el estado de entrada, usando como parámetros al menos una de las otras variables de estado en el conjunto de variables de estado, donde ejecutar la primera opción de bucle comprende cancelar la primera opción de bucle de la secuencia ordenada, realizar etapas de procesamiento utilizando al menos una variable de estado y determinar si alguna de las variables de estado se ha alterado como resultado de las etapas de procesamiento, donde: si ninguna de las variables de estado se ha alterado, se ingresa una siguiente iteración de bucle con un estado de entrada donde la primera opción de bucle se cancela desde la secuencia ordenada, revelando una nueva opción de primer bucle, y; si al menos una de las variables de estado ha cambiado, restablece la opción de primer bucle cancelado en la secuencia ordenada e ingresa a la siguiente iteración del bucle con un estado de entrada definido por la(s) variable(s) de estado alterada(s) y la secuencia ordenada original, por lo que cada bucle siguiente de la iteración recibe un nuevo estado de entrada.

50 El método es particularmente útil en un método implementado por ordenador para identificar en una red de dispositivos interconectados una ruta a través de la red desde un dispositivo de origen a un dispositivo de destino, comprendiendo la ruta una secuencia de dispositivos conectados, comprendiendo el método en un monitor de ordenador conectado a la red: identificar un primer dispositivo conectado al dispositivo fuente; transmitir una consulta al primer dispositivo, incluyendo la consulta un identificador de destino y solicitar la identificación de un puerto de salida para los mensajes dirigidos a un destino identificado por el identificador de destino cuando la consulta se recibe en el primer dispositivo; recibir un mensaje de resultado que identifica el puerto de salida e identifica el segundo dispositivo conectado al primer dispositivo en función de una topología de red a la que puede acceder el monitor de ordenador; y dirigir una próxima consulta al segundo dispositivo y recibir un mensaje de resultado siguiente que identifica un puerto de salida del segundo dispositivo; e identificar desde la topología de la red un tercer dispositivo conectado al segundo dispositivo, y así sucesivamente, donde la ruta se identifica para incluir el primer, el segundo y los dispositivos posteriores.

65

Cuando se usa el método de ejecución de bucle, la formulación de los mensajes de consulta a los dispositivos puede realizarse mediante la misma opción repetida o diferentes opciones, ejecutadas en iteraciones de bucle sucesivas.

5 El proceso de identificación de rutas utiliza un nuevo enfoque para identificar las rutas tomadas a través de una red de dispositivos interconectados para un flujo concreto de mensajes. El concepto se basa en usar una cantidad mínima de datos recogidos "por adelantado" - expeditamente la topología de red estática y la posición de host final (que clientes y servidores están conectados a que conmutadores de acceso/borde), y recoge cualquier otra cosa necesaria al vuelo y de forma altamente selectiva según sea preciso para tales datos altamente dinámicos. Para modernos entornos dinámicos, la capacidad de calcular la ruta de extremo a extremo ahora, es decir, en tiempo real, 10 tiene amplia aplicabilidad. La recogida de los datos y su procesado tiene que ser muy rápida para que el algoritmo sea de valor factible cuando se use con redes del mundo real a gran escala.

15 El comportamiento en un dispositivo concreto se denomina "comportamiento por salto" (PHB). PHB en sí mismo no puede proporcionar una ruta de extremo a extremo. Sin embargo, conocer que un paquete sale del dispositivo en una interfaz específica puede ser de valor si no se conoce que dispositivo e interfaz están conectados a esa interfaz. Usando topología de red acoplada con PHB, el cálculo directo de una ruta de extremo a extremo a través de la red para un flujo de aplicación puede realizarse.

20 Así, la topología de red incluye tanto interconectividad de dispositivos de red como posición de host final. A su vez, los dispositivos fuente y destino usados para sembrar el algoritmo de hallazgo de ruta son determinados por el flujo de mensajes de interés. Finalmente, la ruta específica tomada por un paquete hipotético entre dispositivos fuente y destino es calculada dinámicamente.

25 La consulta transmitida a cada dispositivo está adaptada para consultar cada dispositivo para determinar la identificación de un puerto de salida que represente los puertos de salida que el dispositivo usaría para un mensaje hipotético dirigido a un destino identificado por el identificador de destino. Obsérvese que el identificador de destino para cualquier consulta dada puede ser o no ser el identificador de destino del dispositivo terminal dependiendo de la posición en la red del dispositivo consultado. Esto se puede lograr, cuando el dispositivo es un enrutador, consultando que hay en su tabla de enrutamiento activa al tiempo en que se recibe la consulta. 30

35 La consulta propiamente dicha puede acomodarse en un mensaje o señal transmitida desde el ordenador supervisor al dispositivo que este siendo consultado (dispositivo de enfoque). El mensaje o la señal de consulta no constituyen el flujo de mensajes para el que la ruta ha de ser determinada. En cambio, cada consulta contiene un identificador de destino que consulta al dispositivo de enfoque para hallar cómo gestionaría el dispositivo de enfoque un mensaje hipotético dirigido a dicho destino si tuviese que hacer la decisión al tiempo de recibirse la consulta. Así, el dispositivo de enfoque devuelve un resultado que identifica un puerto de salida inmediata que se habría usado entonces para un mensaje real dirigido a ese destino. Las consultas pueden transmitirse mientras la red es activa y mientras tiene lugar el flujo de mensajes. Sin embargo, también pueden transmitirse cuando el flujo de mensajes propiamente dicho no es activo - la técnica puede ser usada en cualquier contexto. 40

45 Donde la consulta tiene forma de un mensaje o paquete, por ejemplo, el mensaje puede ser un mensaje SNMP con una dirección IP de destino, llevará su propia dirección de destino y será distribuido por la red del ordenador supervisor al dispositivo de enfoque. En ese caso, la dirección de destino del mensaje de consulta es la del dispositivo de enfoque. Esto no es lo mismo que el identificador de destino que se incluye en la consulta propiamente dicha. En una disposición alternativa, se puede enviar una señal o señales de consulta desde el ordenador supervisor a través de conexiones directas a los dispositivos de enfoque, tal como a través de un mecanismo CLI o XML API.

50 El método aquí descrito permite varias técnicas de análisis de red útiles. Permite una determinación de ruta a demanda de modo que un administrador que intente determinar la ruta para una aplicación particular puede preguntar de forma más o menos instantánea al ordenador supervisor y recibir un resultado de la ruta.

55 Permite el descubrimiento de rutas múltiples. Es decir, debido a cambios en el entorno en la red, los dispositivos de enrutamiento pueden enrutar un flujo de mensajes de forma diferente dependiendo de dichos cambios. Así, un primer conjunto de consultas para identificar una ruta podría registrar una primera ruta, mientras que un segundo conjunto de consultas podría identificar una segunda ruta, incluso cuando el conjunto de consultas primero y segundo estén muy cerca uno de otro en el tiempo. La información acerca de múltiples rutas entre puntos finales comunes (es decir, el mismo dispositivo fuente y el mismo dispositivo destino) puede presentarse grafica o visualmente para mostrar al usuario no solamente la naturaleza de la ruta, sino el porcentaje de tiempo que cada ruta es adoptada para un flujo de mensajes concreto. Esto se puede lograr fácilmente porque las consultas propiamente dichas no representan una carga significativa para la red, y por lo tanto pueden enviarse múltiples conjuntos de consultas sin afectar de forma significativa al rendimiento. 60

65 El método permite la detección de cambio de ruta legitimado rápido. Es decir, un ajuste en la red puede hacer que la ruta cambie y esto puede ser detectado y señalado al usuario en una interfaz de usuario grafica visual.

Donde hay múltiples rutas entre dispositivos fuente y destino comunes, las rutas pueden llevar diferentes latencias. A veces, un dispositivo de enrutamiento que realiza enrutamiento inteligente puede producir un fenómeno conocido como "aleteo de ruta" donde un flujo concreto de mensajes cambia continuamente de una ruta a otra. Puede ser útil para que un administrador de red identifique estos casos debido a las implicaciones de tales cambios de ruta en latencia de extremo a extremo y las implicaciones de tal "oscilación" de conversaciones telefónicas de voz por IP, por ejemplo.

El método puede ser usado para localizar un fallo de ruta. Es decir, en la realización preferida del método, se envían consultas y se reciben y analizan resultados para identificar el dispositivo siguiente hasta que un dispositivo es identificado como el dispositivo destino. A veces, sin embargo, hay un fallo en la red, de tal manera que la red no llevara el flujo de mensajes al dispositivo destino. El método permite la identificación de esa situación porque opera a lo largo de una ruta de extremo a extremo hasta que la ruta no puede seguir y esta posición de red se le puede notificar después a un administrador.

Además, el método puede permitir la posibilidad de volver a arrancar en un dispositivo posterior en dicha ruta, usando estimaciones basadas en la topología de red (indicado en el presente documento como "salto"). El método de identificación de ruta puede adoptarse entonces de nuevo hasta que se llegue al dispositivo destino desde el punto de fallo. De esta forma, porciones de la red de las que el ordenador supervisor no tiene visibilidad (por ejemplo, dispositivos que no tienen una interfaz de gestión apropiada, o que pertenecen a una organización diferente) se pueden dejar a un lado y continuar el análisis de ruta.

El método también permite la identificación de enrutamiento asimétrico. No es insólito que el flujo de mensajes entre un dispositivo fuente y un dispositivo destino adopte rutas diferentes dependiendo de su dirección. Es decir, se puede utilizar una ruta directa desde el dispositivo fuente al dispositivo destino para el flujo de mensajes, y una ruta de retorno desde el dispositivo destino al dispositivo fuente que sea diferente.

La ruta es registrada en una memoria o almacenada en el ordenador supervisor o accesible por el ordenador supervisor. El registro de ruta incluye un conjunto de dispositivos conectados e interfaces. Esto puede presentarse en forma de un inventario ordenado de los dispositivos (componentes de red) entre los dos puntos finales. Esto permite supervisar la disponibilidad de rutas de red, incluyendo notificación de eventos, reportes, SLA (Acuerdos de nivel de servicio) ; gestión de red proactiva incluyendo reportes de dispositivos con fallo, CPU de dispositivo alta, memoria de dispositivo baja, congestión de puertos, etc., y análisis de impacto (planificación de capacidad, análisis "que si").

Una ventaja significativa de aspectos de la presente invención es que un mapeado entre la aplicación o el servicio distribuido por la red y los dispositivos de red o componentes propiamente dichos puede ser conocido a través de la identificación de la ruta. Esto representa un avance significativo en la gestión de redes.

De acuerdo con otro aspecto de la presente invención, se proporciona un sistema informático para ejecutar un programa informático de bucle, comprendiendo el sistema informático un procesador operable para ejecutar un método de ejecución de bucle y medios de almacenamiento para registrar el estado de entrada.

El sistema se puede utilizar como un sistema informático de supervisión para identificar en una red de dispositivos interconectados una ruta a través de la red desde un dispositivo de origen a un dispositivo de destino, comprendiendo el sistema informático: una interfaz conectada a la red para transmitir consultas y recibir respuestas; un segundo medio de almacenamiento para almacenar un registro de ruta; y un tercer medio de almacenamiento para almacenar una topología de red.

Un aspecto adicional de la presente invención proporciona un producto de programa informático que comprende instrucciones legibles por computadora que, cuando se ejecutan por un procesador, implementan el método como se define aquí.

Para una mejor comprensión de la presente invención y para mostrar cómo se puede poner en práctica, ahora se hará referencia, a modo de ejemplo, a los dibujos acompañantes, en los que:

- La figura 1 es un diagrama esquemático de una red;
- Las figuras 2a a 2c son una ilustración esquemática de un algoritmo de descubrimiento de ruta en proceso;
- La figura 3 es un diagrama de flujo para un algoritmo de descubrimiento de ruta;
- La figura 4 representa una ruta descubierta;
- La figura 5 es la estructura de una tabla de enrutamiento lineal;
- La figura 6 ilustra un conjunto de resultados que surgen de combinar una dirección de destino con múltiples máscaras de ruta;
- La figura 7 representa una estructura de una tabla ARP;
- La figura 8 es un diagrama esquemático de un ordenador supervisor;
- La figura 9 es un diagrama esquemático de un enrutador de capa 3;
- La figura 10 es un diagrama esquemático de un conmutador de capa 2;

Las figuras 11a a 11d son un diagrama de flujo de una utilidad ejecutada en el ordenador supervisor;  
 La figura 12 es un diagrama de flujo de un programa de ejecución de bucle;  
 La figura 13 es un diagrama de flujo que representa un proceso de terminación del programa de ejecución de bucle;  
 5 La figura 14 es un diagrama de flujo que representa la opción C del programa de ejecución de bucle;  
 La figura 15 es un diagrama de flujo que representa la opción S del programa de ejecución de bucle;  
 La figura 16 es un diagrama de flujo que representa la continuación del proceso en la opción S;  
 La figura 17 es un diagrama de flujo que representa el proceso en la opción del programa de ejecución de bucle;  
 La figura 18 es un diagrama de flujo que representa una continuación del proceso de la figura 17;  
 10 La figura 19 es un diagrama de flujo que representa el proceso de la opción C del programa de ejecución de bucle;  
 La figura 20 es un diagrama de flujo que representa el proceso de la opción A;  
 La figura 21 ilustra un proceso para obtener una indicación VLAN;  
 La figura 22 es un proceso para obtener un puerto conectado y dispositivo conectado a almacenar en un registro de ruta;  
 15 La figura 23 es un diagrama de flujo que representa un proceso iterativo para hallar una ruta que es utilizada en algunas de las opciones anteriores;  
 Las figuras 24, 25 y 26 ilustran tres procesos iniciadores;  
 La figura 27 es un diagrama de flujo que ilustra un proceso de hallazgo de ruta que se utiliza en el proceso iterativo de hallazgo de ruta de la figura P; y  
 20 La figura 28 ilustra un proceso para buscar una entrada de base de datos de envío de un dispositivo de enfoque.

La figura 1 es un diagrama esquemático de una red. La red se extiende por varios lugares geográficos diferentes. En cada lugar geográfico de extremo hay dispositivos de punto final y dispositivos de red o nodos. Los dispositivos de red incluyen enrutadores y conmutadores. El núcleo de la red incluye una pluralidad de dispositivos de red. Con respecto al lugar geográfico indicado Londres, terminales de cliente 2 pueden actuar como dispositivos de punto final. Igualmente, un servidor 4 puede actuar como un dispositivo de punto final y la impresora 6 puede considerarse un dispositivo de punto final. Se representan dispositivos similares en los lugares geográficos Pans y Nueva York con diferentes configuraciones (Nueva York representa un centro de servidores o centro de datos). Obsérvese que en el lugar Nueva York una pluralidad de servidores 8 representa dispositivos de punto final de servicio o aplicación clave.

Se deberá apreciar que la red representada en la figura 1 se ofrece a modo de ejemplo. Hay una amplia variedad de redes posibles y la presente invención puede usarse en cualquier red de dispositivos interconectados. En particular, la naturaleza de los dispositivos de punto final y los dispositivos de red específicos o nodos puede variar. En la red concreta que se describe, los dispositivos de red pueden ser dispositivos de capa 3 o de capa 2.

El modelo OSI (interconexión de sistemas abiertos) define siete capas dentro de las que los protocolos de los sistemas de comunicaciones pueden caracterizarse. El algoritmo de hallazgo de ruta aquí descrito calcula rutas de red usando información disponible en las capas 2 y 3.

Los dispositivos que operan en la capa 2 (la capa de enlace de datos) tienen conocimiento de dispositivos inmediatamente adyacentes y tienen la responsabilidad de llevar paquetes desde un dispositivo de capa 2 al dispositivo de capa 2 siguiente (en base a dirección MAC (control de acceso a medio) de capa 2).

Los dispositivos que operan en la capa 3 (la capa de red) son responsables de propagar paquetes desde un punto en una red a otro punto en la red -a menudo separados muchas decenas o cientos de dispositivos. Para calcular que dispositivos deberán participar en una ruta de capa 3 dada (aquí denominados los saltos de capa 3) , los dispositivos de capa 3 intercambian información de enrutamiento y usan protocolos de enrutamiento para calcular la (s) ruta (s) más deseable (s). Para pasar paquetes entre dispositivos de capa 3 consecutivos en una ruta, se usan dispositivos que operan en la capa 2; a menudo con muchos dispositivos de capa 2 (aquí denominados los saltos de capa 2) entre cada dispositivo de capa 3.

Así, las redes grandes están subdivididas realmente en múltiples segmentos, conteniendo típicamente cada uno múltiples dispositivos de capa 2, conectados por dispositivos de capa 3.

La figura 9 es un diagrama altamente esquemático de un dispositivo de enrutamiento de capa 3. El dispositivo incluye un controlador 90 por ejemplo, en forma de un microprocesador que ejecuta código de control, microprogramas o cualquier otra implementación adecuada. El controlador 90 puede acceder a una tabla de enrutamiento 92 que se explica con más detalle más adelante con referencia a la figura 5. El dispositivo de enrutamiento de capa 3 tiene puertos Pi/Po. Cada puerto está conectado a un enlace físico como se ilustra en la red de la figura 1. En esta notación, Pi denota un puerto de "entrada" y Po denota un puerto de "salida". Esto es por razones de conveniencia de la notación, en la práctica, los dispositivos no tienen generalmente puertos dedicados como puertos de entrada o de salida - que sean de entrada o de salida depende de los datos que entonces transfieran. La mayor parte de los puertos funcionan como de salida y entrada todo el tiempo.

Los identificadores de destino, por ejemplo, IP (direcciones de protocolo de Internet) de los paquetes que llegan a un puerto de entrada Pi pueden ser leídos por el controlador 90 mediante un bus 94. El controlador 90 accede a la tabla de enrutamiento 92 y en base a información derivada de ella controla un conmutador de enrutamiento 96 al que el paquete entrante es dirigido.

5 El conmutador de enrutamiento 96 enruta entonces el paquete entrante a un puerto de salida apropiado Po dependiendo de la información presente en la tabla de enrutamiento. El dispositivo de enrutamiento incluye una tabla de mapeado 91 que mapea direcciones de capa 3 a capa 2 para enrutamiento posterior. La operación de tales dispositivos de enrutamiento es conocida en la técnica y por ello no se describirá más aquí. Se indica en este contexto que la tabla de enrutamiento puede ser consultada por paquetes procedentes del ordenador supervisor que llegan por los enlaces al puerto de entrada Pi interceptando tales paquetes en el controlador 90. Tales paquetes de consulta no son suministrados al conmutador de enrutamiento 96 para enrutamiento adicional, sino que en cambio generan una respuesta que es enviada desde el dispositivo de enrutamiento y devuelta a la entidad consultante por la red desde un puerto de salida. En este caso, la entidad consultante es el ordenador supervisor 16.

15 Todos los paquetes transportados por la red (incluyendo los paquetes de consulta) contienen una dirección de fuente y otra de destino - el paquete de consulta tiene una dirección de fuente correspondiente al ordenador supervisor y una dirección de destino correspondiente al dispositivo que es consultado. Cuando la respuesta ha de ser enviada, las direcciones de fuente y de destino se cambian haciendo que la dirección de fuente sea el dispositivo consultado y que la dirección de destino sea el ordenador supervisor.

20 La figura 10 es una versión altamente esquematizada de un conmutador de capa 2. Al igual que un dispositivo de enrutamiento de capa 3, el conmutador de capa 2 tiene puertos Pi/Po, cada uno conectado a un enlace físico como se representa, por ejemplo, en la red de la figura 1. Como se ha mencionado antes, los puertos no están dedicados en general como de entrada o de salida. Los paquetes entrantes en un puerto de entrada Pi son dirigidos a un conmutador 100 que puede acceder a una base de datos de envío (FDB) de capa 2 102 para determinar cómo enrutar los paquetes en base a identificadores de destino (normalmente cabeceras) en los paquetes. Una base de datos de envío de capa 2 mapea un identificador de paquete entrante a un puerto de salida donde el paquete deberá ser enviado. Como ya se ha explicado anteriormente, según el modelo OSI, los identificadores para los dispositivos de capa 3 de enrutamiento son direcciones IP, mientras que los identificadores para los dispositivos de capa 2 son direcciones MAC.

25 Como con los dispositivos de capa 3, los de capa 2 son conocidos en la técnica y por ello no se explicarán más aquí. Sin embargo, se indica que, de nuevo de forma análoga a los dispositivos de capa 3, pueden recibir una consulta en un paquete en un puerto de entrada Pi y generar una respuesta a esa consulta en la salida del conmutador de capa en un puerto de salida Po. Así, los paquetes de consulta propiamente dichos no son enrutados en el conmutador, sino que, en cambio, generan una respuesta que es devuelta al dispositivo consultante, en este caso el ordenador supervisor 16.

40 Un controlador de conmutador 101 en el conmutador es responsable de enviar tráfico y de generar respuestas.

Algunos dispositivos más recientes pueden realizar la función de capa 3 y capa 2.

45 Las realizaciones de la presente invención descritas a continuación proporcionan un método de identificar una ruta tomada por un flujo de mensajes entre un dispositivo fuente dado y un dispositivo destino dado. Por ejemplo, el punto final X podría considerarse un dispositivo fuente y el punto final Y podría considerarse un dispositivo destino. Considerando una red de la figura 1, dista mucho de ser una tarea trivial, como ya se ha explicado anteriormente, establecer que ruta se adoptara a través de la red entre los puntos finales en cualquier tiempo dado y en cualquier conjunto dado de condiciones medioambientales. La figura 1 representa un ordenador supervisor 16 que ejecuta un programa de descubrimiento de ruta que permite descubrir y registrar esa ruta. La figura 8 es una versión altamente esquemática de un ordenador supervisor 16. El ordenador 16 incluye un microprocesador 80 que puede acceder a una memoria 82 donde se guarda un código para ejecución por el procesador. En el caso presente, el código incluye el programa de descubrimiento de ruta. La memoria 82 también guarda un registro de ruta 81 tal como lo crea el programa de descubrimiento de ruta. El ordenador tiene una interfaz de usuario 84 que puede incluir un dispositivo de entrada de usuario, un ratón o un teclado, y una pantalla para presentar información a un usuario. En particular, como se explica aquí con más detalle, pueden presentarse al usuario en la interfaz de usuario 84 alertas después del programa de descubrimiento de ruta o información con relación al programa de descubrimiento de ruta. Las figuras 2a a 2c ilustran etapas de la ruta que se describirán ahora.

60 En un nivel alto, el algoritmo usa la noción de un "dispositivo de enfoque" que es el dispositivo actualmente consultado acerca de donde enviaría un paquete hipotético siguiente (es decir, desde que interfaz enviaría el paquete hipotético). Comenzando en el dispositivo fuente y el dispositivo terminal (es decir, el último destino del paquete) el algoritmo adopta sucesivas opciones en un bucle para evaluar cada dispositivo de enfoque. De acuerdo con dos de las opciones ("S" y "R") si el dispositivo está operando en la capa 3, se le consulta que interfaz (puerto de salida) utilizaría para enviar paquetes unidos. Para el salto siguiente en la capa 3 (NHL3); si el dispositivo está operando en la capa 2, se le consulta que interfaz (puerto de salida) utilizaría para enviar paquetes unidos para la

siguiente dirección de capa 2 (MAC) de salto a capa (NHL2). Usando la respuesta del dispositivo de enfoque en unión con una topología de red, se puede determinar el dispositivo siguiente en la ruta. De esta forma, el algoritmo opera a lo largo de la ruta de capa 3, usando los dispositivos de capa 2 para navegar entre los nodos de capa 3 consecutivos.

5 Otras opciones se pueden implementar en el bucle como sea necesario, como se describe a continuación.

Como parte del proceso de preparación (definido posteriormente), se localizan el dispositivo de fuente y el dispositivo terminal. Esto puede no ser directo y técnicas para conseguir esto se describen posteriormente.

10 Según el algoritmo principal, se localiza el primer salto. Se siembra la ruta y el recuento de bucle se pone a cero. El límite de bucle controla el número de veces que se ejecuta un bucle de identificación de ruta (que se explica más adelante).

15 A continuación, se hará referencia a la figura 12 para describir el proceso de iteración en bucle. La entrada en el bucle principal se indica en la parte superior de la figura 12 mediante la flecha de entrada 4. La flecha de entrada 4 indica el estado al final de los procesos de preparación que se describirán más adelante. El estado de entrada comprende:

20 <Opciones, dispositivo de enfoque, NHL 3, NHL 2, VLAN>

Estos elementos se describirán en más detalle y se establecen por los procesos de preparación que se describirán más adelante. Estos se denominan a continuación como "variables de estado". La variable de estado titulada "opciones" tiene una secuencia ordenada de opciones de bucle. En el presente ejemplo, la secuencia ordenada comprende CSRAcr.

25 La variable de estado, titulada "VLAN" es el identificador de la red de área local virtual (número) de la VLAN con la cual el paquete hipotético está etiquetado actualmente en este punto en la ruta.

30 En la etapa L01 del bucle, la primera opción (cabeza de lista) se selecciona y se ejecuta. Estas opciones se describen más adelante. Después de la ejecución de la opción, el proceso vuelve a regresar al punto L02 y se crea un nuevo estado (L03), siguiendo las etapas de procesamiento implementadas por la cabeza de la opción de la lista. Se determina (L04) de si este estado se ha producido antes, y si no, el estado se almacena (L05). Si el estado se ha producido antes, se genera un informe de "bucle descubierto" y el límite del bucle se establece en cero, lo que tendrá por efecto la ruptura del bucle.

35 En la etapa L07, se disminuye el límite de bucle. Si el límite de bucle es menor que o igual a cero o no hay más opciones disponibles en la secuencia de opciones, o el dispositivo de enfoque es igual al dispositivo terminal, entonces se establece una condición de "terminación es igual a verdadero". En la etapa L08, se realiza una comprobación de la condición de terminación, y si la condición de terminación es verdadera, termina el bucle principal. De lo contrario, vuelve al punto de entrada de bucle principal usando el nuevo estado que fue creado en la etapa L03.

40 Obsérvese que, en la ejecución de cada opción en una iteración del bucle, la primera etapa en la ejecución de la opción es eliminar esa opción de la secuencia ordenada en la variable de estado de opciones.

45 Al final de las etapas de procesamiento de la opción, puede ser que esa opción se restablezca de nuevo en la secuencia, o puede haber sido eliminada de forma permanente, dependiendo de la opción y de los resultados de las etapas de procesamiento.

50 Obsérvese también que, en la ejecución de las etapas de procesamiento de una opción, las otras variables de estado (dispositivo de enfoque, NHL3, NHL2, VLAN) pueden alterarse de forma individual o en total. La alteración de cualesquiera variables de estado resulta en un nuevo estado, que puede constituir un nuevo estado de entrada para una siguiente iteración del bucle.

55 Ahora se hará referencia a la figura 13 para describir la segunda parte del proceso de bucle principal. En la etapa L09 se determina si se ha alcanzado o no el dispositivo terminal esperado. Si lo ha hecho, en la etapa L10 se determina si el terminal tiene un conmutador de acceso conectado al IP del servidor. Si no lo es, entonces el proceso termina después de que se indica un descubrimiento de ruta completa con éxito y devuelve una ruta completa. Si el dispositivo terminal es un conmutador de acceso conectado a la IP del servidor, se añade la conexión del puerto de acceso y la IP del servidor a la ruta, y entonces el proceso pasa a completar el descubrimiento de la ruta con éxito y devuelve una ruta completa antes de terminar deteniéndose.

60 Si en la etapa L09 se determina que el dispositivo terminal esperado no se ha alcanzado, entonces en la etapa L14, se preguntó si NHL3 es la IP del servidor o no. Si no lo es, entonces se determina que el descubrimiento de la ruta completa no ha tenido éxito, y una ruta parcial se devuelve antes de la detención. Si NHL fue la IP del servidor, esto

indica que el proceso en el segmento L2 final, por lo que el proceso puede saltar hasta el destino mediante el incremento del contador de segmentos de salto y estableciendo el dispositivo de enfoque en NHL3.

5 La figura 14 ilustra la Opción C. En la primera etapa C1, la opción se elimina de la secuencia en la variable de opciones. En la etapa C2, se realiza una comprobación para una única interfaz de red cuya dirección coincide con el destino (IP del servidor). Si es así, esto indica que el algoritmo ha llegado a la porción de red de conmutación final, y NHL3 se establece en la IP del servidor. En la etapa C3, el dispositivo de enfoque se consulta para encontrar la entrada ARP para la IP del servidor, y NHL2 se establece en el resultado. El proceso vuelve entonces al bucle principal (C4) para permitir que otra opción decida la interfaz de salida. La consulta al dispositivo de enfoque se realiza mediante SNMP a la tabla ARP en el dispositivo de enfoque, o si no se encuentra, se consulta el sistema de gestión de la red ARP en caché. Estas consultas son de acuerdo con técnicas más plenamente descritas más adelante.

15 En la etapa C2, si la comprobación para la interfaz única para la IP del servidor de destino falla, no existe ninguna interfaz única identificada. Las variables de estado no se actualizan en absoluto. En este caso, todo lo que hemos hecho es la opción "C" evaluada (y descartada) y pareció que es improductiva.

20 La figura 15 ilustra opción S. Según la etapa S101, la opción S se elimina de la secuencia ordenada. En la etapa S102, se realiza una consulta al sistema de gestión de red o mediante consultas SNMP al dispositivo de enfoque para encontrar si el dispositivo de enfoque aloja NHL3. Si NHL3 está en el dispositivo de enfoque, el proceso vuelve al punto de retorno del bucle principal (S104). Si la dirección de enrutamiento NHL3 no está en el dispositivo de enfoque, se determina si se establece el NHL2 de la dirección de conmutación, y el dispositivo de enfoque se consulta en la tabla de asignación (ARP) para el NHL2 dado el NHL3. Las consultas se realizan como se describe más completamente más adelante. En la etapa S106 se determina si se establece o no el indicio de VLAN indirecto. Indicios de VLAN se discutirán más adelante. Si no, se determina una lista de las VLAN en el dispositivo de enfoque, ya sea desde el dispositivo de enfoque o desde el sistema de gestión de red. Una VLAN se selecciona de la lista y una búsqueda de la entrada de la base de datos de reenvío se realiza utilizando el dispositivo de enfoque, NHL2 y VLAN. La búsqueda de la entrada de base de datos de reenvío que se ilustra en la etapa S109 se muestra en un diagrama de flujo en la figura X. Volviendo a la etapa S106, si el indicio de VLAN se establece a continuación, el proceso pasa directamente a la etapa 110, que es una búsqueda de la entrada FDB como en la etapa S109 usando el indicio de VLAN que la VLAN ha consultado dentro de la FDB. En la etapa S112 (también S111), se determina si hay o no hay una entrada encontrada en la base de datos de reenvío. Si la hay, el proceso pasa a la segunda parte de la opción S que se muestra en la figura 16 (flecha de entrada 5). Si después de la etapa S111, no se encuentra ninguna entrada FDB, se entra en un bucle de VLAN hasta que se determina si existe una entrada FDB o que el proceso debe volver al bucle principal. El punto de entrada a la segunda parte de la opción S se muestra en la flecha 5 en la parte inferior de la figura 15. Esto también se muestra en la parte superior de la figura 16. Como se describe, anteriormente, si se encuentra una entrada de base de datos de reenvío, esto indica el puerto de salida para la ruta S115. Esto se puede utilizar para obtener el siguiente dispositivo conectado desde la topología de la red, como se muestra en la etapa S117, y se describe más completamente más adelante. La etapa S116 es la etapa de obtener un indicio de VLAN que se muestra en la figura 21 y se describirá más adelante.

45 La etapa S117 se muestra en la figura 22, que es un diagrama de flujo que ilustra el proceso para la obtención del puerto conectado y, por lo tanto, el dispositivo conectado posterior del sistema de gestión de red basado en el puerto de salida regresa desde la base de datos de reenvío: Si en la etapa S118 se encuentra el puerto conectado, el puerto conectado y dispositivo conectado se añaden a la ruta identificada (etapa S119) y en la etapa S120 el dispositivo de enfoque se cambia al dispositivo conectado. En la etapa S121, las opciones de bucle se restablecen a CSRACr. A continuación, el procesador vuelve al bucle principal en la etapa S122. Volviendo a la etapa S118, si no se encuentra el puerto conectado, el proceso salta adelante a NHL3, incrementando un contador de segmento de salto 89 (figura 8) y cambiando el dispositivo de enfoque a NHL3. El contador de segmento de salto se implementa en el ordenador de administración de hardware, firmware o software y permite a los segmentos de la ruta que se distinguen por ser saltados cuando está claro que el siguiente dispositivo conectado no puede determinarse fácilmente a partir de las etapas de proceso anteriores. En la etapa S124, se determina si el dispositivo de enfoque no es el destino (IP del servidor). Si no es así, las opciones de bucle se restablecen a CSRACr en la etapa S125. Si el dispositivo de enfoque es el servidor de destino, en la etapa 126 se determina si el destino está en un conmutador de acceso conocido, y si es así, NHL 3 se establece en la dirección del conmutador de acceso al servidor. Después de configurar las opciones de bucle en la etapa S125, en la etapa S128 el dispositivo de enfoque realiza una consulta para NHL2 utilizando la dirección NHL3 que fue creada en la etapa S127, mediante la consulta de la tabla ARP del dispositivo de enfoque o un NMS.

60 Debe tenerse en cuenta que la opción S incluye la etapa de eliminarlo de las opciones disponibles en la etapa S101, y luego restablecerse de nuevo en las opciones disponibles en la etapa S121 y la etapa S125 en base a los resultados de las etapas de procesamiento.

65 Ahora se hará referencia a la figura 17 para describir las opciones R y r. Cada una de esas opciones comienza con la eliminación de esa opción de la secuencia ordenada en la opción variable de la etapa R1, r1. En la etapa R2, r2, un proceso es instigado a buscar la ruta a la IP de destino (IP del servidor) utilizando un proceso de búsqueda de

rutas iterativo que se ilustra en la figura P. En la opción R, el proceso opera donde no hay ruta por defecto permitida (ruta por defecto permitida es igual a falso). En la opción R, el proceso permite una ruta por defecto (la ruta por defecto permitida es igual a verdadero). Si no se encuentra ninguna ruta, etapa R3, el procesador vuelve al bucle principal. Si se encuentra una ruta, a continuación, se envía una consulta al dispositivo de enfoque mediante un

5 NHL3 candidato, que se ha determinado a partir de la tabla de enrutamiento desde la que se encontró la ruta. Esta consulta es la tabla ARP del dispositivo para determinar un NHL2 candidato que corresponde al NHL3 candidato. Si no se encuentra ningún NHL2 candidato, el proceso se mueve al punto de entrada 7 para la segunda parte de la opción R/r. Si un NHL2 candidato se encuentra a partir de la consulta ARP, a continuación, se realiza una comprobación en la etapa R8 para determinar si el NHL2 candidato es el mismo que el NHL2 que se registra como

10 la variable de estado en el estado de la entrada para el proceso R/r. Si son iguales, entonces el proceso pasa al punto de entrada 6. Si no lo son, siguiendo la etapa R8, si el NHL2 candidato no es el mismo que el estado de entrada NHL2, a continuación, NHL3 se establece en siguiente salto de IP de la ruta candidata, y NHL2 se establece en el NHL2 candidato. En la etapa R10 se determina si las consultas de la tabla de enrutamiento en R2 y r2 también proporcionan un puerto de salida. Si es así, el proceso prosigue al punto de entrada 6. Si no es así, las opciones se restablecen a CSRACr en la etapa R11. La figura 18 ilustra el punto de entrada 6 en la parte superior de la figura. En la siguiente etapa R12, se determina si la tabla de enrutamiento proporciona un puerto de salida. Si no, el proceso vuelve al bucle principal. En la etapa R13, el proceso de obtener un indicio de VLAN se realiza de acuerdo con la figura 21 y se describirá posteriormente.

20 A continuación, el proceso de obtener el puerto conectado se realiza como se ilustra en la figura 22. En la etapa R15, se determina si se encuentra o no el puerto conectado. Si lo es, el puerto de salida, el puerto conectado, y el dispositivo conectado se añaden a la ruta. El dispositivo de enfoque se cambia al dispositivo conectado y las opciones de bucle se restablecen en CSRACr. Si no se encuentra un puerto conectado, no se hace nada en esta etapa. El proceso pasa a la etapa R20, donde se determina si NHL3 se ha actualizado. Si no es así, el proceso

25 vuelve al bucle principal. Si es así, el dispositivo de enfoque anterior se consulta para el candidato NHL2, dado el candidato NHL3 de la tabla de enrutamiento. Si, después de esta etapa, la NHL2 se ha resuelto, el proceso vuelve al bucle principal. Si no es así, el nuevo dispositivo de enfoque se consulta para el candidato NHL 2 dado el candidato NHL3.

30 Se hará referencia a la opción c con referencia a la figura 19. En la primera etapa c1, c se elimina de las opciones en la variable de estado. En la etapa C2 se realiza una comprobación para una única interfaz, cuya dirección de red coincide con la dirección de red NHL3. Si no se encuentra una interfaz única, el proceso vuelve al bucle principal. Si se encuentra una interfaz única, y es un nombre de interfaz de salida, se realiza el proceso de indicio de VLAN que se describirá con referencia a la figura 21. Después de la etapa c4, se obtiene el puerto conectado empleando el

35 proceso de obtener el puerto conectado que se muestra en la figura 22.

En la etapa C7, se determina si se encuentra un puerto emparejado. Si es así, el puerto de salida, el puerto emparejado y el dispositivo emparejado se añaden a la ruta y el dispositivo de enfoque se fija en el dispositivo emparejado. Las opciones disponibles se restablecen a CSRACr en C8. Si ningún puerto emparejado se encuentra

40 en la etapa C7, el proceso vuelve al bucle principal.

Ahora se hará referencia a la figura 20 para describir la Opción A. En la etapa A1, la opción A se retira de la secuencia de opción en la variable de estado. En la etapa A2, se encuentra la entrada ARP SNMP para asignación de NHL3 a NHL2. Si no se encuentra ninguna asignación, el proceso vuelve al bucle principal.

45 Si se encuentra una asignación, el proceso utiliza SNMP para encontrar el ifIndex de la interfaz desde la que se aprendió la relación. En la etapa A5, se determina si una interfaz única se ha encontrado o no. Si no es así, el proceso vuelve al bucle principal. Si es así, el proceso pasa a la etapa A6, donde se determina si hay un nombre de interfaz de salida disponible. Si lo hay, el proceso de indicio de VLAN es instigado como se describirá con referencia a la figura 21. Luego, en A8, se instiga el proceso de obtener el puerto conectado, como se ilustra en la figura 22.

50 Si como resultado del proceso de obtener el puerto conectado se encuentra un puerto emparejado, se añade el puerto de salida a la ruta, el puerto emparejado y el dispositivo emparejado se añaden a la ruta y el dispositivo de enfoque se fija en el dispositivo emparejado. Además, las opciones disponibles se restablecen a CSRACr. Si no se encuentra ningún puerto emparejado, el proceso vuelve al bucle principal.

55 Ahora se hará referencia a la figura 21 para explicar el proceso de indicio de VLAN. Este proceso se utilizó en la Opción A en la etapa A7, en la Opción C en la etapa c5, en la Opción R/r en la etapa R13 y en la Opción S en la etapa S116. Además, se utiliza en uno de los procesos de preparación que aún no ha sido abordado. El proceso comienza en la etapa VL1 con un nombre de puerto de acceso. En la etapa VL2 se determina si el nombre está en la forma de "VL + un número", y si el número se extrae y se almacena como un indicio de VLAN en la etapa VL3.

60 Si no es así, entonces en el sistema de gestión de la red se solicita una lista de todas las VLAN en la interfaz. La etapa VL5 comprueba cualesquiera VLAN en conflicto. Si no hay ninguna, entonces la única VLAN se almacena como un indicio de VLAN en la etapa VL6. Si hay VLAN en conflicto, cualquier indicio de VLAN que ya ha sido almacenado se deja sin cambios.

65

Los indicios de VLAN abordan un problema que puede surgir en determinadas redes que utilizan una técnica de conmutación llamada STP (protocolo de árbol de extensión) que se utiliza para evitar tener bucles lógicos en porciones conmutadas (capa 2) de la red (para evitar el bucle de tráfico alrededor de manera infinita). El protocolo se utiliza para decidir a dónde un dispositivo de conmutación debe enviar un paquete dado.

5 Es decir, si el dispositivo se está conmutando, el conmutador mirará al encabezado de la capa 2 (MAC/Ethernet) y mirará a la dirección de la capa 2 de destino (NHL2) y luego consultará una base de datos interna (la FDB - base de datos de reenvío) para determinar desde cuál de sus puertos se debe enviar el paquete. Muchas empresas utilizan una extensión de STP llamada PVSTP (protocolo de árbol de expansión por VLAN), por lo que cada paquete está marcado con un identificador de VLAN también. Entonces el conmutador mantiene FDB separadas - una por cada VLAN.

15 Esto se hace en parte por eficiencia y en parte para permitir topologías virtuales más complejas. Así, es perfectamente posible (y no es raro) que dos paquetes con la misma capa 2 de destino abandonen por diferentes puertos, ya que están etiquetados como que están en diferentes VLAN, a pesar de que su destino sea el mismo dispositivo/puerto.

20 La consecuencia de esto es que el proceso no puede simplemente arrastrar todos los FDB por VLAN hasta que se encuentra una coincidencia. Es importante saber a priori qué VLAN del paquete se etiqueta como un miembro.

Este etiquetado de la VLAN puede producirse en diferentes lugares de la red, por ejemplo, en el puerto de acceso de origen - es decir, cuando el dispositivo de origen está conectado físicamente, o en otro lugar en la red - no es raro que una etiqueta de VLAN sea sustituida por otra (esto se llama enrutamiento entre VLAN).

25 Por ejemplo, dado un paquete que llega al dispositivo D de la red (en la ruta A->B->C->D) D sólo puede ser consultado por la interfaz de salida correcta si se conoce la VLAN del paquete desde A estuviera en el punto cuando alcanza D. Podría ser, por ejemplo, que A coloque el paquete en la VLAN 100, B pase (usando la VLAN 100), entonces C cambie de 100 a 200 y luego D lo conmute usando la VLAN 200.

30 Por esta razón, es necesario 'llevar' un indicio de VLAN en toda la red desde nuestro dispositivo de origen a nuestro dispositivo de destino como parte del paquete hipotético que rastreamos. Así, en su caso el indicio de VLAN se usa, anula, restablece o actualiza.

35 Como ya se ha mencionado, la figura 23 ilustra el proceso de findRoutel iterative que se utiliza en las opciones R y r. El proceso consiste en un bucle de ruta de hallazgo que comienza en la etapa F1 y termina en una verificación del límite de la ruta F2. El proceso findRoutel iterative determina entonces si se ha encontrado una ruta y permite un índice de salida que se encuentra perteneciente a la ruta.

40 Antes de embarcarse en el bucle principal, hay tres procesos de preparación que se implementan para establecer el estado de entrada para la primera iteración del bucle. Un primer proceso de preparación se muestra en la figura 24, que establece como punto inicial el conmutador de acceso o dispositivo de red identificado por el dispositivo de origen (indicado en la figura 24 como el lado del cliente). Del mismo modo, un conmutador de acceso o dispositivo de red se almacena como un punto de detención, en base al dispositivo de destino, que se refiere en la figura 24 como el lado del servidor. En la Figura 24, la IP del cliente y la IP del servidor son el origen y destino respectivamente.

La figura 25 es un proceso de preparación para el establecimiento de las direcciones NHL3 y NHL2 de estado de entrada inicial.

50 La figura 26 ilustra un tercer proceso de preparación que establece el dispositivo de enfoque para el estado de la entrada inicial.

55 Obsérvese que el primer proceso de preparación de la figura 24 conduce al segundo proceso de preparación de la figura 25, y el segundo proceso de preparación de la figura 25 conduce al tercer proceso de preparación de la figura 26. El tercer proceso de preparación lleva al punto de entrada principal 4 del bucle principal que se muestra en la figura A.

60 Un algoritmo alternativo para identificar una ruta de acceso se describe ahora. Será evidente que las explicaciones detalladas en partes de la siguiente descripción se aplican a las partes equivalentes del proceso de iteración de bucle que se acaba de describir.

### **Hallar el primer salto en la capa 3**

65 El primer salto se localiza hallando el salto siguiente inicial (el salto siguiente desde el dispositivo fuente) en la capa 3 (NHL3). En la explicación siguiente, se utiliza frecuentemente el término "consulta". Las consultas son generadas y estructuradas como se describe con más detalle más adelante. La finalidad de una consulta es localizar una

dirección de salto siguiente y el puerto de salida de un dispositivo de enfoque al que se dirige la consulta. La dirección NHL3 inicial puede determinarse consultando en primer lugar un dispositivo fuente X usando la dirección IP de destino. Es decir, se intenta consultar la tabla de enrutamiento en el dispositivo fuente para NHL3 y el puerto de salida. Si no se halla ninguna ruta, y el dispositivo fuente tiene un conmutador de acceso a capa 3, este conmutador de acceso a capa 3 es consultado con respecto a NHL3 usando la dirección IP de destino. Si eso no tiene éxito, se consulta la puerta de enlace por defecto del dispositivo fuente para conocer la NHL3. Si eso no tiene éxito, se realiza una consulta usando la dirección IP de destino al conmutador de acceso para la puerta de enlace por defecto. Si no se halla la dirección NHL3, esto se considera como un fallo. Esto no quiere decir que el algoritmo ha fallado, sino que en este punto puede haberse identificado un punto de fallo en la ruta. Alternativamente, puede haber otras razones por las que no se halló NHL3.

**Sembrar la ruta**

Para sembrar la ruta, se añade el dispositivo fuente a la ruta cuando se haya localizado. La interfaz de salida del dispositivo fuente se localiza y añade a la ruta. Si se halla NHL3 a partir de la tabla de enrutamiento en el dispositivo fuente, se añade a la ruta la interfaz de salida de dispositivo fuente para esta dirección NHL3. Como se explica más adelante, la dirección de capa 2 (NHL2) correspondiente a la dirección de capa 3 (NHL3) puede averiguarse. Si no se halla ningún puerto de salida para NHL3 a partir de la tabla de enrutamiento en el dispositivo fuente, se usa la tabla de envío de capa 2 en el dispositivo fuente para NHL2 para hallar el puerto de salida. Si se halla, entonces se añade dicho puerto de salida a la ruta.

**Algoritmo de descubrimiento de ruta**

La consulta enviada desde el ordenador supervisor 16 al dispositivo fuente X se representa como una flecha directa en la figura 2a, pero, de hecho, la podría implementar en la red de la figura 1 el ordenador supervisor 16 emitiendo un mensaje o paquete dirigido al dispositivo fuente X. La consulta pregunta al dispositivo fuente la IP de salto siguiente (y puerto de salida) para la IP terminal (IP de destino), que es la dirección de capa 3 del punto de destino Y. La finalidad es hacer que el dispositivo fuente X suministre una respuesta que incluya nHL3 y el puerto de salida para NHL3 (la dirección de IP terminal). Véase la etapa S1 de la figura 3 y la figura 2a.

Como se ha explicado anteriormente, puede haber situaciones en las que el dispositivo fuente no pueda suministrar la información necesaria. Otras posibilidades mencionadas anteriormente para obtener el primer dispositivo de "enfoque" incluyen consultar al conmutador de acceso conectado sobre información de enrutamiento de capa 3 (en caso de que el conmutador de acceso sea un conmutador de capa 3); si esto falla, el algoritmo consulta el conmutador de acceso conectado con respecto a una puerta de enlace por defecto y la dirección IP de la puerta de enlace por defecto usada como la primera NHL3.

En la etapa S2, la dirección de capa 2 (MAC) de salto siguiente se resuelve a partir de la dirección NHL3 y NHL2 se pone a esta dirección MAC. Esto se puede lograr consultando una tabla de mapeado 91 que mapea direcciones L3 a L2. Tal tabla de mapeado es una tabla ARP (otras incluyen "mapeado directo" y descubrimiento de vecinos). Esto puede ser la ARP de dispositivo fuente, ARP de dispositivo de salto L3 siguiente o ARP en cache global usando una consulta ARP descrita más adelante. El puerto de salida identificado en la etapa S1 se añade al registro de ruta S1A. En la etapa S3, el conmutador de red de inicio (y puerto) se halla usando la posición de host final en cache (a partir de consultas CAM de conmutador), y se pone como el dispositivo de enfoque. En la etapa S4, se halla el conmutador de red terminal usando la posición de host final en cache (a partir de consultas CAM de conmutador). El conmutador de inicio se añade al registro de ruta.

El método está preparado ahora para entrar en un bucle de identificación de ruta. En la etapa S5 se determina si NHL2 es conocido. En caso afirmativo, el bucle pasa a la etapa S5A. En caso negativo, el proceso lleva a cabo la etapa S5B para resolver NHL2 por una consulta ARP en el dispositivo de enfoque o el dispositivo NHL3. La generación de una consulta para correlacionar una dirección de capa 3 con una dirección de capa 2 se explica con más detalle más adelante con referencia a la figura 7. En resumen, para el dispositivo que es consultado, se obtiene una lista de índices de interfaz (ifíndices) a partir de la topología de red o recorriendo ifIndex a partir de la tabla de interfaces del dispositivo propiamente dicho. Cada ifIndex para el dispositivo se combina con la dirección NHL 3 para generar un conjunto de claves a incluir en la consulta al dispositivo. Así, una consulta conteniendo estas claves es formulada y transmitida al dispositivo de enfoque. El dispositivo de enfoque produce cero o una respuesta exitosa.

Si fallan las dos técnicas anteriores para resolver NHL2, se accede a ARP global. En la etapa S5A, se determina si la dirección NHL3 está o no en el dispositivo de enfoque actual.

Si NHL3 no está en el dispositivo actual, en la etapa S6, el proceso envía una consulta para hallar la entrada FDB de capa 2 para que NHL2 obtenga el puerto de salida. La generación de una consulta en la capa 2 se explica más adelante. Si tiene éxito, se añade el puerto de salida al registro de ruta (S6A), se usa la topología en cache 3 para hallar el puerto y el dispositivo en el extremo del enlace (S7) , se añade el dispositivo a la ruta (S7A) , y el dispositivo de enfoque se pone al dispositivo que acaba de localizarse en el extremo del enlace (salto L2). Las etapas S6A, S7 y S7A pueden denominarse un salto L2. En este punto, consúltese la figura 2b. En la etapa S5A, el dispositivo de

enfoque es el dispositivo A. este recibe una consulta para hallar la entrada FDB de capa 2 y devuelve el puerto de salida. El dispositivo que se determina que está en el extremo de dicho enlace es el dispositivo B (figura 2c) que recibe una consulta con NHL3 todavía puesto a la dirección IP de destino.

5 Si no se halló una entrada FDB de capa 2, o si en S5A se determinó que NHL3 se alojaba en el dispositivo de enfoque, en la etapa S8 se realiza una consulta de ruta para determinar si la ruta L3 se halla en el dispositivo de enfoque a la dirección IP de destino. La consulta de ruta puede ser una consulta de ruta única o recursiva, que se explican más adelante. Esto establece una IP de salto siguiente y una interfaz de salida. Si no se halla la ruta L3, se indica una ruta rota y el proceso se para - S8A. En la etapa S9 (salto L3) se añade la interfaz de salida de tabla de enrutamiento a la ruta, NHL3 se pone a la nueva dirección IP de salto siguiente, y el proceso consulta al dispositivo para averiguar la dirección de capa 2 de NHL3. Si no se puede resolver NHL2, NHL2 se pone a "desconocido".

10 En la etapa S10, la dirección NHL3 actual se compara con la dirección IP de destino. Si NHL3 no es la IP de destino (es decir, el algoritmo de identificación de ruta aún no está en el segmento L2 final) , en la etapa S11 se usa la topología en cache para hallar el puerto y dispositivo en el extremo de enlace, el dispositivo se añade al registro de ruta y el enfoque se pone a este dispositivo. El proceso consulta entonces (S12) si el dispositivo de enfoque es el dispositivo terminal. Si el dispositivo de enfoque no es el dispositivo terminal, el proceso vuelve a la etapa S5, pero usando NHL3 y NHL2 puestos en la etapa 9.

## 20 **Terminación**

El algoritmo termina cuando se llega al dispositivo terminal y se añaden el puerto terminal y el servidor de destino a la ruta. Otras condiciones de terminación evitan que el algoritmo itere indefinidamente. En cada iteración de la ruta, se inicia una iteración poniendo un señalizador conmutado a falso y un señalizador enrutado a falso. Cuando tiene lugar un salto L2 (S7) el señalizador conmutado se pone a verdadero; cuando tiene lugar un salto L3 (S9) , el señalizador enrutado se pone a verdadero. Como ya se ha mencionado, el puerto de salida se determina a partir de un dispositivo de enfoque, y la topología de red se usa para hallar el dispositivo unido y el puerto de entrada del dispositivo unido. Por cada iteración se almacena la combinación de:

30 "Dispositivo de enfoque, NHL2, NHL3".

Si el dispositivo de enfoque, NHL2 o NHL3 han cambiado y la nueva combinación de "dispositivo de enfoque, NHL2, NHL3" se ha visto antes, aparece un evento de bucle detectado y el bucle se para. Si no se ha alcanzado el límite de bucle, y se ha producido enrutamiento o conmutación (es decir, los señalizadores enrutado o conmutado son verdaderos) y el dispositivo de enfoque no es igual al dispositivo terminal, itera de nuevo. Cada vez se averigua si se ha alcanzado el límite de bucle de iteración. Si se ha alcanzado, el algoritmo termina.

40 Cuando cesa la iteración, si el dispositivo de enfoque es el dispositivo terminal, el dispositivo terminal se añade a la ruta. Si el dispositivo de enfoque no es el dispositivo terminal, pero el algoritmo se ha parado, se reporta un error de que el algoritmo de hallazgo de ruta habrá terminado en un lugar inesperado. Si el dispositivo terminal es un conmutador de acceso, el puerto de salida de conmutador de acceso se añade desde "localizar destino" (S4) a la ruta y el dispositivo de destino derivado del puerto de salida de conmutador de acceso se añade a la ruta - el algoritmo termina entonces. Si el dispositivo terminal es igual al dispositivo destino, el algoritmo termina. El detalle del algoritmo se explicará ahora con más detalle.

## 45 **Ejemplo específico**

La figura 4 representa un resultado de la operación del algoritmo de identificación de ruta. Es decir, proporciona la ruta que un paquete de datos procedente del dispositivo fuente X dirigido a dispositivo destino Y tomaría por la red al tiempo en que el algoritmo de identificación de ruta consulta la red. La ruta se representa incluyendo dispositivos A-J que forman parte del registro de ruta. El registro de ruta incluye los puertos de entrada y de salida de cada uno de los dispositivos.

55 Observando de nuevo la red original de la figura 1, se puede ver que la primera parte del registro de ruta representado en la figura 4 deriva de la red de la figura 1, donde se han usado letras correspondientes para denotar los dispositivos seleccionados por el conmutador o dispositivo de enrutamiento previos. Cuando el algoritmo de identificación de ruta opero, el dispositivo de enrutamiento B había determinado enviar el paquete al conmutador C. Sin embargo, sin usar la presente invención, habría sido sumamente difícil hacerlo en tiempo real. El dispositivo de enrutamiento B tema igualmente una opción de enrutar el paquete al enrutador F en la red de núcleo. Consultando el dispositivo de enrutamiento B en tiempo real (o más o menos en tiempo real) , en base al paquete hipotético dirigido al destino Y, el dispositivo de enrutamiento B devuelve la decisión que habría tomado si hubiese llegado un paquete real con esa dirección. Averiguando que el dispositivo de enrutamiento B enviara el paquete al conmutador C, y estableciendo a continuación que el conmutador C está conectado, el extremo lejano de su dispositivo de enrutamiento de puerto de salida D, C y D se ha añadido al registro de ruta 81. De esta forma, el algoritmo de identificación de paquete ha pasado a través de la ruta que el paquete hipotético habría tomado al tiempo en que el algoritmo de identificación de ruta consulta los dispositivos en la red. El recuadro adyacente al dispositivo de

enrutamiento D denota los parámetros para NHL3 y NHL2, es decir, NHL3 se pone a la dirección IP del dispositivo E que se ha establecido como el dispositivo de extremo lejano para el dispositivo de enrutamiento D en base a la tabla de enrutamiento actualmente activa en D, y NHL2 ha sido establecida como la dirección MAC para el dispositivo E por el dispositivo consultante D para su entrada ARP para el dispositivo E.

5

### **Topología de red**

Como se ha mencionado previamente, la topología de red incluye tanto interconectividad de dispositivos de red como localización de host final. La topología de red 3 puede facilitar un servidor de topología que proporcione detalles de conexiones de puerto a puerto. Así, cuando un puerto de salida es identificado en un dispositivo, el puerto de entrada del dispositivo conectado se puede conocer usando conexión de puerto a puerto identificada en la topología. Ambos puertos de salida y de entrada pueden añadirse al registro de ruta. El servidor de topología también proporciona una CAM global, un ARP global y credenciales de dispositivo. Además, por cada dispositivo registrado en la topología hay preferiblemente una lista de índices de interfaz (IfIndex), y una lista VLAN (red de área local virtual). Se explican mejor aquí. Cuando se devuelve una respuesta al ordenador supervisor 16, el ordenador supervisor consulta la topología 3 en el orden siguiente al manejar respuestas de capa 2. En este contexto, una respuesta de capa 2 es una respuesta que ha identificado un puerto de salida de un dispositivo conmutador de capa 2. El orden de consulta es CDP, LLDP, STP y SONMP, IPv6 ND.

10

15

20

### **Localización de dispositivo fuente**

Como se ha mencionado antes, la localización del primer dispositivo en la ruta (el dispositivo conectado al dispositivo fuente) no es necesariamente sencilla. En una realización, el ordenador supervisor 16 implementa el algoritmo para intentar hallar en primer lugar la fuente como un host conectado y, si eso falla, intenta hallar la fuente como un dispositivo de red. Al intentar hallar la fuente como un host conectado, consulta al dispositivo fuente con respecto a la dirección de capa 2 (MAC) para la fuente IP. Esto se puede realizar de la misma forma que la consulta en un dispositivo de enfoque como se ha descrito anteriormente en la etapa S5B. Es decir, el proceso envía una consulta para hallar la entrada ARP para la dirección IP fuente.

25

30

Si no hay dirección de capa 2 procedente del dispositivo fuente, se consulta la tabla ARP en cache global en el servidor de topología. En la realización descrita, estas se denominan tablas ARP, pero se puede utilizar cualesquiera tablas que mapean las direcciones de capa 3 a capa 2. Si se halla una dirección MAC que corresponde a la dirección IP fuente, se consulta el servidor de topología con respecto a la localización MAC de IP fuente consultando tablas de envío de capa 2 en cache globales en el servidor de topología para hallar puertos que hayan tenido tráfico procedente de esta dirección MAC. Se espera que el servidor de topología devuelva una localización de MAC fuente única quitando múltiples coincidencias (la fuente MAC vista en muchos puertos), filtrando puertos señalizados como líneas principales, puertos con excesivos números de MAC (las entradas FDB de los puertos de conmutador de acceso tienen típicamente una sola dirección MAC 'vista'), puertos con topología entre redes (por ejemplo, si un puerto tiene información de adyacencia CDP no puede ser un puerto en un conmutador de acceso), etc.

35

40

Si no se puede hallar la fuente como un host conectado, se intenta hallar la fuente como un dispositivo de red. Esto se puede lograr consultando el servidor de topología con respecto a todas las direcciones IP halladas en todos los dispositivos de red gestionados para ver si la dirección IP está en un dispositivo de red. Si esta, ese dispositivo de red se pone como el dispositivo de enfoque.

45

### **Localización de dispositivo destino**

Se aplican consideraciones similares a la localización del dispositivo destino. En primer lugar, se intenta hallar el dispositivo destino como un host conectado, y si eso falla, se intenta hallar el destino como un dispositivo de red. Para hallar el dispositivo destino como un host conectado, se consulta el dispositivo destino con respecto a su dirección de capa 2, o se consultan tablas de mapeado de capa 3 a capa 2 en cache globales en el servidor de topología (igual que con respecto al dispositivo fuente explicado anteriormente). Después se consultan tablas de envío de capa 2 en cache globales en el servidor de topología para hallar puertos que han tenido tráfico procedente de esta MAC (de nuevo, como se ha descrito anteriormente con referencia a la localización de dispositivo fuente).

50

55

Para hallar el destino como un dispositivo de red si falla lo anterior, el servidor de topología puede ser consultado con respecto a todas las direcciones IP halladas en todos los dispositivos gestionados para ver si la dirección IP está en un dispositivo de red. El dispositivo de red se puede poner entonces como el dispositivo terminal.

60

### **Utilidad por salto**

Con el fin de implementar el algoritmo de identificación de ruta, el ordenador supervisor 16 ejecuta un programa de ordenador como se ha explicado. Este programa de ordenador proporciona una utilidad que maneja consultas "por salto". Es decir, el algoritmo de identificación se basa en enviar una consulta desde el ordenador supervisor a un dispositivo de enfoque y recibir del dispositivo de enfoque un puerto de salida que puede ser usado para acceder a la topología. Esto no se puede lograr necesariamente con una sola consulta. Esto se encuentra en el proceso de la

65

figura 23 (con el subproceso de la figura 27), y la figura 28, más de una consulta puede ser generada de forma autónoma por el proceso de ejecución del bucle para lograr un resultado. El principio de una utilidad "por salto" se describe más claramente con referencia al algoritmo alternativo mencionado anteriormente. Como se ha descrito anteriormente, el algoritmo requiere un salto siguiente inicial en la capa 3 (NHL3). La utilidad intenta consultar una

5 tabla de enrutamiento en el dispositivo fuente para NHL3 y el puerto de salida, usando la dirección IP de destino. Si no se halla ninguna ruta, consulta la tabla de enrutamiento en el conmutador de acceso en caso de que sea un conmutador de capa 3 (que es el primer dispositivo conectado al dispositivo fuente para NHL3). Si no se halla ninguna ruta, el dispositivo fuente es consultado con respecto a la puerta de enlace por defecto para NHL3. Si no se halla ninguna ruta, el primer dispositivo es consultado con respecto a una puerta de enlace por defecto.

10 Para consultar una tabla de enrutamiento para hallar NHL3 (como se ha descrito anteriormente) , se halla una ruta para la dirección IP en cuestión (la dirección IP 'buscada') consultando el dispositivo de enrutamiento usando una técnica de manipulación especulativa explicada más adelante. Si se halla la ruta, pero no se especifica el puerto de salida, se devuelve la dirección IP de salto siguiente y se usa como NHL3. Si se halla la ruta con una interfaz de salida ifIndex superior a cero, el puerto de salida es devuelto con la dirección NHL3 y se añade el puerto de salida a la ruta. Si se halla la ruta con la interfaz de salida ifIndex igual a cero, la utilidad reitera poniendo la IP buscada a la IP de salto siguiente (de la consulta previa) y hallando la ruta para la IP buscada consultando el dispositivo usando manipulación especulativa (como se explica más adelante). Esto se repite hasta que el ifIndex devuelto sea no cero.

20 La etapa de hallar la ruta para la IP buscada usa la técnica de manipulación especulativa para devolver una entrada de ruta. Si se halla la entrada de ruta, la utilidad sondea la dirección de salto siguiente a partir de ipRouteNextHop.NetworkAddress. La utilidad también sondea la interfaz de salida a partir de ipRouteIfIndex.NetworkAddress y sondea ipRouteType.NetworkAddress. Si ipRouteType es 'directo', la IP buscada se pone al salto siguiente, puesto que un tipo de ruta IP de directo indica que está conectado directamente al segmento de red.

25 Es posible que se devuelvan múltiples coincidencias de una tabla de enrutamiento en un dispositivo. En ese caso, es apropiado determinar si se están utilizando múltiples rutas, por ejemplo, cuando un dispositivo es responsable de tráfico de equilibrio de carga. Si solamente se está usando activamente una sola ruta, deberá determinarse la ruta activa. Si se están utilizando múltiples rutas, la ruta podría dividirse en este punto y el registro de ruta podría

30 contener los resultados del algoritmo de hallazgo de ruta aplicado a toda y cada ruta hallada desde este punto en adelante. En muchos casos, múltiples opciones de enrutamiento en un dispositivo son indicativas de un dispositivo que enruta inteligentemente en base a diversa métrica. Esta métrica también puede ser consultada y devuelta para registro en el ordenador supervisor.

35 La utilidad también es responsable de hallar el salto siguiente inicial en la capa 2 consultando la tabla de mapeado de capa 3 a capa 2 91 en el dispositivo de enfoque. Si no se halla la dirección de capa 2, donde el dispositivo de enfoque es el dispositivo fuente, la utilidad consulta el conmutador de acceso (si es un conmutador de capa 3 deberá proporcionar un mapeado de capa 3 a capa 2). Si no se halla la dirección de capa 2, la utilidad consulta las tablas ARP en cache global en el servidor de topología 3. Una consulta de una dirección de capa 2 en un dispositivo se lleva a cabo como se ha explicado anteriormente con referencia a la etapa S5B.

45 Si la dirección NHL 3 no está en el dispositivo de enfoque, la utilidad sondea el dispositivo de enfoque con respecto a un puerto de salida para la dirección de capa 2 NHL2. La etapa de sondear el dispositivo de enfoque con respecto al puerto de salida NHL2 incluye sondeo específico VLAN (red de área local virtual). Es decir, incluye la etapa de establecer en que VLAN está participando el dispositivo según la topología 3 y como se ha registrado en el dispositivo. Estas VLAN se usan para ayudar a hallar entradas de tabla de envío para VLAN específicas (las FDB se dividen a menudo según las VLAN con las que están relacionadas - por ejemplo, para el Protocolo de árbol de expansión por VLAN (PVSTP) es necesario realizar las consultas FDB en el contexto de cada VLAN para intentar hallar una coincidencia).

50 Si no se halla el puerto de salida a partir de la capa 2 FDB (usando una VLAN específica o la VLAN nativa) , entonces la utilidad intenta hallar que interfaz se dirige hacia NHL2 a partir de registros ARP sondeando ipNetToMedia-PhysAddress 71 (figura 7). Es decir, la utilidad intenta aprender de que interfaz se aprendió la relación de capa 2 a capa 3.

55 Una vez que la utilidad ha hallado un puerto de salida usando la dirección de capa 2, añade el puerto de salida al registro de ruta y usa el servidor de topología 3 para hallar el puerto remoto unido al puerto de salida. Este puerto remoto se registra como el puerto de entrada en el dispositivo siguiente.

60 **Canales de puerto/Puertos multiplexados**

Si no se halla ningún puerto remoto, o el nombre de puerto de salida impone el uso de puertos de capa superior o inferior, la utilidad comprueba los puertos de capa inferior o los puertos de capa más alta. Es decir, puede haber un escenario donde haya un mapeado de salidas de ruta virtual a puertos físicos. Para que el algoritmo de identificación de ruta tenga éxito, tiene que identificar un puerto de salida físico para acceder al servidor de topología. En un escenario donde la comprobación de puertos de capa inferior revela la presencia de puertos de capa inferior, estos

puertos de capa inferior pueden ser usados como los puertos de salida y se accede al servidor de topología para hallar los puertos remotos (puertos de entrada del dispositivo siguiente) unidos a los puertos de salida. En este punto, la ruta se divide en múltiples rutas separadas, cada una de las cuales es seguida independientemente desde este punto en adelante.

5 Si se identifican puertos de capa más alta, el puerto de capa más alta se usa para el puerto de salida. El servidor de topología se usa para hallar el puerto remoto unido a este puerto de salida de capa más alta.

### 10 **Salto siguiente**

Poner los señalizadores enrutado y conmutado a falso. Usando el servidor de topología o consultas directas al dispositivo de enfoque, conocer si el dispositivo de enfoque aloja o no la dirección NHL3 IP en alguno de sus puertos. Si aloja la dirección NHL3 IP, la utilidad pasa entonces a consultar la tabla de enrutamiento de dispositivo de enfoque con respecto a rutas a la IP de destino usando la técnica de manipulación especulativa. Si la utilidad localiza una ruta candidata, la dirección de capa 2 NHL2 siguiente se pone consultando el dispositivo de enfoque (o tablas ARP en cache global) para mapeado de capa 3 a capa 2 y el señalizador enrutado se pone a verdadero. Si NHL3 es igual a la IP de destino, eso indica que la utilidad ha llegado al último dispositivo de capa 3 más próximo al destino de modo que todavía no hay necesidad de mover este dispositivo puesto que el salto siguiente sería un salto de capa 2. Por lo tanto, la utilidad añade los puertos de salida de ruta candidato a la ruta. Si NHL3 no es igual a la IP de destino, indica que no está en el segmento de capa 2 final y el puerto de salida de ruta candidato se añade a la ruta.

25 Si no se produjo enrutamiento durante esta iteración (el señalizador enrutado todavía está puesto a falso) , entonces la utilidad sondea el dispositivo de enfoque con respecto a un puerto de salida para la dirección de capa 2 NHL2. La etapa de sondear el dispositivo de enfoque con respecto al puerto de salida NHL2 incluye sondeo específico de VLAN (red de área local virtual) (como se ha descrito anteriormente). Si el puerto de salida no se halla a partir de la FDB de capa 2 (usando una VLAN específica o la VLAN nativa), la utilidad intenta hallar que interfaz se dirige hacia NHL2 desde registros ARP sondeando con respecto a ipNetToMediaPhysAddress 71. Es decir, la utilidad intenta aprender de que interfaz se aprendió la relación de capa 2 a capa 3. Una vez que la utilidad ha hallado un puerto de salida usando la dirección de capa 2, añade el puerto de salida al registro de ruta y usa el servidor de topología 3 para hallar el puerto remoto unido al puerto de salida. Este puerto remoto se registra como el puerto de entrada en el dispositivo siguiente. Si se halla un puerto de salida usando consultas FDB o consultas ARP, el señalizador conmutado se pone a verdadero.

35 Si, al consultar el servidor de topología, no se halla ningún puerto remoto, o el nombre de puerto de salida impone el uso de puertos de capa superior o inferior, entonces se realiza una comprobación de puertos de capa inferior o más alta, como se ha descrito anteriormente. Si se halla un puerto de salida, se añade a la ruta, el dispositivo conteniendo el puerto se añade a la ruta y el dispositivo de enfoque se pone al dispositivo remoto.

40 Esta etapa de "Salto siguiente" se repite hasta que se llega a un límite preestablecido en el número de iteraciones o la ruta llega a un final (es decir, no se produjo conmutación ni enrutamiento).

45 Si el proceso termina en el dispositivo terminal previamente identificado y es dispositivo es un conmutador de acceso, el puerto de salida se añade desde "localizar destino" al registro de ruta, y el dispositivo destino se añade al registro de ruta. Si el dispositivo terminal es el dispositivo destino propiamente dicho, la utilidad termina.

Las figuras 11A a 11D muestran un diagrama de flujo de la operación de la utilidad ejecutada en el ordenador supervisor.

### 50 **Equilibrador de carga**

Como se ha mencionado anteriormente, si el dispositivo de enfoque es el dispositivo terminal, el dispositivo terminal se añade con el destino al registro de ruta. Si el dispositivo terminal es un equilibrador de carga, entonces se obtiene la IP virtual para mapeado de grupo de servidores para el equilibrador de carga. Esto permite identificar el servidor para mapeado de servidor físico para el equilibrador de carga. La ruta se retiene hasta la ruta "raíz" (hasta que el dispositivo equilibrador de carga). Entonces, por cada dirección IP de servidor físico, se ejecuta una utilidad de descubrimiento de ruta adicional desde el equilibrador de carga a la dirección IP de servidor físico, anteponiendo la ruta "raíz" a cada ruta adicional.

### 60 **Consulta de tabla de enrutamiento**

65 Uno de los factores que hacen el algoritmo de ruta especialmente eficiente es la capacidad de generar eficientemente una consulta a un dispositivo de enrutamiento, es decir, generar una consulta a la que el dispositivo de enrutamiento puede responder en un corto período de tiempo sin carga significativa. La figura 5 ilustra la estructura de una tabla de ruta lineal direccionable mediante SNMP. Para establecer una ruta a un destino concreto, ipRouteDest es el índice requerido a la tabla de ruta. Esto se indica con 48 en la figura 5. Las entradas de interés en

la tabla son ipRouteIndex 50 que define la interfaz de salida, ipRouteNextHop 52 que define la dirección IP del salto siguiente (IP de salto siguiente) e ipRouteType 54 que define el tipo de entrada de enrutamiento (no valida/directa/indirecta). El acceso a la tabla requiere normalmente el conocimiento de ipRouteMask 56: esto permitiría localizar una dirección IP de red específica. Sin embargo, como se puede ver en la figura 5, IpRouteMask  
 5 propiamente dicho esta embebido en ipRouteEntr y por lo tanto no se conoce que está puesto en la consulta. Lo que hay que hacer es hallar una coincidencia para:

<IP de interés> & <ipRouteMask.X> == <ipRouteDest.X>  
 con el fin de hallar la clave IpRouteDest 48 que representa el índice a la tabla.

10 La figura 27 ilustra el proceso Como observaron los autores de la invención, solamente hay 33 posibilidades de IpRouteMask (/32.../0) , es decir 255.255.255.255, 255.255.255.254, 255.255.255.252,... 0.0.0.0. Un numero de estos produce ID de red duplicadas para la misma dirección IP, a causa del número de ceros en la dirección IP. Se produce una lista de las 33 máscaras de red posibles (Z2), y se aplica a la dirección IP (Z3). La figura 6 representa la aplicación de las 33 máscaras de red a la dirección IP 10.44.1.213 = OA.2C.01.D5 = 0000 1010 0010 1100 0000 0001 1101 0101.

15 Esto genera 12 valores únicos (etiquetados 32, 31, 29, 27, 25, 24, 23, 13, 12, 10, 6, 4). Así, ahora solamente hay que hacer 12 consultas SNMP (que pueden presentarse en un solo paquete de consulta) para hallar la ruta. Después de las etapas Z4 a y Z5 para determinar si están permitidas rutas por defecto y quitar redes consiguientemente, los 12 resultados se comparan con la tabla de ruta del dispositivo de enfoque y cuando se halla una coincidencia, los elementos requeridos ipRouteIndex (egressIndex) , ipRouteNextHop e ipRouteType son recuperados (Z12) y devueltos en una respuesta al ordenador supervisor 16.

20 La interfaz de resultado se pone a egressInterface (Z13).

La reducción del número de consultas requerido para hallar la ruta se denomina aquí "manipulación especulativa" y permite realizar la consulta de tabla de ruta en tiempo real de manera muy eficiente.

30 Al examinar tablas de enrutamiento reales, no es insólito que la ruta hallada para una dirección IP dada no tenga una interfaz de salida valida y solamente proporcione una dirección de salto siguiente. En estos casos, la dirección de salto siguiente se usa para una consulta posterior de la tabla de enrutamiento para intentar obtener una interfaz de salida para dicha dirección de salto siguiente. Esta reutilización de la dirección de salto siguiente se repite hasta que se obtiene una interfaz de salida.

35 Según este enfoque, en una primera etapa una consulta de hallazgo de ruta única usa manipulación especulativa para hallar una entrada de enrutamiento para la dirección IP especificada (IPx) como acaba de esbozarse. Si el ipRouteType asociado es "directo", se devuelven IPx (e ipRouteIndexx) en una respuesta al ordenador supervisor como el salto siguiente. Es decir, está conectado directamente y por lo tanto no tiene salto siguiente de capa 3.

40 Si el ipRouteType asociado no es directo, se devuelven ipRouteNextHop e ipRouteIndex en respuesta al ordenador supervisor.

45 El proceso de hallazgo de ruta también toma en cuenta tablas de enrutamiento entre dominios sin clase IP que son más difíciles de consultar. En este caso, si la etapa Z10 no da lugar a una dirección IP, el proceso pasa a la etapa Z14 donde se envía una consulta SNMP (Obtener siguiente) al dispositivo, usando IPcidrRouteDest + dirección de red + mascara de red. Si el resultado no es una dirección IP, el proceso itera de nuevo a la etapa Z7 y realiza las etapas Z8, Z9, Z10 de nuevo. Si el resultado es una dirección IP, se extrae la dirección de red del OID devuelto. Entonces se determina si la dirección de red de OID coincide con la dirección de red de consulta. En caso negativo, el proceso vuelve a la etapa Z7. En caso positivo, la ruta hallada se pone a verdadera, la clave CIDR se pone al OID de la consulta devuelta con IPcidrRouteDest quitado, es decir, el índice a la tabla de ruta CIDR. El proceso prosigue después permitiendo una consulta SNMP para obtener salto siguiente, ifIndex de salida y tipo de ruta.

50 Como se representa en la figura P, en el proceso FindRouteIterative, se realiza la etapa FindRoute F1 para la dirección IP requerida (IPx). Si no se halla ruta, se devuelve un fallo. Si se halla una ruta, pero no hay interfaz de salida, se devuelve ipRouteNextHop. Si se halla la ruta e ipRouteIndex es igual a cero, entonces se realiza una etapa FindRouteIterative posterior para la dirección IP de ipRouteNextHop, con los mismos cuatro resultados posibles.

60 Aunque la manipulación especulativa es una técnica especialmente buena para la consulta eficiente de grandes conjuntos de datos, su principal aplicabilidad es cuando se consultan datos que están indexados con una clave derivada de la que ya se conoce una clave parcial. Por esa razón es especialmente útil en el contexto del análisis de tabla de ruta SNMP y la consulta de tabla SNMP ARP. Sin embargo, el comportamiento de envío rápido por dispositivo de red también se puede conocer usando otras técnicas de consulta, por ejemplo, acceso CLI y XML API.

**Consulta ARP**

Ahora se hará referencia a la figura 7 para describir una técnica eficiente de consultar una tabla ARP usando manipulación especulativa. La generación de una consulta se explica con más detalle más adelante con referencia a la figura 7. Para el dispositivo consultado, se obtiene una lista de índices de interfaz (IfIndices) de la topología de red recorriendo IfIndex del dispositivo propiamente dicho. Cada ifIndex del dispositivo se combina con la dirección NHL 3 para generar un conjunto de claves a incluir en la consulta al dispositivo. Así, se formula una consulta conteniendo dichas claves y se transmite al dispositivo de enfoque. El dispositivo de enfoque produce cero o una respuesta exitosa. La figura 7 ilustra un formato de tabla ipNetToMediaEntr y que permitiría en principio determinar la dirección MAC para cualquier dirección IP dada. Dado que no puede hallarse una única entrada para una dirección IP específica a no ser que se conozca de que interfaz se aprendió la entrada ARP, se usa manipulación especulativa combinando la dirección IP con todos y cada ifIndex en el dispositivo. Es decir, cada clave de consulta puede ser creada combinando la dirección IP con un ifIndex. De esta forma, el número de consultas SNMP es el número de interfaces en el dispositivo que típicamente es mucho menor que el número de entradas ARP en el dispositivo y por ello es significativamente más eficiente.

En manipulación especulativa, múltiples claves de consulta pueden contenerse en un solo mensaje de consulta.

**Tecnologías/protocolos adicionales**

El algoritmo de identificación de ruta utilizado anteriormente proporciona una forma efectiva de identificar una ruta concreta que probablemente seguirá un paquete o mensaje concreto a través de la red de dispositivos interconectados que operan según protocolos de red conocidos en general. Surgen situaciones en las que, por una u otra razón, el algoritmo de identificación de ruta se enfrenta a un reto particular. Algunos de estos retos se explican a continuación.

En algunos casos, la utilidad ejecutada en el algoritmo tiene que atravesar un segmento de red conmutado etiquetado multiprotocolo (MPLS). Lo lleva a cabo hallando la asignación de etiqueta inicial (en el punto donde el tráfico entra en el segmento MPLS) y rastreando a través de la red MPLS por salto usando detalles por salto de despliegue, empuje y envío de etiqueta hasta que el tráfico haya desplegado su etiqueta final y salga del segmento MPLS.

Otro reto es atravesar límites NAT que pueden ser realizados sondeando tablas NAT del dispositivo NAT. Esto puede requerir sondeo especulativo en tiempo real para NAT dinámico, pero podría ser posible usar sondeo de fondo para NAT estático.

Para protocolos de túnel, tal como IPSEC/GRE/SSL, etc., la utilidad comprueba una ruta directa desde un extremo del túnel al otro (típicamente con un salto de capa 3 desconocido que representa todos los nodos entremedios). La utilidad también comprueba información topológica específica de protocolo y comprueba en las tablas de enrutamiento/interfaces la presencia de saltos de cripto/tunelización.

Otro reto es la virtualización. Es importante que el algoritmo identifique puertos de salida físicos de modo que a un dispositivo físico conectado al puerto de salida pueda accederse desde la topología. Muchas redes operan en varias capas de virtualización diferentes. Los conmutadores virtuales pueden ser consultados usando API adicionales, y para asegurar que el servidor de topología tenga información oportuna acerca de la posición de host final, podría ser necesario integrar el servidor de topología con plataformas de gestión de virtualización para obtener actualizaciones relativas a reasignación de máquina virtual para permitir un sondeo proactivo de la posición de host final en conmutadores virtuales afectados.

La utilidad negocia tablas de enrutamiento y envío virtualizadas (VRF) consultando la tabla de envío (enrutamiento) de IP apropiada requerida para un identificador VRF específico. En SNMP, por ejemplo, esto se puede hacer usando cadenas comunitarias contextualizadas VRF.

## REIVINDICACIONES

1. Un método para identificar una ruta desde un dispositivo de origen (X) a un dispositivo terminal en una red de dispositivos interconectados, que comprende:

5 identificar como dispositivo de enfoque un primer dispositivo conectado al dispositivo fuente (X);  
 transmitir una primera consulta de identificación de ruta al primer dispositivo, incluyendo la consulta un  
 identificador de destino y solicitar la identificación de un puerto de salida para los mensajes dirigidos al destino  
 10 identificado por el identificador de destino cuando la consulta se recibe en el primer dispositivo; y  
 recibir un mensaje de resultado que identifica el puerto de salida e identifica como dispositivo de enfoque un  
 segundo dispositivo conectado al primer dispositivo basado en una topología de red a la que puede acceder el  
 ordenador del monitor (16);  
 15 dirigir una consulta de identificación de ruta adicional al segundo dispositivo y recibiendo un mensaje de  
 resultado siguiente que identifica un puerto de salida desde el segundo dispositivo; y  
 identificar desde la topología de la red un tercer dispositivo conectado al segundo dispositivo, donde la ruta se  
 identifica para incluir el primer, segundo y tercer dispositivos;  
 donde la formulación de la primera y otras consultas se lleva a cabo mediante un programa informático de bucle  
 de acuerdo con las siguientes etapas:

20 recibir en una unidad de ejecución en un ordenador de monitor (16) que tiene una ruta de consulta a los  
 dispositivos interconectados un conjunto de variables de estado que definen un estado de entrada, donde una  
 de las variables de estado del conjunto define una secuencia ordenada de opciones de bucle (C, S, R, A, c,  
 r);  
 registrar el estado de entrada en una unidad de almacenamiento;  
 25 en la unidad de ejecución (16), ejecutar una primera opción de bucle en la secuencia ordenada de opciones  
 de bucle (C, S, R, A, c, r) en el estado de entrada, utilizando como parámetros al menos una de las otras  
 variables de estado en el conjunto de variables de estado, donde la ejecución de la primera opción de bucle  
 comprende cancelar la primera opción de bucle de la secuencia ordenada, realizar etapas de procesamiento  
 utilizando al menos una de las otras variables de estado y determinar si alguna de estas variables de estado  
 30 se ha alterado como resultado de las etapas de procesamiento, donde:

si ninguna de las variables de estado ha cambiado, ingresar la siguiente iteración de bucle con un estado  
 de entrada en el cual la primera opción de bucle se cancela desde la secuencia ordenada de opciones de  
 bucle, revelando una nueva opción de primer bucle que se ejecutará con las variables no alteradas, y  
 35 si al menos una de las variables de estado ha cambiado, restablecer la opción de primer bucle cancelado  
 en la secuencia ordenada para proporcionar una secuencia ordenada original de opciones de bucle e  
 ingresar una iteración de bucle siguiente con un estado de entrada definido por la(s) variable(s) de estado  
 alterado y la secuencia original ordenada de opciones de bucle,  
 por lo que cada iteración de bucle siguiente recibe un nuevo estado de entrada, de las variables de estado  
 40 alteradas, donde una de las variables de estado es el dispositivo de enfoque para recibir la consulta de  
 identificación de ruta generada por la opción de bucle ejecutada en el ordenador de monitor; otra de las  
 variables de estado es una dirección de red de enrutamiento como identificador de destino y otra de las  
 variables de estado es una dirección de conmutación como identificador de destino, y una de las opciones  
 45 determina si la dirección de enrutamiento del estado de entrada está en el dispositivo de enfoque del  
 estado de entrada y, si no, utiliza una tabla de asignación para traducir la dirección de enrutamiento para  
 proporcionar una nueva dirección de conmutación como una variable de estado alterada.

2. El método de la reivindicación 1, donde las variables de estado incluyen un identificador de red de área local  
 virtual que es relevante para la ruta en la ubicación identificada por el dispositivo de enfoque.

50 3. El método de la reivindicación 1 o 2, donde las opciones incluyen: una opción que determina si hay una interfaz  
 única en el dispositivo de enfoque que coincida con la dirección de la red de enrutamiento en el estado de entrada, o  
 cada nuevo estado de entrada, e identifica un puerto de salida del dispositivo de enfoque desde la interfaz única, y  
 una opción que ubica en una tabla de asignación una entrada de asignación entre la dirección de enrutamiento y la  
 55 dirección de conmutación en el estado de entrada (o nuevo estado de entrada), utiliza la entrada de asignación para  
 determinar desde qué interfaz fue derivada la entrada de asignación, y por lo tanto deriva un puerto de salida.

4. El método de cualquier reivindicación anterior, donde la dirección de enrutamiento (L3) se establece en la  
 dirección de enrutamiento (L3) del dispositivo terminal (Y).

60 5. El método de cualquier reivindicación anterior, donde la dirección de conmutación (L2) se usa para acceder a una  
 base de datos de reenvío para localizar un puerto de salida.

6. El método de cualquier reivindicación anterior, donde la opción determina la ruta mediante la consulta iterativa de  
 65 una tabla de enrutamiento (92) en el dispositivo de enfoque para generar una dirección de enrutamiento candidata.

7. El método de la reivindicación 6, que comprende usar una tabla de mapeo (91) para traducir la dirección de enrutamiento candidata (L3) a una dirección de conmutación candidata (L2), y comparar la dirección de conmutación candidata (L2) con la dirección de conmutación (L2) en el estado de entrada (o el nuevo estado de entrada).
- 5 8. El método de la reivindicación 1, que comprende procesos de preparación que incluyen al menos uno de un primer proceso de preparación que almacena los puntos de inicio y parada identificando ubicaciones de red para cada uno de los dispositivos fuente (X) y dispositivo terminal (Y); un segundo proceso de preparación que establece una dirección de enrutamiento (L3) y una dirección de conmutación (L2) para el estado de entrada de una primera iteración de bucle; y un tercer proceso de preparación que establece un dispositivo de enfoque para el estado de
- 10 entrada de una primera iteración de bucle como el dispositivo de origen (X) o un conmutador de acceso para el dispositivo de origen (X).
9. El método de la reivindicación 8, que comprende: obtener una sugerencia para ayudar a identificar una red de área local virtual donde está conectado el dispositivo de enfoque.
- 15 10. El método de la reivindicación 9, donde obtener una sugerencia comprende determinar un puerto de acceso para la salida de mensajes del dispositivo fuente (X), identificar el nombre del puerto de acceso y extraer un número del nombre del puerto de acceso para almacenar como el indicio.
- 20 11. El método de la reivindicación 9, donde obtener una sugerencia comprende determinar un puerto de acceso para la salida de mensajes del dispositivo de origen (X), identificar asignaciones explícitas de VLAN para el puerto (a través de SNMP o desde un NMS) y si la asignación de VLAN es única, almacenar el número de VLAN único como el indicio.
- 25 12. El método de las reivindicaciones 3 o 5, donde el puerto de salida y el indicio de VLAN se utilizan para ubicar el siguiente dispositivo conectado desde la topología de la red.
13. Un método según la reivindicación 1, que comprende registrar un conjunto de dispositivos y puertos identificados para estar en la ruta en una tienda a la que puede acceder el ordenador monitor (16) como un registro de ruta.
- 30 14. Un método según la reivindicación 13, donde el registro de ruta incluye puertos de entrada identificados desde la topología de red para dispositivos identificados.
15. Un método según la reivindicación 1, donde la consulta de identificación de ruta se transmite desde el ordenador monitor (16) al dispositivo de enfoque que se consulta en un mensaje que se transmite a través de la red.
- 35 16. Un método según la reivindicación 1, donde la consulta de identificación de ruta incluye una pluralidad de claves para consultar el dispositivo.
- 40 17. Un método según la reivindicación 1, que comprende identificar una primera ruta a través de la red desde el dispositivo de origen (X) hasta el dispositivo de destino (Y) utilizando un primer conjunto de consultas en una primera vez, y rutas adicionales a través de la red desde el dispositivo fuente (X) al dispositivo de destino final (Y) utilizando conjuntos de consultas posteriores.
- 45 18. Un método según la reivindicación 1, cuando se usa en una red de dispositivos interconectados que incluyen dispositivos de capa 2 y de capa 3, comprendiendo el método, después de identificar el primer dispositivo conectado al dispositivo fuente (X):
- 50 identificar un siguiente identificador de destino de capa 3, a lo largo de la ruta hacia el dispositivo de destino final (Y); entonces  
identificar de un identificador de destino de capa 2 del siguiente identificador de destino de capa 3.
19. Un método según la reivindicación 1, donde cada consulta de identificación de ruta se genera para una tabla de reenvío de tráfico en el dispositivo al que se transmite la consulta de identificación de ruta, utilizando la consulta al menos una de un conjunto de claves que se han generado al combinar el identificador de destino con una pluralidad de índices incrustados de la tabla de reenvío de tráfico.
- 55 20. Un sistema informático para ejecutar un programa informático de bucle, comprendiendo el sistema informático un procesador que puede ejecutarse para ejecutar un método de ejecución de bucle de acuerdo con cualquiera de las reivindicaciones 1 a 19; y medios de almacenamiento para registrar el estado de entrada.
- 60 21. Un sistema informático según la reivindicación 20, que comprende, además:
- 65 una interfaz (84) conectada a una red de dispositivos interconectados para transmitir consultas y recibir respuestas a un dispositivo de enfoque;  
un segundo medio de almacenamiento para almacenar un registro de trayectoria; y

un tercer medio de almacenamiento que almacena una topología de red.

22. Un programa informático que comprende un código que, cuando se ejecuta mediante un procesador, realiza las etapas del método de cualquiera de las reivindicaciones 1 a 19.

5

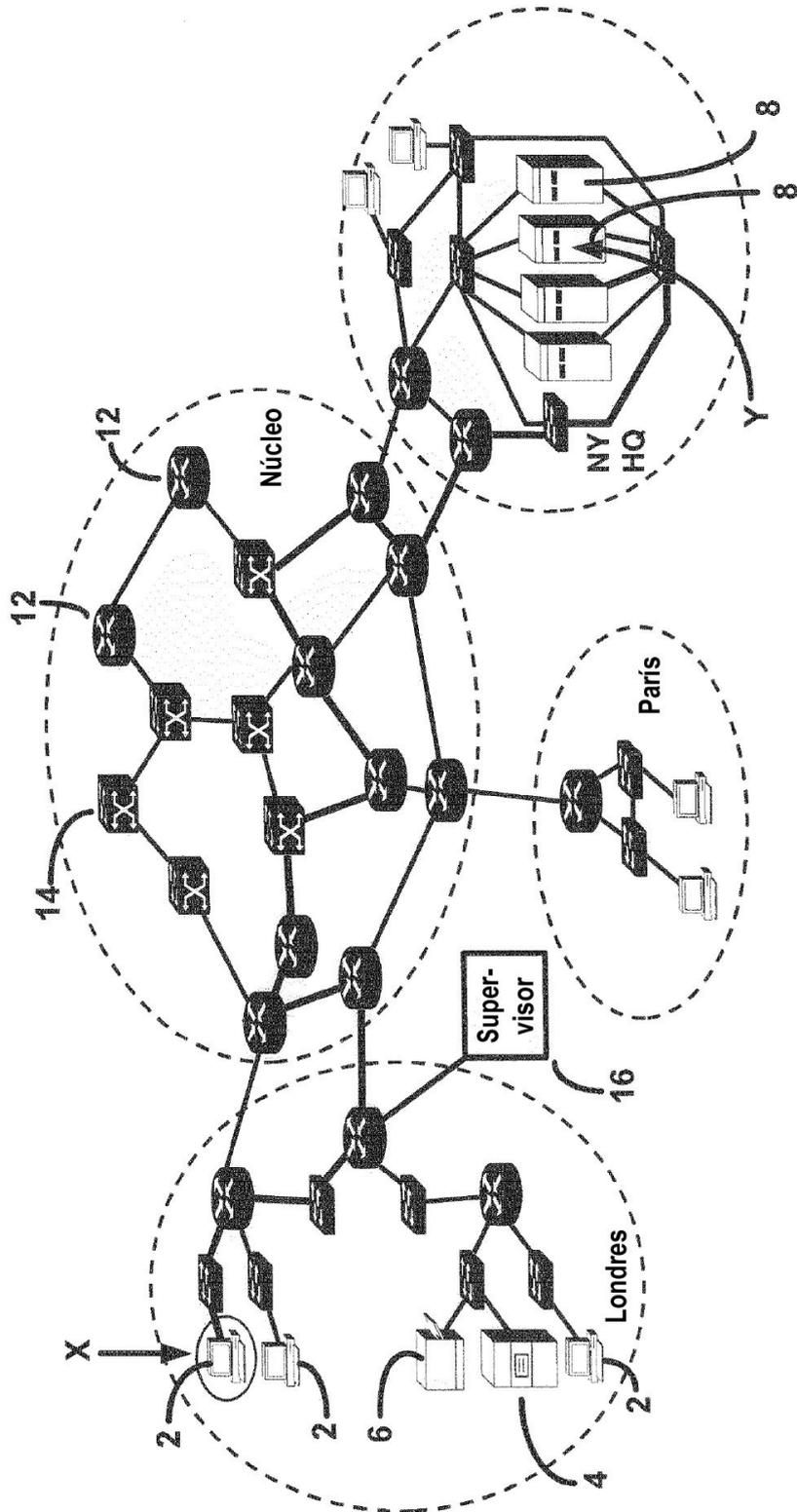


Fig. 1

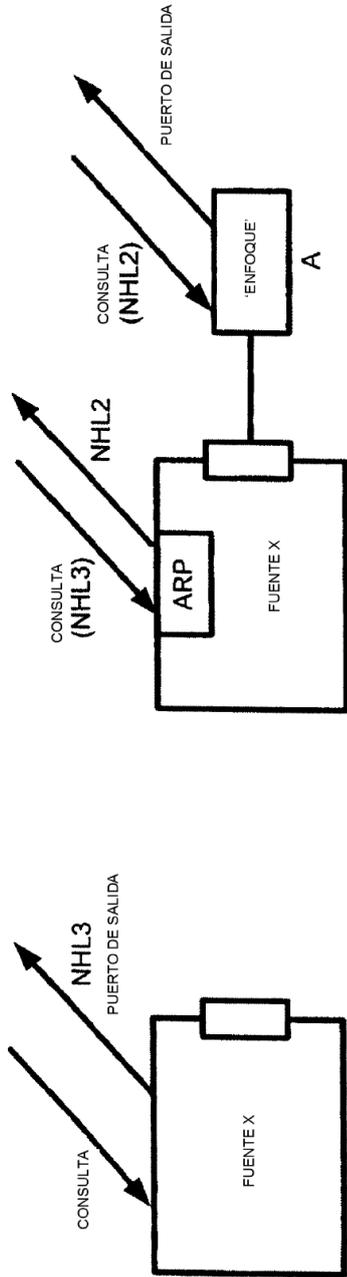


FIG. 2b

FIG. 2a

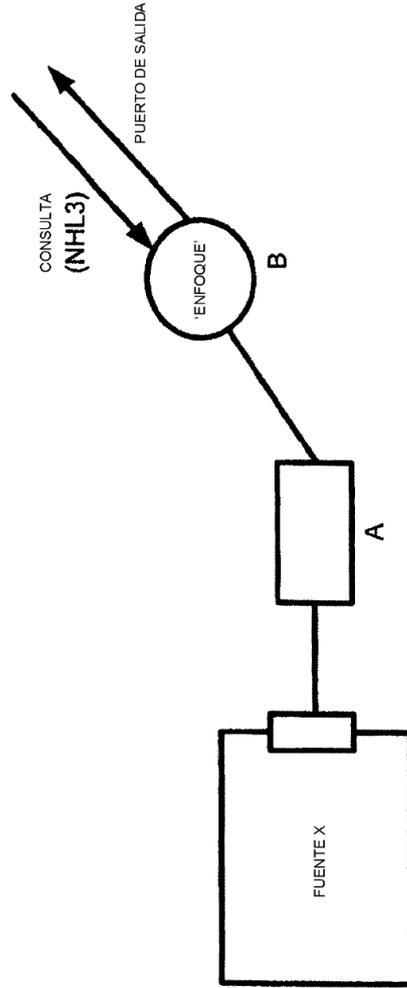


FIG. 2c

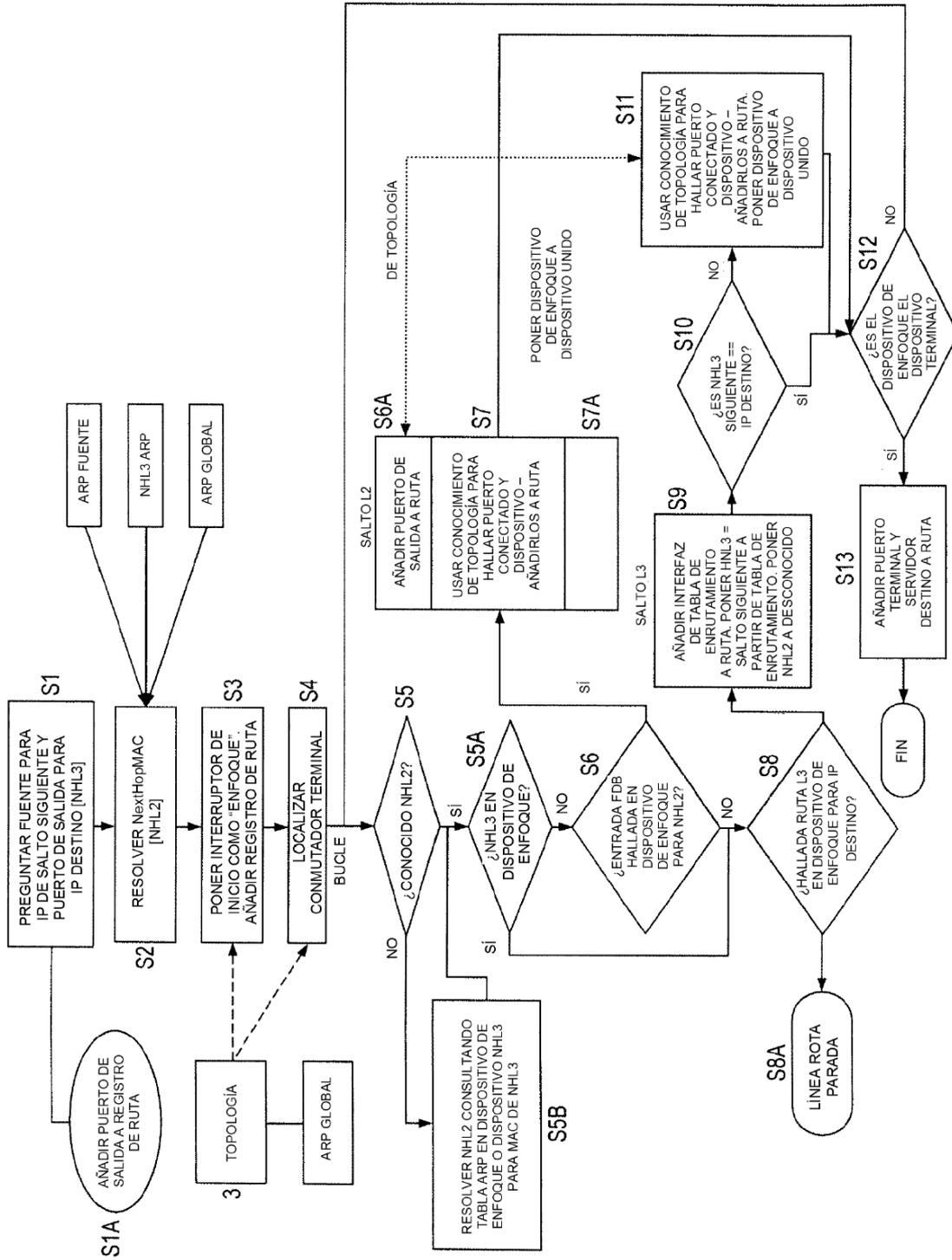


FIG. 3

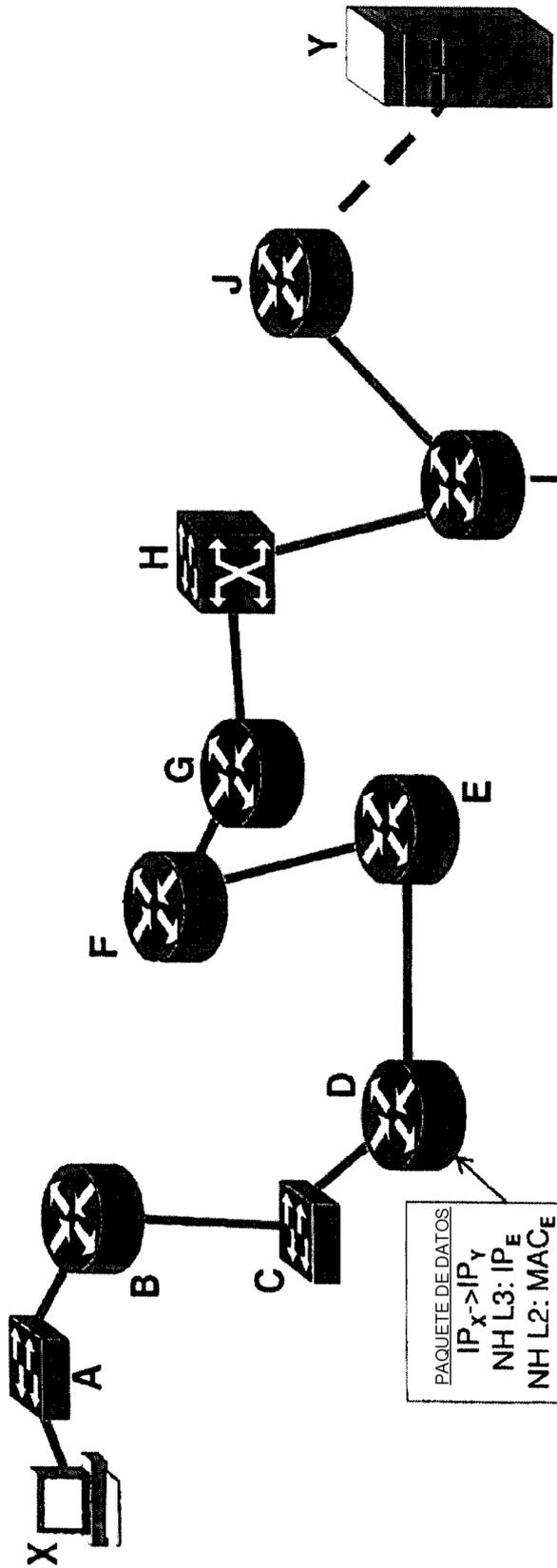


FIG. 4

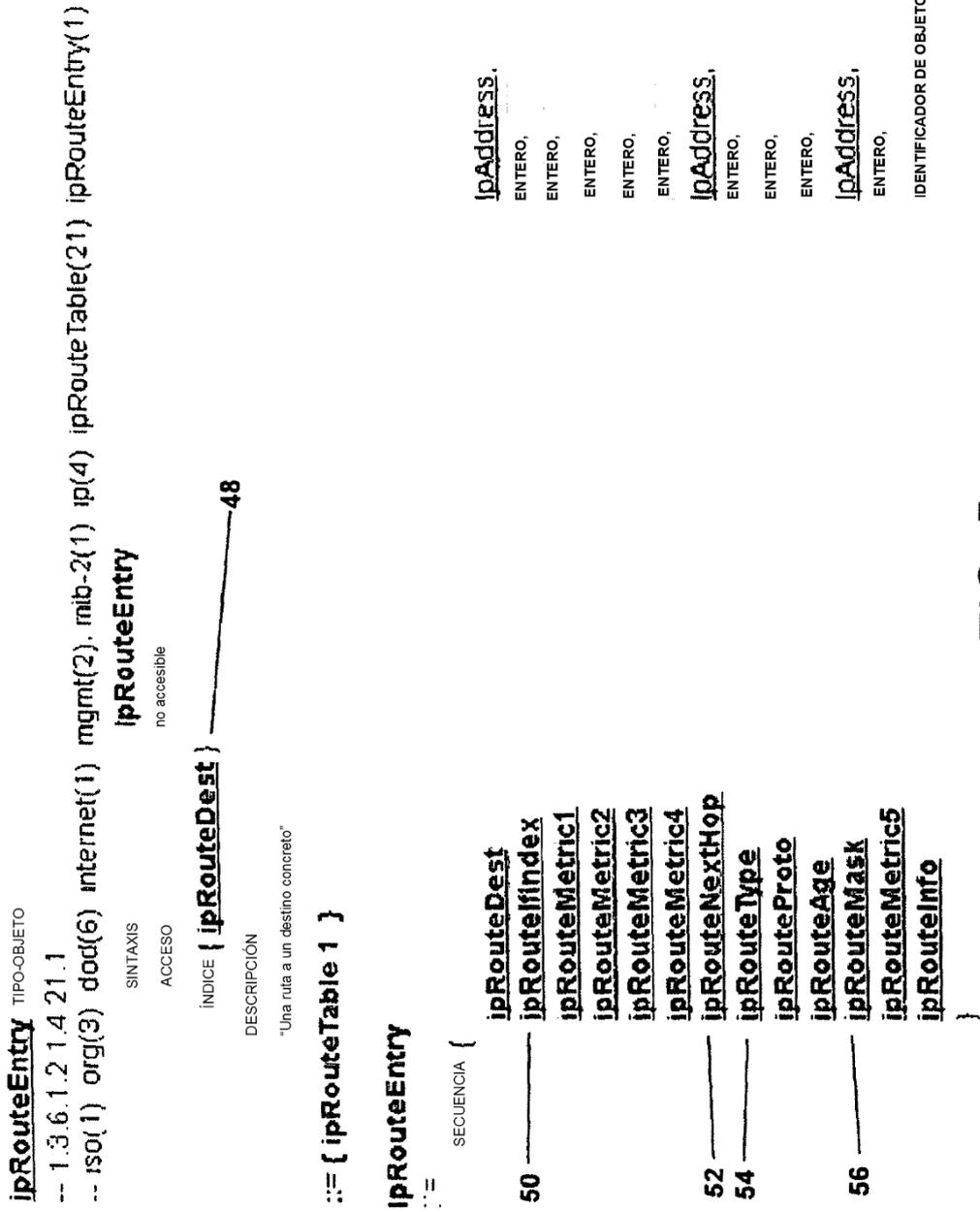


FIG. 5

DIRECCIÓN IP 10.44.1.213 = 0A.2C.01.D5 = 0000 1010 0010 1100 0000 0001 1101 0101

/32: 0000 1010 0010 1100 0000 0001 1101 0101 /15: 0000 1010 0010 1100 0000 0000 0000 0000  
/31: 0000 1010 0010 1100 0000 0001 1101 0100 /14: 0000 1010 0010 1100 0000 0000 0000 0000  
/30: 0000 1010 0010 1100 0000 0001 1101 0100 /13: 0000 1010 0010 1000 0000 0000 0000 0000  
/29: 0000 1010 0010 1100 0000 0001 1101 0000 /12: 0000 1010 0010 0000 0000 0000 0000 0000  
/28: 0000 1010 0010 1100 0000 0001 1101 0000 /11: 0000 1010 0010 0000 0000 0000 0000 0000  
/27: 0000 1010 0010 1100 0000 0001 1100 0000 /10: 0000 1010 0000 0000 0000 0000 0000 0000  
/26: 0000 1010 0010 1100 0000 0001 1100 0000 /09: 0000 1010 0000 0000 0000 0000 0000 0000  
/25: 0000 1010 0010 1100 0000 0001 1000 0000 /08: 0000 1010 0000 0000 0000 0000 0000 0000  
/24: 0000 1010 0010 1100 0000 0001 0000 0000 /07: 0000 1010 0000 0000 0000 0000 0000 0000  
/23: 0000 1010 0010 1100 0000 0000 0000 0000 /06: 0000 1000 0000 0000 0000 0000 0000 0000  
/22: 0000 1010 0010 1100 0000 0000 0000 0000 /05: 0000 1000 0000 0000 0000 0000 0000 0000  
/21: 0000 1010 0010 1100 0000 0000 0000 0000 /04: 0000 0000 0000 0000 0000 0000 0000 0000  
/20: 0000 1010 0010 1100 0000 0000 0000 0000 /03: 0000 0000 0000 0000 0000 0000 0000 0000  
/19: 0000 1010 0010 1100 0000 0000 0000 0000 /02: 0000 0000 0000 0000 0000 0000 0000 0000  
/18: 0000 1010 0010 1100 0000 0000 0000 0000 /01: 0000 0000 0000 0000 0000 0000 0000 0000  
/17: 0000 1010 0010 1100 0000 0000 0000 0000 /00: 0000 0000 0000 0000 0000 0000 0000 0000  
/16: 0000 1010 0010 1100 0000 0000 0000 0000

FIG. 6

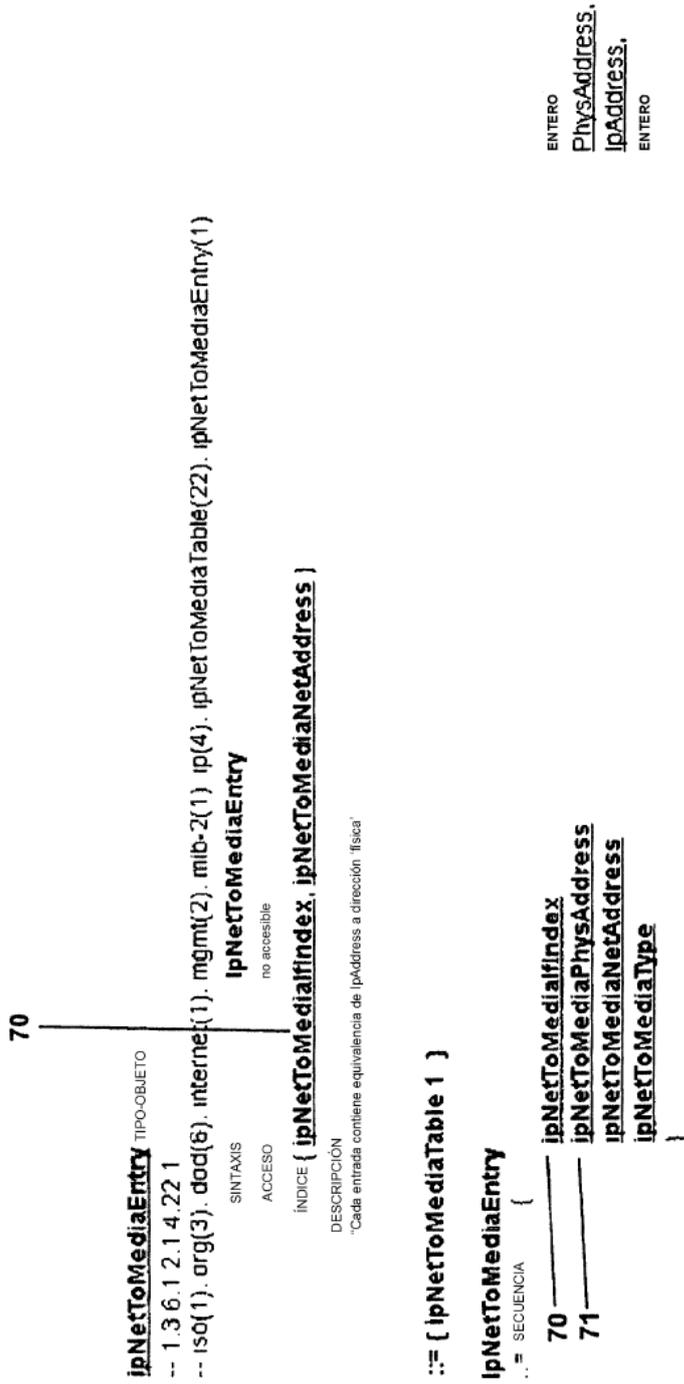


FIG. 7

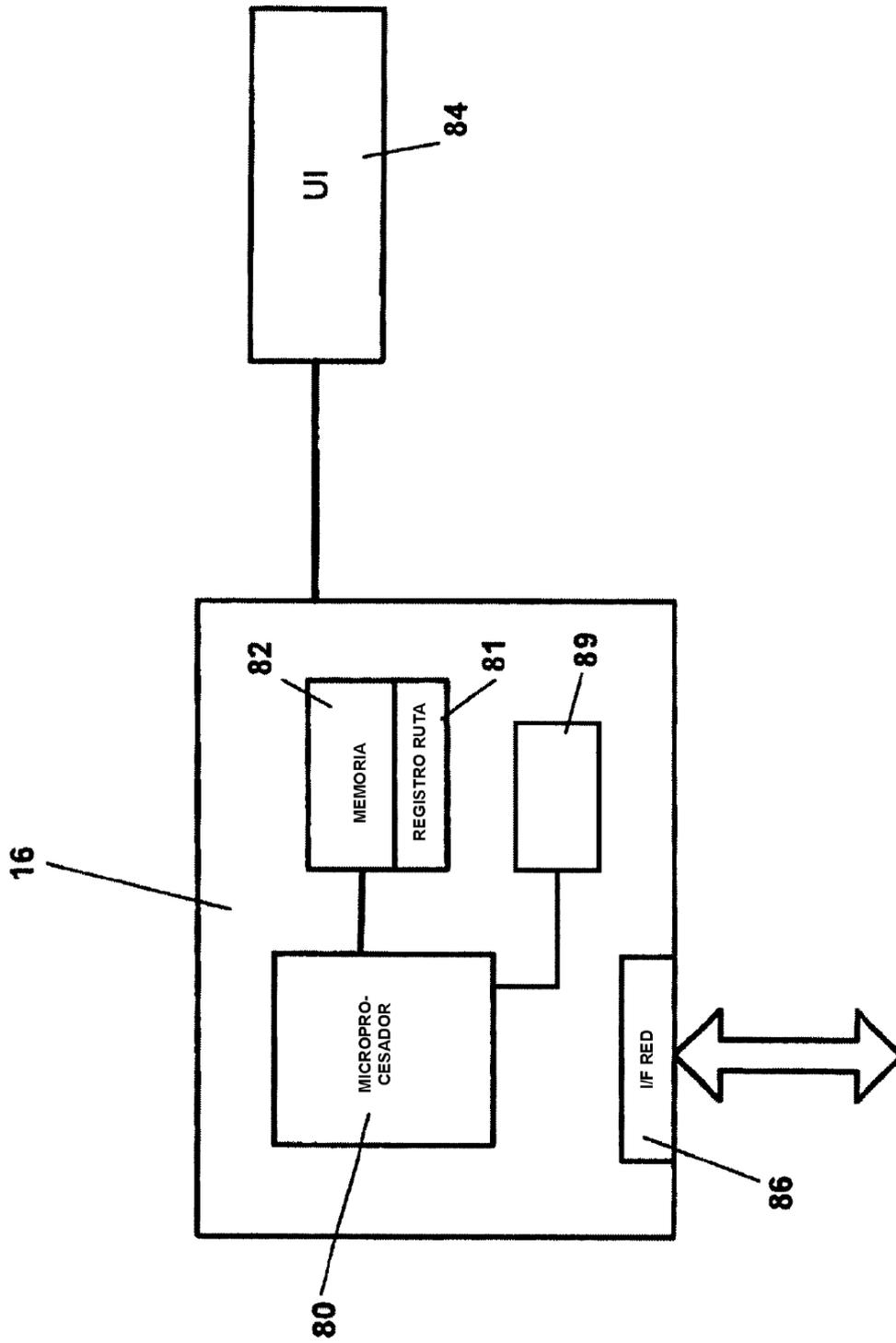


FIG. 8

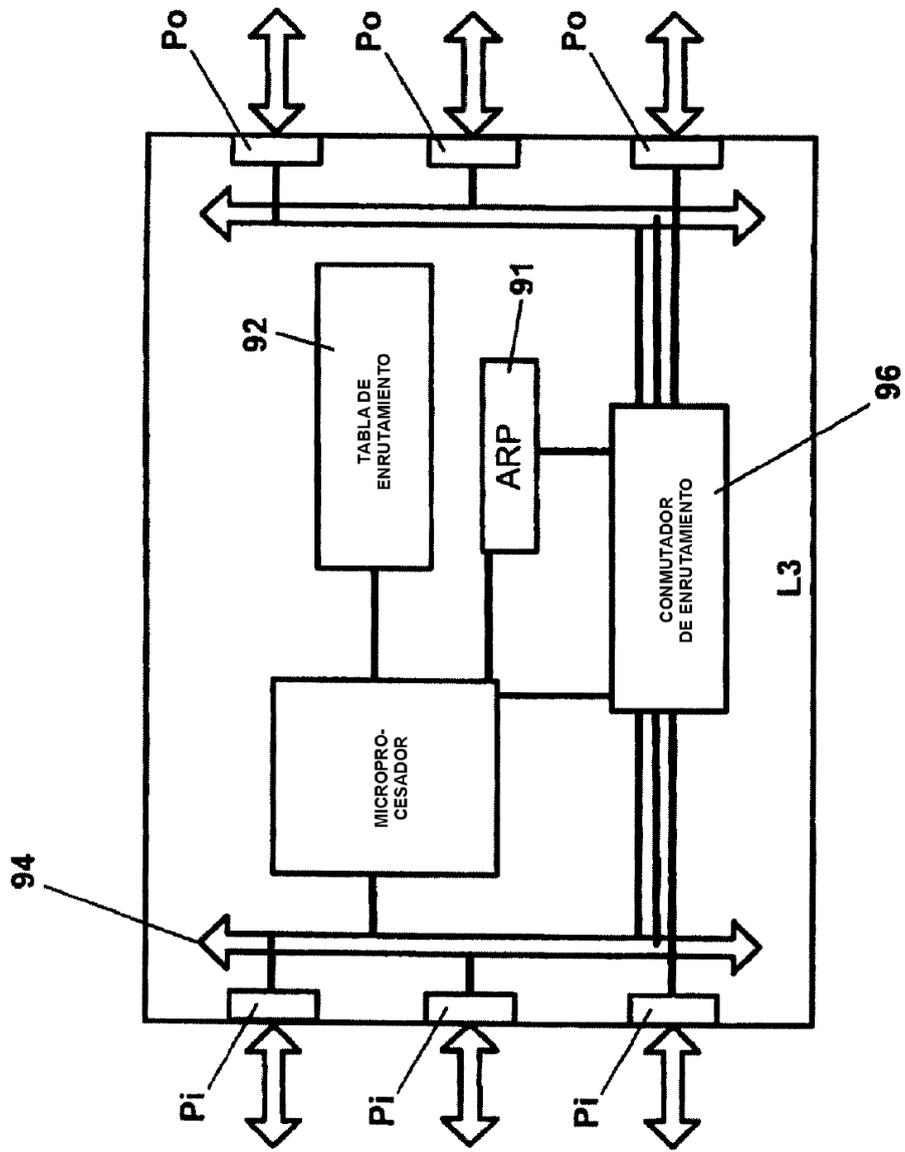


FIG. 9

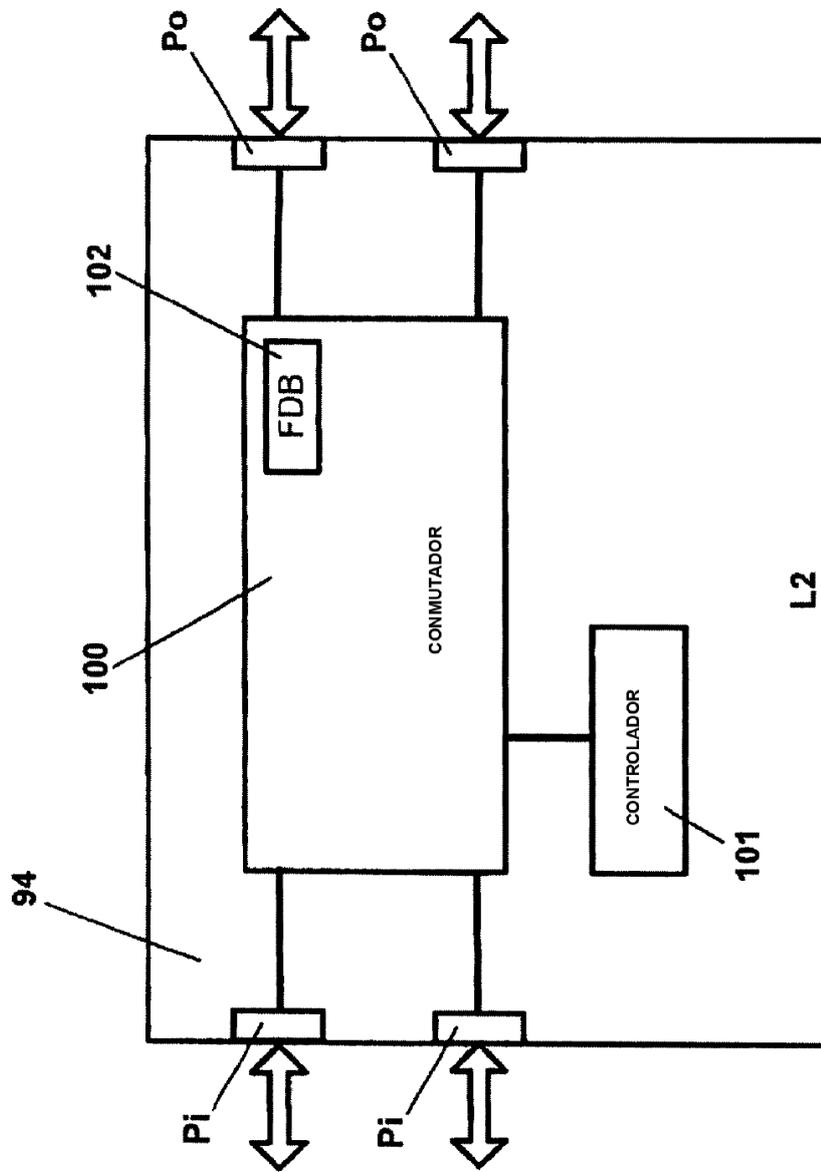


FIG. 10

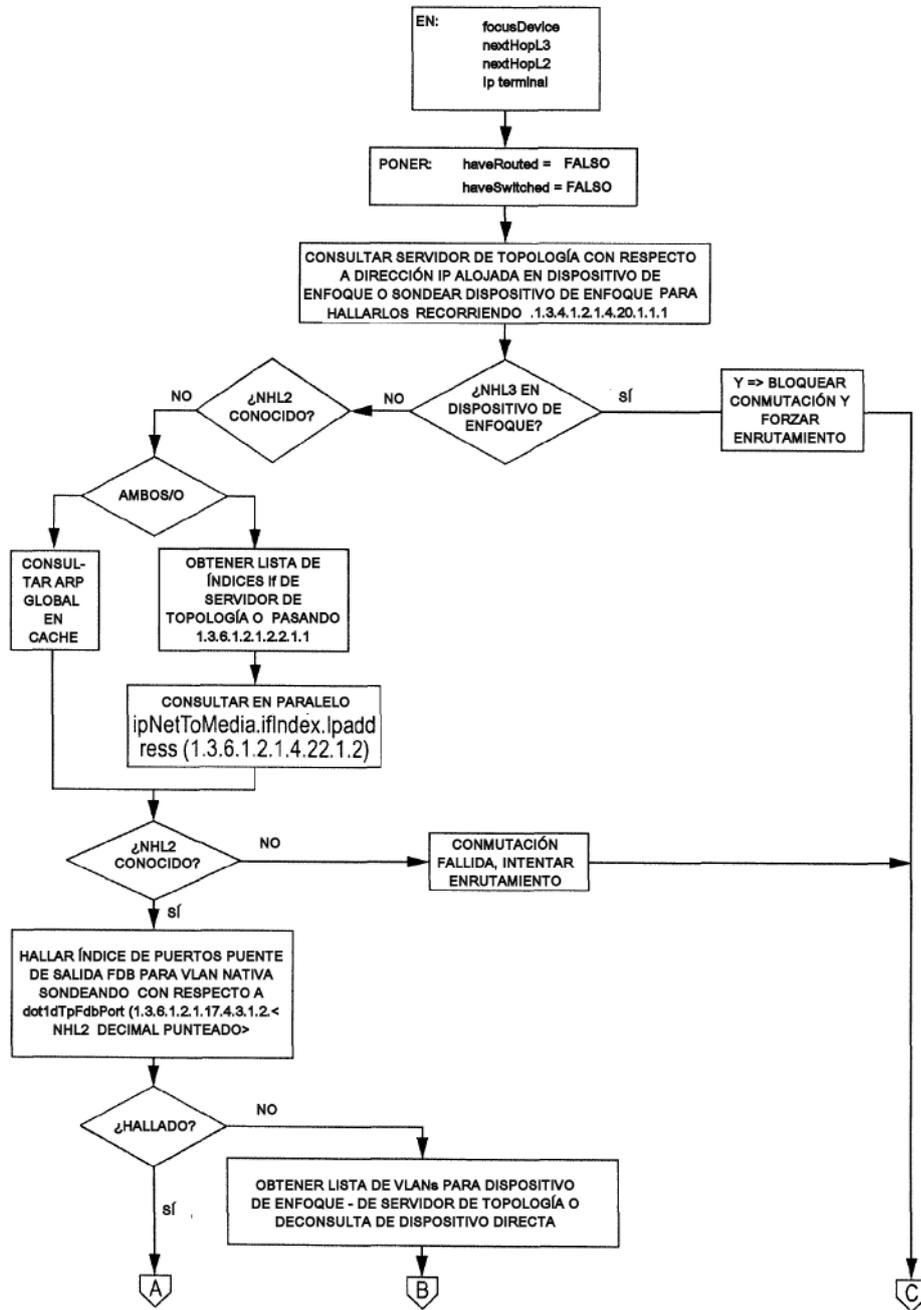


FIG. 11a

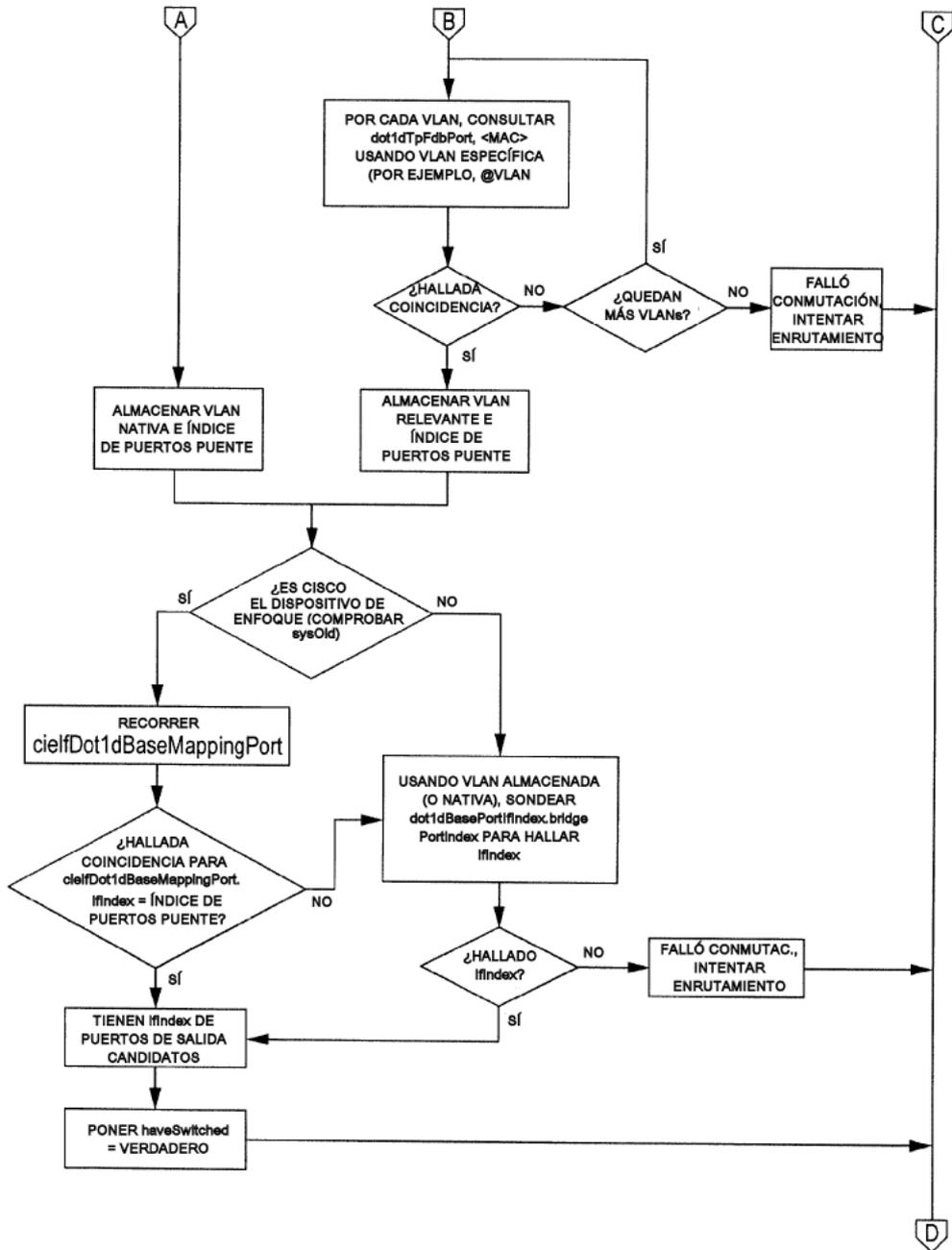


FIG. 11b

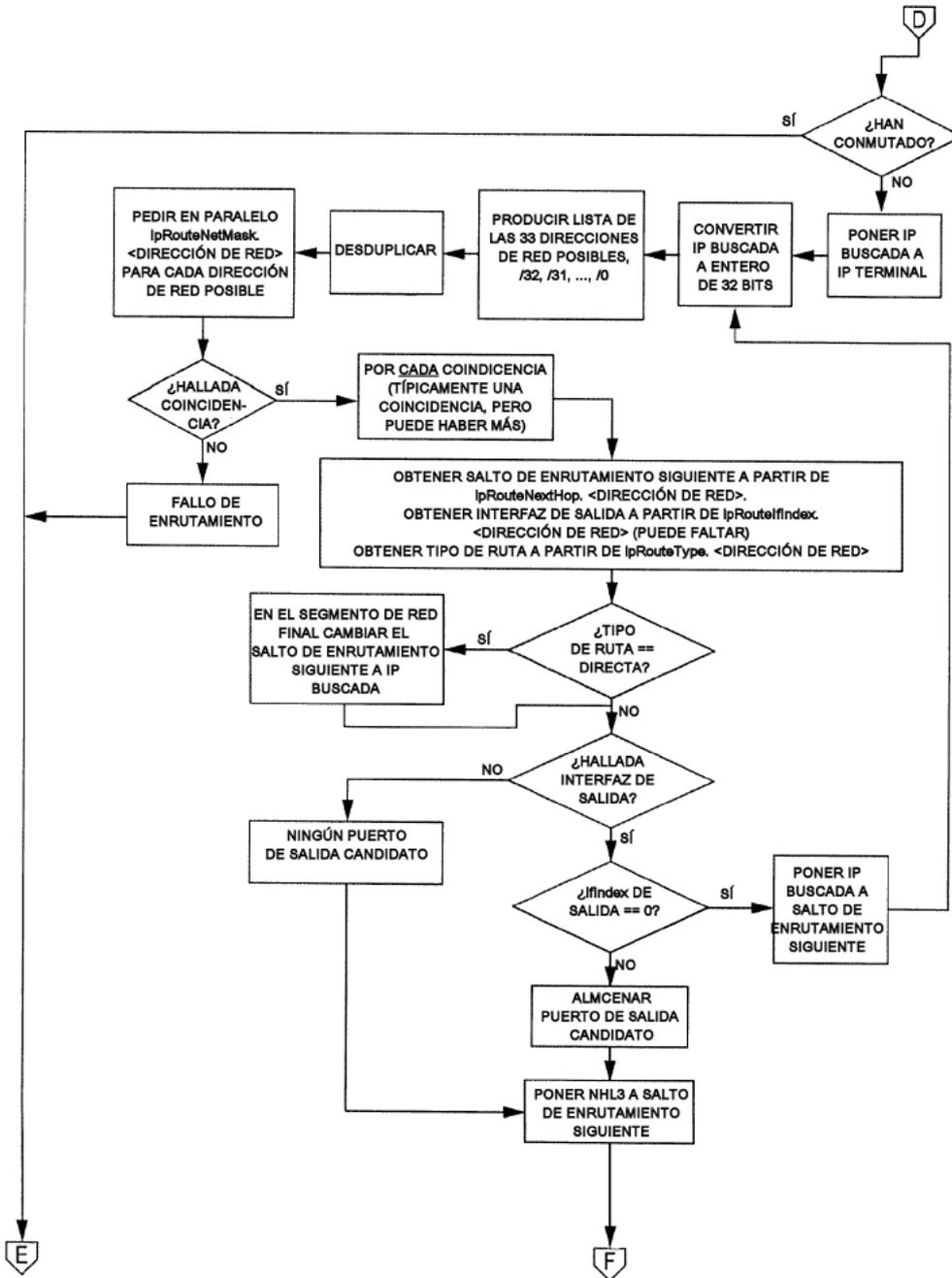


FIG. 11c

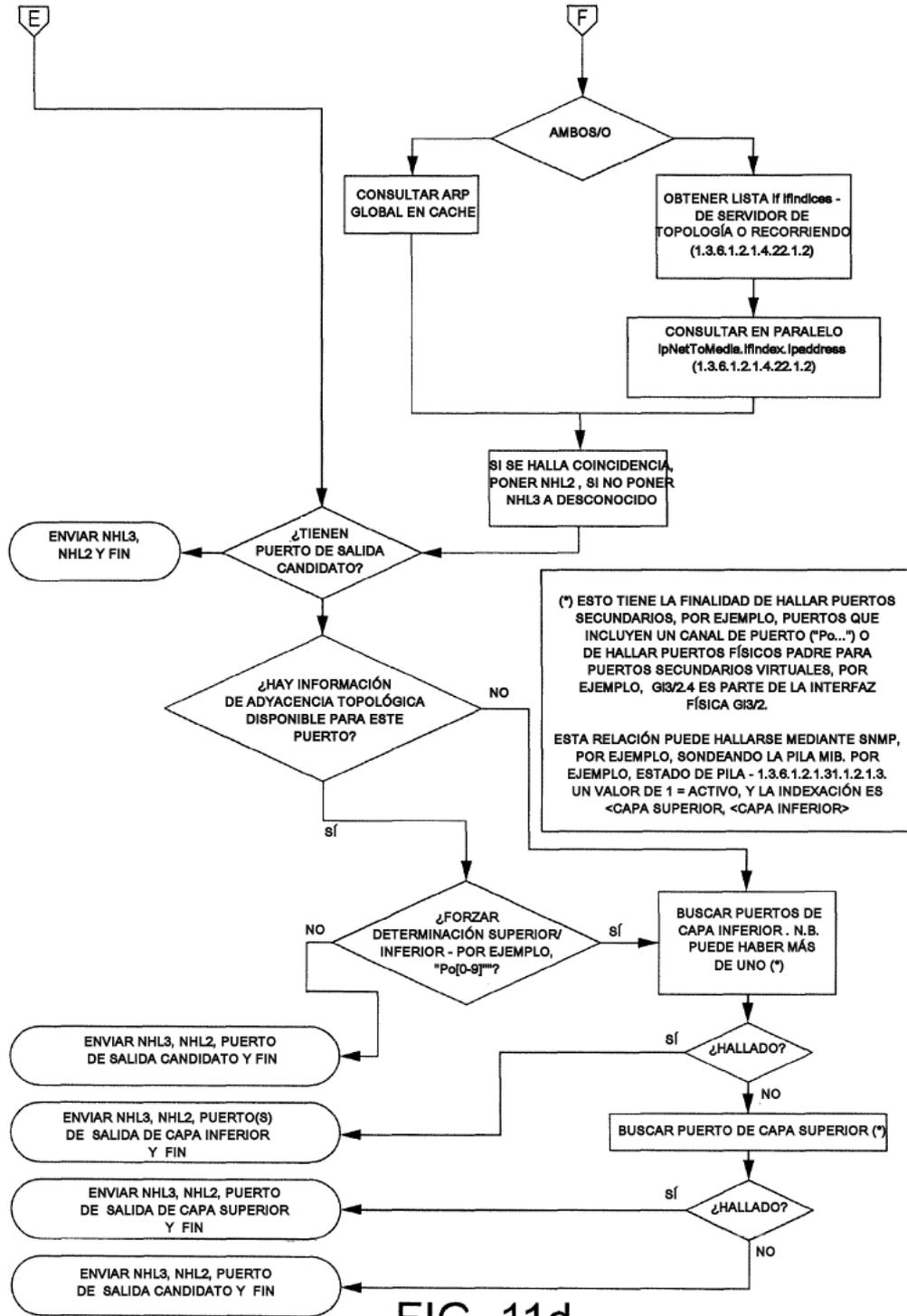


FIG. 11d

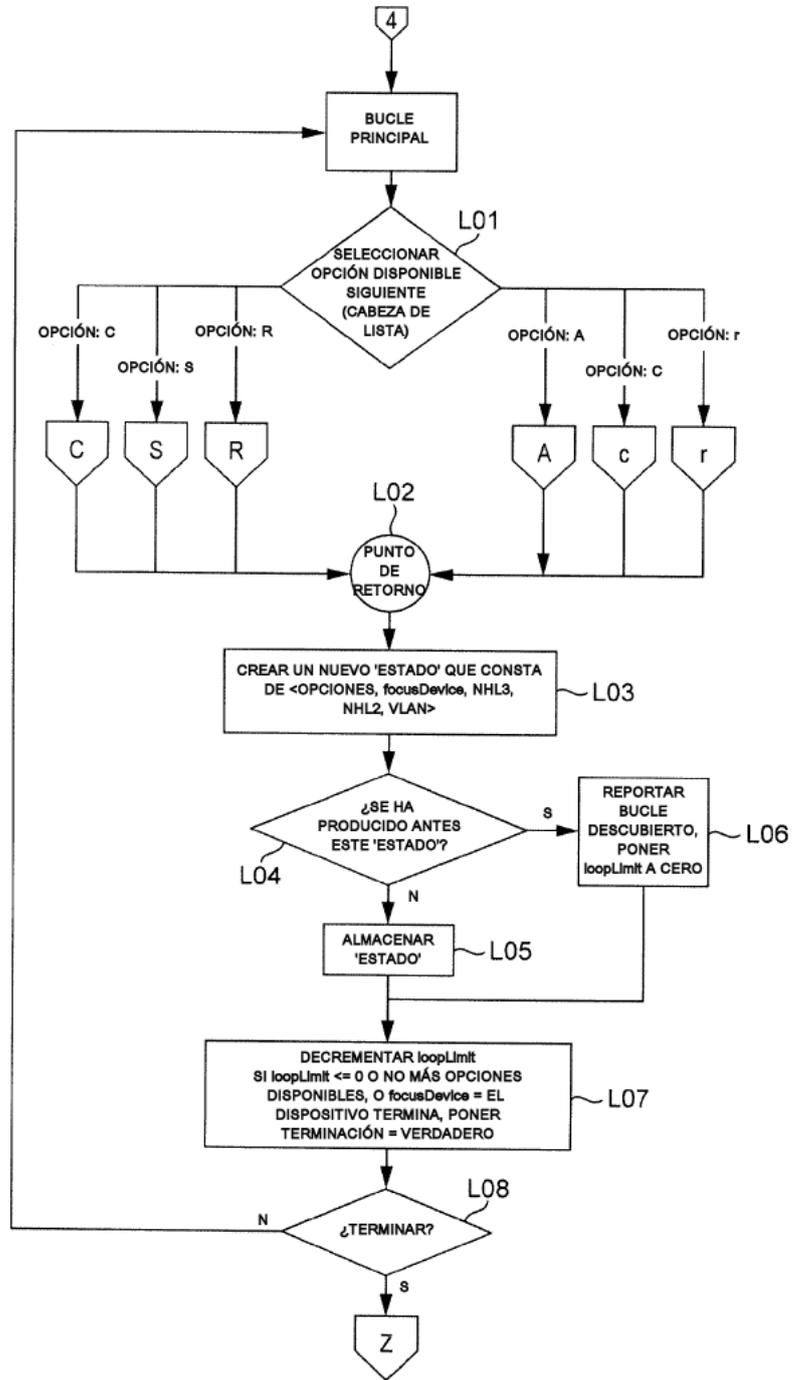


FIG. 12

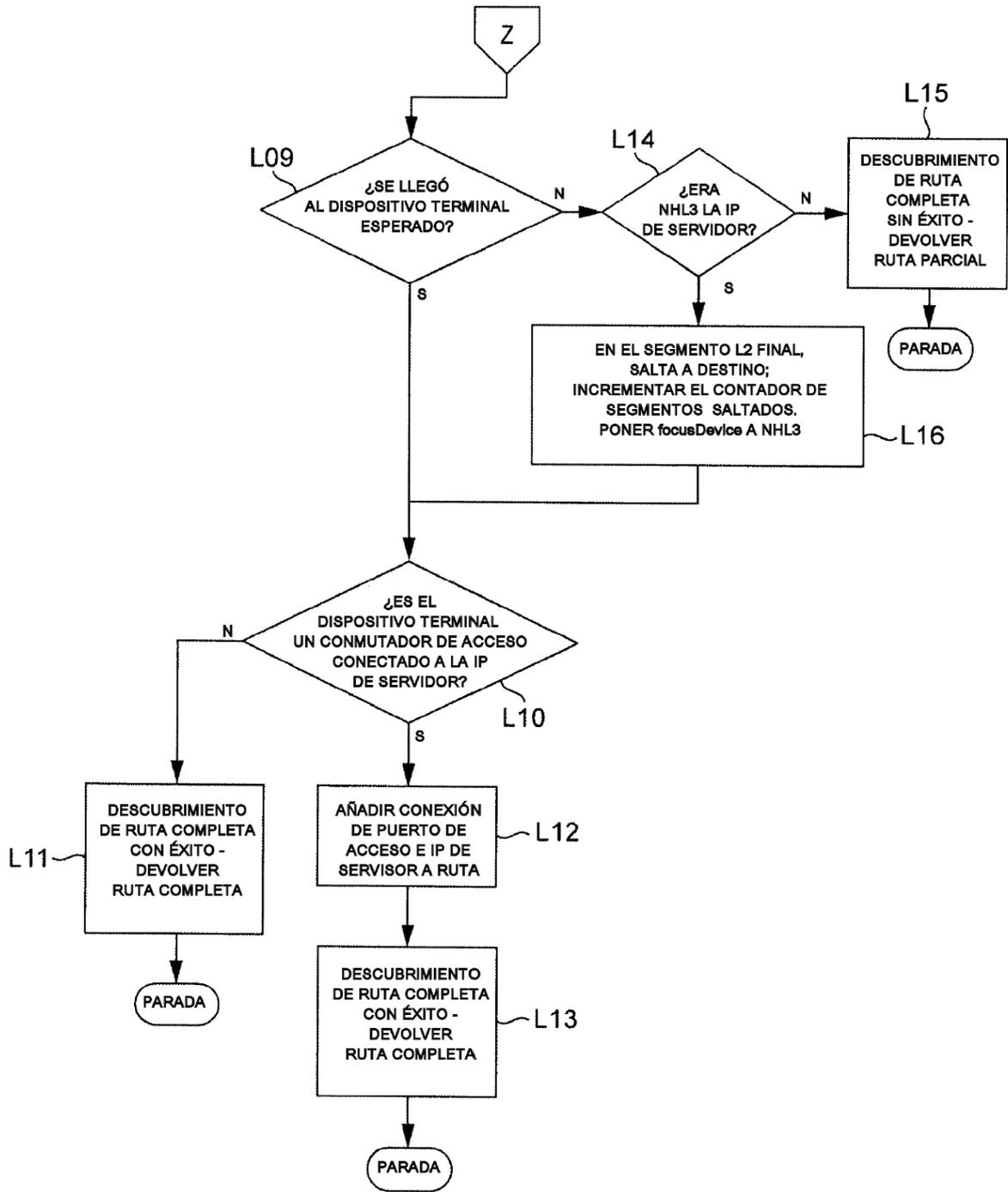


FIG. 13

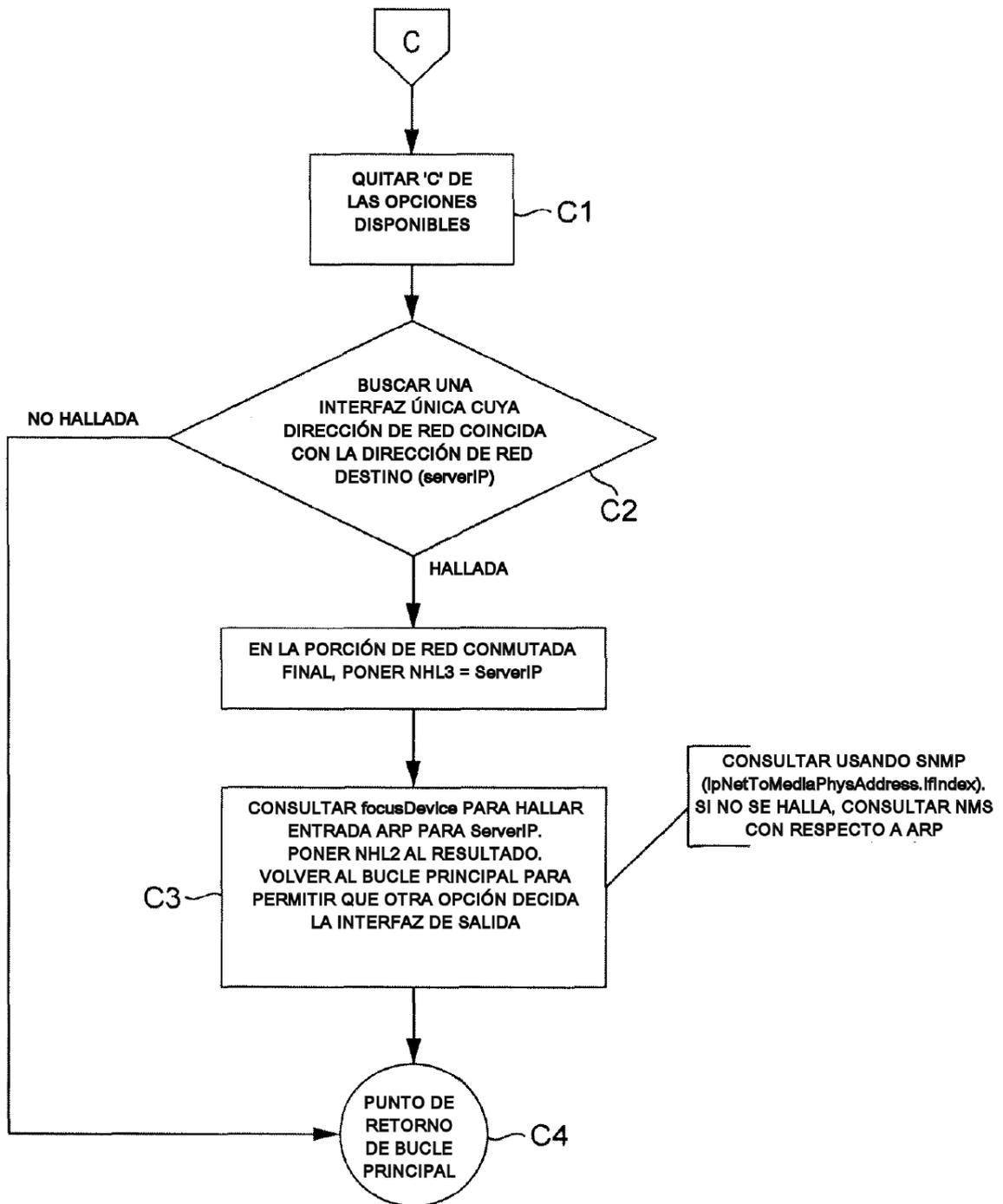


FIG. 14

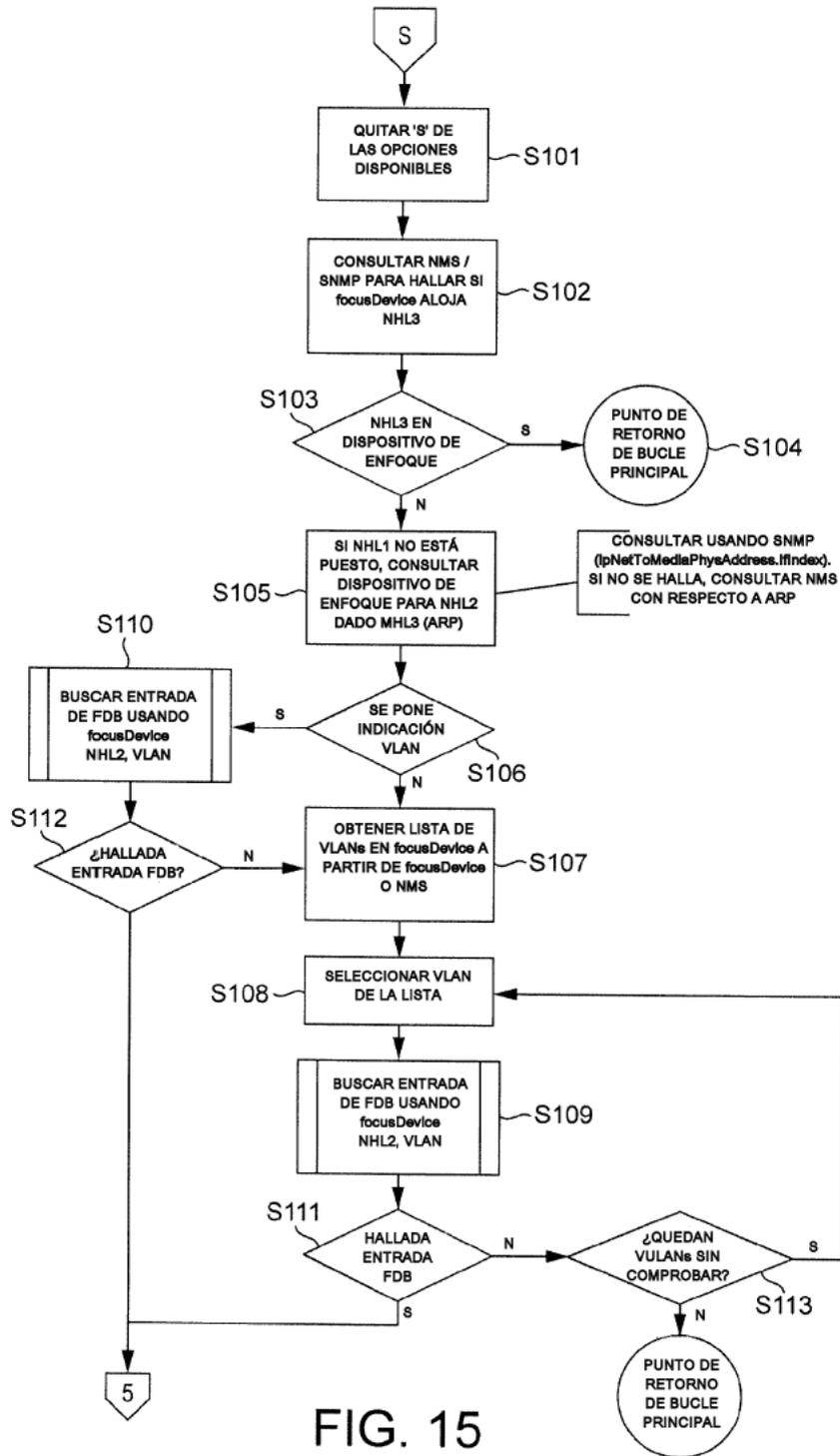


FIG. 15

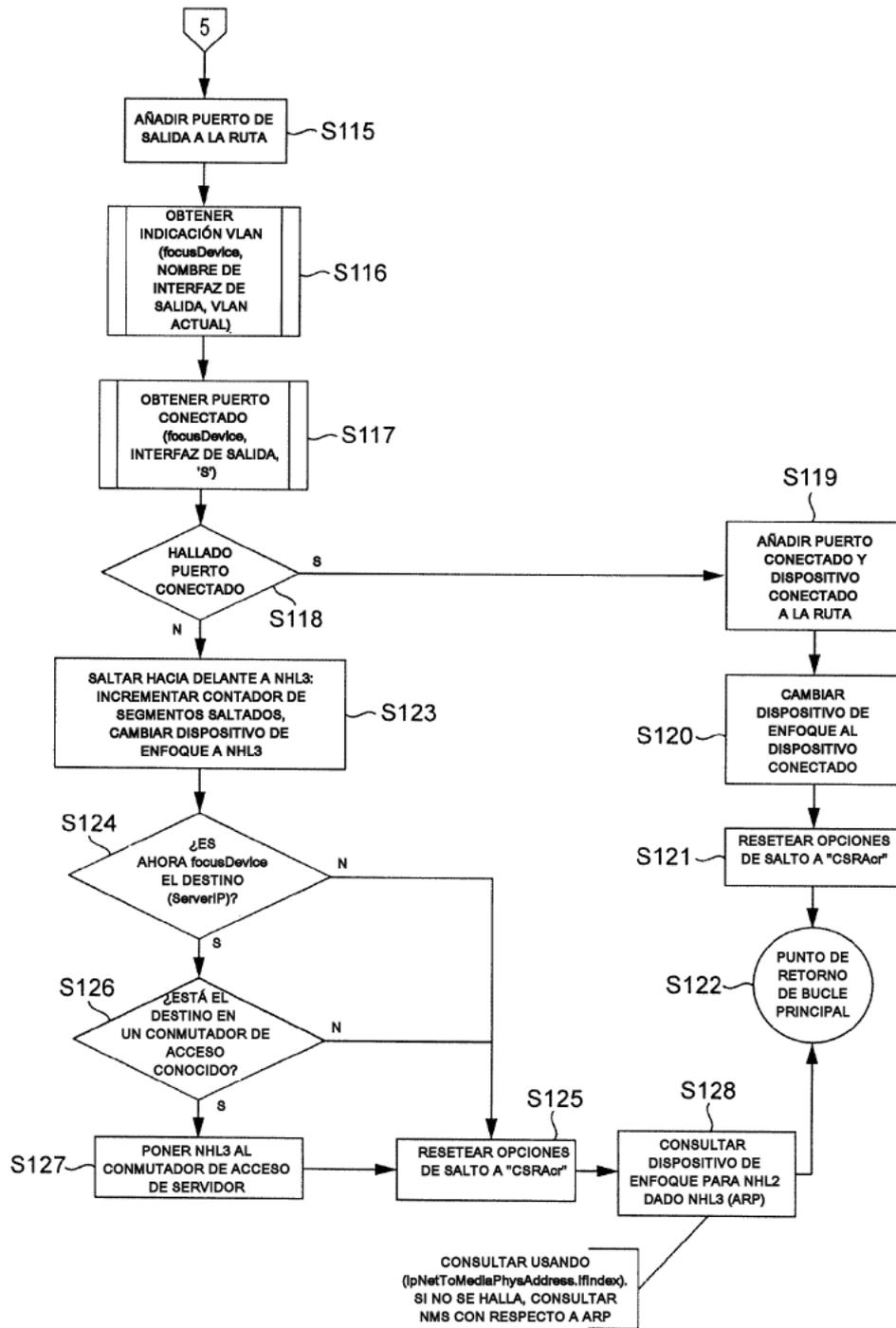


FIG. 16

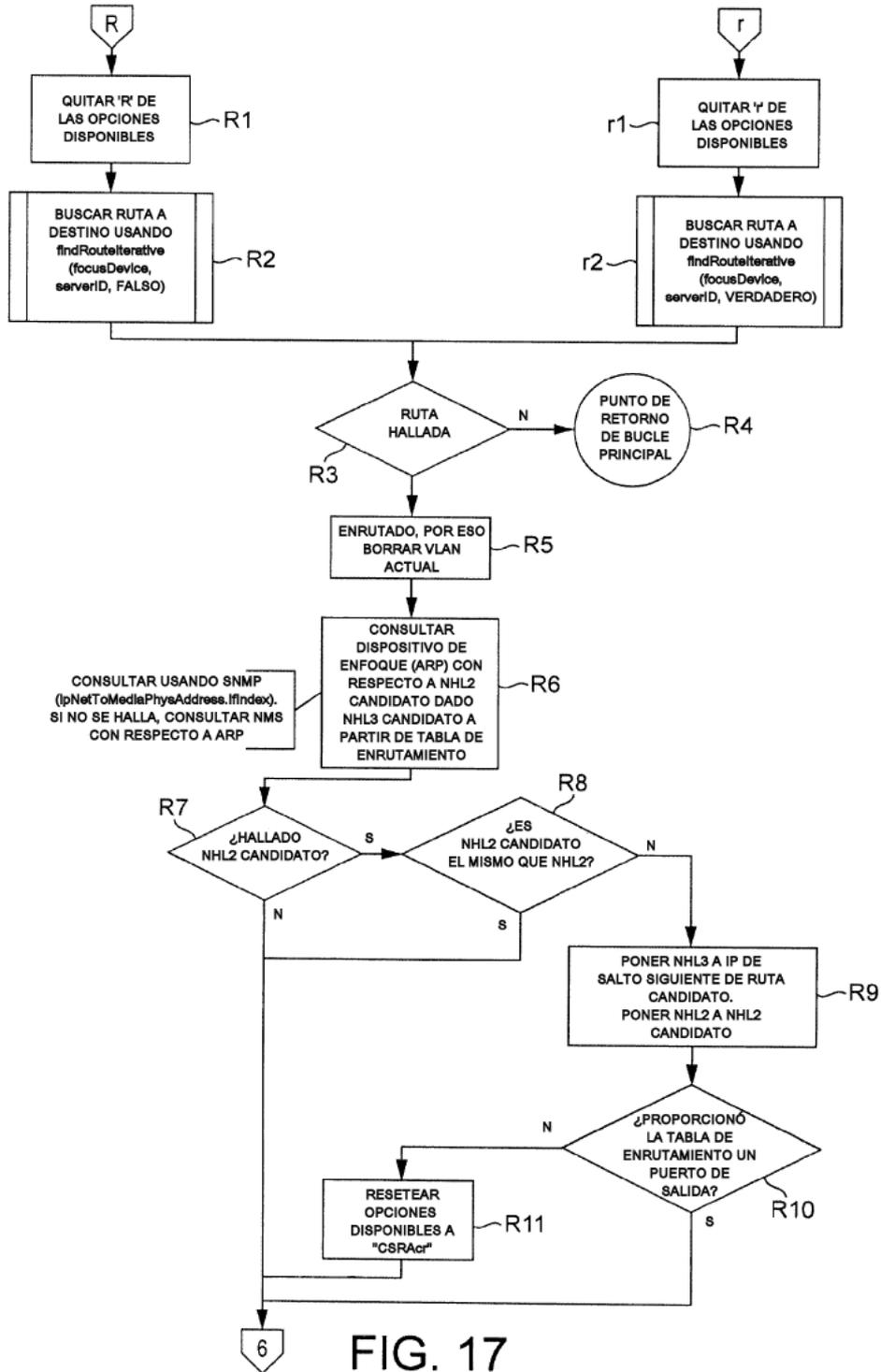


FIG. 17

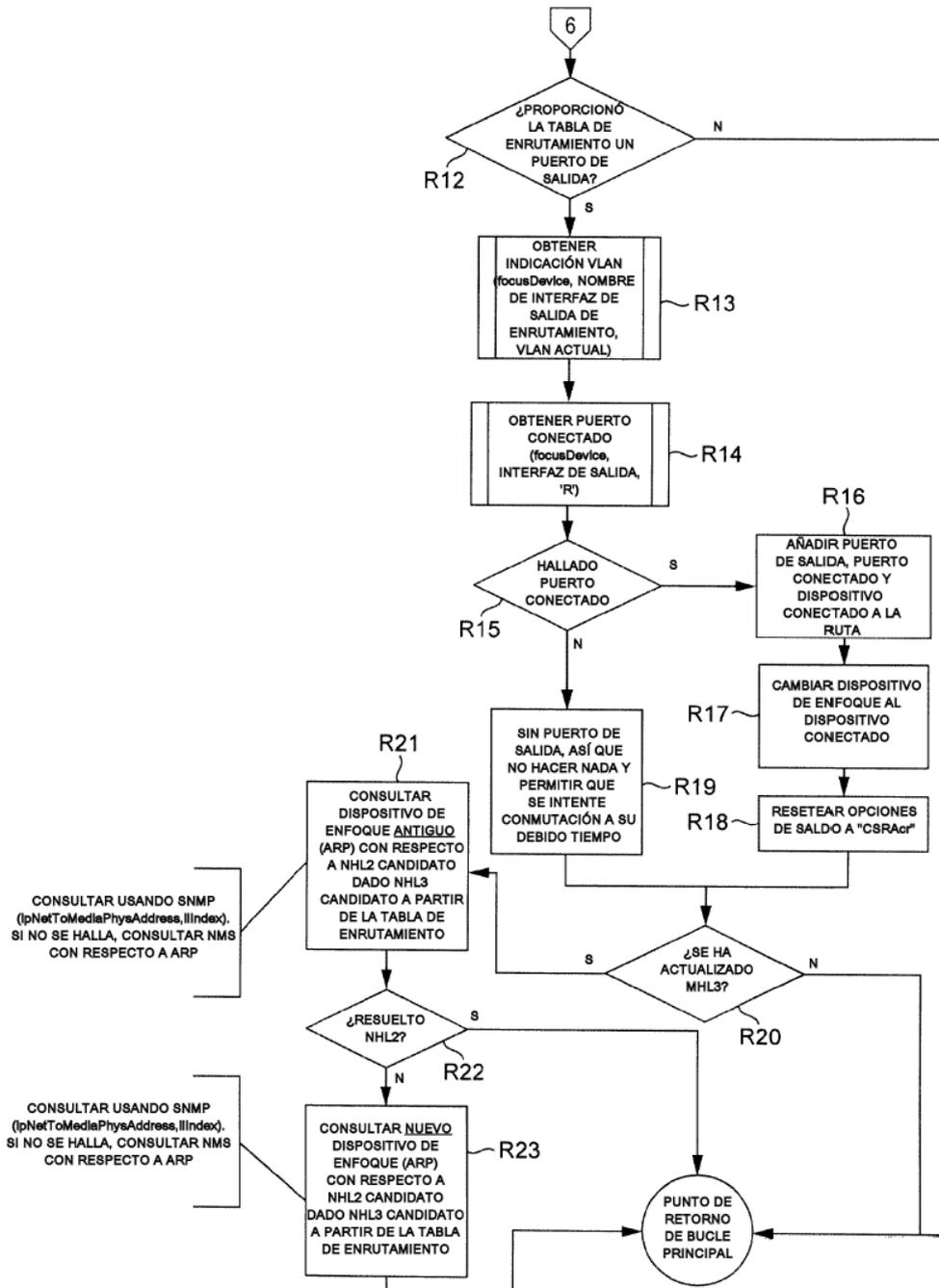


FIG. 18



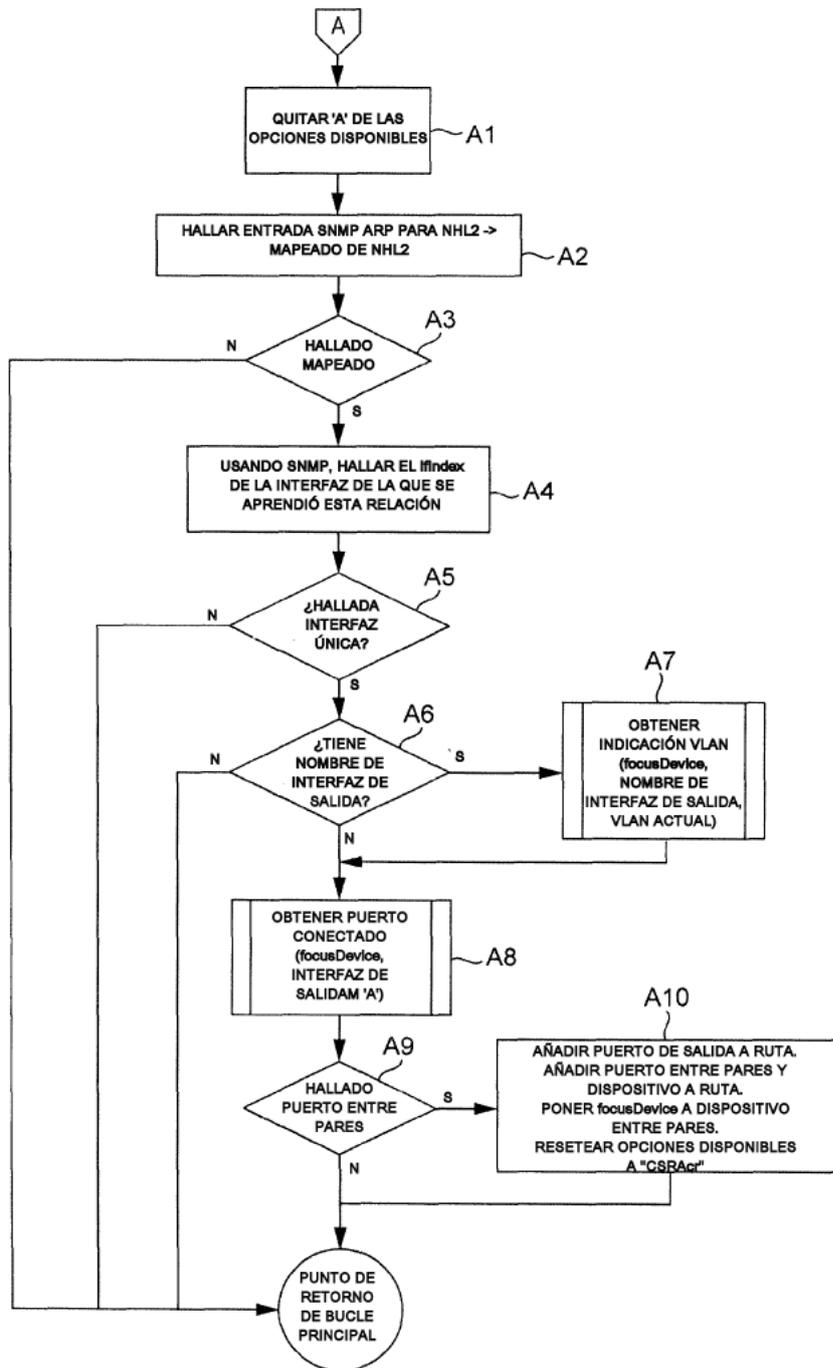


FIG. 20

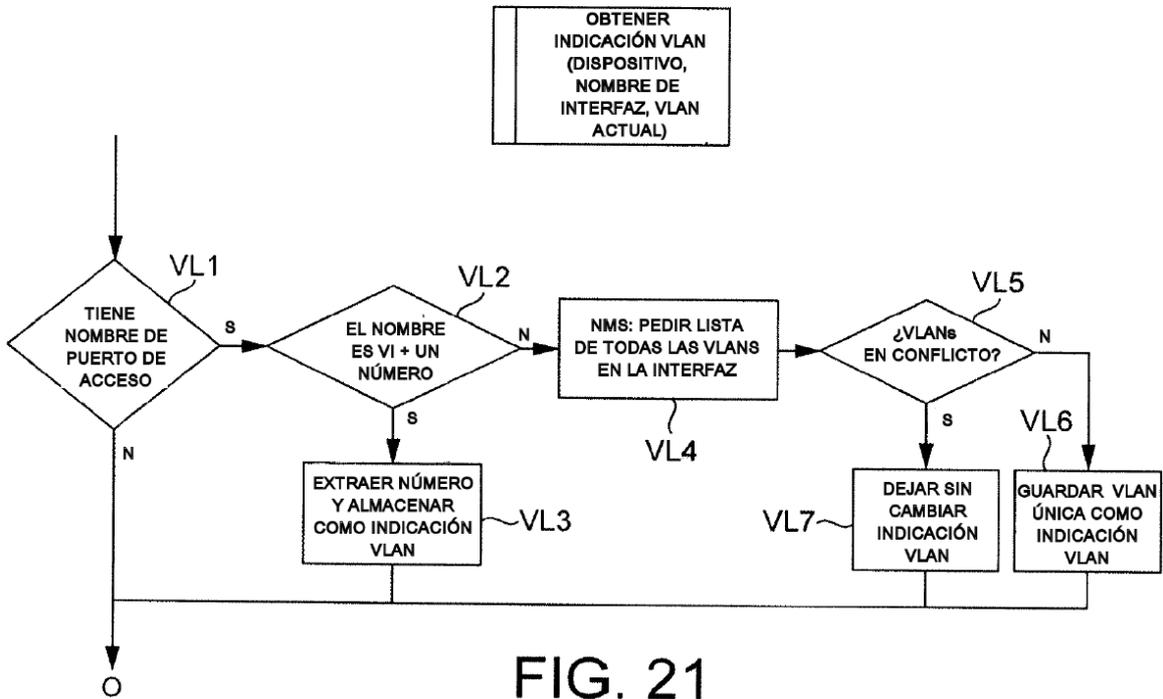


FIG. 21

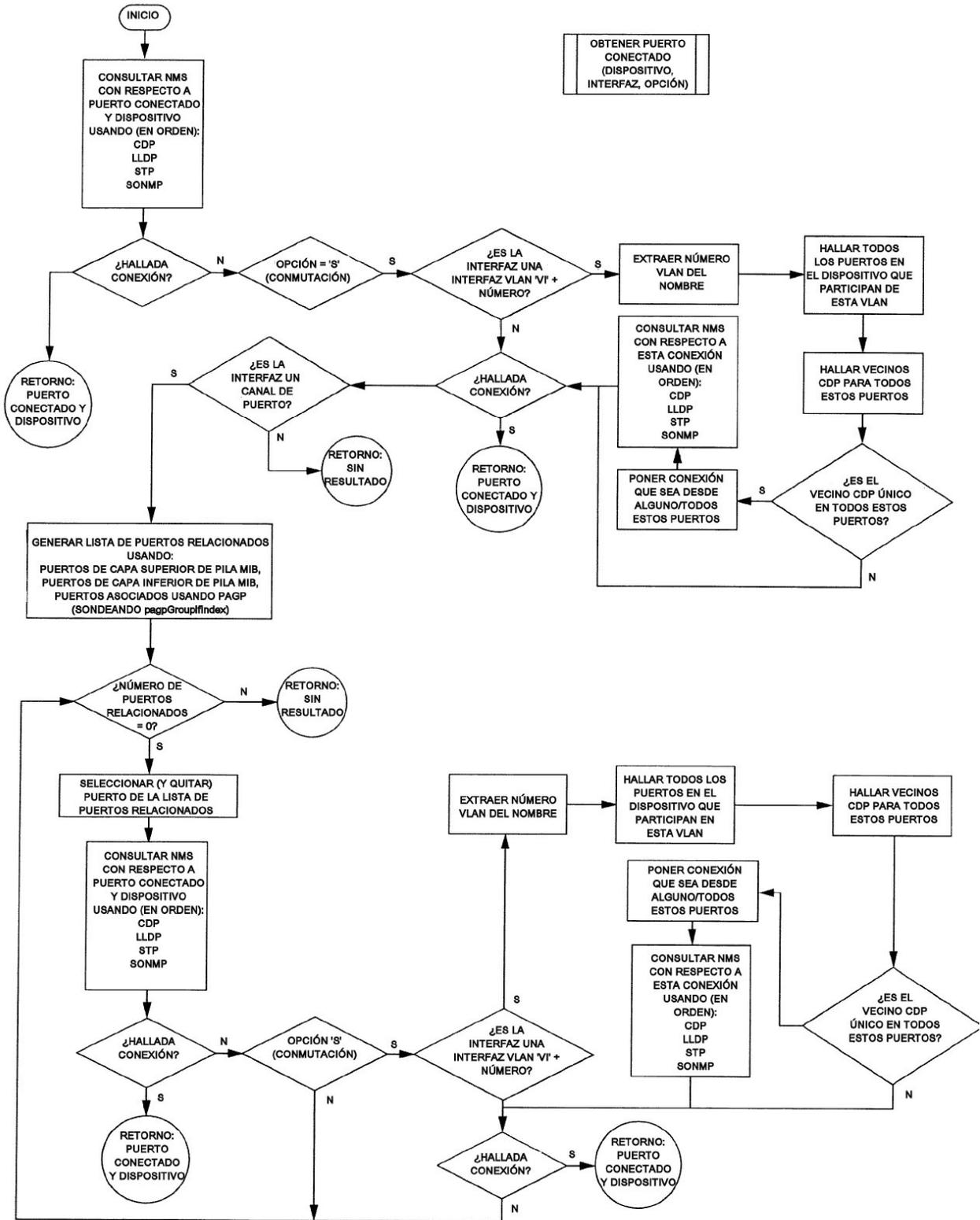


FIG. 22

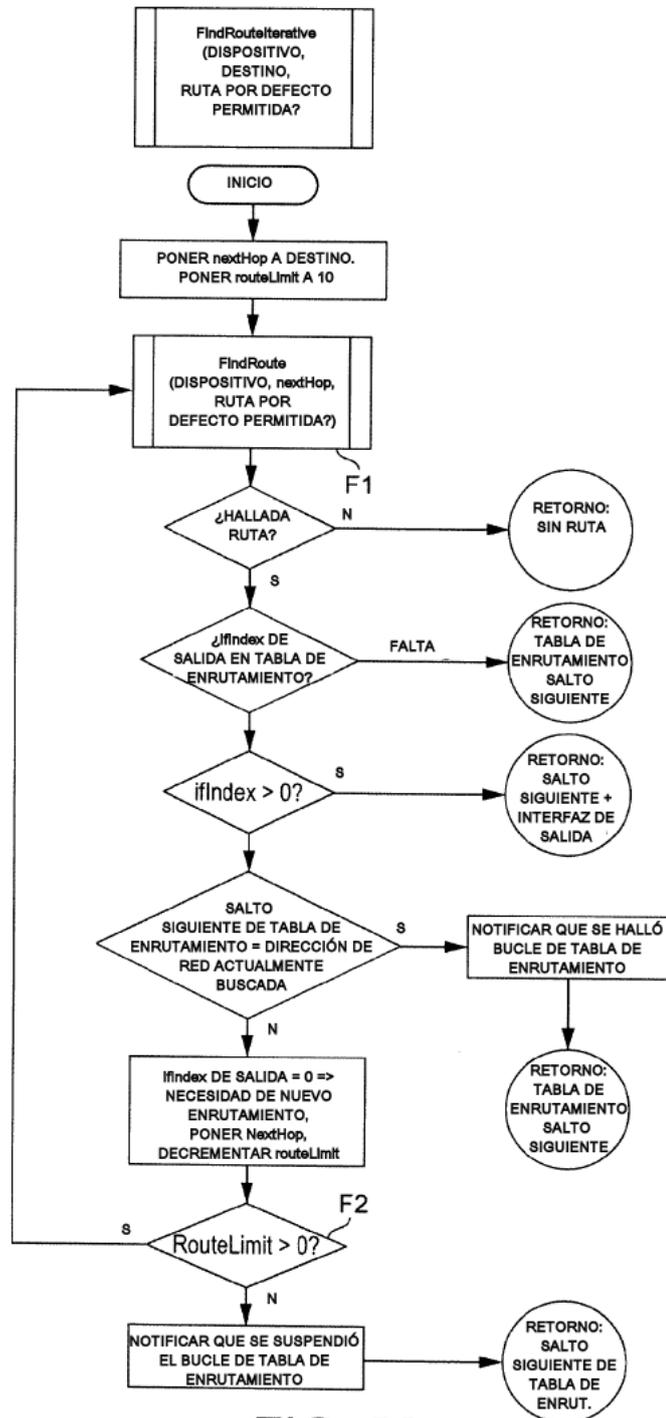


FIG. 23

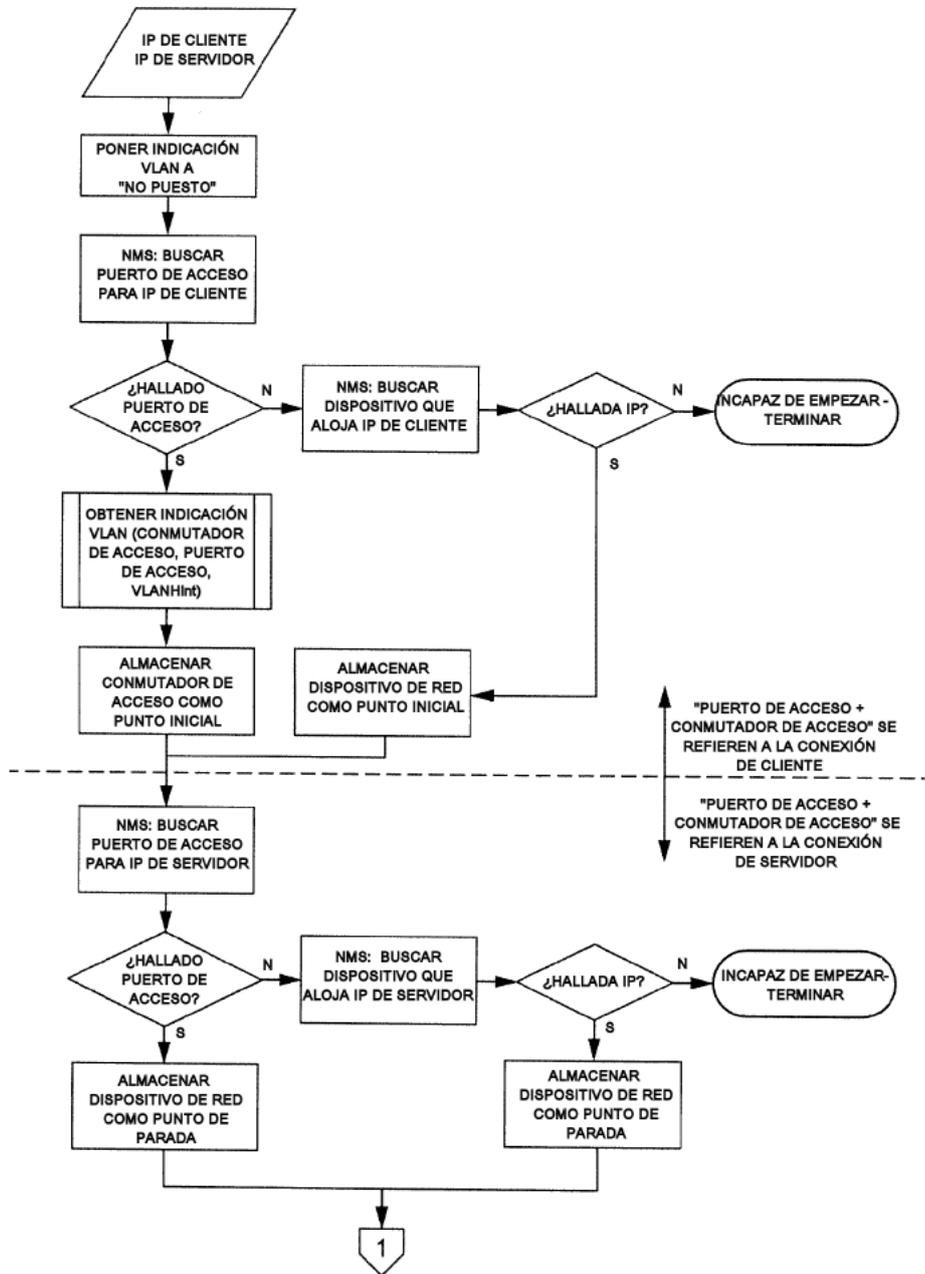


FIG. 24



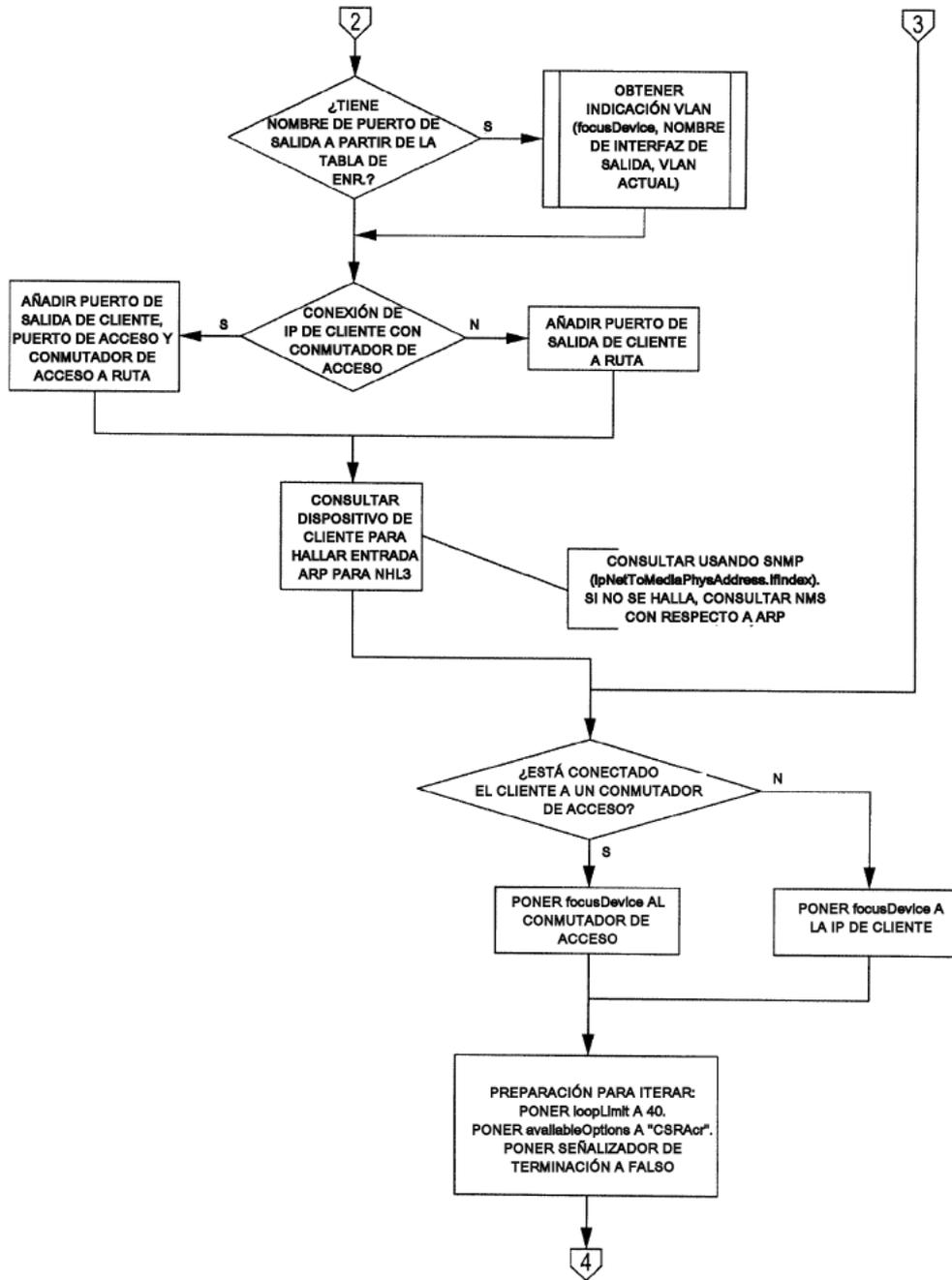


FIG. 26

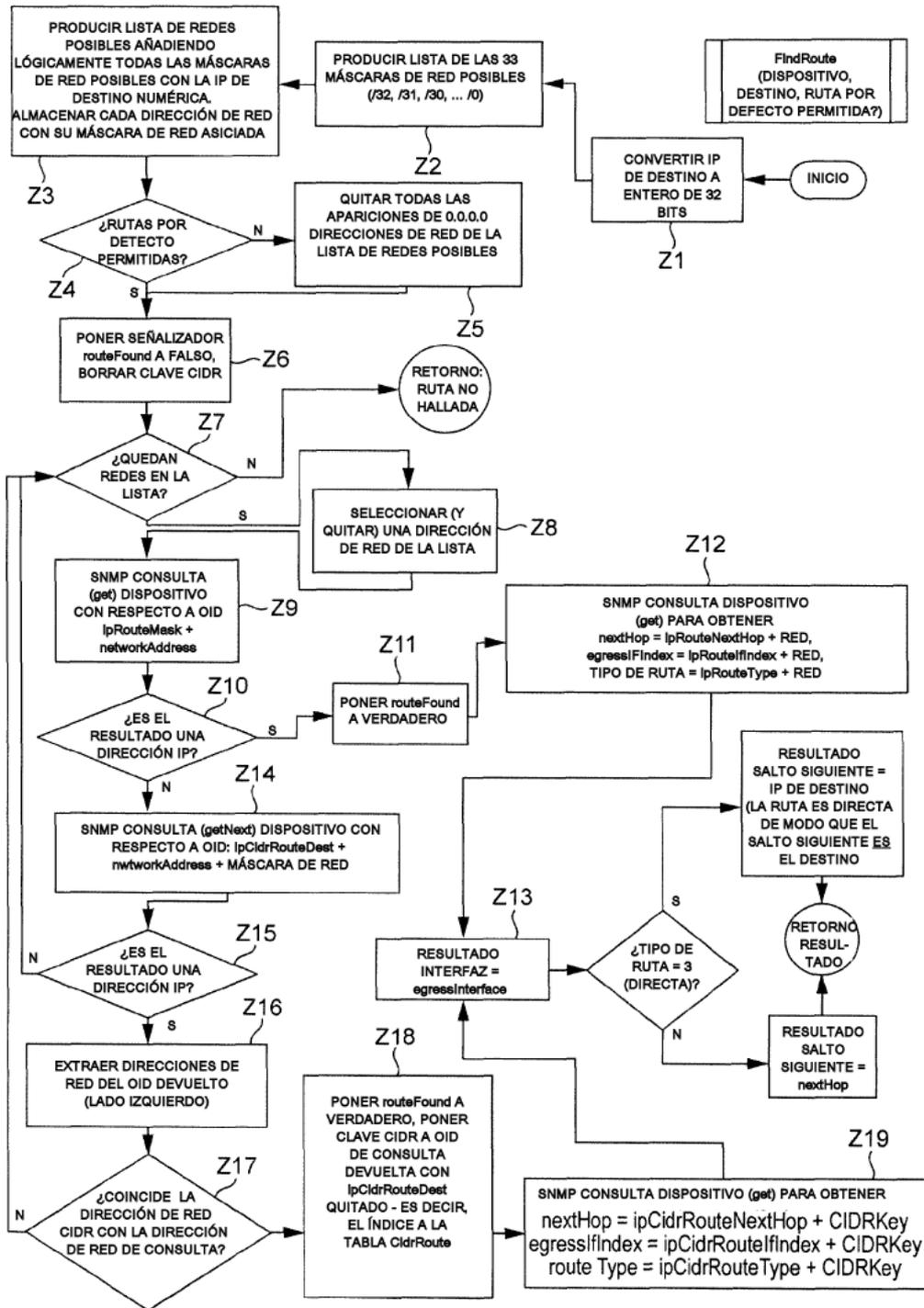


FIG. 27

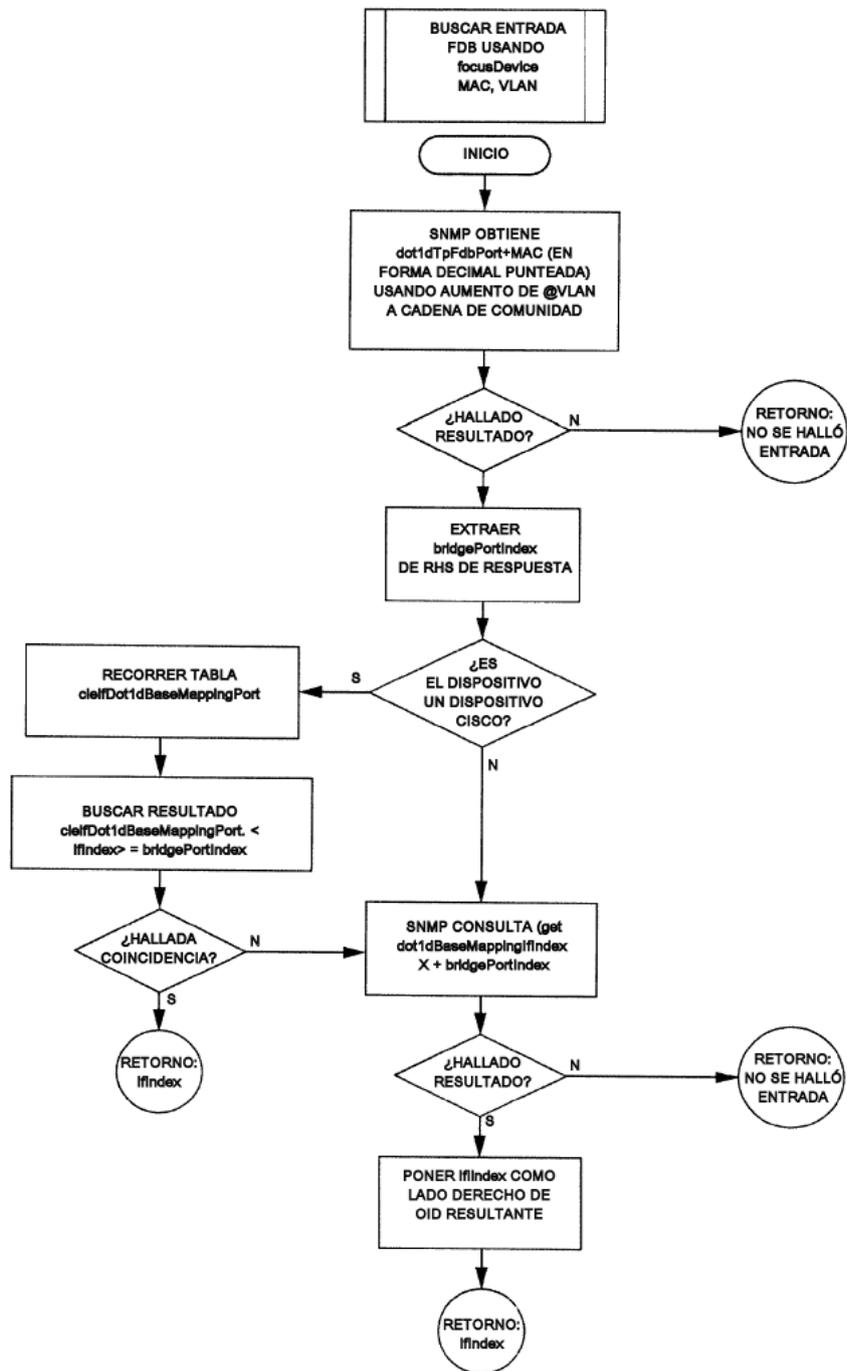


FIG. 28