(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 626 578**

(51) Int. CI.:

H04L 12/751 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **17.04.2014 PCT/EP2014/057962**

(87) Fecha y número de publicación internacional: **23.10.2014 WO14170457**

(96) Fecha de presentación y número de la solicitud europea: **17.04.2014 E 14719282 (7)**

(97) Fecha y número de publicación de la concesión europea: **03.05.2017 EP 2984800**

(54) Título: **Identificación de un puerto de salida de un dispositivo**

(30) Prioridad:

19.04.2013 GB 201307131
11.04.2014 GB 201406568

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
25.07.2017

(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

DESCRIPCIÓN

Identificación de un puerto de salida de un dispositivo

La presente invención se refiere a identificar un puerto de salida en un dispositivo en una red de ordenadores.

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, 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 cómo 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 computerizadas 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.

GB2462493A describe un método para descubrir y actualizar los puertos de salida en una red con el fin de realizar enrutamiento. Un ordenador supervisor envía una consulta a un dispositivo de enfoque/prueba y una clave de consulta con hash para hallar rápidamente entradas de tabla. El dispositivo de enfoque/prueba responde al ordenador supervisor con la información de puerto de salida pedida.

Resumen

Los autores de la presente invención han desarrollado un acercamiento para identificar un puerto de salida de un dispositivo consultando un dispositivo de enfoque para determinar qué haría con un paquete hipotético (en contraposición a consultar elementos específicos de un protocolo de enrutamiento). Si una primera consulta no devuelve una respuesta utilizable, se formula automáticamente una segunda consulta por una utilidad ejecutada en un ordenador supervisor.

Según un aspecto de la presente invención se facilita un método implementado por ordenador que consiste en identificar un puerto de salida de un dispositivo de enfoque conectado en una red de ordenadores, implementándose el método en un ordenador supervisor conectado a la red e incluyendo: generar un mensaje de consulta al dispositivo de enfoque, incluyendo el mensaje de consulta una dirección que identifica el dispositivo y una clave de consulta formulada en base a un identificador de destino, e incluyendo una instrucción legible en el dispositivo para devolver el mensaje de resultado incluyendo la identificación de un puerto de salida para mensajes dirigidos al destino identificado en el identificador de destino cuando el mensaje de consulta es recibido en el dispositivo; recibir un mensaje de resultado en el ordenador supervisor; leer el mensaje de resultado, y donde el mensaje de resultado no identifica un puerto de salida, generar de forma autónoma al menos un mensaje de consulta posterior; incluyendo al menos una de (i) una dirección diferente para un dispositivo diferente conectado en la red de ordenadores, y (ii) una clave de consulta diferente, seleccionada por el ordenador supervisor, donde se generan suficientes mensajes de consulta para identificar el puerto de salida del dispositivo de enfoque.

La invención también proporciona un producto de programa de ordenador que, cuando está instalado en un ordenador, implementa el método definido anteriormente. La invención también proporciona un ordenador supervisor con una interfaz para conexión a una red incluyendo al menos un dispositivo de enfoque y un procesador dispuesto para ejecutar un programa de ordenador que implementa el método definido anteriormente.

La clave de consulta diferente puede formularse en base al mismo identificador de destino, para acceder a una tabla de envío diferente en el mismo dispositivo. Alternativamente, la clave de consulta diferente puede formularse en base a un identificador de destino diferente, para acceder al mismo dispositivo.

Donde el dispositivo de enfoque es un dispositivo de enrutamiento, el mensaje de consulta puede ser dirigido a una tabla de enrutamiento del dispositivo de enrutamiento. En ese caso, el mensaje de resultado puede contener un mensaje de consulta posterior con una clave de consulta formulada en base a un identificador de destino diferente.

Donde el mensaje de resultado contiene una dirección de enrutamiento para un salto siguiente desde el dispositivo de enfoque en la red de ordenadores, el mensaje de consulta posterior puede ser formulado en base a la dirección de salto siguiente como el identificador de destino diferente. El mensaje de consulta posterior puede incluir una dirección que identifica el dispositivo de enfoque, con una clave de consulta diferente. Alternativamente, el mensaje de consulta posterior puede incluir una dirección diferente que identifica un dispositivo diferente, con la misma clave de consulta.

El mensaje de consulta posterior puede ser dirigido a una tabla de mapeado en el dispositivo identificado por la dirección, mapeando la tabla de mapeado identificadores de destino según un protocolo de dirección de

enrutamiento al identificador de destino según un protocolo de dirección de conmutación.

Donde el identificador de destino en el mensaje de consulta es según un protocolo de dirección de enrutamiento, la clave de consulta diferente puede ser formulada en base a un identificador de destino diferente que es según un protocolo de dirección de conmutación. Esto permite realizar de forma autónoma las denominadas investigaciones de capa 2/capa 3 en el ordenador supervisor.

Donde se formula al menos un mensaje de consulta posterior para consultar la tabla de mapeado, puede establecerse para conocer de qué interfaz en el dispositivo de enfoque derivó el dispositivo de enfoque un mapeado entre una dirección de enrutamiento y una dirección de conmutación para la dirección de salto siguiente.

Si el mensaje de resultado identifica un puerto de salida, el método puede estar configurado para determinar que el puerto de salida que ha sido devuelto en el mensaje de resultado no identifica de forma única un dispositivo conectado. En ese caso, el mensaje de consulta posterior generado de forma autónoma puede pedir asociaciones de puerto más alto y/o más bajo del dispositivo de enfoque.

La consulta que se transmite a cada dispositivo está adaptada para consultar cada dispositivo para determinar la identificación de un puerto de salida que representa los puertos de salida que el dispositivo utilizaría 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 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 router consultando qué hay en su tabla de enrutamiento activa al tiempo en que se recibe la consulta. El identificador de destino es la dirección de envío que se usa para la tabla de enrutamiento o tabla ARP, por ejemplo una dirección IP (protocolo de Internet).

La consulta propiamente dicha puede estar alojada en un mensaje o señal transmitida desde el ordenador supervisor al dispositivo que se consulta (dispositivo de enfoque). El mensaje de consulta o señal no constituye el flujo de mensajes para el que se ha de determinar la ruta. En cambio, cada consulta contiene un identificador de destino (dirección de envío) que consulta la tabla de envío de un dispositivo de enfoque para hallar cómo el dispositivo de enfoque manejaría un mensaje hipotético dirigido a ese destino si tuviese que hacer la decisión en el tiempo en que se recibió la consulta. Así, el dispositivo de enfoque devuelve un resultado que identifica un puerto de salida inmediato que se habría usado entonces para un mensaje real dirigido a dicho destino. Las consultas pueden transmitirse mientras la red está activa y mientras tiene lugar el flujo de mensajes. Sin embargo, también pueden transmitirse cuando el flujo de mensajes propiamente dicho no está activo - la técnica puede usarse en ambos contextos.

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 desde el ordenador supervisor al dispositivo de enfoque. En ese caso, la dirección de destino del mensaje de consulta es la del dispositivo de enfoque. Éste no es el mismo que el identificador de destino (dirección de envío) que se incluye en la consulta propiamente dicha. En una disposición alternativa, una señal o señales de consulta pueden ser enviadas 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.

La utilidad puede mejorarse usando una técnica definida aquí como “manipulación especulativa”. Permite generar una lista delimitada de claves especulativas para simplificar la búsqueda de la tabla de envío de tráfico, y reducir el tráfico necesario para tales consultas por una red de ordenadores. Generando una lista delimitada pequeña de claves especulativas, todas las peticiones pueden ser enviadas en paralelo (es decir, un mensaje de consulta común puede contener un número de claves de consulta), más bien que de forma secuencial.

El identificador de destino (dirección de envío) puede combinarse con cada índice embebido combinando lógicamente una secuencia de bits que representa la dirección de envío con una secuencia de bits que representa el índice embebido.

Donde la tabla de envío es una tabla de enrutamiento, cada índice en la tabla de enrutamiento es una máscara neta. Las claves que se usan en el mensaje de consulta pueden seleccionarse de claves solamente únicas que son generadas combinando lógicamente una secuencia de bits que representa la dirección de envío con una secuencia de bits que representa la máscara neta. La dirección de envío puede ser una dirección IP (protocolo de Internet), y la tabla de envío puede ser para un dispositivo de enrutamiento de capa 3.

La técnica de manipulación especulativa también es aplicable donde el índice embebido es un índice de interfaz de una tabla ARP en un dispositivo de envío de tráfico, tal como un router.

La utilidad es especialmente útil cuando se usa en un nuevo acercamiento desarrollado por los autores de la invención para identificar la ruta seguida a una red de dispositivos interconectados para un flujo de mensajes concreto. Dicha técnica se basa en usar una cantidad mínima de datos recogidos “desde el inicio” - específicamente

la topología de red estática y la posición de host final (qué clientes y servidores están conectados a qué conmutadores de acceso/borde), y recoge cualquier otra cosa que sea necesaria al vuelo y de forma altamente selectiva según sea preciso para tales datos altamente dinámicos. Para entornos dinámicos modernos, la capacidad de calcular la ruta de extremo a extremo ahora, es decir, en tiempo real, tiene amplia aplicabilidad. La recogida de los datos y su procesado tienen que ser muy rápidos para que el algoritmo sea un valor útil cuando se use con redes del mundo real y a gran escala.

El comportamiento en un dispositivo concreto se denomina “comportamiento por salto” (PHB). Aquí es donde la presente utilidad de identificación de puerto de salida es especialmente valiosa. PHB por sí solo no puede proporcionar una ruta de extremo a extremo. Sin embargo, conocer que un paquete sale del dispositivo en una interfaz específica puede ser útil si no se conoce qué dispositivo e interfaz están conectados a qué interfaz. Usando topología de red acoplada con PHB, puede realizarse un cálculo directo de una ruta de extremo a extremo a través de la red para un flujo de aplicación.

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)
- Protocolo de árbol de expansión (STP)
- IP Traceroute
- Descubrimiento de vecinos IPv6
- Adiciones / modificaciones / borrados de usuario

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 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.

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 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, routers, 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 cómo el mensaje de dicho punto final será dirigido a través de la red a un punto final de destino dado. La complejidad de tal determinación de ruta es exacerbada por el uso de múltiples rutas alternas, rutas redundantes, equilibrio de carga, etc.

Se ha intentado predecir cómo 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:

• Interfaz de entrada y tecnología ide interfaz de entrada

• Cabeceras de paquete (L2, L3, MPLS, ATM, etc)

• Rutas estáticas y conectadas directamente

• 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 VLANs), etc.

• Tecnologías de prevención de bucle - por ejemplo, PVSTP

• Protocolos de tunelización (MPLS, IPSEC, SSL, GRE)

• Carga equilibrada / enlaces redundantes

• Puertas de enlace por defecto

Sin embargo, aunque en principio era posible en el pasado predecir dónde se enviaría un paquete dado a continuación en un dispositivo concreto, esto requería una amplia cantidad de datos cuya recogida es lenta, y que podían 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.

La utilidad aquí descrita 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.

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 gráfica 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.

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 señalizado al usuario en una interfaz de usuario gráfica 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 llevará 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. 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, SLAs (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 “qué 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.

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 diagramática 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 router 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.

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.

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.

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 precedentes.

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 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 routers 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 París 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 qué 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. 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. 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.

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 en el que el paquete deberá ser enviado. Como ya se ha explicado anteriormente, según el modelo OSI, los identificadores para los dispositivos de enrutamiento de capa 3 son direcciones IP, mientras que los identificadores para los dispositivos de capa 2 son direcciones MAC.

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 2 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.

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

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

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 qué ruta se adoptará 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 en la que 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 pasos de la ruta que se describirán ahora.

En un nivel alto, el algoritmo usa la noción de un “dispositivo de enfoque” que es el dispositivo actualmente consultado acerca de dónde enviaría un paquete hipotético siguiente (es decir, desde qué interfaz enviaría el paquete hipotético). Comenzando en el dispositivo fuente, el algoritmo avanza hacia el dispositivo terminal (es decir, el último destino del paquete) evaluando cada dispositivo de enfoque por orden - si el dispositivo está operando en la capa 3, se le consulta qué 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 qué interfaz (puerto de salida) utilizaría para enviar paquetes unidos para la siguiente dirección de capa 2 (MAC) de salto a capa 3 (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.

Antes de comenzar el algoritmo iniciador, se localizan el dispositivo fuente y el dispositivo terminal. Esto puede no ser sencillo y más adelante se explican técnicas para lograrlo.

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).

Hallar el primer salto en la capa 3

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.

Visión general del 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. Como se explica, 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 el paso 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 el paso 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 el paso S1 se añade al registro de ruta S1A. En el paso 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 el paso 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 el paso S5 se determina si NHL2 es conocido. En caso afirmativo, el bucle pasa al paso S5A. En caso negativo, el proceso lleva a cabo el paso 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 (ifindices) a partir de la topología de red o recorriendo iflndex a partir de la tabla de interfaces del dispositivo propiamente dicho. Cada iflndex 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 el paso 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 el paso 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). Los pasos S6A, S7 y S7A pueden denominarse un salto L2. En este punto, consúltese la figura 2b. En el paso S5A, el dispositivo de enfoque es el dispositivo A. Éste 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.

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 el paso 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 el paso 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”.

En el paso 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 el paso 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 al paso S5, pero usando NHL3 y NHL2 puestos en el paso 9.

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:

“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.

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.

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.

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 operó, 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 tenía igualmente una opción de enrutar el paquete al router 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 enviará 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.

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 facilitarla 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). Los dispositivos VLAN no se han explicado todavía. 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.

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 el paso S5B. Es decir, el proceso envía una consulta para hallar la entrada ARP para la dirección IP fuente.

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, éstas 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 IPs 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 MACs (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.

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 está, ese dispositivo de red se pone como el dispositivo de enfoque.

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).

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.

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. Como se ha descrito anteriormente, el algoritmo requiere un salto siguiente inicial en la capa 3 (NHL3). La utilidad intenta consultar una 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.

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 iflndex 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 iflndex 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 iflndex devuelto sea no cero.

El paso 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 ipRouteIflindex.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.

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 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 es indicativo 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.

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 al paso S5B.

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. El paso 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 el paso de establecer en qué VLANs está participando el dispositivo según la topología 3 y como se ha registrado en el dispositivo. Estas VLANs se usan para ayudar a hallar entradas de tabla de envío para VLANs específicas (las FDBs 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).

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 qué interfaz se dirige hacia NHL2 a partir de registros ARP sondeando ipNetToMedia-PhysAddress 71 (figura 7). Es decir, la utilidad intenta aprender de qué 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.

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.

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.

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.

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. El paso 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 qué interfaz se dirige hacia NHL2 desde registros ARP sondeando con respecto a ipNetToMediaPhysAddress 71. Es decir, la utilidad intenta aprender de qué 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.

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.

Este paso “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).

Si el proceso termina en el dispositivo terminal previamente identificado y ese 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.

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 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.

Consulta de tabla de enrutamiento

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 ipRouteIfindex 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 válida/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 propiamente dicho está embebido en ipRouteEntry 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. 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 número de estos produce IDs 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.

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 los pasos 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 ipRouteIfIndex (egresslfindex), ipRouteNextHop e ipRouteType son recuperados (Z12) y devueltos en una respuesta al ordenador supervisor 16.

La interfaz de resultado se pone a egresslnterface (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.

Al examinar tablas de enrutamiento reales, no es insólito que la ruta hallada para una dirección IP dada no tenga una interfaz de salida válida 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. Según este acercamiento, en un primer paso una consulta de hallazgo de ruta única usa manipulación especulativa para hallar una entrada de enrutamiento para la dirección IP especificada (IP_X) como acaba de esbozarse. Si el ipRouteType asociado es “directo”, se devuelven IP_X (e $ipRouteIfIndex_X$) 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.

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

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 el paso Z10 no da lugar a una dirección IP, el proceso pasa al paso Z14 donde se envía una consulta SNMP (Obtener siguiente) al dispositivo, usando IPcidrRouteDest + dirección de red + máscara de red. Si el resultado no es una dirección IP, el proceso itera de nuevo al paso Z7 y realiza los pasos 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 al paso 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.

Como se representa en la figura P, en el proceso FindRouteIterative, se realiza el paso FindRoute F1 para la dirección IP requerida (IP_x). 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 ipRouteIfindex es igual a cero, entonces se realiza un paso FindRouteIterative posterior para la dirección IP de ipRouteNextHop, con los mismos cuatro resultados posibles.

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 o 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 ipNetToMediaEntry 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 qué 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.

Sigue una descripción de algoritmos alternativos para identificación de ruta. Ahora se hará referencia a la figura 12 para describir un proceso de iteración de bucle. La entrada al bucle principal se indica en la parte superior de la figura 12 con la flecha de entrada 4. La flecha de entrada 4 denota el estado al final de los procesos iniciadores que se describirán más adelante. El estado de entrada incluye:

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

Estos elementos se ponen por los procesos iniciadores que se describirán más adelante. A continuación se denominan “variables de estado”. La variable de estado titulada “opciones” tiene una secuencia ordenada de opciones de bucle. En el ejemplo presente, la secuencia ordenada incluye CSRAcr.

La variable de estado titulada “VLAN” es el identificador de red de área local virtual (número) para la VLAN con el que el paquete hipotético es etiquetado actualmente en este punto en la ruta.

En el paso L01 del bucle, se selecciona y ejecuta la primera opción (cabeza de la lista). Estas opciones se explican más adelante. Después de la ejecución de la opción, el proceso vuelve al punto de retorno L02 y se crea un nuevo estado (L03) siguiendo los pasos de procesado implementados por la opción de cabeza de lista. Se determina (L04) si este estado se ha producido antes, y en caso negativo, se almacena el estado (L05). Si el estado se ha producido antes, se genera un informe “bucle descubierto” y el límite de bucle se pone a cero, lo que tendrá el efecto de terminar el bucle.

En el paso L07, se decrementa el límite de bucle. Si el límite de bucle es menor 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 pone una condición “terminar es igual a verdadero”. En el paso L08, se lleva a cabo una comprobación en la condición de terminación, y si la condición de terminación es verdadera, el bucle principal termina. De otro modo, vuelve al punto de entrada de bucle principal usando el nuevo estado que se creó en el paso L03.

Obsérvese que en la ejecución de cada opción en una iteración de bucle, el primer paso en la ejecución de la opción es quitar dicha opción de la secuencia ordenada en la variable de estado opciones.

Al final de los pasos de procesado de la opción, puede ser que dicha opción se resetee de nuevo a la secuencia, o puede haberse quitado permanentemente, dependiendo de la opción y los resultados de los pasos de procesado.

Obsérvese también que en la ejecución de los pasos de procesado de una opción, las otras variables de estado (dispositivo de enfoque, NHL3, NHL3, VLAN) pueden alterarse individualmente o en total. La alteración de cualquier variable de estado da lugar a un nuevo estado, que puede constituir un estado de entrada nuevo para una iteración de bucle siguiente.

Ahora se hará referencia a la figura 12 para describir la segunda parte del proceso de bucle principal. En el paso L09 se determina si se ha llegado o no al dispositivo terminal esperado. En caso afirmativo, en el paso L10 se determina si el dispositivo terminal es un conmutador de acceso conectado a la IP de servidor. Si no lo es, entonces el proceso termina después de indicar un descubrimiento exitoso de ruta completa y devuelve una ruta completa. Si el dispositivo terminal es un conmutador de acceso conectado a la IP de servidor, se añade a la ruta la conexión de puerto de acceso y la IP de servidor, y entonces el proceso procede a completar el descubrimiento de ruta exitoso y devuelve una ruta completa antes de terminar en parada.

Si se determina en el paso L09 que no se ha llegado al dispositivo terminal esperado, entonces en el paso L14 se pregunta si NHL3 es la IP de servidor o no. Si no lo es, se determina que el descubrimiento de ruta completa ha tenido éxito, y se devuelve una ruta parcial antes de parar. Si NHL era la IP de servidor, esto indica que el proceso en el segmento L2 final y por ello el proceso puede saltar al destino incrementando el contador de segmentos de salto y poniendo el dispositivo de enfoque a NHL3.

La figura 14 ilustra la opción C. En el primer paso C1, se quita la opción de la secuencia en la variable opciones. En el paso C2, se realiza una comprobación para una sola interfaz cuya dirección de red coincide con el destino (IP de servidor). Si es así, esto indica que el algoritmo ha llegado a la porción de red de conmutación final, y NHL3 se pone a la IP de servidor. En el paso C3, el dispositivo de enfoque es consultado para hallar la entrada ARP para IP de servidor, y NHL2 se pone al 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 usando SNMP a la tabla ARP en el dispositivo de enfoque, o si no se halla, se consulta el cache ARP del sistema de gestión de red. Estas consultas se realizan según técnicas que se describen más plenamente más adelante.

En el paso C2, si falla la comprobación de la única interfaz para IP de servidor de destino, no se identifica una sola interfaz. Las variables de estado no son actualizadas. En este caso, todo lo realizado es evaluar (y desechar) la opción “C” y hallar que no es productiva.

La figura 15 ilustra la opción S. Según el paso S101, la opción S se quita de la secuencia ordenada. En el paso S102, se realiza una consulta al sistema de gestión de red o se usan consultas SNMP al dispositivo de enfoque para hallar si el dispositivo de enfoque aloja NHL3. Si NHL3 está en el dispositivo de enfoque, el proceso vuelve al punto de retorno de bucle principal (S104). Si la dirección de enrutamiento NHL3 no está en el dispositivo de enfoque, se determina si la dirección de conmutación NHL2 está puesta, y el dispositivo de enfoque es consultado en la tabla de mapeado (ARP) con respecto a NHL2 dado NHL3. Las consultas se hacen como se describe más plenamente más adelante. En el paso S106 se determina si la indicación VLAN está puesta o no. Las indicaciones VLAN se explicarán más adelante. En caso negativo, en el dispositivo de enfoque se determina una lista de VLANs a partir del dispositivo de enfoque o a partir del sistema de gestión de red. Se selecciona una VLAN de la lista y se realiza una búsqueda para la entrada de base de datos de envío usando el dispositivo de enfoque, NHL2 y VLAN. La búsqueda de la entrada de base de datos de envío ilustrada en el paso S109 se representa en un diagrama de flujo en la figura X. Volviendo al paso S106, si la indicación VLAN está puesta, el proceso va directamente al paso 110, que es una búsqueda de entrada FDB como en el paso S109 usando la indicación VLAN como la VLAN consultada dentro de la FDB. En el paso S112 (también S111), se determina si hay o no una entrada hallada en la base de datos de envío. Si la hay, el proceso pasa a la segunda parte de la opción S que se representa en la figura 16 (flecha de entrada 5). Si, después del paso S111, no se halla entrada FDB, se introduce un bucle VLAN hasta que se determina si hay una entrada FDB o que el proceso deberá 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 ha descrito, antes, si se halla una entrada de base de datos de envío, esto indica el puerto de salida para la ruta S115. Esto puede usarse para obtener el dispositivo conectado siguiente de la topología de red, como se representa en el paso S117, y se describe más plenamente más adelante. El paso S116 es el paso de obtener una indicación VLAN que se representa en la figura 2 y se explicará más adelante.

El paso S117 se representa en la figura 22, que es un diagrama de flujo que ilustra el proceso para obtener el puerto conectado y por lo tanto el dispositivo conectado posterior del sistema de gestión de red en base al puerto de salida devuelto de la base de datos de envío: si se halla el puerto conectado en el paso S118, el puerto conectado y el dispositivo conectado se añaden a la ruta identificada (paso S119) y en el paso S120 el dispositivo de enfoque se cambia al dispositivo conectado. En el paso S121, las opciones de bucle se resetean a CSRAcr. El procesador vuelve entonces al bucle principal en el paso S122. Volviendo al paso S118, si no se halla el puerto conectado, el proceso salta hacia delante a NHL3 incrementando un contador de segmentos de salto 89 (figura 8) y cambiando el dispositivo de enfoque a NHL3. El contador de segmentos de salto se implementa en el ordenador de gestión en hardware, microprogramas o software y permite indicar los segmentos de la ruta como saltados donde sea claro que el dispositivo conectado siguiente no se pueda conocer fácilmente a partir de los pasos de proceso precedentes. En el paso S124 se determina si el dispositivo de enfoque no es el destino (IP de servidor). Si no lo es, las opciones de bucle se resetean a CSRAcr en el paso S125. Si el dispositivo de enfoque es el servidor destino, en el paso 126 se determina si el destino está en un conmutador de acceso conocido, y si es así, NHL 3 se pone a la dirección de conmutación de acceso de servidor. Después de poner las opciones de bucle en el paso S125, en el paso S128 el dispositivo de enfoque es consultado con respecto a NHL2 usando la dirección NHL3 que se puso en el paso S127, consultando la tabla ARP del dispositivo de enfoque o un NMS.

Obsérvese que la opción S incluye el paso de quitarla de las opciones disponibles en el paso S101, y luego resetearla de nuevo a las opciones disponibles en los pasos S121 y S125 en base a los resultados de los pasos de procesado.

Ahora se hará referencia a la figura 17 para describir las opciones R y r. Cada una de dichas opciones comienza con la extracción de dicha opción de la secuencia ordenada en la variable opción, pasos R1, r1. En el paso R2, r2, se provoca un proceso para buscar la ruta a la IP de destino (IP de servidor) usando un proceso iterativo de hallazgo de

ruta que se ilustra en la figura P. En la opción R, el proceso opera donde no hay ruta permitida (la ruta por defecto permitida es igual a falso). En la opción r, el proceso permite una ruta por defecto (ruta por defecto permitida es igual a verdadero). Si no se halla ruta, paso R3, el procesador vuelve al bucle principal. Si se halla una ruta, entonces se envía una consulta al dispositivo de enfoque usando un NHL3 candidato que ha sido determinado a partir de la tabla de enrutamiento a partir de la se halló la ruta. Esta consulta se hace a la tabla ARP del dispositivo para determinar un NHL2 candidato que corresponda al NHL3 candidato. Si no se halla ningún NHL2 candidato, el proceso pasa al punto de entrada 6 para la segunda parte de la opción R/r. Si se halla un NHL2 candidato a partir de la consulta ARP, entonces se realiza una comprobación en el paso R8 para determinar si el NHL2 candidato es el mismo que NHL2 que está registrado como la variable de estado en el estado de entrada al proceso R/r. Si son los mismos, entonces el proceso pasa al punto de entrada 6. Si no son los mismos, paso R8 siguiente, si el NHL2 candidato no es el mismo que el estado de entrada NHL2, entonces NHL3 se pone a la IP de salto siguiente de la ruta candidato, y NHL2 se pone al NHL2 candidato. En el paso R10 se determina si las consultas de tabla de enrutamiento en R2 y r2 también proporcionó un puerto de salida. En caso afirmativo, el proceso pasa al punto de entrada 6. En caso negativo, las opciones se resetean a CSRAcr en el paso R11. La figura 18 ilustra el punto de entrada 6 en la parte superior de la figura. En el paso siguiente R12, se determina si la tabla de enrutamiento proporciona un puerto de salida. En caso negativo, el proceso vuelve al bucle principal. En el paso R13, el proceso de obtención de indicación VLAN se realiza según la figura 21 y se describirá más adelante.

Entonces se realiza el proceso de obtención de puerto conectado como se ilustra en la figura 22. En el paso R15, se determina si el puerto conectado se halla o no. En caso afirmativo, 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 resetean a CSRAcr. Si no se halla puerto conectado, no se realiza nada en esta etapa. El proceso pasa al paso R20 donde se determina si se ha actualizado NHL3. Si no se ha actualizado, el proceso vuelve al bucle principal. Si se ha actualizado, se consulta el dispositivo de enfoque antiguo con respecto al NHL2 candidato, dado el NHL3 candidato de la tabla de enrutamiento. Si, después de dicho paso, se ha resuelto NHL2, el proceso vuelve al bucle principal. Si no, se consulta el nuevo dispositivo de enfoque con respecto al candidato NHL 2 dado el NHL3 candidato.

Se hará referencia a la opción c con referencia a la figura 19. En el primer paso c1, se quita c de las opciones en la variable de estado. En el paso c2 se lleva a cabo una comprobación con respecto a una sola interfaz cuya dirección de red coincide con la dirección NHL3 de red. Si no se halla una única interfaz, el proceso vuelve al bucle principal. Si se halla una única interfaz, y hay un nombre de interfaz de salida, se lleva a cabo el proceso de indicación VLAN que se describirá con referencia a la figura 21. Después del paso C4, el puerto conectado se obtiene usando el proceso de obtención de puerto conectado representado en la figura 22.

En el paso c7, se determina si se halla un puerto entre pares. Si se halla, el puerto de salida, el puerto entre pares y el dispositivo entre pares se añaden a la ruta y el dispositivo de enfoque se pone al dispositivo entre pares. Las opciones disponibles se resetean a CSRAcr en c8. Si no se halla puerto entre pares en el paso c7, el proceso vuelve al bucle principal.

Ahora se hará referencia a la figura 20 para describir la opción A. En el paso A1, se quita la opción A de la secuencia de opciones en la variable de estado. En el paso A2, se halla la entrada SNMP ARP para mapeado de NHL3 a NHL2. Si no se halla mapeado, el proceso vuelve al bucle principal.

Si se halla mapeado, el proceso usa SNMP para hallar el iflndex de la interfaz de la que se aprendió la relación. En el paso A5, se determina si se ha hallado o no una única interfaz. En caso negativo, el proceso vuelve al bucle principal. En caso positivo, el proceso pasa al paso A6 donde se determina si hay disponible un nombre de interfaz de salida. Si lo hay, se provoca el proceso de indicación VLAN como se describirá con referencia a la figura 21. Entonces, en A8, se provoca el proceso de obtención de puerto conectado como se ilustra en la figura 22.

Si, como resultado del proceso de obtención de puerto conectado, se halla un puerto entre pares, el puerto de salida se añade a la ruta, el puerto entre pares y el dispositivo entre pares se añaden a la ruta y el dispositivo de enfoque se pone al dispositivo entre pares. Además, las opciones disponibles se resetean a CSRAcr. Si no se halla puerto entre pares, el proceso vuelve al bucle principal.

Ahora se hará referencia a la figura 21 para explicar el proceso de indicación VLAN. Este proceso se usó en la opción A en el paso A7, la opción c en el paso c5, la opción R/r en el paso R13 y la opción S en el paso S116. Además, se usa en uno de los procesos iniciadores que todavía no se han explicado. El proceso empieza en el paso VL1 con un nombre de puerto de acceso. En el paso VL2 se determina si el nombre tiene forma de “VL+ un número”, y, en caso afirmativo, se extrae el número y se almacena como una indicación VLAN en el paso VL3.

En caso negativo, entonces en el sistema de gestión de red se pide una lista de todas las VLANs en la interfaz. El paso VL5 comprueba cualesquiera VLANs contradictorias. Si no hay ninguna, entonces se almacena la única VLAN como una indicación VLAN en el paso VL6. Si hay VLANs contradictorias, cualquier indicación VLAN que ya haya sido almacenada se deja sin cambiar.

Las indicaciones VLAN resuelven el problema que puede surgir en ciertas redes que usan una técnica de conmutación llamada STP (Protocolo de árbol de expansión) que se usa para no tener bucles lógicos en porciones conmutadas (capa 2) de la red (para evitar que el tráfico itere de forma infinita). El protocolo se usa para decidir adonde un dispositivo de conmutación deberá enviar un paquete dado.

Es decir, si el dispositivo está conmutando, el conmutador observará la cabecera de capa 2 (MAC/Ethernet) y observará la dirección destino de capa 2 (NHL2) y luego consultará una base de datos interna (FDB – base de datos de envío) para conocer desde cuál de sus puertos se deberá enviar el paquete. Muchas empresas usan una extensión a STP llamada PVSTP (Protocolo de árbol de expansión por VLAN) por lo que cada paquete se marca también con un identificador VLAN. El conmutador mantiene entonces FDBs separadas – una por VLAN.

Esto se hace parcialmente por razones de eficiencia y en parte para permitir topologías virtuales más complejas. Así, es perfectamente posible (y no insólito) que dos paquetes con el mismo destino de capa 2 salgan por puertos diferentes cuando se les etiqueta como que están en VLANs diferentes aunque su destino sea el mismo dispositivo/puerto.

La consecuencia de esto es que el proceso no puede rastrear simplemente todas las FDBs por VLAN hasta que se halle una coincidencia. Es importante conocer a priori de qué VLAN es el paquete etiquetado como elemento de la misma.

Este etiquetado VLAN puede tener lugar en diferentes lugares en la red, por ejemplo, en el puerto de acceso fuente - es decir, donde el dispositivo fuente está físicamente conectado, o en algún otro lugar en la red - no es insólito que una etiqueta VLAN sea sustituida por otra (esto se denomina enrutamiento entre VLAN).

Por ejemplo, dado un paquete que llega al dispositivo de red D (en la ruta A->B->C->D), D solamente puede ser consultado con respecto a la interfaz de salida correcta si se conoce la VLAN en la que estaría el paquete procedente de A en el punto en que llegue a D. Puede ser, por ejemplo, que A ponga el paquete en VLAN 100, B lo pase (usando VLAN 100), luego C cambie 100 a 200 y luego D lo conmute usando VLAN 200.

Por esta razón hay que 'llevar' una indicación VLAN a través de la red desde nuestro dispositivo fuente a nuestro dispositivo destino como parte del paquete hipotético que se rastrea. Así, donde sea aplicable, la indicación VLAN se usa, anula, resetea o actualiza.

Como ya se ha mencionado, la figura 23 ilustra el proceso findRouteIterative que se usa en las opciones R y r. El proceso implica un bucle de hallazgo de ruta que comienza en el paso F1 y termina en una comprobación de límite de ruta F2. El proceso findRouteIterative determina entonces si se ha hallado una ruta y permite que se localice un índice de salida perteneciente a la ruta.

Antes de embarcarse en el bucle principal, hay tres procesos iniciadores que se implementan con el fin de poner el estado de entrada para la primera iteración del bucle. Un primer proceso iniciador se representa en la figura 24, que pone como punto de inicio el conmutador de acceso o el dispositivo de red identificado para el dispositivo fuente (que en la figura 24 se denomina el lado de cliente. Igualmente, un conmutador de acceso o dispositivo de red es almacenado como un punto de parada, en base al dispositivo destino, denominado en la figura 24 el lado de servidor. En la figura 24, la IP de cliente y la IP de servidor son las direcciones fuente y destino respectivamente.

La figura 25 es un proceso iniciador para establecer las direcciones de estado de entrada inicial NHL3 y NHL2.

La figura 26 ilustra un tercer proceso iniciador que pone el dispositivo de enfoque para el estado de entrada inicial.

Obsérvese que el primer proceso iniciador de la figura 24 conduce al segundo proceso iniciador de la figura 25, y el segundo proceso iniciador de la figura 25 conduce al tercer proceso iniciador de la figura 26. El tercer proceso iniciador conduce al punto de entrada principal 4 del bucle principal representado en la figura A.

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 multi-protocolo (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 entremedio). 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 APIs 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 implementado por ordenador que consiste en identificar un puerto de salida (Po) de un dispositivo de enfoque conectado en una red de ordenadores, implementándose el método en un ordenador supervisor (16) conectado a la red e incluyendo:

generar un mensaje de consulta al dispositivo de enfoque, incluyendo el mensaje de consulta una dirección que identifica el dispositivo y una clave de consulta formulada en base a un identificador de destino, e incluyendo una instrucción legible en el dispositivo para devolver un mensaje de resultado incluyendo la identificación de un puerto de salida (Po) para mensajes dirigidos al destino identificado en el identificador de destino cuando el mensaje de consulta es recibido en el dispositivo;

recibir un mensaje de resultado en el ordenador supervisor (16);

leer el mensaje de resultado, y donde el mensaje de resultado no identifica un puerto de salida (Po) generar de forma autónoma al menos un mensaje de consulta posterior;

incluyendo al menos una de (i) una dirección diferente para un dispositivo diferente conectado en la red de ordenadores, y (ii) una clave de consulta diferente, seleccionada por el ordenador supervisor (16), donde se generan suficientes mensajes de consulta para identificar el puerto de salida (Po) del dispositivo de enfoque.

2. Un método según la reivindicación 1, donde la clave de consulta diferente se formula en base al mismo identificador de destino, para acceder a una tabla de envío diferente en el mismo dispositivo.

3. Un método según la reivindicación 1, donde la clave de consulta diferente se formula en base a un identificador de destino diferente, para acceder al mismo dispositivo.

4. Un método según la reivindicación 3, donde el dispositivo de enfoque es un dispositivo de enrutamiento (12), y el mensaje de consulta es dirigido a una tabla de enrutamiento (92) del dispositivo de enrutamiento (12), conteniendo el mensaje de resultado una indicación de que el tipo de ruta al identificador de destino es indirecto, donde el paso de leer el mensaje de resultado incluye detectar el tipo indirecto y generar un mensaje de consulta posterior con una clave de consulta formulada en base a un identificador de destino diferente.

5. Un método según la reivindicación 3, donde el mensaje de resultado contiene una dirección de enrutamiento para un salto siguiente desde el dispositivo de enfoque en la red de ordenadores, y donde el mensaje de consulta posterior es formulado en base a la dirección de salto siguiente como el identificador de destino diferente.

6. Un método según la reivindicación 5, donde el mensaje de consulta posterior incluye una dirección que identifica el dispositivo de enfoque, con una clave de consulta diferente.

7. Un método según la reivindicación 5, donde el mensaje de consulta posterior incluye una dirección diferente que identifica el dispositivo diferente, con la misma clave de consulta.

8. Un método según la reivindicación 5, 6 o 7, donde el mensaje de consulta posterior es dirigido a una tabla de mapeado (91) en el dispositivo identificado por la dirección, mapeando la tabla de mapeado (91) identificadores de destino según un protocolo de dirección de enrutamiento a identificadores de destino según un protocolo de dirección de conmutación.

9. Un método según la reivindicación 1, donde el identificador de destino en el mensaje de consulta es según un protocolo de dirección de enrutamiento, y donde la clave de consulta diferente es formulada en base a un identificador de destino diferente que es según un protocolo de dirección de conmutación.

10. Un método según la reivindicación 8, donde el al menos único mensaje de consulta posterior es formulado para consultar la tabla de mapeado (91) para conocer de qué interfaz en el dispositivo de enfoque derivó un mapeado entre una dirección de enrutamiento y una dirección de conmutación para la dirección de salto siguiente.

11. Un método según la reivindicación 1, donde si el mensaje de resultado identifica un puerto de salida (Po), el método busca identificar un dispositivo conectado al puerto de salida (Po) en base a una topología de red (3).

12. Un método según la reivindicación 11, donde si se determina que el puerto de salida (Po) que ha sido devuelto en el mensaje de resultado no identifica de forma única un dispositivo conectado, el al menos único mensaje de consulta posterior pide asociaciones de puerto más alto y/o más bajo del dispositivo de enfoque.

13. Un producto de programa de ordenador incluyendo un programa de ordenador que implementa una utilidad de identificación de puerto de salida, que, cuando es ejecutado por un ordenador, implementa un método de alguna de las reivindicaciones 1 to12.

14. Un ordenador que tiene una interfaz de red (86) para conectar el ordenador a una red incluyendo al menos un dispositivo de enfoque; y

un procesador (80) dispuesto para ejecutar un programa de ordenador para implementar el método de alguna de las reivindicaciones 1 a 12.

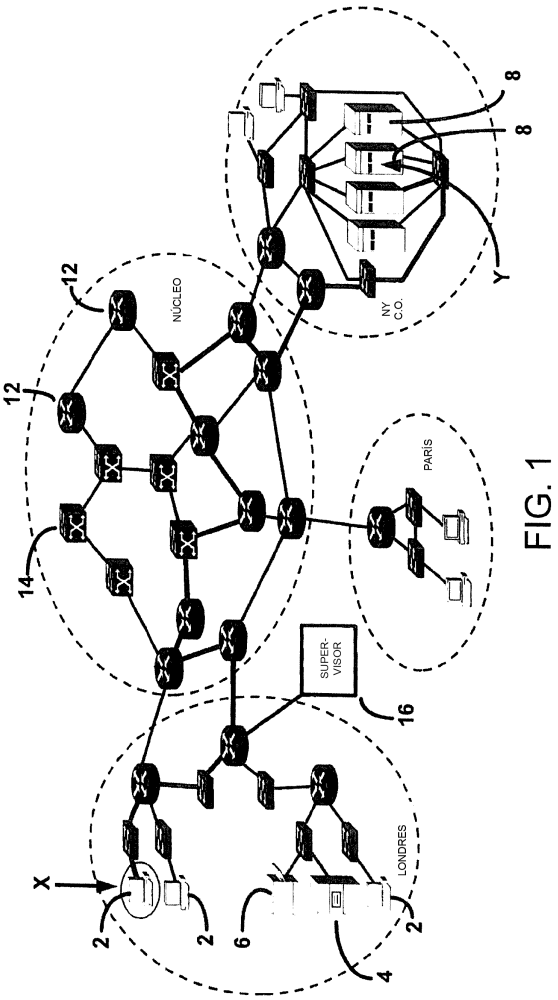

FIG. 1

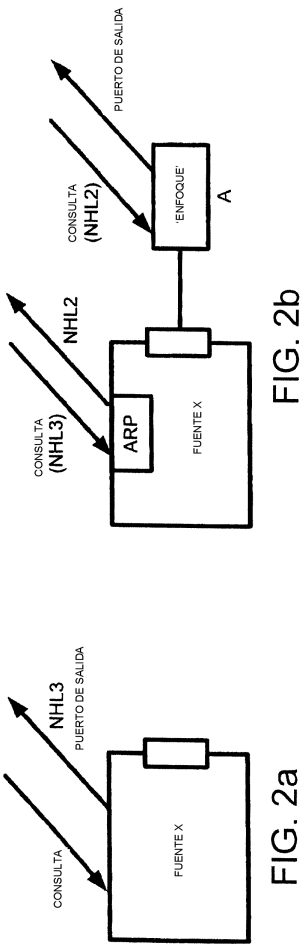

FIG. 2a

FIG. 2b

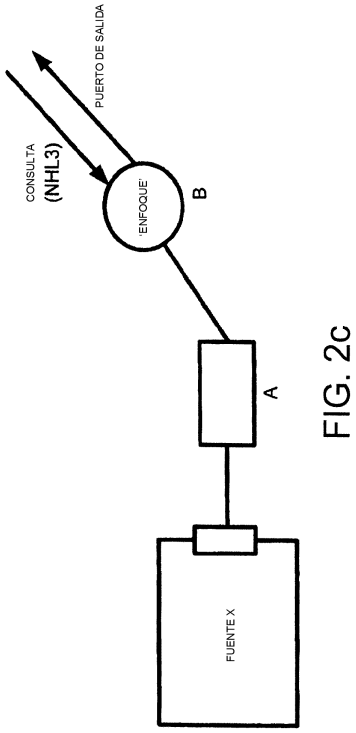

FIG. 2c

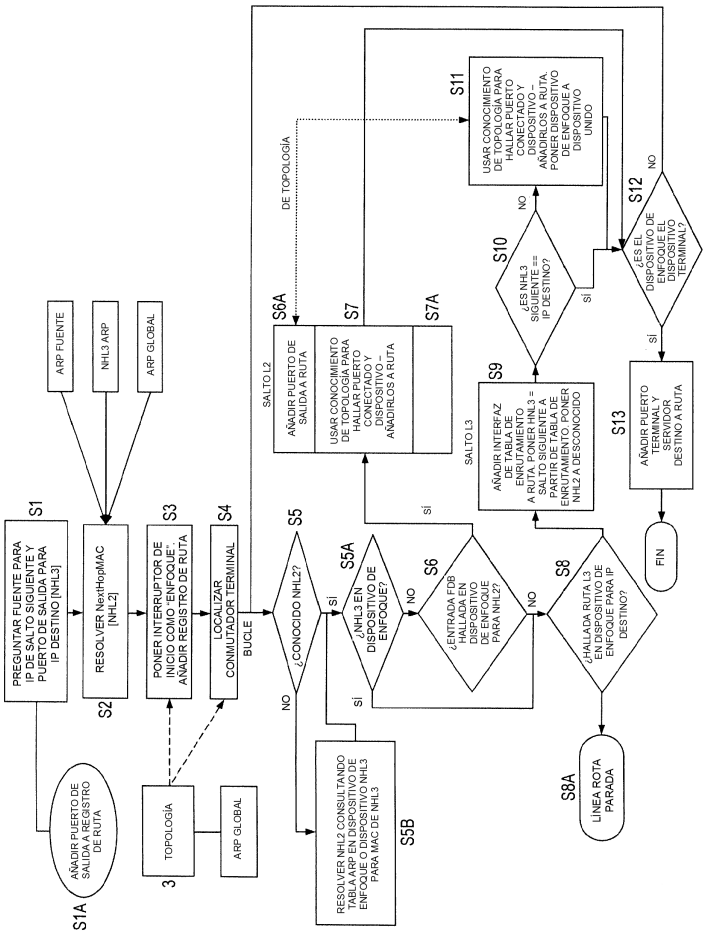
ARP FUENTE
NHL3 ARP
ARP GLOBAL
S1
PREGUNTAR FUENTE PARA IP DE SALTO SIGUIENTE Y PUERTO DE SALIDA PARA IP DESTINO [NHL3]
S2
RESOLVER NextHopMAC [NHL2]
S3
PONER INTERRUPTOR DE INICIO COMO 'ENFOQUE'. AÑADIR REGISTRO DE RUTA
S4
LOCALIZAR CONMUTADOR TERMINAL
BUCLE
S5
¿CONOCIDO NHL2?
SÍ
NO
S5A
¿NHL3 EN DISPOSITIVO DE ENFOQUE?
SÍ
NO
S6
¿ENTRADA FDB HALLADA EN DISPOSITIVO DE ENFOQUE PARA NHL2?
SÍ
NO
S8
¿HALLADA RUTA L3 EN DISPOSITIVO DE ENFOQUE PARA IP DESTINO?
SÍ
NO
S8A
LÍNEA ROTA PARADA
S1A
AÑADIR PUERTO DE SALIDA A REGISTRO DE RUTA
3
TOPOLOGÍA
ARP GLOBAL
S5B
RESOLVER NHL2 CONSULTANDO TABLA ARP EN DISPOSITIVO DE ENFOQUE Y DISPOSITIVO NHL3 PARA MAC DE NHL3
S6A
AÑADIR PUERTO DE SALIDA A RUTA
SALTO L2
S7
USAR CONOCIMIENTO DE TOPOLOGÍA PARA HALLAR PUERTO CONECTADO Y DISPOSITIVO – AÑADIRLOS A RUTA
S7A
SALTO L3
DE TOPOLOGÍA
S9
AÑADIR INTERFAZ DE TABLA DE ENRUTAMIENTO A RUTA. PONER NHL3 = SALTO SIGUIENTE A PARTIR DE TABLA DE ENRUTAMIENTO. PONER NHL2 A DESCONOCIDO
S10
¿ES NHL3 SIGUIENTE == IP DESTINO?
SÍ
NO
S11
USAR CONOCIMIENTO DE TOPOLOGÍA PARA HALLAR PUERTO CONECTADO Y DISPOSITIVO Y AÑADIRLOS A RUTA. PONER DISPOSITIVO DE ENFOQUE A DISPOSITIVO UNIDO
S12
¿ES EL DISPOSITIVO DE ENFOQUE EL DISPOSITIVO TERMINAL?
SÍ
NO
S13
AÑADIR PUERTO TERMINAL Y SERVIDOR DESTINO A RUTA
FIN

FIG. 3

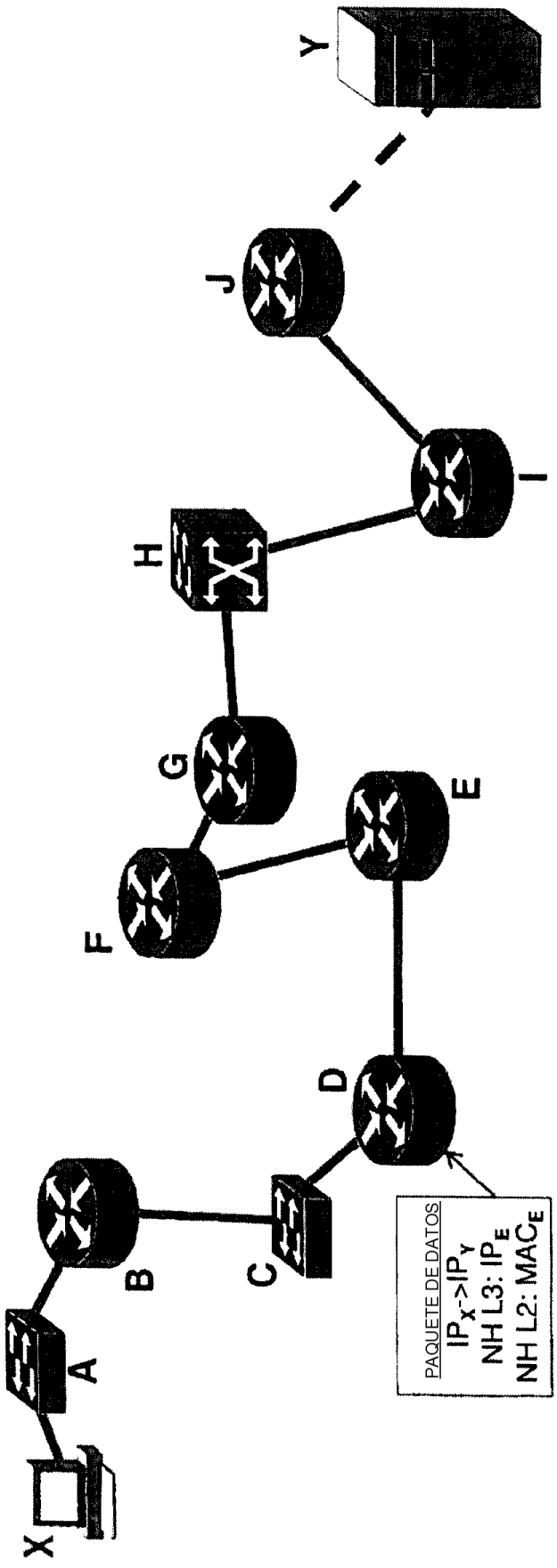

FIG. 4

ipRouteEntry TIPO-OBJETO
-- 1.3.6.1.2.1.4.21.1
-- iso(1) org(3) dod(6) internet(1) mgmt(2), mib-2(1) ip(4) ipRouteTable(21) ipRouteEntry(1)
SINTAXIS ipRouteEntry
ACCESO no accesible
ÍNDICE { ipRouteDest }
48
DESCRIPCIÓN
"Una ruta a un destino concreto"
::= { ipRouteTable 1 }
ipRouteEntry
::=
SECUENCIA {
ipRouteDest IpAddress,
50 ipRouteIfIndex ENTERO,
ipRouteMetric1 ENTERO,
ipRouteMetric2 ENTERO,
ipRouteMetric3 ENTERO,
ipRouteMetric4 ENTERO,
52 ipRouteNextHop IpAddress,
54 ipRouteType ENTERO,
ipRouteProto ENTERO,
ipRouteAge ENTERO,
56 ipRouteMask IpAddress,
ipRouteMetric5 ENTERO,
ipRouteInfo IDENTIFICADOR DE OBJETO
}

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

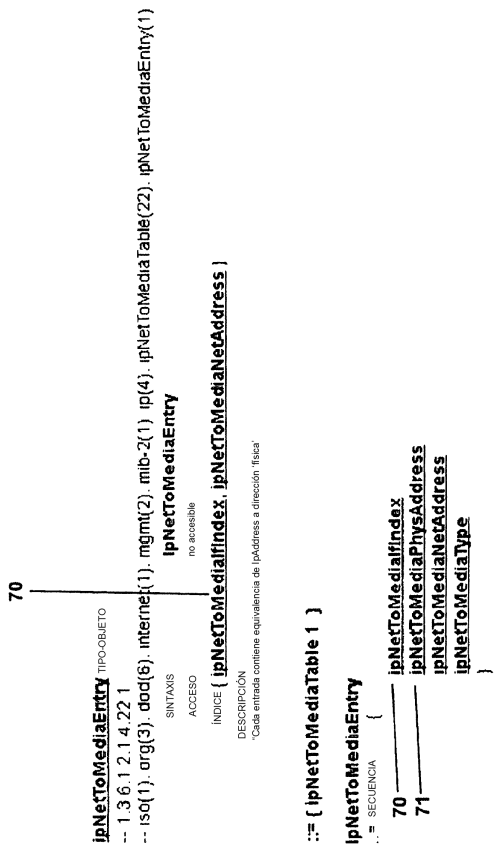
70
ipNetToMediaEntry TIPO-OBJETO
-- 1.3.6.1.2.1.4.22.1
-- iso(1). org(3). dod(6). internet(1). mgmt(2). mib-2(1). ip(4). ipNetToMediaTable(22). ipNetToMediaEntry(1)
SINTAXIS ipNetToMediaEntry
ACCESO no accesible
ÍNDICE { ipNetToMediaIfIndex, ipNetToMediaNetAddress }
DESCRIPCIÓN
"Cada entrada contiene equivalencia de IpAddress a dirección 'física'
::= { ipNetToMediaTable 1 }
ipNetToMediaEntry
::= SECUENCIA {
70 ipNetToMediaIfIndex ENTERO
71 ipNetToMediaPhysAddress PhysAddress,
ipNetToMediaNetAddress IpAddress,
ipNetToMediaType ENTERO
}

FIG. 7

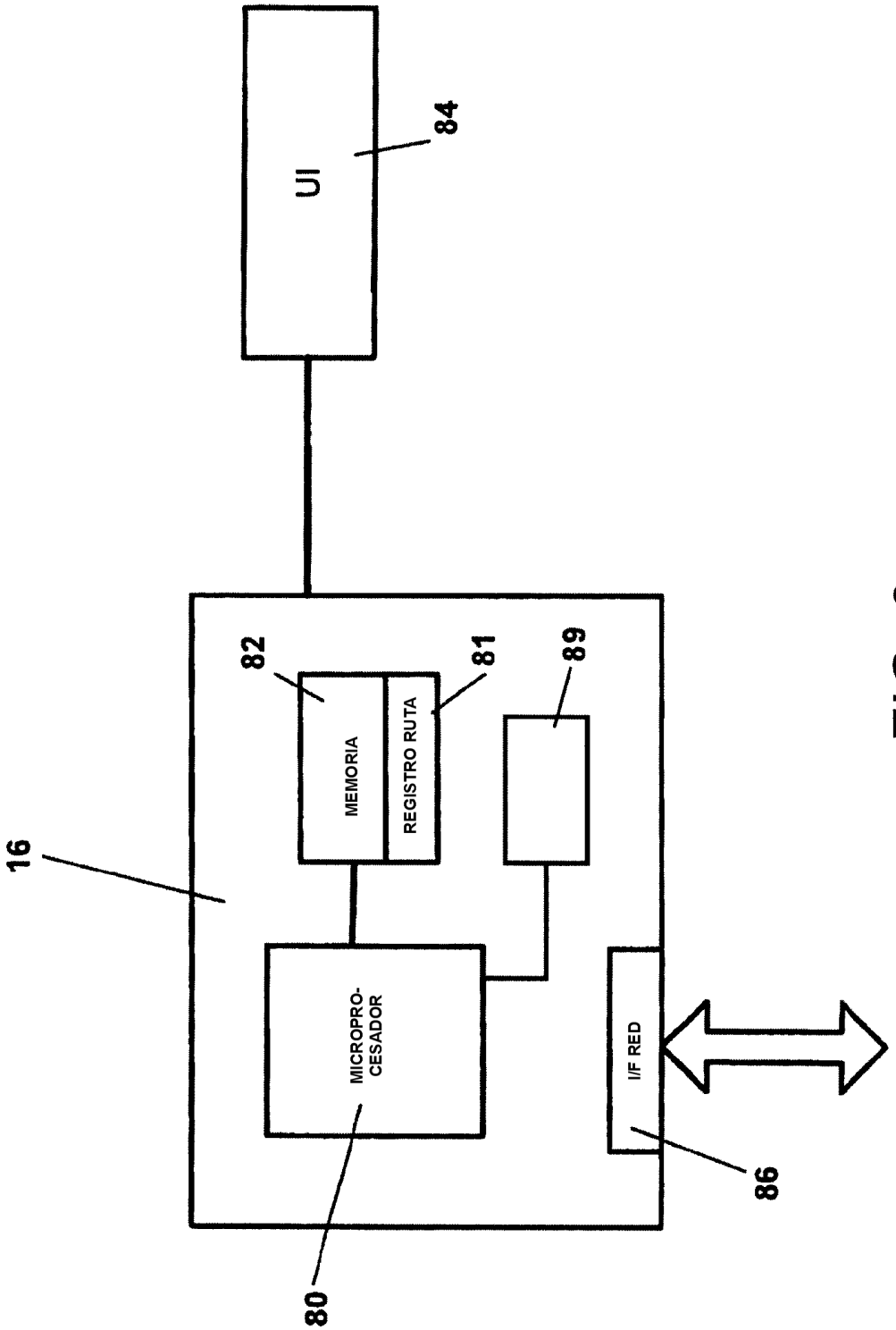
UI
84
82
81
89
MEMORIA
REGISTRO RUTA
16
MICROPRO-
CESADOR
I/F RED
80
86

FIG. 8

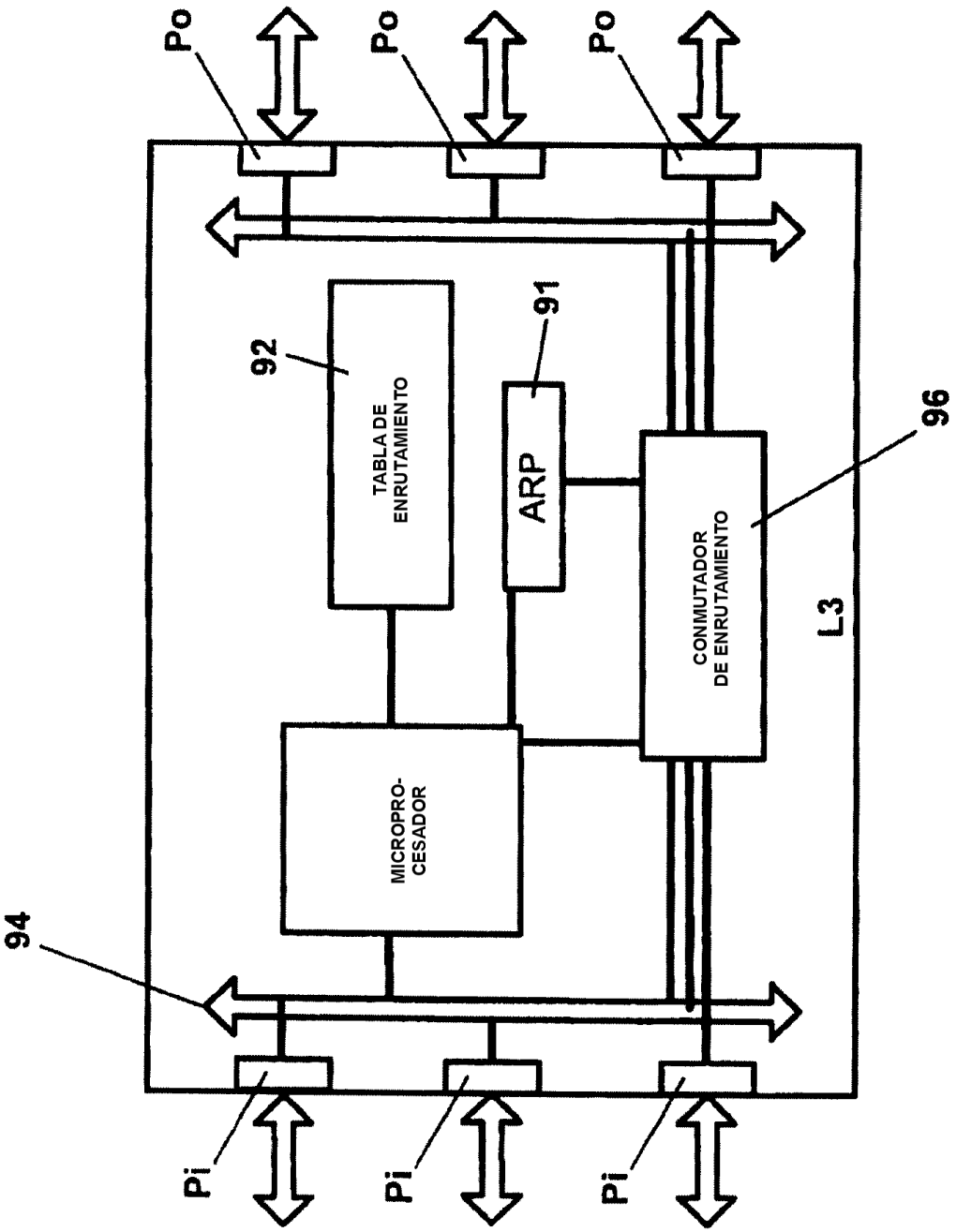
Po
Po
Po
92
TABLA DE ENRUTAMIENTO
91
ARP
96
CONMUTADOR DE ENRUTAMIENTO
L3
MICROPRO-CESADOR
94
Pi
Pi
Pi

FIG. 9

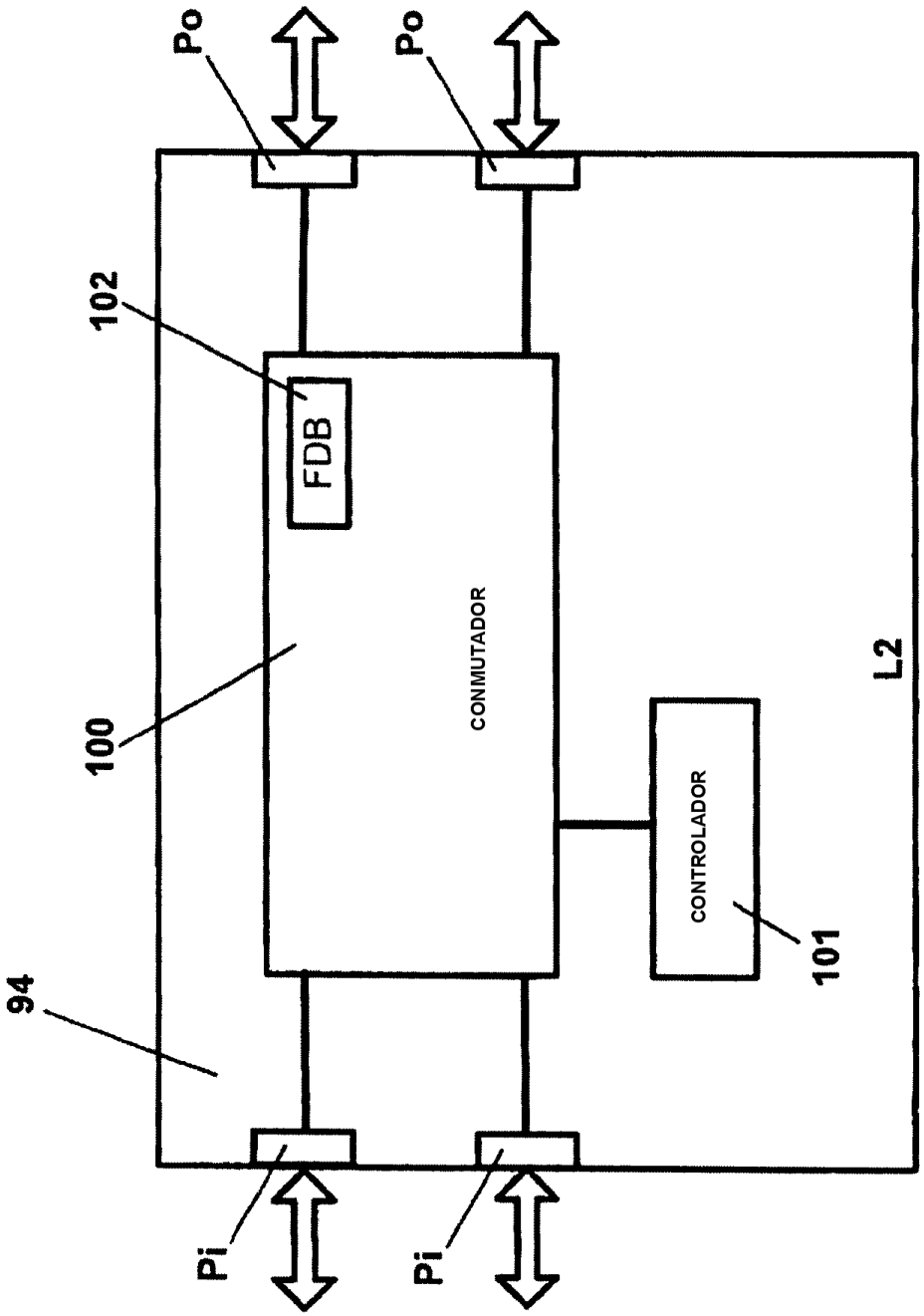

FIG. 10

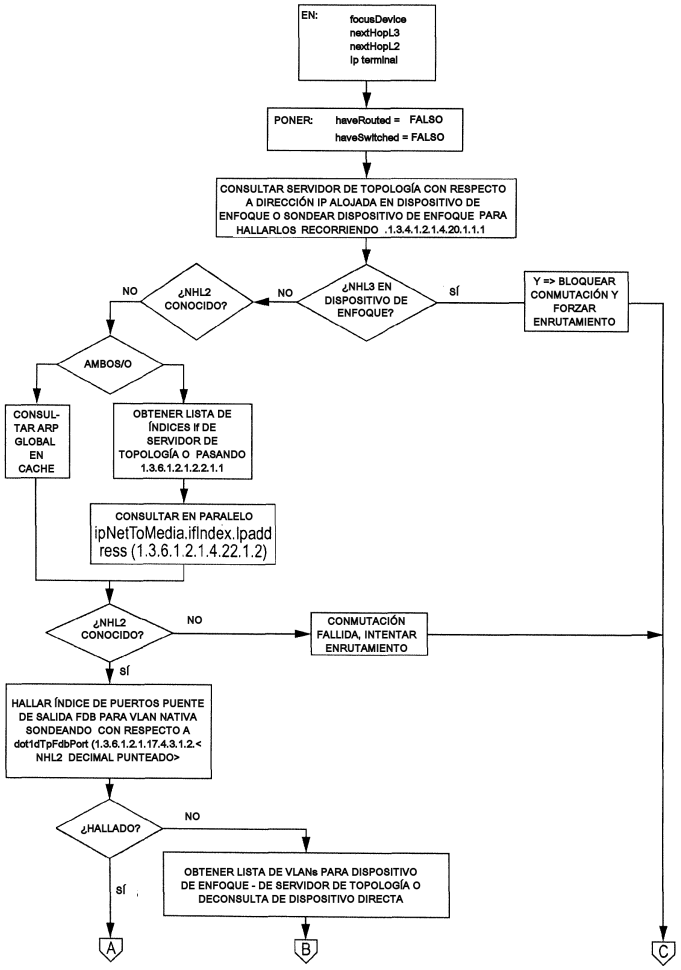
EN:
focusDevice
nextHopL3
nextHopL2
Ip terminal
PONER:
haveRouted = FALSO
haveSwitched = FALSO
CONSULTAR SERVIDOR DE TOPOLOGÍA CON RESPECTO A DIRECCIÓN IP ALOJADA EN DISPOSITIVO DE ENFOQUE O SONDEAR DISPOSITIVO DE ENFOQUE PARA HALLARLOS RECORRIENDO .1.3.4.1.2.1.4.20.1.1.1
¿NHL3 EN DISPOSITIVO DE ENFOQUE?
NO
SÍ
Y => BLOQUEAR CONMUTACIÓN Y FORZAR ENRUTAMIENTO
¿NHL2 CONOCIDO?
NO
AMBOS/O
CONSUL-TAR ARP GLOBAL EN CACHE
OBTENER LISTA DE ÍNDICES If DE SERVIDOR DE TOPOLOGÍA O PASANDO 1.3.6.1.2.1.2.2.1.1
CONSULTAR EN PARALELO
ipNetToMedia.ifIndex.Ipaddress (1.3.6.1.2.1.4.22.1.2)
¿NHL2 CONOCIDO?
NO
CONMUTACIÓN FALLIDA, INTENTAR ENRUTAMIENTO
SÍ
HALLAR ÍNDICE DE PUERTOS PUENTE DE SALIDA FDB PARA VLAN NATIVA SONDEANDO CON RESPECTO A dot1dTpFdbPort (1.3.6.1.2.1.17.4.3.1.2.< NHL2 DECIMAL PUNTEADO>
¿HALLADO?
NO
SÍ
OBTENER LISTA DE VLANs PARA DISPOSITIVO DE ENFOQUE - DE SERVIDOR DE TOPOLOGÍA O DECONSULTA DE DISPOSITIVO DIRECTA
A
B
C

FIG. 11a

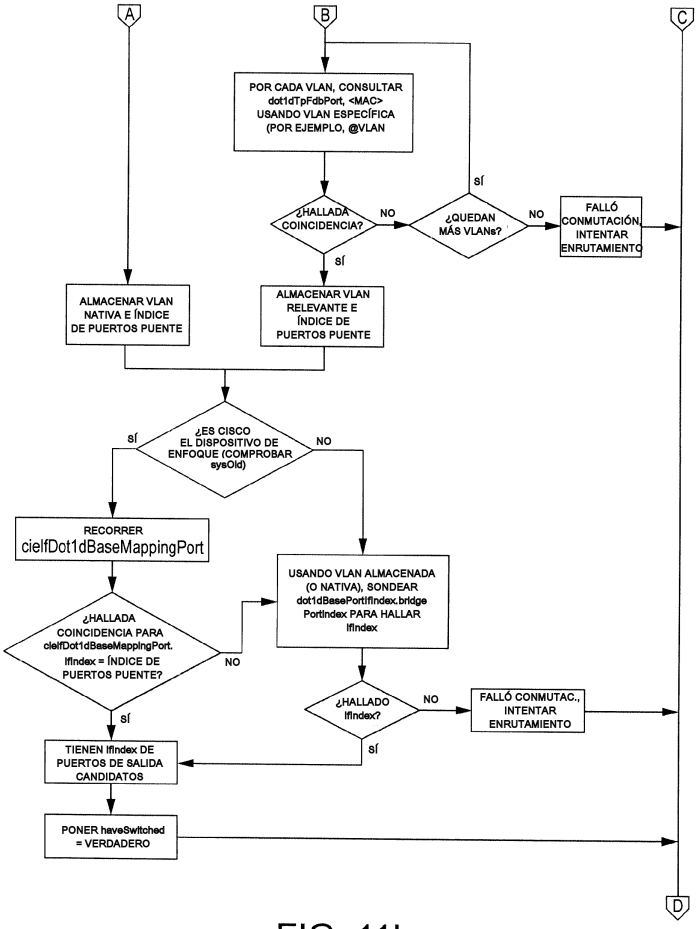
A
B
C
POR CADA VLAN, CONSULTAR dot1dTpFdbPort, <MAC> USANDO VLAN ESPECÍFICA (POR EJEMPLO, @VLAN
¿HALLADA COINCIDENCIA?
NO
¿QUEDAN MÁS VLANs?
NO
FALLÓ CONMUTACIÓN, INTENTAR ENRUTAMIENTO
SÍ
SÍ
ALMACENAR VLAN NATIVA E ÍNDICE DE PUERTOS PUENTE
ALMACENAR VLAN RELEVANTE E ÍNDICE DE PUERTOS PUENTE
¿ES CISCO EL DISPOSITIVO DE ENFOQUE (COMPROBAR sysOId)
SÍ
NO
RECORRER cielfDot1dBaseMappingPort
USANDO VLAN ALMACENADA (O NATIVA), SONDEAR dot1dBasePortIfIndex.bridge PortIndex PARA HALLAR IfIndex
¿HALLADA COINCIDENCIA PARA cielfDot1dBaseMappingPort. IfIndex = ÍNDICE DE PUERTOS PUENTE?
NO
SÍ
¿HALLADO IfIndex?
NO
FALLÓ CONMUTAC., INTENTAR ENRUTAMIENTO
SÍ
TIENEN IfIndex DE PUERTOS DE SALIDA CANDIDATOS
PONER haveSwitched = VERDADERO
D

FIG. 11b

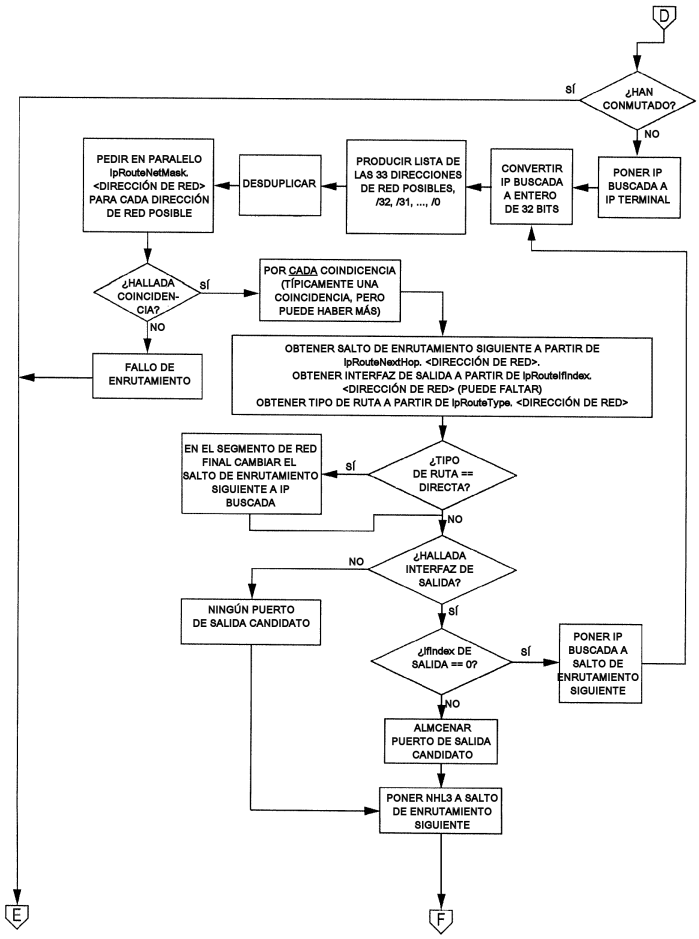
D
¿HAN CONMUTADO?
sí
NO
PONER IP BUSCADA A IP TERMINAL
CONVERTIR IP BUSCADA A ENTERO DE 32 BITS
PRODUCIR LISTA DE LAS 33 DIRECCIONES DE RED POSIBLES, /32, /31, ..., /0
DESDUPLICAR
PEDIR EN PARALELO IpRouteNetMask. <DIRECCIÓN DE RED> PARA CADA DIRECCIÓN DE RED POSIBLE
¿HALLADA COINCIDENCIA?
sí
NO
POR CADA COINDICENCIA (TÍPICAMENTE UNA COINCIDENCIA, PERO PUEDE HABER MÁS)
FALLO DE ENRUTAMIENTO
OBTENER SALTO DE ENRUTAMIENTO SIGUIENTE A PARTIR DE IpRouteNextHop. <DIRECCIÓN DE RED>.
OBTENER INTERFAZ DE SALIDA A PARTIR DE IpRouteIfIndex. <DIRECCIÓN DE RED> (PUEDE FALTAR)
OBTENER TIPO DE RUTA A PARTIR DE IpRouteType. <DIRECCIÓN DE RED>
¿TIPO DE RUTA == DIRECTA?
sí
EN EL SEGMENTO DE RED FINAL CAMBIAR EL SALTO DE ENRUTAMIENTO SIGUIENTE A IP BUSCADA
NO
¿HALLADA INTERFAZ DE SALIDA?
NO
sí
NINGÚN PUERTO DE SALIDA CANDIDATO
¿IfIndex DE SALIDA == 0?
sí
PONER IP BUSCADA A SALTO DE ENRUTAMIENTO SIGUIENTE
NO
ALMCENAR PUERTO DE SALIDA CANDIDATO
PONER NHL3 A SALTO DE ENRUTAMIENTO SIGUIENTE
E
F

FIG. 11c

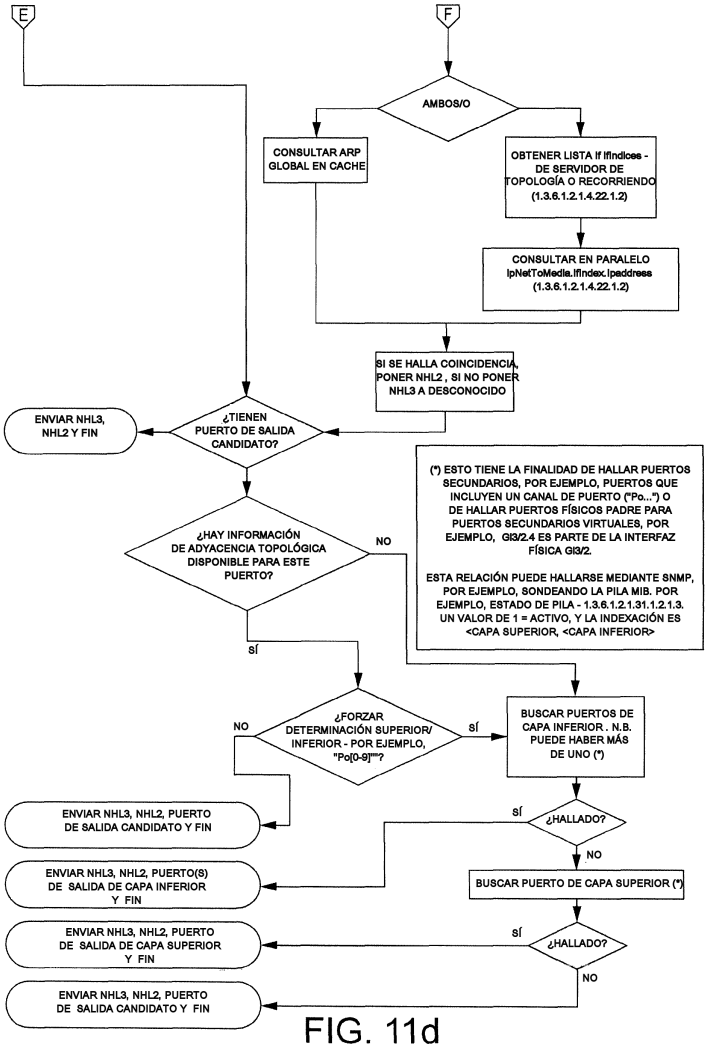
E
F
AMBOS/O
CONSULTAR ARP GLOBAL EN CACHE
OBTENER LISTA If IfIndices - DE SERVIDOR DE TOPOLOGÍA O RECORRIENDO (1.3.6.1.2.1.4.22.1.2)
CONSULTAR EN PARALELO IpNetToMedia.IfIndex.IpAddress (1.3.6.1.2.1.4.22.1.2)
SI SE HALLA COINCIDENCIA, PONER NHL2 , SI NO PONER NHL3 A DESCONOCIDO
ENVIAR NHL3, NHL2 Y FIN
¿TIENEN PUERTO DE SALIDA CANDIDATO?
¿HAY INFORMACIÓN DE ADYACENCIA TOPOLÓGICA DISPONIBLE PARA ESTE PUERTO?
NO
SÍ
(*) ESTO TIENE LA FINALIDAD DE HALLAR PUERTOS SECUNDARIOS, POR EJEMPLO, PUERTOS QUE INCLUYEN UN CANAL DE PUERTO ("Po...") O DE HALLAR PUERTOS FÍSICOS PADRE PARA PUERTOS SECUNDARIOS VIRTUALES, POR EJEMPLO, GI3/2.4 ES PARTE DE LA INTERFAZ FÍSICA GI3/2.
ESTA RELACIÓN PUEDE HALLARSE MEDIANTE SNMP, POR EJEMPLO, SONDEANDO LA PILA MIB. POR EJEMPLO, ESTADO DE PILA - 1.3.6.1.2.1.31.1.2.1.3. UN VALOR DE 1 = ACTIVO, Y LA INDEXACIÓN ES <CAPA SUPERIOR, <CAPA INFERIOR>
¿FORZAR DETERMINACIÓN SUPERIOR/ INFERIOR - POR EJEMPLO, "Po[0-9]"?
NO
SÍ
BUSCAR PUERTOS DE CAPA INFERIOR . N.B. PUEDE HABER MÁS DE UNO (*)
ENVIAR NHL3, NHL2, PUERTO DE SALIDA CANDIDATO Y FIN
¿HALLADO?
SÍ
NO
ENVIAR NHL3, NHL2, PUERTO(S) DE SALIDA DE CAPA INFERIOR Y FIN
BUSCAR PUERTO DE CAPA SUPERIOR (*)
ENVIAR NHL3, NHL2, PUERTO DE SALIDA DE CAPA SUPERIOR Y FIN
¿HALLADO?
SÍ
NO
ENVIAR NHL3, NHL2, PUERTO DE SALIDA CANDIDATO Y FIN

FIG. 11d

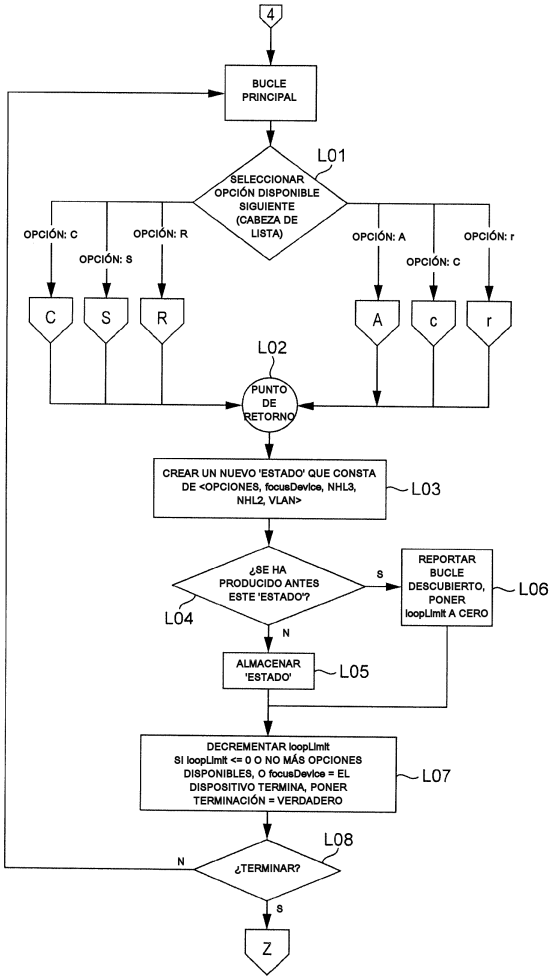
4
BUCLE PRINCIPAL
L01
SELECCIONAR OPCIÓN DISPONIBLE SIGUIENTE (CABEZA DE LISTA)
OPCIÓN: C
OPCIÓN: S
OPCIÓN: R
OPCIÓN: A
OPCIÓN: C
OPCIÓN: r
C
S
R
A
c
r
L02
PUNTO DE RETORNO
CREAR UN NUEVO 'ESTADO' QUE CONSTA DE <OPCIONES, focusDevice, NHL3, NHL2, VLAN>
L03
¿SE HA PRODUCIDO ANTES ESTE 'ESTADO'?
L04
S
N
REPORTAR BUCLE DESCUBIERTO, PONER loopLimit A CERO
L06
ALMACENAR 'ESTADO'
L05
DECREMENTAR loopLimit SI loopLimit <= 0 O NO MÁS OPCIONES DISPONIBLES, O focusDevice = EL DISPOSITIVO TERMINA, PONER TERMINACIÓN = VERDADERO
L07
L08
¿TERMINAR?
N
S
Z

FIG. 12

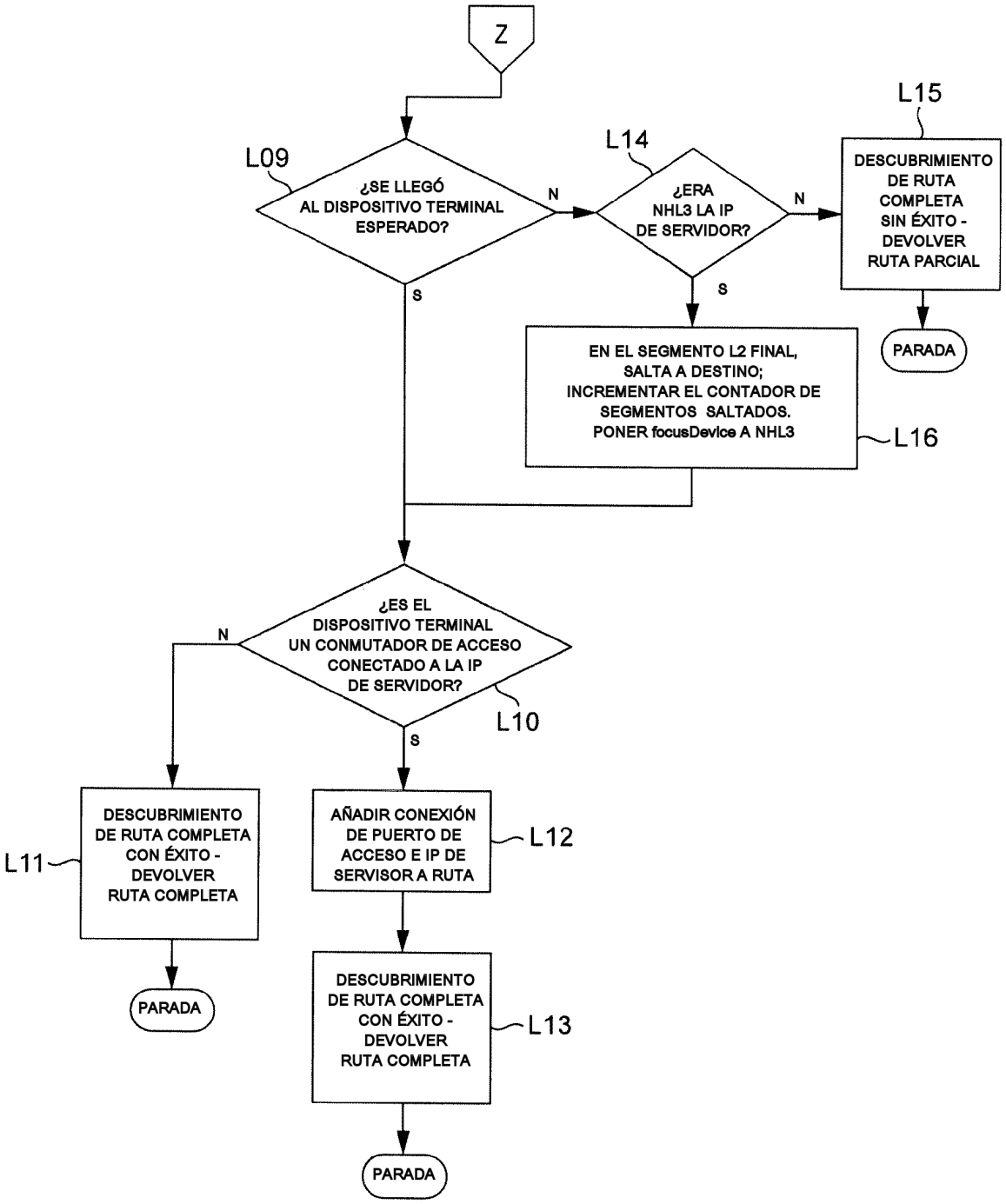
Z
L09
¿SE LLEGÓ AL DISPOSITIVO TERMINAL ESPERADO?
N
S
L14
¿ERA NHL3 LA IP DE SERVIDOR?
N
S
L15
DESCUBRIMIENTO DE RUTA COMPLETA SIN ÉXITO - DEVOLVER RUTA PARCIAL
PARADA
EN EL SEGMENTO L2 FINAL, SALTA A DESTINO; INCREMENTAR EL CONTADOR DE SEGMENTOS SALTADOS. PONER focusDevice A NHL3
L16
¿ES EL DISPOSITIVO TERMINAL UN CONMUTADOR DE ACCESO CONECTADO A LA IP DE SERVIDOR?
L10
N
S
DESCUBRIMIENTO DE RUTA COMPLETA CON ÉXITO - DEVOLVER RUTA COMPLETA
L11
PARADA
AÑADIR CONEXIÓN DE PUERTO DE ACCESO E IP DE SERVISOR A RUTA
L12
DESCUBRIMIENTO DE RUTA COMPLETA CON ÉXITO - DEVOLVER RUTA COMPLETA
L13
PARADA

FIG. 13

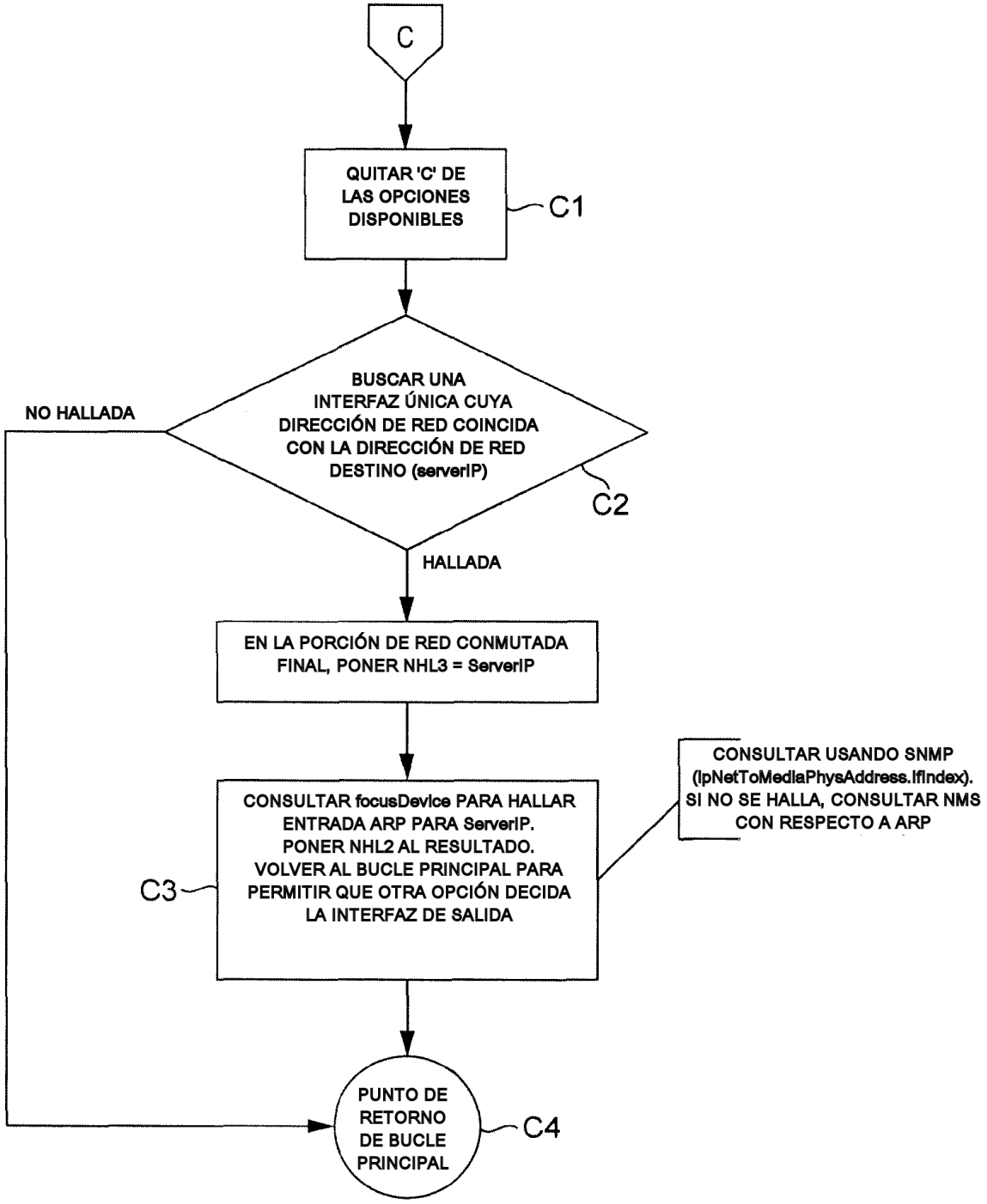
C
QUITAR 'C' DE LAS OPCIONES DISPONIBLES
C1
BUSCAR UNA INTERFAZ ÚNICA CUYA DIRECCIÓN DE RED COINCIDA CON LA DIRECCIÓN DE RED DESTINO (serverIP)
C2
NO HALLADA
HALLADA
EN LA PORCIÓN DE RED CONMUTADA FINAL, PONER NHL3 = ServerIP
CONSULTAR focusDevice PARA HALLAR ENTRADA ARP PARA ServerIP. PONER NHL2 AL RESULTADO. VOLVER AL BUCLE PRINCIPAL PARA PERMITIR QUE OTRA OPCIÓN DECIDA LA INTERFAZ DE SALIDA
C3
CONSULTAR USANDO SNMP (IpNetToMediaPhysAddress.IfIndex). SI NO SE HALLA, CONSULTAR NMS CON RESPECTO A ARP
PUNTO DE RETORNO DE BUCLE PRINCIPAL
C4

FIG. 14

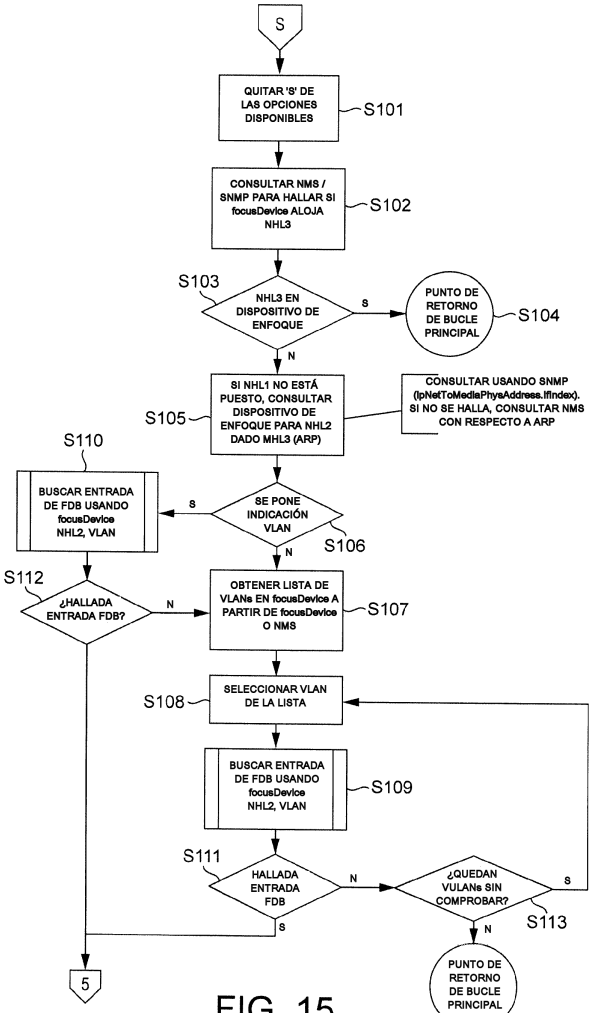
S
QUITAR '6' DE LAS OPCIONES DISPONIBLES
S101
CONSULTAR NMS / SNMP PARA HALLAR SI focusDevice ALOJA NHL3
S102
S103
NHL3 EN DISPOSITIVO DE ENFOQUE
S
PUNTO DE RETORNO DE BUCLE PRINCIPAL
S104
N
SI NHL1 NO ESTÁ PUESTO, CONSULTAR DISPOSITIVO DE ENFOQUE PARA NHL2 DADO MHL3 (ARP)
S105
CONSULTAR USANDO SNMP (IpNetToMediaPhysAddress.IfIndex). SI NO SE HALLA, CONSULTAR NMS CON RESPECTO A ARP
S110
BUSCAR ENTRADA DE FDB USANDO focusDevice NHL2, VLAN
S
SE PONE INDICACIÓN VLAN
S106
N
S112
¿HALLADA ENTRADA FDB?
N
OBTENER LISTA DE VLANs EN focusDevice A PARTIR DE focusDevice O NMS
S107
SELECCIONAR VLAN DE LA LISTA
S108
BUSCAR ENTRADA DE FDB USANDO focusDevice NHL2, VLAN
S109
S111
HALLADA ENTRADA FDB
N
¿QUEDAN VULANs SIN COMPROBAR?
S
S113
N
PUNTO DE RETORNO DE BUCLE PRINCIPAL
S
5

FIG. 15

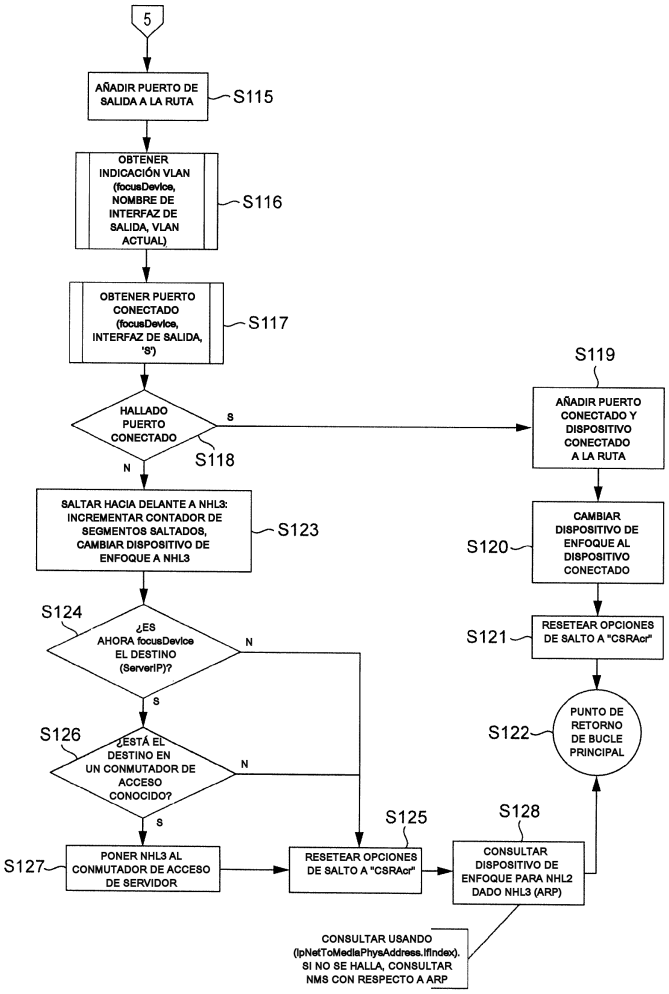
5
AÑADIR PUERTO DE SALIDA A LA RUTA
S115
OBTENER INDICACIÓN VLAN (focusDevice, NOMBRE DE INTERFAZ DE SALIDA, VLAN ACTUAL)
S116
OBTENER PUERTO CONECTADO (focusDevice, INTERFAZ DE SALIDA, 'S')
S117
HALLADO PUERTO CONECTADO
S
N
S118
S119
AÑADIR PUERTO CONECTADO Y DISPOSITIVO CONECTADO A LA RUTA
CAMBIAR DISPOSITIVO DE ENFOQUE AL DISPOSITIVO CONECTADO
S120
RESETEAR OPCIONES DE SALTO A "CSRAcr"
S121
PUNTO DE RETORNO DE BUCLE PRINCIPAL
S122
SALTAR HACIA DELANTE A NHL3: INCREMENTAR CONTADOR DE SEGMENTOS SALTADOS, CAMBIAR DISPOSITIVO DE ENFOQUE A NHL3
S123
S124
¿ES AHORA focusDevice EL DESTINO (ServerIP)?
N
S
S126
¿ESTÁ EL DESTINO EN UN CONMUTADOR DE ACCESO CONOCIDO?
N
S
S125
S128
S127
PONER NHL3 AL CONMUTADOR DE ACCESO DE SERVIDOR
RESETEAR OPCIONES DE SALTO A "CSRAcr"
CONSULTAR DISPOSITIVO DE ENFOQUE PARA NHL2 DADO NHL3 (ARP)
CONSULTAR USANDO (IpNetToMediaPhysAddress.IfIndex). SI NO SE HALLA, CONSULTAR NMS CON RESPECTO A ARP

FIG. 16

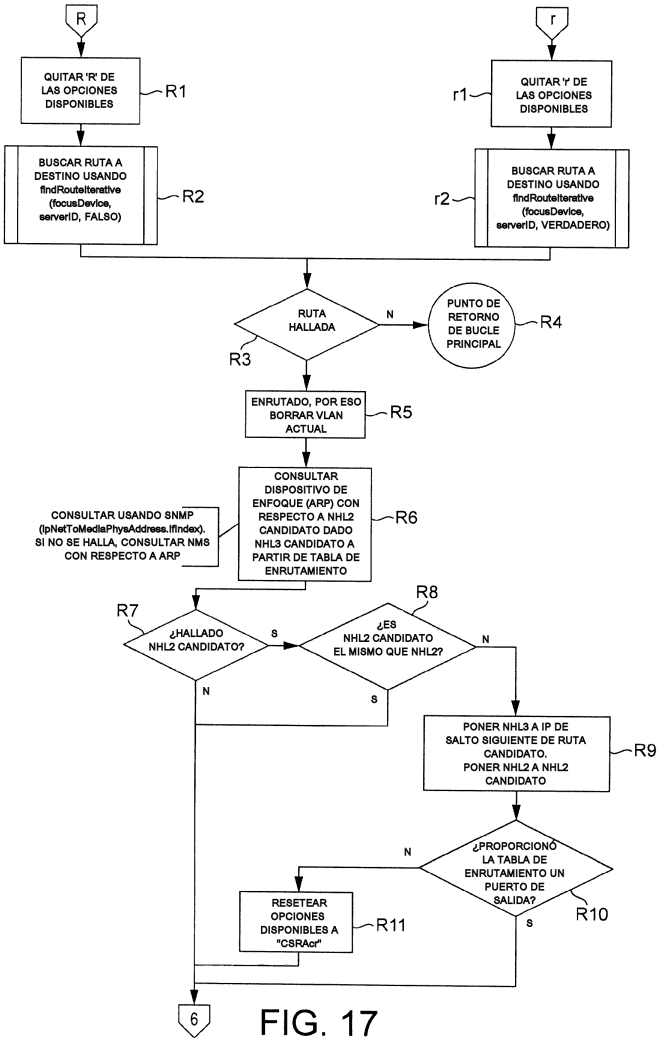
R
QUITAR 'R' DE LAS OPCIONES DISPONIBLES
R1
BUSCAR RUTA A DESTINO USANDO findRouteIterative (focusDevice, serverID, FALSO)
R2
r
r1
QUITAR 'r' DE LAS OPCIONES DISPONIBLES
r2
BUSCAR RUTA A DESTINO USANDO findRouteIterative (focusDevice, serverID, VERDADERO)
RUTA HALLADA
R3
N
PUNTO DE RETORNO DE BUCLE PRINCIPAL
R4
ENRUTADO, POR ESO BORRAR VLAN ACTUAL
R5
CONSULTAR DISPOSITIVO DE ENFOQUE (ARP) CON RESPECTO A NHL2 CANDIDATO DADO NHL3 CANDIDATO A PARTIR DE TABLA DE ENRUTAMIENTO
R6
CONSULTAR USANDO SNMP (IpNetToMediaPhysAddress.IfIndex). SI NO SE HALLA, CONSULTAR NMS CON RESPECTO A ARP
R7
¿HALLADO NHL2 CANDIDATO?
S
N
R8
¿ES NHL2 CANDIDATO EL MISMO QUE NHL2?
N
S
PONER NHL3 A IP DE SALTO SIGUIENTE DE RUTA CANDIDATO. PONER NHL2 A NHL2 CANDIDATO
R9
¿PROPORCIONÓ LA TABLA DE ENRUTAMIENTO UN PUERTO DE SALIDA?
N
S
R10
RESETEAR OPCIONES DISPONIBLES A "CSRAcr"
R11
6

FIG. 17

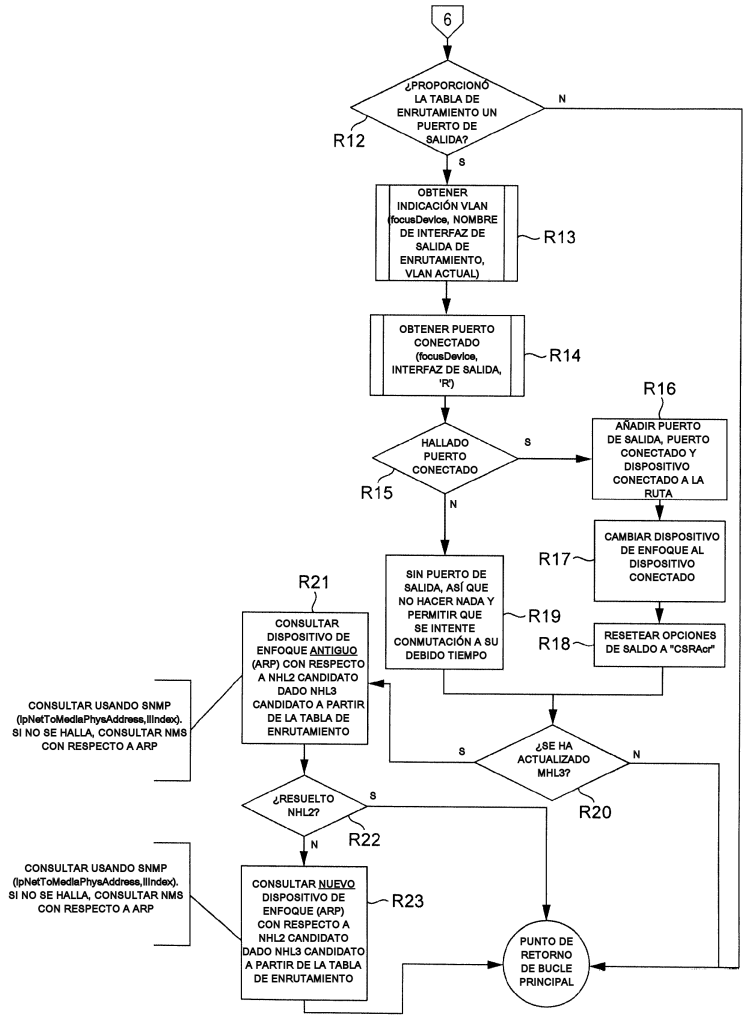
6
¿PROPORCIONÓ LA TABLA DE ENRUTAMIENTO UN PUERTO DE SALIDA?
R12
N
S
OBTENER INDICACIÓN VLAN (focusDevice, NOMBRE DE INTERFAZ DE SALIDA DE ENRUTAMIENTO, VLAN ACTUAL)
R13
OBTENER PUERTO CONECTADO (focusDevice, INTERFAZ DE SALIDA, 'R')
R14
R16
HALLADO PUERTO CONECTADO
R15
S
N
AÑADIR PUERTO DE SALIDA, PUERTO CONECTADO Y DISPOSITIVO CONECTADO A LA RUTA
CAMBIAR DISPOSITIVO DE ENFOQUE AL DISPOSITIVO CONECTADO
R17
SIN PUERTO DE SALIDA, ASÍ QUE NO HACER NADA Y PERMITIR QUE SE INTENTE CONMUTACIÓN A SU DEBIDO TIEMPO
R19
RESETEAR OPCIONES DE SALDO A "CSRAcr"
R18
R21
CONSULTAR DISPOSITIVO DE ENFOQUE ANTIGUO (ARP) CON RESPECTO A NHL2 CANDIDATO DADO NHL3 CANDIDATO A PARTIR DE LA TABLA DE ENRUTAMIENTO
CONSULTAR USANDO SNMP (IpNetToMediaPhysAddress,IIIndex). SI NO SE HALLA, CONSULTAR NMS CON RESPECTO A ARP
¿SE HA ACTUALIZADO MHL3?
S
N
R20
¿RESUELTO NHL2?
R22
S
N
CONSULTAR USANDO SNMP (IpNetToMediaPhysAddress,IIIndex). SI NO SE HALLA, CONSULTAR NMS CON RESPECTO A ARP
CONSULTAR NUEVO DISPOSITIVO DE ENFOQUE (ARP) CON RESPECTO A NHL2 CANDIDATO DADO NHL3 CANDIDATO A PARTIR DE LA TABLA DE ENRUTAMIENTO
R23
PUNTO DE RETORNO DE BUCLE PRINCIPAL

FIG. 18

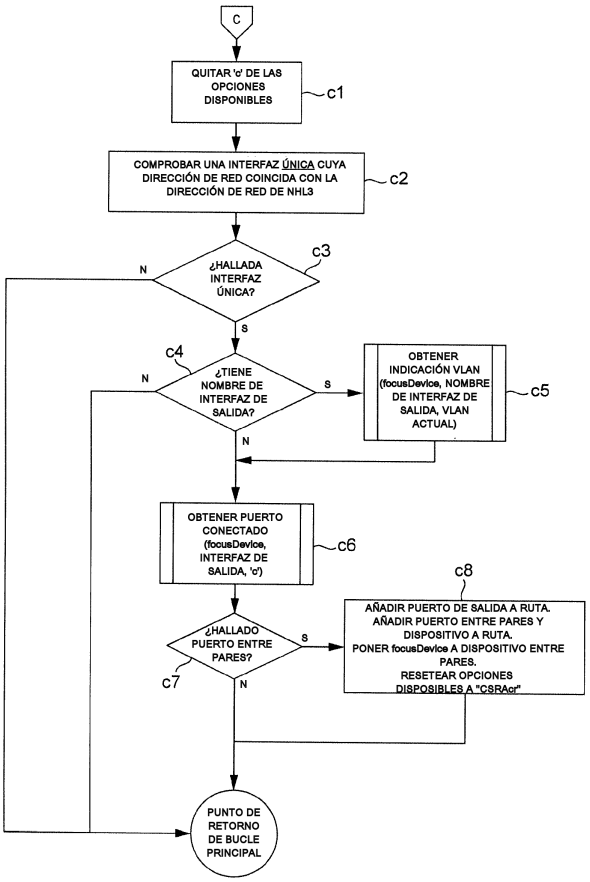
C
QUITAR 'c' DE LAS OPCIONES DISPONIBLES
c1
COMPROBAR UNA INTERFAZ ÚNICA CUYA DIRECCIÓN DE RED COINCIDA CON LA DIRECCIÓN DE RED DE NHL3
c2
¿HALLADA INTERFAZ ÚNICA?
c3
N
S
¿TIENE NOMBRE DE INTERFAZ DE SALIDA?
c4
N
S
OBTENER INDICACIÓN VLAN (focusDevice, NOMBRE DE INTERFAZ DE SALIDA, VLAN ACTUAL)
c5
N
OBTENER PUERTO CONECTADO (focusDevice, INTERFAZ DE SALIDA, 'c')
c6
¿HALLADO PUERTO ENTRE PARES?
c7
S
N
c8
AÑADIR PUERTO DE SALIDA A RUTA.
AÑADIR PUERTO ENTRE PARES Y DISPOSITIVO A RUTA.
PONER focusDevice A DISPOSITIVO ENTRE PARES.
RESETEAR OPCIONES DISPOSIBLES A "CSRAcr"
PUNTO DE RETORNO DE BUCLE PRINCIPAL

FIG. 19

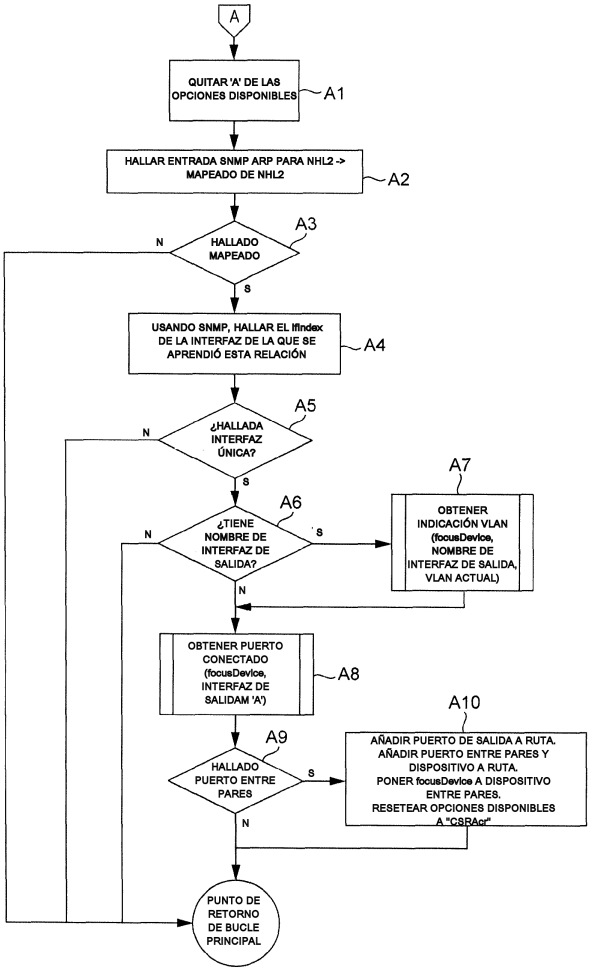
A
QUITAR 'A' DE LAS OPCIONES DISPONIBLES
A1
HALLAR ENTRADA SNMP ARP PARA NHL2 -> MAPEADO DE NHL2
A2
A3
HALLADO MAPEADO
N
S
USANDO SNMP, HALLAR EL IfIndex DE LA INTERFAZ DE LA QUE SE APRENDIÓ ESTA RELACIÓN
A4
A5
¿HALLADA INTERFAZ ÚNICA?
N
S
A7
A6
¿TIENE NOMBRE DE INTERFAZ DE SALIDA?
N
S
OBTENER INDICACIÓN VLAN (focusDevice, NOMBRE DE INTERFAZ DE SALIDA, VLAN ACTUAL)
N
OBTENER PUERTO CONECTADO (focusDevice, INTERFAZ DE SALIDAM 'A')
A8
A10
A9
HALLADO PUERTO ENTRE PARES
S
AÑADIR PUERTO DE SALIDA A RUTA.
AÑADIR PUERTO ENTRE PARES Y DISPOSITIVO A RUTA.
PONER focusDevice A DISPOSITIVO ENTRE PARES.
RESETEAR OPCIONES DISPONIBLES A "CSRAct"
N
PUNTO DE RETORNO DE BUCLE PRINCIPAL

FIG. 20

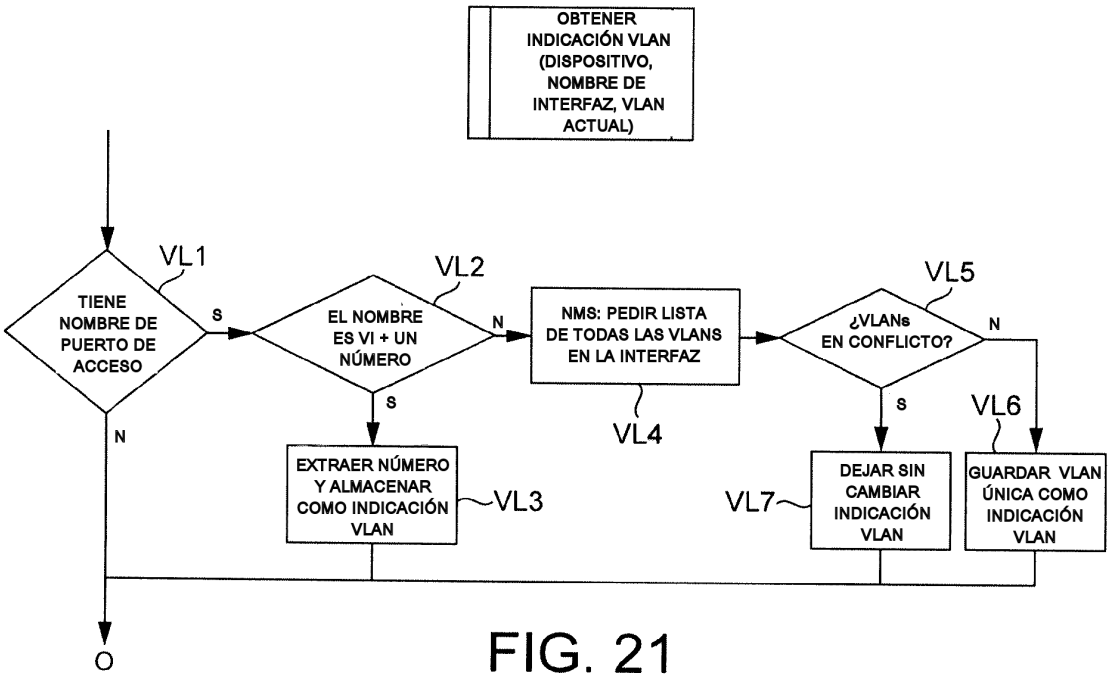
OBTENER INDICACIÓN VLAN (DISPOSITIVO, NOMBRE DE INTERFAZ, VLAN ACTUAL)
VL1
TIENE NOMBRE DE PUERTO DE ACCESO
S
N
VL2
EL NOMBRE ES VI + UN NÚMERO
N
S
NMS: PEDIR LISTA DE TODAS LAS VLANS EN LA INTERFAZ
VL4
VL5
¿VLANs EN CONFLICTO?
N
S
VL6
EXTRAER NÚMERO Y ALMACENAR COMO INDICACIÓN VLAN
VL3
VL7
DEJAR SIN CAMBIAR INDICACIÓN VLAN
GUARDAR VLAN ÚNICA COMO INDICACIÓN VLAN
O

FIG. 21

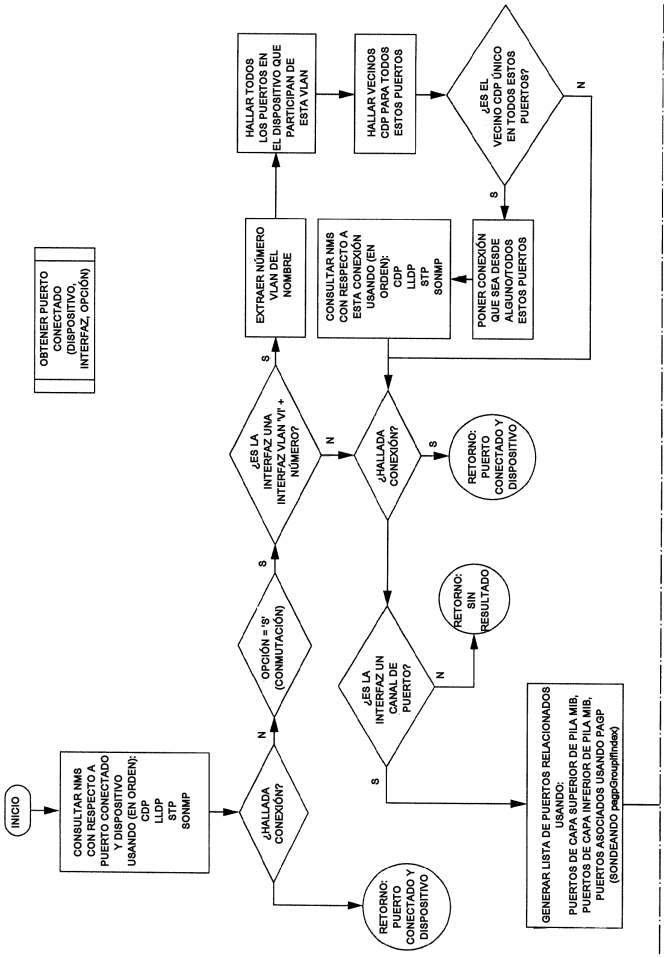
INICIO
CONSULTAR NMS CON RESPECTO A PUERTO CONECTADO Y DISPOSITIVO USANDO (EN ORDEN): CDP LLDP STP SONMP
¿HALLADA CONEXIÓN?
N
S
RETORNO: PUERTO CONECTADO Y DISPOSITIVO
OPCIÓN = 'S' (CONMUTACIÓN)
N
S
¿ES LA INTERFAZ UNA INTERFAZ VLAN 'VI' + NÚMERO?
N
S
EXTRAER NÚMERO VLAN DEL NOMBRE
HALLAR TODOS LOS PUERTOS EN EL DISPOSITIVO QUE PARTICIPAN DE ESTA VLAN
HALLAR VECINOS CDP PARA TODOS ESTOS PUERTOS
¿ES EL VECINO CDP ÚNICO EN TODOS ESTOS PUERTOS?
S
N
PONER CONEXIÓN QUE SEA DESDE ALGUNO/TODOS ESTOS PUERTOS
CONSULTAR NMS CON RESPECTO A ESTA CONEXIÓN USANDO (EN ORDEN): CDP LLDP STP SONMP
¿HALLADA CONEXIÓN?
N
S
RETORNO: PUERTO CONECTADO Y DISPOSITIVO
¿ES LA INTERFAZ UN CANAL DE PUERTO?
N
S
RETORNO: SIN RESULTADO
GENERAR LISTA DE PUERTOS RELACIONADOS USANDO:
PUERTOS DE CAPA SUPERIOR DE PILA MIB,
PUERTOS DE CAPA INFERIOR DE PILA MIB,
PUERTOS ASOCIADOS USANDO PAGP
(SONDEANDO pagpGroupIfIndex)
OBTENER PUERTO CONECTADO (DISPOSITIVO, INTERFAZ, OPCIÓN)

FIG. 22

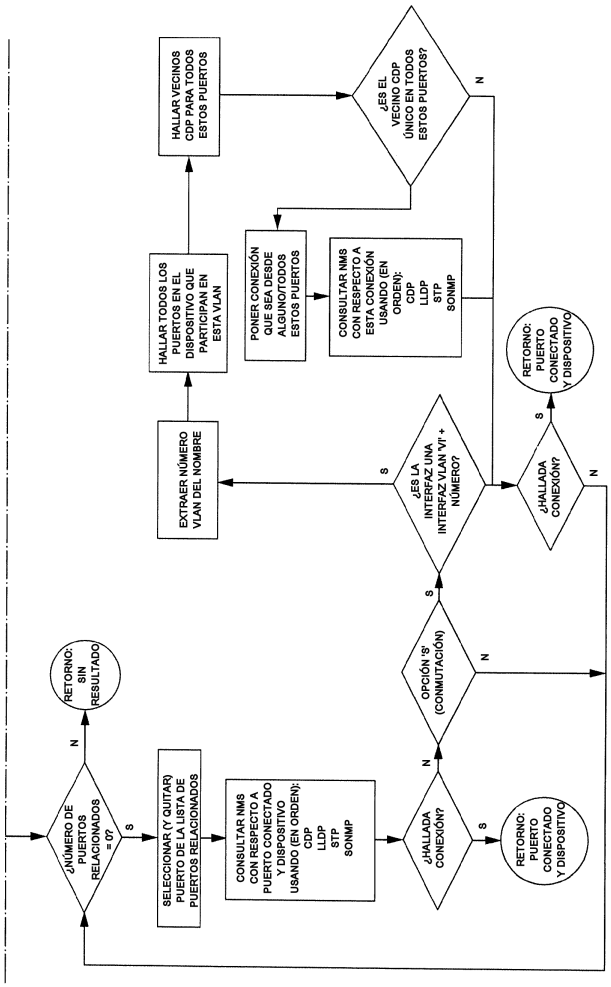

FIG. 22 CONTINUACIÓN

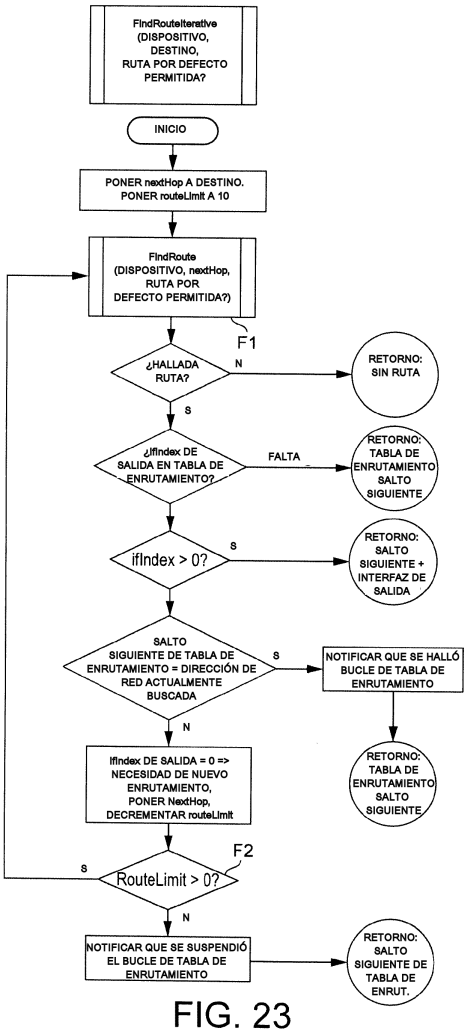
FindRouteIterative (DISPOSITIVO, DESTINO, RUTA POR DEFECTO PERMITIDA?
INICIO
PONER nextHop A DESTINO. PONER routeLimit A 10
FindRoute (DISPOSITIVO, nextHop, RUTA POR DEFECTO PERMITIDA?)
F1
¿HALLADA RUTA?
N
RETORNO: SIN RUTA
S
¿IfIndex DE SALIDA EN TABLA DE ENRUTAMIENTO?
FALTA
RETORNO: TABLA DE ENRUTAMIENTO SALTO SIGUIENTE
ifIndex > 0?
S
RETORNO: SALTO SIGUIENTE + INTERFAZ DE SALIDA
SALTO SIGUIENTE DE TABLA DE ENRUTAMIENTO = DIRECCIÓN DE RED ACTUALMENTE BUSCADA
S
NOTIFICAR QUE SE HALLÓ BUCLE DE TABLA DE ENRUTAMIENTO
RETORNO: TABLA DE ENRUTAMIENTO SALTO SIGUIENTE
N
IfIndex DE SALIDA = 0 => NECESIDAD DE NUEVO ENRUTAMIENTO, PONER NextHop, DECREMENTAR routeLimit
F2
S
RouteLimit > 0?
N
NOTIFICAR QUE SE SUSPENDIÓ EL BUCLE DE TABLA DE ENRUTAMIENTO
RETORNO: SALTO SIGUIENTE DE TABLA DE ENRUT.

FIG. 23

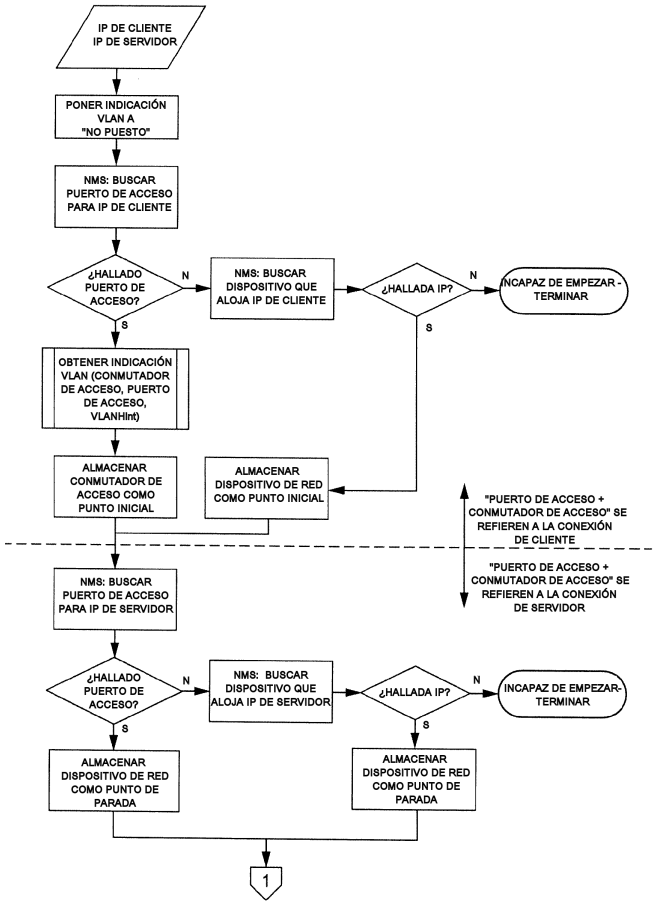
IP DE CLIENTE
IP DE SERVIDOR
PONER INDICACIÓN VLAN A "NO PUESTO"
NMS: BUSCAR PUERTO DE ACCESO PARA IP DE CLIENTE
¿HALLADO PUERTO DE ACCESO?
N
S
NMS: BUSCAR DISPOSITIVO QUE ALOJA IP DE CLIENTE
¿HALLADA IP?
N
S
INCAPAZ DE EMPEZAR - TERMINAR
OBTENER INDICACIÓN VLAN (CONMUTADOR DE ACCESO, PUERTO DE ACCESO, VLANHint)
ALMACENAR CONMUTADOR DE ACCESO COMO PUNTO INICIAL
ALMACENAR DISPOSITIVO DE RED COMO PUNTO INICIAL
"PUERTO DE ACCESO + CONMUTADOR DE ACCESO" SE REFIEREN A LA CONEXIÓN DE CLIENTE
"PUERTO DE ACCESO + CONMUTADOR DE ACCESO" SE REFIEREN A LA CONEXIÓN DE SERVIDOR
NMS: BUSCAR PUERTO DE ACCESO PARA IP DE SERVIDOR
¿HALLADO PUERTO DE ACCESO?
N
S
NMS: BUSCAR DISPOSITIVO QUE ALOJA IP DE SERVIDOR
¿HALLADA IP?
N
S
INCAPAZ DE EMPEZAR- TERMINAR
ALMACENAR DISPOSITIVO DE RED COMO PUNTO DE PARADA
ALMACENAR DISPOSITIVO DE RED COMO PUNTO DE PARADA
1

FIG. 24

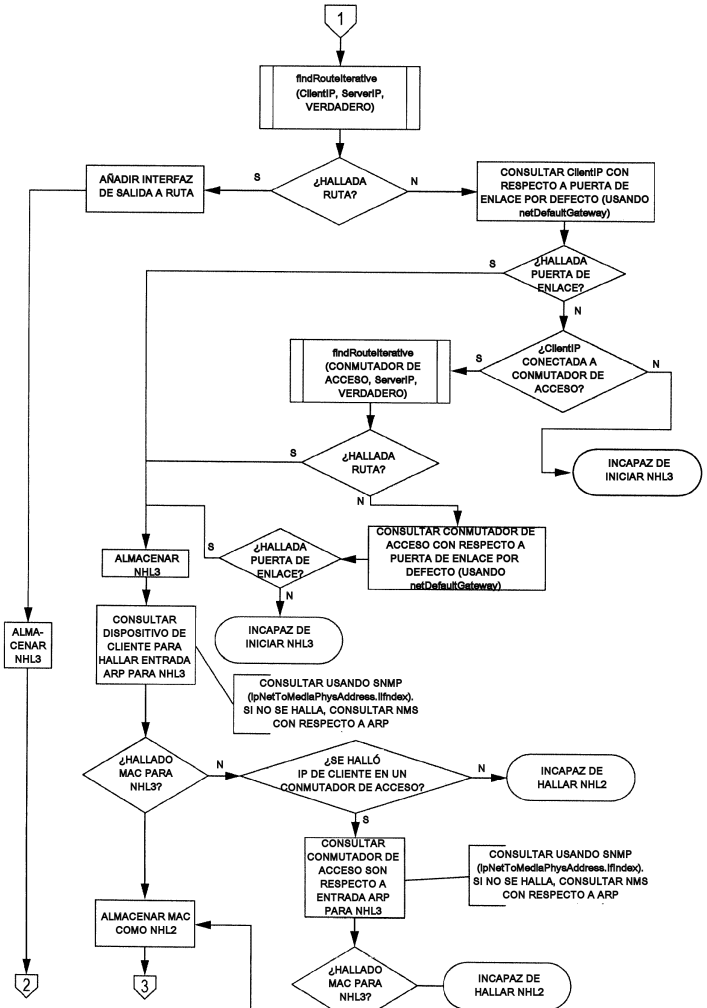
1
findRouteIterative (ClientIP, ServerIP, VERDADERO)
¿HALLADA RUTA?
S
N
AÑADIR INTERFAZ DE SALIDA A RUTA
CONSULTAR ClientIP CON RESPECTO A PUERTA DE ENLACE POR DEFECTO (USANDO netDefaultGateway)
¿HALLADA PUERTA DE ENLACE?
¿ClientIP CONECTADA A CONMUTADOR DE ACCESO?
findRouteIterative (CONMUTADOR DE ACCESO, ServerIP, VERDADERO)
INCAPAZ DE INICIAR NHL3
ALMACENAR NHL3
CONSULTAR CONMUTADOR DE ACCESO CON RESPECTO A PUERTA DE ENLACE POR DEFECTO (USANDO netDefaultGateway)
ALMA-CENAR NHL3
CONSULTAR DISPOSITIVO DE CLIENTE PARA HALLAR ENTRADA ARP PARA NHL3
CONSULTAR USANDO SNMP (IpNetToMediaPhysAddress.IfIndex). SI NO SE HALLA, CONSULTAR NMS CON RESPECTO A ARP
¿HALLADO MAC PARA NHL3?
¿SE HALLÓ IP DE CLIENTE EN UN CONMUTADOR DE ACCESO?
INCAPAZ DE HALLAR NHL2
CONSULTAR CONMUTADOR DE ACCESO SON RESPECTO A ENTRADA ARP PARA NHL3
ALMACENAR MAC COMO NHL2
2
3

FIG. 25

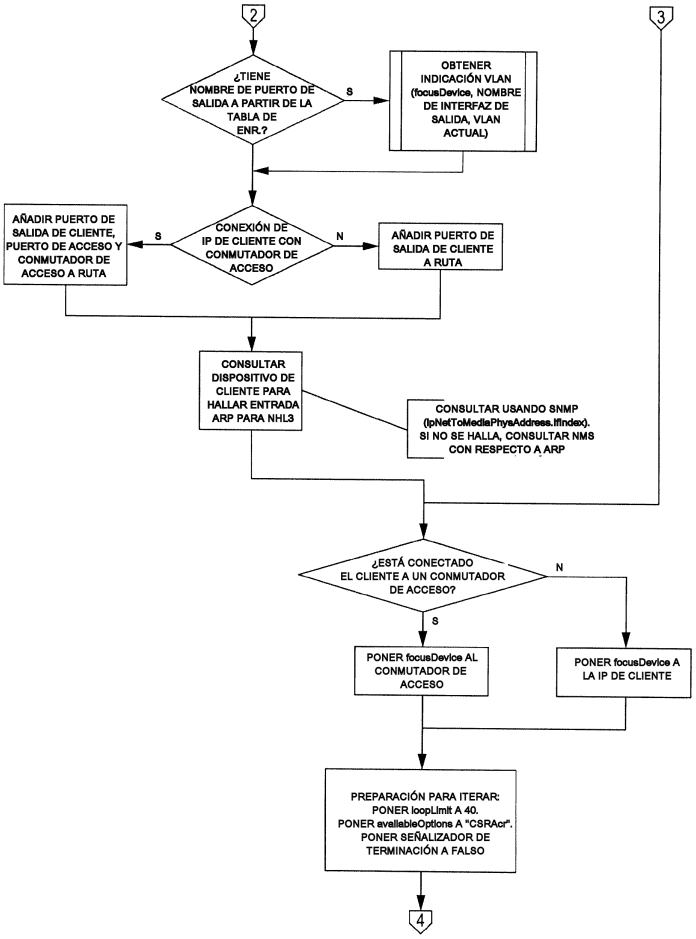
2
3
¿TIENE NOMBRE DE PUERTO DE SALIDA A PARTIR DE LA TABLA DE ENR.?
S
OBTENER INDICACIÓN VLAN (focusDevice, NOMBRE DE INTERFAZ DE SALIDA, VLAN ACTUAL)
AÑADIR PUERTO DE SALIDA DE CLIENTE, PUERTO DE ACCESO Y CONMUTADOR DE ACCESO A RUTA
S
CONEXIÓN DE IP DE CLIENTE CON CONMUTADOR DE ACCESO
N
AÑADIR PUERTO DE SALIDA DE CLIENTE A RUTA
CONSULTAR DISPOSITIVO DE CLIENTE PARA HALLAR ENTRADA ARP PARA NHL3
CONSULTAR USANDO SNMP (ipNetToMediaPhysAddress.ifIndex). SI NO SE HALLA, CONSULTAR NMS CON RESPECTO A ARP
¿ESTÁ CONECTADO EL CLIENTE A UN CONMUTADOR DE ACCESO?
N
S
PONER focusDevice AL CONMUTADOR DE ACCESO
PONER focusDevice A LA IP DE CLIENTE
PREPARACIÓN PARA ITERAR: PONER loopLimit A 40. PONER availableOptions A "CSRAcr". PONER SEÑALIZADOR DE TERMINACIÓN A FALSO
4

FIG. 26

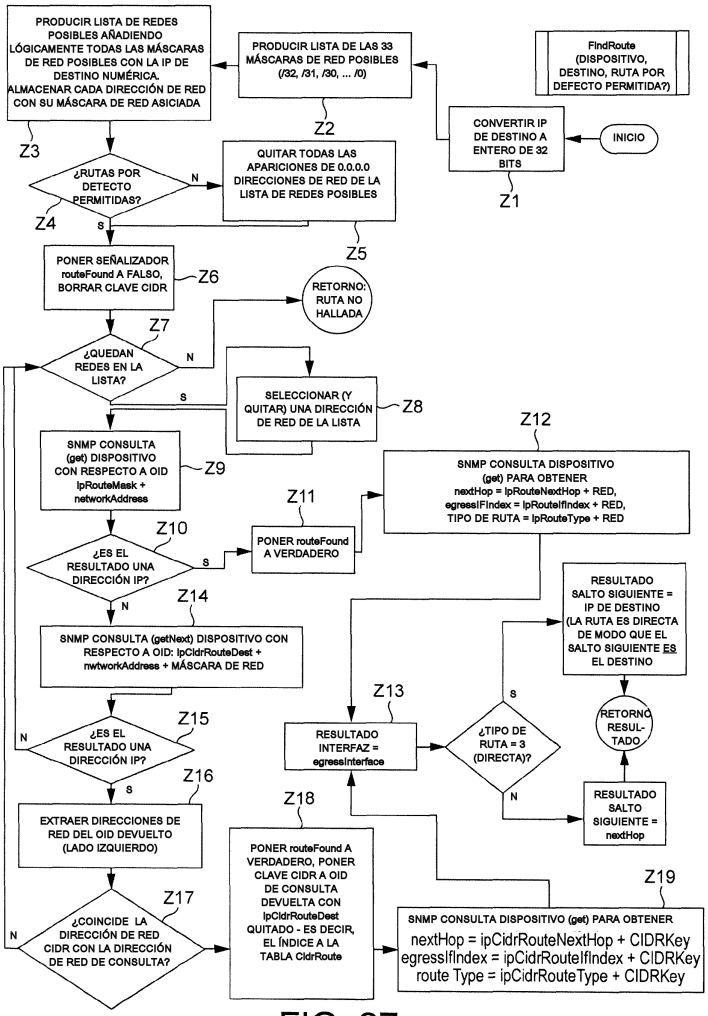
PRODUCIR LISTA DE REDES POSIBLES AÑADIENDO LÓGICAMENTE TODAS LAS MÁSCARAS DE RED POSIBLES CON LA IP DE DESTINO NUMÉRICA. ALMACENAR CADA DIRECCIÓN DE RED CON SU MÁSCARA DE RED ASICIADA
PRODUCIR LISTA DE LAS 33 MÁSCARAS DE RED POSIBLES (/32, /31, /30, ... /0)
FindRoute (DISPOSITIVO, DESTINO, RUTA POR DEFECTO PERMITIDA?)
CONVERTIR IP DE DESTINO A ENTERO DE 32 BITS
INICIO
Z1
Z2
Z3
¿RUTAS POR DETECTO PERMITIDAS?
N
S
Z4
QUITAR TODAS LAS APARICIONES DE 0.0.0.0 DIRECCIONES DE RED DE LA LISTA DE REDES POSIBLES
Z5
PONER SEÑALIZADOR routeFound A FALSO, BORRAR CLAVE CIDR
Z6
RETORNO: RUTA NO HALLADA
Z7
¿QUEDAN REDES EN LA LISTA?
N
S
SELECCIONAR (Y QUITAR) UNA DIRECCIÓN DE RED DE LA LISTA
Z8
Z12
SNMP CONSULTA (get) DISPOSITIVO CON RESPECTO A OID IpRouteMask + networkAddress
Z9
SNMP CONSULTA DISPOSITIVO (get) PARA OBTENER nextHop = IpRouteNextHop + RED, egressIFIndex = IpRouteIfIndex + RED, TIPO DE RUTA = IpRouteType + RED
Z11
Z10
¿ES EL RESULTADO UNA DIRECCIÓN IP?
S
PONER routeFound A VERDADERO
N
Z14
SNMP CONSULTA (getNext) DISPOSITIVO CON RESPECTO A OID: IpCidrRouteDest + nwtworkAddress + MÁSCARA DE RED
RESULTADO SALTO SIGUIENTE = IP DE DESTINO (LA RUTA ES DIRECTA DE MODO QUE EL SALTO SIGUIENTE ES EL DESTINO
Z13
S
RETORNO RESUL-TADO
Z15
¿ES EL RESULTADO UNA DIRECCIÓN IP?
N
RESULTADO INTERFAZ = egressInterface
¿TIPO DE RUTA = 3 (DIRECTA)?
N
RESULTADO SALTO SIGUIENTE = nextHop
S
Z16
EXTRAER DIRECCIONES DE RED DEL OID DEVUELTO (LADO IZQUIERDO)
Z18
Z19
Z17
¿COINCIDE LA DIRECCIÓN DE RED CIDR CON LA DIRECCIÓN DE RED DE CONSULTA?
N
PONER routeFound A VERDADERO, PONER CLAVE CIDR A OID DE CONSULTA DEVUELTA CON IpCidrRouteDest QUITADO - ES DECIR, EL ÍNDICE A LA TABLA CidrRoute
SNMP CONSULTA DISPOSITIVO (get) PARA OBTENER
nextHop = ipCidrRouteNextHop + CIDRKey
egressIfIndex = ipCidrRouteIfIndex + CIDRKey
route Type = ipCidrRouteType + CIDRKey

FIG. 27

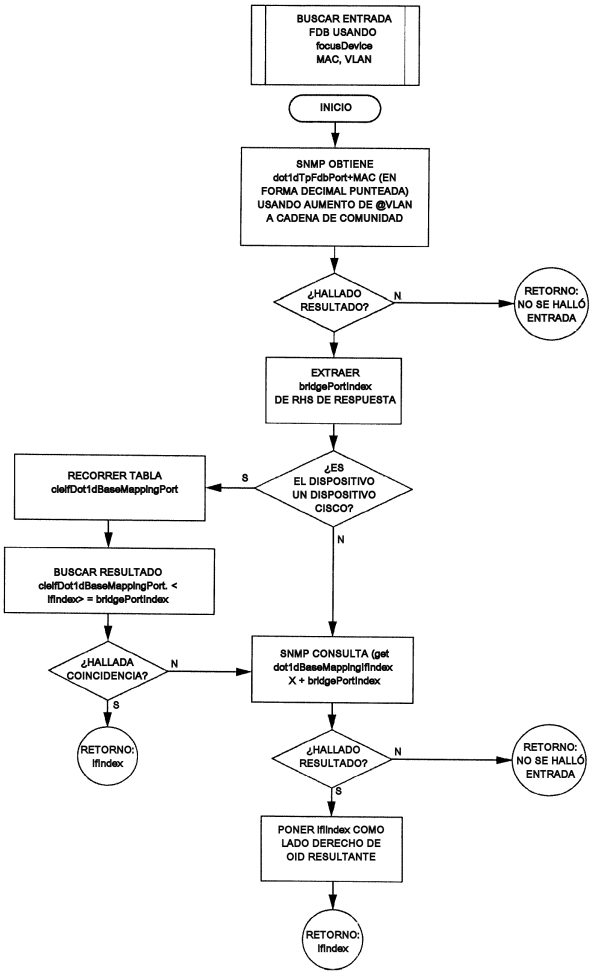
BUSCAR ENTRADA FDB USANDO focusDevice MAC, VLAN
INICIO
SNMP OBTIENE dot1dTpFdbPort+MAC (EN FORMA DECIMAL PUNTEADA) USANDO AUMENTO DE @VLAN A CADENA DE COMUNIDAD
¿HALLADO RESULTADO?
N
RETORNO: NO SE HALLÓ ENTRADA
EXTRAER bridgePortIndex DE RHS DE RESPUESTA
¿ES EL DISPOSITIVO UN DISPOSITIVO CISCO?
S
RECORRER TABLA cleIfDot1dBaseMappingPort
N
BUSCAR RESULTADO cleIfDot1dBaseMappingPort. < Ifindex> = bridgePortIndex
¿HALLADA COINCIDENCIA?
N
S
RETORNO: Ifindex
SNMP CONSULTA (get dot1dBaseMappingIfIndex X + bridgePortIndex
¿HALLADO RESULTADO?
N
S
RETORNO: NO SE HALLÓ ENTRADA
PONER Ifindex COMO LADO DERECHO DE OID RESULTANTE
RETORNO: Ifindex

FIG. 28