(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 689 913**

(51) Int. CI.:

H04L 12/751 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **28.05.2013 E 17155921 (4)**

(97) Fecha y número de publicación de la concesión europea: **15.08.2018 EP 3190755**

(54) Título: **Identificación de rutas tomadas a través de una red de dispositivos interconectados**

(30) Prioridad:

19.04.2013 GB 201307129

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
16.11.2018

(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 rutas tomadas a través de una red de dispositivos interconectados

La presente invención se refiere a la identificación de una ruta en una red de dispositivos interconectados.

Las redes informáticas constituyen la base de la infraestructura de TI (tecnología de la información) en una amplia variedad de contextos. Dichas redes informáticas comprenden dispositivos interconectados de diversos tipos. El propósito de la red es soportar el flujo de mensajes entre esos dispositivos para entregar información, aplicaciones y servicios, etc., a través de la red. Hay varias técnicas disponibles para administrar una red. En este contexto, la administración de una red incluye la monitorización de la red para identificar puntos de fallo y otras áreas problemáticas, tales como puntos de acceso inalámbrico, y proporcionar información a los administradores y usuarios de la red para permitir que se resuelvan los problemas. Hay una serie de herramientas disponibles para proporcionar una topología de red. La topología de una red identifica cómo los dispositivos de la red están físicamente o lógicamente conectados entre sí. Por lo tanto, cualquier dispositivo individual particular puede tener una o más conexiones a un dispositivo vecino. Las herramientas computarizadas que “descubren” una red están disponibles, y crean topologías de red que definen la interconexión de los dispositivos en la red, y la naturaleza de esos dispositivos.

Determinar la topología de la red se puede hacer de muchas maneras. Las técnicas que pueden utilizarse por separado o en combinación para dar una buena representación de la conectividad de red incluyen, por ejemplo:

- *Cisco Discovery Protocol (CDP)*
- *Link Layer Discovery Protocol (LLDP)*
- *SynOptics Network Management Protocol (SoNMP)*
- *Spanning Tree Protocol (STP)*
- *Traceroute IP*
- *IPv6 Neighbour Discovery*
- *Modificaciones/supresiones de adiciones de usuarios*

Saber la topología de una red es extremadamente útil, pero no proporciona una solución a todos los problemas que pueden ocurrir. Las redes se utilizan cada vez más para proporcionar la infraestructura para soportar la entrega de aplicaciones y servicios entre ubicaciones geográficas remotas, ya sea a largas distancias o a redes extremadamente complejas con un gran número de dispositivos interconectados. Cada vez más, los administradores de red y los usuarios están interesados en conocer no necesariamente los detalles completos de la red, sino para comprender el rendimiento de la entrega de aplicaciones y servicios a través de una red. Por lo tanto, la monitorización de “extremo a extremo” se está volviendo cada vez más popular. Con la monitorización de “extremo a extremo”, las aplicaciones que implican el flujo de mensajes desde un dispositivo fuente a un dispositivo de destino tienen su desempeño monitorizado a medida que se entregan entre ese dispositivo de origen y destino. Los parámetros de rendimiento se pueden utilizar para estimar o adivinar posibles fallos en la red, aunque no proporcionan ninguna información específica sobre la ubicación de esos fallos y, por tanto, no apuntan directamente a una solución.

A menudo, un dispositivo de origen es un servidor que proporciona un servicio particular y el dispositivo de destino es un terminal de cliente que está conectado al servidor a través de la red y que requiere utilizar dicho servicio. El término “dispositivo” utilizado en la presente memoria pretende cubrir cualquier dispositivo que pueda conectarse en una red. El término “servidor” se utiliza para denotar un dispositivo que es responsable de la entrega de un servicio o aplicación y el término “cliente” se utiliza para denotar un dispositivo (ya sea basado en el usuario u otra máquina o servidor dependiente) que depende de esa aplicación o servicio.

Una dificultad significativa en adivinar dónde podría estar un problema cuando se puede ver que el rendimiento de una aplicación se está deteriorando es una falta de comprensión sobre la ruta a través de la red que el flujo de mensajes para esa aplicación podría haber tomado.

Las redes dependen de muchos tipos de dispositivos de red (por ejemplo, enrutadores, conmutadores, cortafuegos, equilibradores de carga, etc.) para conectar sus dispositivos de punto final, de tal manera que es extremadamente difícil decir para cualquier punto final de origen cómo se dirigirá el mensaje desde ese punto final a través de la red a un punto final de destino determinado. La complejidad de tal determinación de ruta se ve exacerbada por el uso de múltiples rutas alternas, rutas redundantes, equilibrio de carga, etc.

Se han hecho intentos para predecir cómo un determinado paquete se encaminará a través de una red. Dichas predicciones se basan en un modelo complejo de la topología de la red junto con indicaciones sobre una base por dispositivo en cuanto a cómo un dispositivo particular se comportará en la red. Los dispositivos de red pueden ser altamente sofisticados, y se ha desarrollado un gran número de algoritmos complejos para determinar una estrategia de enrutamiento en cualquier dispositivo en particular. Además, esa estrategia de enrutamiento puede depender del

tráfico y de otras consideraciones medioambientales que afectan a la red (como fallos 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 de interfaz de entrada*
- *Cabeceras de paquetes (L2, L3, MPLS, ATM, etc.)*
- *Rutas estáticas y conectadas directamente*
- *Tablas de enrutamiento compartidas (conocimiento completo de BGP, OSPF, RIP, EIGRP, etc. - vecinos activos, estados de enlaces, costes de rutas, pesos de rutas, etc.)*
- *Tablas de reenvío MAC aprendidas*
- *Listas de control de acceso*
- *Tecnologías de superposición de red (p. MPLS, VLAN 802.1q), etc.*
- *Tecnologías de evitación de bucle - p. PVSTP*
- *Protocolos de tunelización (MPLS, IPSEC, SSL, GRE)*
- *Enlaces de carga equilibrada/redundantes*
- *Puertas de enlace predeterminadas*

Sin embargo, incluso si en principio es posible predecir dónde se enviará un paquete dado al siguiente en un dispositivo particular, esto requiere una gran cantidad de datos que es lenta de recopilar y puede estar obsoleta en cuestión de segundos debido a la naturaleza de tiempo real de la operación de los dispositivos de enrutamiento. Además, la mera adquisición de estos datos puede suponer una carga significativa tanto en los dispositivos de red como en las redes.

Además, los modelos necesarios para simular dispositivos de enrutamiento modernos son extremadamente complejos y, sin el modelo completo, su comportamiento no puede predecirse correctamente. Para mantener el modelo completo, es necesario que tales soluciones sean actualizadas regularmente y rápidamente para incluir desarrollos en las tecnologías de enrutamiento.

El documento US2011/142054A1 describe un método para consultar tablas de enrutamiento con el fin de obtener información sobre puertos de entrada y salida. Esta información se utiliza para configurar nuevas rutas de enrutamiento. El proceso de consulta se realiza mediante un nuevo enrutador configurado que le envía consultas a toda la red de una manera difundida. La información recibida es entonces utilizada por dicho enrutador para aprender la topología de red y para configurar rutas de acuerdo con criterios tales como costes de ruta, ancho de banda disponible, seguridad de ruta y así sucesivamente.

Sumario

Con el enfoque novedoso detallado aquí, los dispositivos de red se consultan en cuanto a lo que harían con un paquete hipotético (en lugar de consultar las especificaciones del protocolo de enrutamiento). Los protocolos de enrutamiento que se alimentan en las tablas de enrutamiento y reenvío de los dispositivos no necesitan ser conocidos o mantenidos.

De acuerdo con un primer aspecto de la presente invención, se proporciona un método implementado por ordenador para identificar en una red de dispositivos interconectados una ruta a través de la red desde un dispositivo fuente a un dispositivo destino, comprendiendo la ruta una secuencia conectada de dispositivos, comprendiendo el método en un ordenador monitor conectado a la red: identificar un primer dispositivo de conmutación en la trayectoria, teniendo el primer dispositivo una tabla de reenvío de tráfico con entradas partidas de acuerdo con la VLAN con la que están relacionadas; establecer en cuál de las múltiples redes de área local virtuales (VLAN) participa el primer dispositivo; transmitir una primera consulta al primer dispositivo, incluyendo la consulta un identificador de destino en forma de una dirección de conmutación, y solicitar la identificación de un puerto de salida para mensajes dirigidos al destino identificado por el identificador de destino cuando se recibe la consulta en el primer dispositivo, en el que se genera la primera consulta para la tabla de reenvío de tráfico utilizando sondeo específico de VLAN para consultar la tabla de reenvío de tráfico en el contexto de cada VLAN en la que el dispositivo participa; recibir un mensaje de resultado que identifica el puerto de salida e identificar un segundo dispositivo conectado al primer dispositivo basado en una topología de red accesible por el ordenador monitor; y dirigir una consulta siguiente al segundo dispositivo y recibir un mensaje de resultado siguiente que identifica un puerto de salida desde el segundo dispositivo; e identificar desde la topología de red un tercer dispositivo conectado al segundo dispositivo, en el que la ruta se identifica para incluir el primer, el segundo y el tercer dispositivos.

De acuerdo con un segundo aspecto de la presente invención, se proporciona un sistema informático de monitorización para identificar en una red de dispositivos interconectados a una ruta a través de la red desde un dispositivo fuente a un dispositivo de destino, comprendiendo el sistema informático: una interfaz conectada a la red para transmitir consultas y recibir respuestas; un procesador operable para ejecutar una herramienta de identificación de ruta que realiza el método implementado por ordenador del primer aspecto; un primer medio de almacenamiento para almacenar el registro de ruta; y un segundo medio de almacenamiento para almacenar la

topología de red.

El identificador de destino en la primera consulta puede ser una dirección de enrutamiento del dispositivo de destino final.

El identificador de destino en la primera consulta puede ser una derivación de conmutación derivada de una dirección de enrutamiento para un dispositivo de enrutamiento en la red.

El método implementado por ordenador puede comprender registrar un conjunto de dispositivos y puertos identificados para estar en la ruta en un almacén accesible para el ordenador monitor como un registro de ruta. El registro de ruta puede incluir puertos de ingreso identificados desde la topología de red para dispositivos identificados.

El método implementado por ordenador puede comprender la transmisión desde el ordenador monitor al dispositivo que se está consultando en un mensaje que se transmite a través de la red. El mensaje puede tomar la forma de al menos un paquete dirigido al dispositivo que se está consultando. La consulta puede incluir una pluralidad de claves para consultar el dispositivo. Una única consulta que contiene dicha pluralidad de claves puede provocar una respuesta con la identificación de un puerto de salida que se devolverá.

El primer dispositivo puede identificarse basándose en la topología de red. El primer dispositivo conectado al dispositivo fuente puede identificarse ubicando en la topología de red un puerto de red en el que se ha visto una dirección de conmutación del dispositivo fuente, y a su vez qué dispositivo forma parte de la topología de red a la que pertenece este puerto de red.

El método implementado por ordenador puede comprender identificar una primera ruta a través de la red desde el dispositivo fuente hasta el dispositivo de destino usando un primer conjunto de consultas por primera vez, y rutas adicionales a través de la red desde el dispositivo fuente hasta el dispositivo de destino final utilizando conjuntos de consultas posteriores. El método implementado por ordenador puede comprender identificar repetidamente la primera y la segunda ruta y determinar el uso de las rutas.

El método implementado por ordenador cuando se usa en una red de dispositivos interconectados que incluye dispositivos de capa 2 y capa 3, puede comprender, después de identificar el primer dispositivo conectado al dispositivo fuente: identificar un siguiente identificador de destino de capa 3, a lo largo de la ruta hacia el dispositivo de destino final; a continuación, identificar un identificador de destino de capa 2 del siguiente identificador de destino de capa 3. Si el primer dispositivo aloja el siguiente identificador de destino de capa 3, entonces el implementado por ordenador puede comprender enviar un mensaje de consulta al primer dispositivo que contiene el identificador de destino del dispositivo de destino final, para determinar un nuevo identificador de destino de capa 3 siguiente. junto con un nuevo identificador de destino de capa 2 de este nuevo identificador de destino de la siguiente capa 3, y el puerto de salida para los mensajes dirigidos a este nuevo identificador de destino de la siguiente capa 3.

El método implementado por ordenador puede comprender enviar una consulta al dispositivo de origen, que contiene el identificador de destino de la capa 3 final, y determinar a partir de la respuesta el siguiente identificador de destino de la capa 3.

Si el primer dispositivo no aloja el siguiente identificador de destino de capa 3, entonces el método implementado por ordenador puede comprender enviar un mensaje de consulta al primer dispositivo para determinar el puerto de salida para los mensajes dirigidos al identificador de destino de capa 2 del siguiente identificador de destino de capa 3, y luego identificar un nuevo dispositivo de siguiente salto conectado al puerto de salida basado en una topología de red accesible por el ordenador monitor. Las etapas de envío de un mensaje de consulta en función de si el primer dispositivo aloja el siguiente identificador de destino de capa 3 pueden repetirse con cada nuevo dispositivo de salto siguiente en lugar del primer dispositivo hasta que el nuevo dispositivo de salto siguiente sea el dispositivo de destino.

Si el dispositivo de origen no responde a la consulta, el método implementado por ordenador puede comprender enviar una primera consulta al primer dispositivo, que contiene el identificador de destino de capa 3 final, y determinar a partir de la respuesta el siguiente identificador de destino de capa 3; a continuación, si el primer dispositivo no proporciona identificación del siguiente identificador de destino de capa 3 a través de la primera consulta, enviar una consulta posterior al dispositivo de salto siguiente para determinar una puerta de enlace predeterminada del primer dispositivo y utilizar el identificador de destino de capa 3 de la puerta de enlace predeterminada el siguiente identificador de destino de la capa 3.

La implementación de ordenador puede comprender enviar una consulta al dispositivo fuente, que contiene el siguiente identificador de destino de capa 3, y determinar a partir de la respuesta el identificador de destino de la capa 2 del identificador de destino de la siguiente capa 3.

Si la fuente no responde a la consulta, el método implementado por ordenador puede comprender enviar una consulta al primer dispositivo, que contiene el siguiente identificador de destino de capa 3, y determinar a partir de la respuesta el identificador de destino de capa 2 del siguiente identificador de destino de capa 3; a continuación, si el primer dispositivo no proporciona el identificador de destino de capa 2 del siguiente identificador de destino de capa 3, entonces se determina el identificador de destino de capa 2 del siguiente identificador de destino de capa 3 utilizando una tabla de asignación. La tabla de asignación puede asignar direcciones de capa 3 a direcciones de capa 2.

Cada vez que se identifica un dispositivo de salto siguiente utilizando la topología de red, tanto el dispositivo de salto siguiente como el puerto de entrada en el que recibirá los mensajes del dispositivo anterior en la ruta, pueden registrarse como parte de la ruta, junto con el dispositivo anterior y su puerto de salida.

Cada consulta puede generarse para una tabla de reenvío de tráfico en el dispositivo al que se transmite la consulta, utilizando la consulta al menos uno de un conjunto de claves que se han generado combinando el identificador de destino con una pluralidad de índices integrados de la tabla de reenvío de tráfico.

La presente invención utiliza un nuevo enfoque para identificar las rutas tomados a través de una red de dispositivos interconectados para un flujo particular de mensajes. El concepto se basa en el uso de una cantidad mínima de datos recopilados “de antemano”, específicamente la topología de la red estática y la ubicación del host final (cuyos clientes y servidores están conectados a los conmutadores de acceso/borde) y reúne todo lo que se requiere sobre la marcha y de forma altamente selectiva según se requiera para estos datos altamente dinámicos. Para los entornos dinámicos modernos, la capacidad de calcular la ruta de extremo a extremo ahora, es decir, en tiempo real, tiene una amplia aplicabilidad. La recopilación de los datos y su procesamiento tienen que ser muy rápidos para que el algoritmo sea de valor práctico cuando se utiliza con redes de gran escala del mundo real.

El comportamiento en un dispositivo particular se denomina “comportamiento por salto” (PHB). El PHB por sí solo no puede proporcionar una ruta de extremo a extremo. Sin embargo, saber que un paquete deja el dispositivo en una interfaz específica puede ser de valor si no se sabe qué dispositivo e interfaz están conectados a esa interfaz. Mediante el uso de la topología de red acoplada con el PHB, se puede realizar el cálculo directo de una ruta de extremo a extremo a través de la red para un flujo de aplicación.

Por lo tanto, la topología de red incluye tanto la interconectividad del dispositivo de red como la ubicación del host final. A su vez, los dispositivos de origen y de destino utilizados para sembrar el algoritmo de búsqueda de ruta son determinados por el flujo de interés de interés. Finalmente, la ruta específica tomada por un paquete hipotético entre los dispositivos de origen y de destino se calcula dinámicamente.

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. Tenga en cuenta que el identificador de destino para cualquier consulta determinada puede ser o no el identificador de destino del dispositivo terminal dependiendo de la ubicación en la red del dispositivo que se está consultando. Esto se puede lograr cuando el dispositivo es un enrutador consultando lo que está en su tabla de enrutamiento activa en el momento en que se recibe la consulta.

La consulta en sí puede alojarse en un mensaje o señal transmitida desde el ordenador monitor al dispositivo que se está consultando (dispositivo de enfoque). El mensaje o señal de consulta no constituye el flujo de mensajes para el que se ha de determinar la ruta. En su lugar, cada consulta contiene un identificador de destino que consulta el dispositivo de enfoque para averiguar cómo el dispositivo de enfoque manejaría un mensaje hipotético dirigido a ese destino si tuviera que tomar la decisión en el momento en que se recibió la consulta. Por lo tanto, el dispositivo de enfoque devuelve un resultado que identifica un puerto de salida inmediato que habría sido utilizado en ese momento para un mensaje real dirigido a ese destino. Las consultas se pueden transmitir mientras la red está activa y mientras el flujo de mensajes está en su lugar. Sin embargo, también pueden transmitirse cuando el flujo de mensajes en sí no está activo - la técnica puede utilizarse en cualquier contexto.

Cuando la consulta está en forma de 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 entregada a través de la red desde el ordenador monitor al dispositivo de enfoque. En ese caso, la dirección de destino del mensaje de consulta es la del dispositivo de enfoque. Esto no es lo mismo que el identificador de destino que se incluye en la consulta en sí. En una disposición alternativa, una señal o señales de consulta pueden ser enviadas desde el ordenador monitor a través de conexiones directas a los dispositivos de enfoque, por ejemplo, a través de un mecanismo CLI o XML API.

El método descrito en la presente memoria permite una serie de técnicas de análisis de red útiles. Permite la determinación de ruta a petición para que un administrador que intenta determinar la ruta de acceso para una aplicación concreta pueda preguntar de forma más o menos instantánea al ordenador monitor y recibir un resultado de la ruta.

Permite el descubrimiento de rutas múltiples. Es decir, debido a los cambios en el entorno de la red, los dispositivos de enrutamiento pueden dirigir un flujo de mensajes de manera diferente dependiendo de esos cambios. Por lo tanto, un primer conjunto de consultas para identificar una ruta de acceso podría registrar una primera ruta, mientras que un segundo conjunto de consultas podría identificar una segunda ruta de acceso, incluso cuando el primer y el segundo conjunto de consultas están muy próximos entre sí en el tiempo. La información sobre múltiples rutas entre puntos finales comunes (es decir, el mismo dispositivo de origen y el mismo dispositivo de destino) puede presentarse gráfica o visualmente para mostrar al usuario no solo la naturaleza de la ruta, sino el porcentaje de tiempo que cada ruta adopta para un determinado flujo de mensajes. Esto se puede lograr fácilmente porque las propias consultas no representan una sobrecarga significativa para la red y, por lo tanto, se pueden enviar múltiples conjuntos de consultas sin afectar significativamente el rendimiento.

El método permite la detección de un rápido y legítimo cambio de ruta. Es decir, un ajuste a la red puede hacer que la ruta cambie y esto puede ser detectado y marcado a un usuario en una interfaz gráfica de usuario visual.

Cuando hay múltiples rutas de acceso entre los dispositivos comunes de origen y destino, las rutas pueden llevar diferentes latencias. A veces, un dispositivo de enrutamiento que realiza un enrutamiento inteligente puede causar un fenómeno conocido como “variación de ruta” donde un flujo particular de mensajes cambia continuamente de ruta a ruta. Puede ser útil para un administrador de red identificar estas ocurrencias debido a las implicaciones de tales cambios de ruta en la latencia de extremo a extremo y las implicaciones de dicha “inestabilidad” en las conversaciones telefónicas de voz sobre IP, por ejemplo.

El método se puede utilizar para localizar el fallo de ruta. Es decir, en la realización preferida del método, se envían consultas y se reciben resultados y se analizan para identificar el dispositivo siguiente hasta que se identifica un dispositivo como el dispositivo de destino. A veces, sin embargo, hay un fallo en la red, de tal manera que la red no entregaría el flujo de mensajes al dispositivo de destino. El método permite la identificación de esa situación al trabajar a lo largo de una ruta de extremo a extremo hasta que la ruta no se pueda recorrer más y esta ubicación de red puede entonces notificarse a un administrador.

Además, el método puede permitir la posibilidad de reiniciar en un dispositivo subsiguiente en esa ruta, utilizando estimaciones basadas en la topología de red. El método de identificación de ruta puede entonces ser adoptado de nuevo hasta que se alcance el dispositivo de destino desde el punto de fallo. De este modo, las porciones de la red para las que el ordenador de supervisión no tiene visibilidad (por ejemplo, dispositivos que no tienen una interfaz de gestión apropiada o que pertenecen a una organización diferente) pueden circunnavegarse y continuar el análisis de ruta.

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

La ruta se registra en una memoria o se almacena en el ordenador monitor o accesible por el ordenador monitor. El registro de ruta comprende un conjunto de dispositivos e interfaces conectados. Esto puede presentarse en forma de un inventario ordenado de los dispositivos (componentes de red) entre los dos extremos. Esto permite la monitorización de disponibilidad de ruta de red, incluyendo notificación de eventos, informes, SLA (acuerdos de nivel de servicio); administración proactiva de la red incluyendo informes sobre dispositivos que fallan, CPU de alto dispositivo, memoria de dispositivo baja, congestión de puertos, etc., y análisis de impacto (planificación de capacidad, análisis de “qué pasa si”).

Es una ventaja significativa de los aspectos de la presente invención que una asignación entre la aplicación o el servicio suministrado por la red y los dispositivos de red o los componentes mismos se pueda determinar a través de la identificación de la ruta. Esto representa un avance significativo en la gestión de las redes.

Para una mejor comprensión de la presente invención y para mostrar cómo puede llevarse a cabo la misma, se hará ahora referencia a modo de ejemplo, a los dibujos adjuntos, en los que:

la figura 1 es un diagrama esquemático de una red;
las figuras 2a a 2c son una ilustración esquemática de un algoritmo de descubrimiento de ruta en proceso;
la figura 3 es un diagrama de flujo para un algoritmo de descubrimiento de ruta;
la figura 4 muestra una ruta descubierto;
la figura 5 es la estructura de una tabla de enrutamiento lineal;
la figura 6 ilustra un conjunto de resultados que se derivan de la combinación de una dirección de destino con múltiples máscaras de ruta;
la figura 7 muestra una estructura de una tabla ARP;
la figura 8 es un diagrama esquemático de un ordenador monitor;
la figura 9 es un diagrama esquemático de un enrutador de capa 3;
la figura 10 es un diagrama esquemático de un conmutador de capa 2; y

las figuras 11A a 11D son un diagrama de flujo de una utilidad ejecutada en el ordenador monitor.

La figura 1 es un diagrama esquemático de una red. La red se extiende sobre un número de diferentes ubicaciones geográficas. En cada extremo de la ubicación geográfica hay dispositivos de punto final y dispositivos de red o nodos. Los dispositivos de red incluyen enrutadores e conmutadores. El núcleo de la red comprende una pluralidad de dispositivos de red. Teniendo en cuenta la ubicación geográfica marcada en Londres, los terminales del cliente 2 pueden actuar como dispositivos de punto final. De manera similar, un servidor 4 puede actuar como un dispositivo de extremo y la impresora 6 puede considerarse un dispositivo de punto final. Dispositivos similares se muestran en las ubicaciones geográficas de París y Nueva York con diferentes diseños (Nueva York mostrando una granja de servidores o centro de datos). Obsérvese que, en la ubicación de Nueva York, una pluralidad de servidores 8 representa una aplicación clave o dispositivos de punto final de servicio.

Debe apreciarse que la red mostrada en la figura 1 se da por medio de un ejemplo. Existe una amplia variedad de redes posibles y la presente invención puede utilizarse en cualquier red de dispositivos interconectados. En particular, la naturaleza de los dispositivos de punto final y los dispositivos o nodos de red específicos puede variar. En la red particular que se describe, los dispositivos de red pueden ser dispositivos de capa 3 o capa 2.

El modelo OSI (interconexión de sistemas abiertos) define siete capas dentro de las cuales los protocolos de los sistemas de comunicación pueden ser caracterizados. El algoritmo de búsqueda de rutas descrito aquí calcula las rutas de red utilizando la información disponible en las capas 2 y 3.

Los dispositivos que funcionan en la capa 2 (la capa de enlace de datos) tienen conocimiento de dispositivos inmediatamente adyacentes y tienen la responsabilidad de obtener paquetes de un dispositivo de capa 2 al siguiente dispositivo de capa 2 (basado en la dirección de la capa 2 de MAC (control de acceso de medios)).

Los dispositivos que operan en la capa 3 (la capa de red) son responsables de propagar paquetes de un punto en una red a otro punto de la red, a muchas decenas o cientos de dispositivos de distancia. Para calcular qué dispositivos deben participar en una ruta dado de la capa 3 (denominada en la presente memoria como saltos de la capa 3), los dispositivos de la capa 3 intercambian información de enrutamiento y usan protocolos de enrutamiento para calcular las rutas más deseables.

Para pasar paquetes entre dispositivos de capa 3 consecutivos en una ruta, se utilizan dispositivos que funcionan en la capa 2; a menudo con muchos dispositivos de capa 2 (denominados en la presente memoria como saltos de capa 2) entre cada dispositivo de capa 3.

Así, las redes grandes se subdividen efectivamente en múltiples segmentos, cada uno de los cuales conteniendo típicamente dispositivos de capa múltiple 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 comprende un controlador 90 por ejemplo, en forma de microprocesador que ejecuta un código de control, firmware o cualquier otra implementación adecuada. El controlador 90 puede acceder a una tabla de enrutamiento 92 que se analiza 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 para la conveniencia de la notación, en la práctica, los dispositivos no suelen tener puertos que se dedican como puertos de entrada o de salida - si son de entrada o de salida depende de los datos que están transfiriendo en ese momento. La mayoría de los puertos funcionan como de salida y de entrada todo el tiempo.

Los paquetes que llegan a un puerto de entrada Pi pueden tener sus identificadores de destino, por ejemplo, IP (direcciones de protocolo de Internet) leídas por el controlador 90 a través de un bus 94. El controlador 90 accede a la tabla de enrutamiento 92 y basado en la información derivada de la misma controla un conmutador de enrutamiento 96 al que se dirige el paquete entrante. El conmutador de enrutamiento 96 dirige entonces el paquete entrante a un puerto de salida Po adecuado dependiendo de la información en la tabla de enrutamiento. El dispositivo de enrutamiento incluye una tabla de asignación 91 que asigna las direcciones de la capa 3 a la capa 2 para el enrutamiento posterior. El funcionamiento de tales dispositivos de enrutamiento se conoce en la técnica y por lo tanto no se describirán más en el presente documento. Se observa en este contexto que la tabla de enrutamiento puede ser consultada por paquetes desde el ordenador monitor que llegan sobre los enlaces al puerto de entrada Pi interceptando dichos paquetes en el controlador 90. Tales paquetes de consulta no se suministran al conmutador de enrutamiento 96 para el enrutamiento adicional, sino que en su lugar generan una respuesta que es emitida desde el dispositivo de enrutamiento y devuelta a la entidad interrogadora a través de la red desde un puerto de salida. En este caso, esa entidad interrogadora es el ordenador monitor 16. Todos los paquetes transportados a través de la red (incluidos los paquetes de consulta) contienen una fuente y una dirección de destino; el paquete de consulta tiene una dirección de origen correspondiente al ordenador monitor y una dirección de destino correspondiente al dispositivo que se está consultando. Cuando se necesita enviar la respuesta, se intercambian las direcciones de origen y de destino para hacer que la dirección de origen sea el dispositivo que se está consultando y la dirección de destino sea el ordenador monitor.

La figura 10 es una versión altamente esquematizada de un conmutador de capa 2. De manera similar a 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 muestra, por ejemplo, en la red de la figura 1. Como se mencionó anteriormente, los puertos no suelen ser dedicados como entrada o salida. Los paquetes entrantes en un puerto de entrada Pi se dirigen a un conmutador 100 que puede acceder a una base de datos de reenvío de capa 2 (FDB) 102 para determinar cómo encaminar los paquetes basándose en identificadores de destino (normalmente cabeceras) en los paquetes. Las bases de datos de reenvío de la capa 2 asignan el identificador de un paquete entrante a un puerto de salida en el que se debe reenviar el paquete. 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.

Al igual que con los dispositivos de capa 3, la capa 2 es conocida en la técnica y, por lo tanto, no se discutirá más en el presente documento. Sin embargo, se observa que de nuevo como 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 a la salida del conmutador de capa 2 en un puerto de salida Po. Por lo tanto, los propios paquetes de consulta no son dirigidos en el conmutador, sino que en su lugar generan una respuesta que se devuelve al dispositivo de consulta, en este caso el ordenador monitor 16.

Un controlador de conmutación 101 en el conmutador es responsable de reenviar tráfico y de generar respuestas.

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

Las realizaciones descritas a continuación de la presente invención proporcionan un método para identificar una ruta tomado por un flujo de mensajes entre un dispositivo de fuente dado y un dispositivo de destino dado. Por ejemplo, el punto extremo X podría considerarse un dispositivo fuente y el punto final Y podría considerarse un dispositivo de destino. Frente a una red de la figura 1, como ya se ha comentado anteriormente, es una tarea para nada trivial de establecer qué ruta se adoptará a través de la red entre esos extremos en un momento dado y bajo cualquier conjunto de condiciones ambientales. La figura 1 muestra un ordenador monitor 16 que ejecuta un programa de detección de ruta que permite descubrir y registrar dicha ruta. La figura 8 es una versión altamente esquemática de un ordenador monitor 16. El ordenador 16 comprende un microprocesador 80 que puede acceder a la memoria 82 en la que se almacena código para su ejecución por el procesador. En el presente caso, el código incluye el programa de descubrimiento de ruta. La memoria 82 almacena también un registro de ruta 81 cuando es creado por el programa de descubrimiento de ruta. El ordenador tiene una interfaz de usuario 84 que puede incluir un dispositivo de entrada de usuario tal como un ratón o teclado y una pantalla para mostrar información a un usuario. En particular, tal como se describe con más detalle, las alertas que siguen al programa de descubrimiento de ruta o información relativa al programa de descubrimiento de ruta pueden ser mostradas a un usuario en la interfaz de usuario 84. Las figuras 2a a 2c ilustran etapas de la ruta como se describirá ahora.

A un nivel alto, el algoritmo usa la noción de un “dispositivo de enfoque” que es el dispositivo que se está consultando actualmente a dónde enviaría un paquete hipotético a continuación (es decir, de qué interfaz enviaría el paquete hipotético). A partir del dispositivo fuente, el algoritmo viaja hacia el dispositivo terminal (es decir, el destino final del paquete) evaluando cada dispositivo de enfoque a su vez - si el dispositivo está operando en la capa 3, se consulta cuál interfaz (puerto de salida) utilizaría para enviar paquetes enlazados para el salto siguiente en la capa 3 (NHL3); si el dispositivo está funcionando en la capa 2, se consulta cuál interfaz (puerto de salida) se usaría para enviar paquetes enlazados para la capa 2 (MAC) de la capa 3 de la capa siguiente (NHL2). Utilizando la respuesta del dispositivo de enfoque junto con una topología de red, se puede determinar el siguiente dispositivo en la ruta. De esta forma, el algoritmo trabaja a lo largo de la capa 3, utilizando dispositivos de capa 2 para navegar entre los nodos consecutivos de la capa 3.

Antes de comenzar el algoritmo principal, se localizan el dispositivo fuente y el dispositivo terminal. Esto puede no ser sencillo y las técnicas para lograr esto se discuten más adelante.

Según el algoritmo principal, se localiza el primer salto. La ruta se sembró y el recuento de bucle se establece en cero. El límite de bucle controla el número de veces que se ejecuta un bucle de identificación de ruta (discutido más adelante).

Encontrar el primer salto en la capa 3

El primer salto se localiza encontrando el siguiente salto inicial (el salto siguiente desde el dispositivo fuente) en la capa 3 (NHL3). En la siguiente explicación, el término “consulta” se utiliza con frecuencia. Las consultas se generan y estructuran como se describe con más detalle más adelante. El propósito de una consulta es localizar una dirección de salto siguiente y un puerto de salida desde un dispositivo de enfoque al que se dirige la consulta. La dirección del 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 el NHL3 y el puerto de salida. Si no se encuentra ninguna ruta y el dispositivo de origen tiene un conmutador de acceso de capa 3, se consulta este conmutador de acceso de la capa 3 para el NHL3 usando la dirección IP de destino. Si eso

no tiene éxito, se consulta la puerta de enlace predeterminada en el dispositivo de origen para determinar el NHL3. Si no tiene éxito, se realiza una consulta con la dirección IP de destino al conmutador de acceso para la puerta de enlace predeterminada. Si no se encuentra ninguna dirección del NHL3, esto se considera como un error. Esto no significa que el algoritmo ha fallado, sino que un punto de fallo en la ruta puede haber sido identificado en este punto. Alternativamente, puede haber otras razones por las que no se ha encontrado el NHL3.

Sembrar la ruta

Para sembrar la ruta, el dispositivo fuente se agrega a la ruta cuando se ha localizado. La interfaz de salida del dispositivo de origen se encuentra y se agrega a la ruta. Si se encuentra el NHL3 desde la tabla de enrutamiento en el dispositivo fuente, la interfaz de salida del dispositivo fuente para esta dirección del NHL3 se añade a la ruta. Como se explica más adelante, se puede determinar la dirección de la capa 2 (NHL2) correspondiente a la dirección de la capa 3 (NHL3). Si no se encuentra un puerto de salida para el NHL3 desde la tabla de enrutamiento en el dispositivo de origen, se usa la tabla de reenvío de capa 2 en el dispositivo de origen para el NHL2 para encontrar el puerto de salida. Si se encuentra, entonces ese puerto de salida se agrega a la ruta.

Descripción general del algoritmo de detección de ruta

La consulta enviada desde el ordenador monitor 16 al dispositivo fuente X se muestra como una flecha directa en la figura 2a, pero de hecho podría implementarse en la red de la figura 1 por el ordenador monitor 16 que emite un mensaje o paquete dirigido a la dispositivo fuente X. Como se ha explicado, la consulta solicita al dispositivo fuente el siguiente salto de IP (y puerto de salida) para la IP del terminal (IP de destino), que es la dirección de capa 3 del punto de destino Y. El objetivo es provocar que el dispositivo fuente X suministre una respuesta que incluye el NHL3 y el puerto de salida para el NHL3 (la dirección IP terminal). Véase la etapa S1 de la figura 3 y la figura 2a.

Como se ha explicado anteriormente, puede haber situaciones en las que el dispositivo fuente no puede proporcionar la información necesaria. Otras posibilidades mencionadas anteriormente para obtener el primer dispositivo de “enfoque” incluyen consultar el conmutador de acceso conectado para la información de enrutamiento de la capa 3 (en caso de que el conmutador de acceso sea un conmutador de capa 3), si falla el algoritmo consulta el conmutador de acceso conectado para una pasarela predeterminada y la dirección IP de la puerta de enlace predeterminada utilizada como el primer NHL3.

En la etapa S2, la siguiente dirección de la capa de salto 2 (MAC) se resuelve desde la dirección del NHL3 y el NHL2 se establece en esta dirección MAC. Esto se puede conseguir consultando una tabla de asignación 91 que asigna direcciones L3 a L2. Una de estas tablas de asignación es una tabla ARP (otras incluyen “asignación directa” y descubrimiento del vecino). Éste puede ser el dispositivo fuente ARP, el siguiente dispositivo ARP de salto L3 o ARP en caché global utilizando una consulta ARP descrita más adelante. El puerto de salida identificado en la etapa S1 se añade al registro de ruta S1A. En la etapa S3, el conmutador de red de inicio (y el puerto) se encuentra utilizando la ubicación del host final en caché (desde las consultas CAM del conmutador) y se establece como el dispositivo de enfoque. En la etapa S4, el conmutador de red de terminal se encuentra utilizando la ubicación de host final almacenada en caché (desde las consultas CAM de conmutador). El interruptor de inicio se agrega al registro de ruta.

El método ahora está listo para introducir un bucle de identificación de ruta. En la etapa S5 se determina si el NHL2 es conocido. Si es así, el bucle se desplaza a la etapa S5A. Si no es así, el proceso lleva a cabo la etapa S5B para resolver el NHL2 mediante una consulta ARP en el dispositivo de enfoque o el dispositivo de NHL3. La generación de una consulta para correlacionar una dirección de capa 3 con una dirección de capa 2 se discute con más detalle más adelante con referencia a la figura 7. En resumen, para el dispositivo que se está consultando, se obtiene una lista de índices de interfaz (ifIndex) a partir de la topología de red o si se camina ifIndex desde la tabla de interfaz del propio dispositivo. Cada ifIndex para el dispositivo se combina con la dirección del NHL 3 para generar un conjunto de claves para incluir en la consulta al dispositivo. Por lo tanto, una consulta que contiene estas claves se formula y se transmite al dispositivo de enfoque. El dispositivo de enfoque produce cero o una respuesta satisfactoria.

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

Si NHL3 no está en el dispositivo actual, en la etapa S6, el proceso envía una consulta para encontrar la entrada FDB de la capa 2 para el NHL2 para obtener el puerto de salida. La generación de una consulta en la capa 2 se discute más adelante. Si tiene éxito, se agrega el puerto de salida al registro de ruta (S6A), se utiliza la topología en caché 3 para encontrar el puerto y el dispositivo al final del enlace (S7), se añade el dispositivo a la ruta (S7A), y el dispositivo de enfoque se ajusta al dispositivo que acaba de estar situado en el extremo del enlace (L2 HOP). Las etapas S6A, S7 y S7A se pueden denominar L2 HOP. En este punto, consulte la figura 2b. En la etapa S5A, el dispositivo de enfoque es el dispositivo A. Este recibe una consulta para encontrar la entrada FDB de la capa 2 y devuelve el puerto de salida. El dispositivo que se determina que está al final de ese enlace es el dispositivo B (figura 2c) que recibe una consulta con el NHL3 todavía fijado a la dirección IP de destino.

Si no se encontró una entrada FDB de la capa 2, o si en S5A se determinó que el NHL3 estaba alojado en el dispositivo de enfoque, en la etapa S8 se realiza una consulta de ruta para determinar si la ruta L3 se encuentra en el dispositivo de enfoque a la dirección IP de destino. La consulta de ruta puede ser una ruta única o una consulta de ruta recursiva. Esto establece un IP de salto siguiente y una interfaz de salida. Si no se encuentra la ruta L3, se indica una ruta roto y el proceso se detiene - S8A. En la etapa S9 (L3 HOP) se añade la interfaz de salida de la tabla de enrutamiento a la ruta, el NHL3 se establece en la nueva dirección IP de salto siguiente y el proceso consulta el dispositivo para determinar la dirección de la capa 2 del NHL3. Si el NHL2 no puede ser resuelto, el NHL2 se establece en “desconocido”.

En la etapa S10, la dirección actual del NHL3 se compara con la dirección IP de destino. Si el NHL3 no es la IP de destino (es decir que el algoritmo de identificación de ruta todavía no está en el segmento L2 final), en la etapa S11 la topología en caché se utiliza para encontrar el puerto y dispositivo al final del enlace, el dispositivo se añade a la ruta y el enfoque se establece en este dispositivo. El proceso entonces consulta (S12) si el dispositivo de enfoque es el dispositivo terminal. Si el dispositivo de enfoque no es el dispositivo terminal, el proceso vuelve a la etapa S5, pero utilizando el NHL3 y el NHL2 establecidos en la etapa 9.

Terminación

El algoritmo finaliza cuando se alcanza el dispositivo terminal y se añaden el puerto de terminal y el servidor de destino a la ruta. Otras condiciones de terminación impiden que el algoritmo haga un bucle indefinidamente. En cada iteración de la ruta, una iteración comienza colocando un indicador cambiado a falso y un indicador dirigido a falso. Cuando se produce un salto L2 (S7), el indicador cambiado se establece en verdadero; cuando se produce un salto L3 (S9), el indicador dirigido se establece en 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 utiliza para encontrar el dispositivo conectado y el puerto de entrada del dispositivo conectado. Para cada iteración, se almacena la combinación de: “Dispositivo de enfoque NHL2, NHL3”.

Si el dispositivo de enfoque del NHL2 o NHL3 ha cambiado y se ha visto la nueva combinación de “dispositivo de enfoque de NHL2, NHL3”, se produce un evento de bucle detectado y se detiene el bucle. Si no se ha alcanzado el límite de bucle y se ha producido el enrutamiento o la conmutación (es decir, si los indicadores dirigidos o conmutados son verdaderos) y el dispositivo de enfoque no es igual al dispositivo terminal, itera de nuevo. Cada vez se evalúa si se ha alcanzado el límite de bucle de iteración. Si lo ha hecho, el algoritmo termina.

Cuando la iteración cesa, 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 detenido, se informa de un error cuando el algoritmo de búsqueda de ruta se habrá terminado en una ubicación inesperada. Si el dispositivo terminal es un conmutador de acceso, el puerto de salida del conmutador de acceso se añade desde el “destino localizado” (S4) a la ruta y el dispositivo de destino derivado del puerto de salida del conmutador de acceso se añade a la ruta. Si el dispositivo terminal es igual al dispositivo de destino, el algoritmo termina. El detalle del algoritmo será discutido ahora con más detalle.

Ejemplo Específico

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

Observando nuevamente la red original de la figura 1, se puede ver que la primera parte del registro de ruta ilustrado en la figura 4 se deriva de la red de la figura 1, en la que se han utilizado letras correspondientes para designar los dispositivos seleccionados por el conmutador anterior o dispositivo de enrutamiento. Cuando el algoritmo de identificación de ruta funcionaba, el dispositivo de enrutamiento B había determinado enviar el paquete al conmutador C. Sin embargo, sin el uso de la presente invención, hubiera sido extremadamente difícil trabajar en tiempo real. El dispositivo de enrutamiento B tenía similarmente una opción para encaminar el paquete al enrutador F en la red central. Consultando el dispositivo de enrutamiento B en tiempo real (o más o menos tiempo real), basado en el paquete hipotético dirigido al destino Y, el dispositivo de enrutamiento B devuelve la decisión que habría tomado si hubiera llegado un paquete real con esa dirección. Comprobando que el dispositivo de enrutamiento B despachará el paquete al conmutador C y estableciendo entonces que el conmutador C ha conectado en el extremo lejano de su dispositivo de enrutamiento de puertos de salida D, C y D se han añadido al registro de ruta 81. De esta manera, el algoritmo de identificación de paquetes ha pasado por la ruta que el paquete hipotético habría tomado en el momento en que el algoritmo de identificación de ruta consulta los dispositivos en la red. La caja adyacente al dispositivo de enrutamiento D indica los ajustes para el NHL3 y el NHL2, es decir, el NHL3 se ajusta a la dirección IP del dispositivo E que se ha establecido como el dispositivo extremo distante para el dispositivo de enrutamiento D basado en la tabla de enrutamiento actualmente activa en D, el NHL2 se ha establecido como la dirección MAC para el dispositivo E consultando el dispositivo D para su entrada ARP para el

dispositivo E.

Topología de la red

Como se mencionó anteriormente, la topología de red incluye tanto la interconectividad del dispositivo de red como la ubicación del host final. La topología de red 3 puede ser proporcionada por un servidor de topología que proporciona detalles de las conexiones puerto a puerto. De este modo, cuando se identifica un puerto de salida en un dispositivo, el puerto de entrada del dispositivo conectado se puede determinar utilizando una conexión puerto a puerto identificada en la topología. Tanto los puertos de salida como los de entrada pueden agregarse al registro de ruta. El servidor de topología también proporciona una CAM global, una ARP global y credenciales de dispositivo. Además, para cada dispositivo registrado en la topología es preferible una lista de índice de interfaz (IfIndex) y una lista VLAN (red de área local virtual). Los dispositivos VLAN no se han discutido todavía. Se discuten más adelante en este documento. Cuando se devuelve una respuesta al ordenador monitor 16, el ordenador de monitorización consulta la topología 3 en el orden siguiente cuando se manejan respuestas de capa 2. En este contexto, una respuesta de capa 2 es una respuesta que ha identificado un puerto de salida desde un dispositivo de conmutación de capa 2. El orden de consulta es CDP, LLDP, STP y SONMP, IPv6 ND.

Localizar el dispositivo de origen

Como se mencionó anteriormente, la ubicación del primer dispositivo en la ruta (el dispositivo conectado al dispositivo de origen) no es necesariamente sencillo. En una realización, el ordenador monitor 16 implementa el algoritmo para intentar en primer lugar encontrar la fuente como un host conectado y si falla trata de encontrar la fuente como un dispositivo de red. Al intentar encontrar la fuente como un host conectado, consulta el dispositivo de origen para la dirección de la capa 2 (MAC) para la IP de origen. Esto se puede realizar de la misma manera que la consulta en un dispositivo de enfoque como se ha descrito anteriormente en la etapa S5B. Es decir, el proceso envía una consulta para encontrar la entrada ARP para la dirección IP de origen.

Si no hay una dirección de capa 2 del dispositivo de origen, se consultará la tabla ARP en caché global del servidor de topología. En la realización descrita, se hace referencia a éstas como tablas ARP, pero pueden utilizarse cualesquiera tablas que asignen las direcciones de la capa 3 a la capa 2. Si se encuentra una dirección MAC que corresponde a la dirección IP de origen, se consulta el servidor de topología para la ubicación MAC de la IP de origen consultando las tablas de reenvío de capa 2 en caché global en el servidor de topología para encontrar puertos que han visto tráfico desde esta dirección MAC. Se espera que el servidor de topología devuelva una ubicación MAC de origen única eliminando varias coincidencias (el MAC de origen visto en muchos puertos), filtrando puertos marcados como troncos, puertos con un número excesivo de MAC (las entradas FDB de los puertos de conmutador de acceso suelen tener una única 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 encontrar la fuente como un host conectado, se intentan encontrar la fuente como un dispositivo de red. Esto se puede conseguir consultando el servidor de topología para todas las direcciones IP encontradas en todos los dispositivos de red administrados para ver si la dirección IP está en un dispositivo de red. Si lo es, ese dispositivo de red se establece como el dispositivo de enfoque.

Localizar el dispositivo de destino

Consideraciones similares se aplican a la ubicación del dispositivo de destino. En primer lugar, se intentan encontrar el dispositivo de destino como un host conectado y, si falla, se intenta encontrar el destino como un dispositivo de red. Para buscar el dispositivo de destino como un host conectado, se consulta el dispositivo de destino para su dirección de capa 2 o se consultan en el servidor de topología las capas de asignación global de capa en caché 3 a la capa 2 (similarmente al dispositivo de origen discutido anteriormente). A continuación, se consultan las tablas de reenvío de la capa 2 en caché global en el servidor de topología para encontrar puertos que han visto el tráfico de este MAC (de nuevo, como se ha descrito anteriormente con referencia a la ubicación del dispositivo de origen).

Para encontrar el destino como un dispositivo de red si el anterior falla, se puede consultar el servidor de topología para todas las direcciones IP encontradas en todos los dispositivos administrados para ver si la dirección IP está en un dispositivo de red. El dispositivo de red puede configurarse como el dispositivo terminal.

Utilidad por salto

Con el fin de implementar el algoritmo de identificación de ruta, el ordenador monitor 16 ejecuta un programa informático tal como se ha discutido. Este programa de computadora proporciona una utilidad que maneja consultas “por salto”. Es decir, el algoritmo de identificación se basa en enviar una consulta desde el ordenador monitor a un dispositivo de enfoque y recibir desde el dispositivo de enfoque un puerto de salida que puede usarse para acceder a la topología. Esto no puede lograrse necesariamente mediante 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 de origen para el NHL3 y el puerto de salida, utilizando la dirección IP de destino. Si no se encuentra ninguna ruta, se 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 de origen para el NHL3). Si no se encuentra ninguna ruta, se pregunta al dispositivo de origen por la puerta de enlace predeterminada para el NHL3. Si no se encuentra ninguna ruta, se pregunta al primer dispositivo por una puerta de enlace predeterminada. Para consultar una tabla de enrutamiento para encontrar el NHL3 (como se ha descrito anteriormente), se encuentra una ruta para la dirección IP en cuestión (la dirección IP buscada) consultando el dispositivo de enrutamiento utilizando una técnica de clave especulativa discutida más adelante. Si se encuentra la ruta, pero no se especifica ningún puerto de salida, se devuelve la siguiente dirección IP de salto y se utiliza como el NHL3. Si la ruta se encuentra con una interfaz de salida ifIndex mayor que cero, el puerto de salida se devuelve con la dirección del NHL3 y el puerto de salida se agrega a la ruta. Si la ruta se encuentra con la interfaz de salida ifIndex igual a cero, la utilidad se reitera colocando la IP buscada en el siguiente IP de salto (de la consulta anterior) y encontrando la ruta para la IP buscada consultando el dispositivo usando la clave especulativa discutida después). Esto se repite hasta que ifIndex devuelto no es cero.

La etapa de encontrar la ruta para la IP buscada utiliza la técnica de clave especulativa para devolver una entrada de ruta. Si se encuentra la entrada de ruta, la utilidad encuesta para la siguiente dirección de salto de ipRouteNextHop.NetworkAddress. La utilidad también realiza sondeos para la interfaz de salida de ipRouteIfIndex.NetworkAddress y encuesta para ipRouteType.NetworkAddress. Si ipRouteType es “directo”, la IP buscado se establece en el salto siguiente, ya que un tipo de ruta IP de directo indica que está directamente conectado 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 utilizan varias rutas, por ejemplo, cuando un dispositivo es responsable del tráfico de equilibrio de carga. Si se utiliza activamente una única ruta, se debe determinar la ruta activa. Si se están utilizando varias rutas, la ruta podría dividirse en este punto y el registro de ruta podría contener los resultados del algoritmo del buscador de rutas aplicado a cada ruta encontrada a partir de este punto. En muchos casos, las múltiples opciones de enrutamiento en un dispositivo son indicativas de un dispositivo que está enrutamiento inteligente basado en varias métricas. Estas métricas también pueden ser consultadas y devueltas para su grabación en el ordenador monitor.

La utilidad también es responsable de encontrar el salto siguiente inicial en la capa 2 consultando la capa 3 a la tabla de asignación 91 de la capa 2 en el dispositivo de enfoque. Si no se encuentra la dirección de la capa 2, donde el dispositivo de enfoque es el dispositivo de origen, la utilidad consulta el conmutador de acceso (si es un conmutador de capa 3, debe proporcionar una capa 3 a la capa 2). Si no se encuentra la dirección de la capa 2, la utilidad consultará las tablas ARP en caché globales del servidor de topología 3. Una consulta para una dirección de capa 2 en un dispositivo se lleva a cabo como se ha explicado anteriormente con referencia a la etapa S5B.

Si la dirección NHL 3 no está en el dispositivo de enfoque, la utilidad sondea el dispositivo de enfoque para un puerto de salida para la dirección 2 de la capa 2 NHL2. La etapa de sondear el dispositivo de enfoque para el puerto de salida NHL2 incluye la consulta específica VLAN (red de área local virtual). Es decir, incluye la etapa de establecer en qué VLANs participa el dispositivo de acuerdo con la topología 3 y como se registran en el dispositivo. Estas VLAN se utilizan para ayudar a encontrar entradas de la tabla de reenvío para VLANs específicas (los FDBs se dividen a menudo según la VLAN con la que están relacionados); por ejemplo, para el protocolo de árbol de expansión por VLAN (PVSTP) es necesario realizar las consultas FDB en el contexto de cada una VLAN para intentar encontrar una coincidencia).

Si el puerto de salida no se encuentra en el FDB de la capa 2 (utilizando una VLAN específica o la VLAN nativa), la utilidad intentará encontrar qué interfaz se dirige hacia el NHL2 desde los registros ARP mediante encuesta para ipNetToMediaPhysAddress 71 (figura 7). Es decir, la utilidad intenta aprender de qué interfaz se ha aprendido la relación entre la capa 2 y la capa 3.

Una vez que la utilidad ha encontrado un puerto de salida usando la dirección de capa 2, agrega el puerto de salida al registro de ruta y utiliza el servidor de topología 3 para encontrar el puerto remoto conectado 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 encuentra ningún puerto remoto o el nombre del puerto de salida ordena el uso de puertos de capa superior o inferior, la utilidad comprueba si hay puertos de capa inferior o puertos de capa superior. Es decir, puede haber un escenario donde hay una asignación de salidas de ruta virtual a puertos físicos. Para que el algoritmo de identificación de ruta tenga éxito, necesita identificar un puerto de salida físico para acceder al servidor de topología. En un escenario en el que la comprobación de puertos de capa inferior revela la presencia de puertos de capa inferior, estos puertos de capa inferior se pueden utilizar como puertos de salida y se accede al servidor de topología para encontrar 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 independientes, cada una de las cuales se rastrea

independientemente desde este punto en adelante.

Si se identifican puertos de capa superior, se utiliza el puerto de capa superior para el puerto de salida. El servidor de topología se utiliza para encontrar el puerto remoto conectado a este puerto de salida de capa superior.

Siguiente salto

Ajustar los indicadores dirigidos y conmutados a falso. Utilizando el servidor de topología o las consultas directas al dispositivo de enfoque, comprobar si el dispositivo de enfoque aloja o no la dirección IP del NHL3 en cualquiera de sus puertos. Si aloja la dirección IP del NHL3, la utilidad pasa a consultar la tabla de enrutamiento del dispositivo de enfoque para las rutas al IP de destino mediante la técnica de clave especulativa. Si la utilidad localiza una ruta candidata, la siguiente dirección de la capa 2 NHL2 se ajusta mediante la consulta del dispositivo de enfoque (o las tablas ARP globales en caché) para la capa 3 a la capa 2 y el indicador dirigido se establece en verdadero. Si el NHL3 es igual al IP de destino, entonces eso indica que la utilidad ha alcanzado el último dispositivo de capa 3 más cercano al destino, por lo que no hay necesidad de mover este dispositivo todavía, ya que el siguiente salto sería un salto de capa 2. Por lo tanto, la utilidad agrega los puertos de salida de la ruta del candidato a la ruta. Si el NHL3 no es igual al IP de destino, esto indica que no está en el segmento final de la capa 2 y que el puerto de salida de la ruta candidata se agrega a la ruta.

Si no se produjo ningún enrutamiento durante esta iteración (el indicador dirigido todavía se ajusta en falso), la utilidad sondea el dispositivo de enfoque para un puerto de salida para la dirección 2 de la capa 2 del NHL2. La etapa de sondeo del dispositivo de enfoque para el puerto de salida del NHL2 incluye la interrogación específica de VLAN (Red de área local virtual) (como se ha descrito anteriormente). Si el puerto de salida no se encuentra en el FDB de la capa 2 (utilizando una VLAN específica o la VLAN nativa), la utilidad intentará encontrar qué interfaz se dirige hacia el NHL2 desde los registros ARP mediante encuestado para ipNetToMediaPhysAddress 71. Es decir, la utilidad intenta aprender de qué interfaz se ha aprendido la relación entre la capa 2 y la capa 3. Una vez que la utilidad ha encontrado un puerto de salida usando la dirección de capa 2, agrega el puerto de salida al registro de ruta y utiliza el servidor de topología 3 para encontrar el puerto remoto conectado al puerto de salida. Este puerto remoto se registra como el puerto de entrada en el dispositivo siguiente. Si se encuentra un puerto de salida utilizando consultas FDB o consultas ARP, el indicador cambiado se establece en verdadero.

Si, al consultar el servidor de topología, no se encuentra ningún puerto remoto o si el nombre del puerto de salida ordena el uso de puertos de capa superior o inferior, se realiza una comprobación para puertos de capa inferior o superior como se describe anteriormente. Si se encuentra un puerto de salida, se agrega a la ruta, el dispositivo que contiene el puerto se agrega a la ruta y el dispositivo de enfoque se establece en el dispositivo remoto.

Esta etapa de “salto siguiente” se repite hasta que se alcanza un límite prescrito en el número de iteraciones o la ruta llega a su fin (es decir, no se han producido conmutaciones 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 agrega desde “localizar destino” al registro de ruta y el dispositivo de destino se añade al registro de ruta. Si el dispositivo terminal es el dispositivo de destino en sí, la utilidad termina.

Las figuras 11A a 11D muestran un diagrama de flujo del funcionamiento de la utilidad ejecutada en el ordenador monitor.

Equilibrador de carga

Como se mencionó 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 asignación de la IP virtual al grupo de servidores para el equilibrador de carga. Esto permite que se identifique el servidor físico asignado al equilibrador de carga. La ruta se conserva hasta la ruta “raíz” (hasta el dispositivo de equilibrio de carga). A continuación, para cada dirección IP del servidor físico, se ejecuta una utilidad de detección de ruta adicional desde el equilibrador de carga a la dirección IP del servidor físico, con cada ruta adicional pendiente previamente con la ruta “raíz”.

Consulta de tabla de enrutamiento

Uno de los factores que hacen que el algoritmo de ruta sea particularmente eficiente es la capacidad de generar una consulta a un dispositivo de enrutamiento de manera eficiente, es decir, generar una consulta a la que el dispositivo de enrutamiento puede responder en un corto período de tiempo sin una sobrecarga significativa. La figura 5 ilustra la estructura de una tabla de rutas lineales direccionable a través de SNMP. Para establecer una ruta a un destino determinado, ipRouteDest es el índice requerido en la tabla de rutas. Esto se denomina 48 en la figura 5. Las entradas de interés en la tabla son ipRoutelfindex 50 que define la interfaz de salida, ipRouteNextHop 52, que define la dirección IP del siguiente salto (IP del siguiente salto) e ipRouteType 54 que define el tipo de entrada de enrutamiento (inválido/directo/indirecto). El acceso a la tabla normalmente requiere el conocimiento del ipRouteMask

56: esto permitiría localizar una dirección IP de red específica. Sin embargo, como se puede ver en la figura 5, el propio IpRouteMask está incrustado en ipRouteEntry y por lo tanto no se sabe que se configure en la consulta. Lo que se requiere es encontrar una coincidencia para:
<IP de interés> & <ipRouteMask.X> == <ipRouteDest.Z>

para encontrar la clave IpRouteDest 48 que representa el índice a la tabla.

Como observado por los inventores, solo hay 33 posibilidades para el IpRouteMask (/32.../0), que es 255.255.255.255, 255.255.255.254, 255.255.255.252, 0.0.0.0. Algunos de estos producen ID de red duplicadas para la misma dirección IP, debido al número de ceros en la dirección IP. La figura 6 muestra 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). Por lo tanto, ahora solo es necesario hacer 12 consultas SNMP (que se pueden presentar en un solo paquete de consulta) para encontrar la ruta. Es decir, los 12 resultados se emparejan en la tabla de ruta del dispositivo de enfoque y cuando se encuentra una coincidencia los elementos requeridos ipRouteIfIndex, ipRouteNextHop y ipRouteType se recuperan y devuelven en una respuesta al ordenador monitor 16.

La reducción en el número de consultas requeridas para encontrar la ruta se denomina en este documento “codificación especulativa” y permite la realización de consultas en tiempo real de la tabla de rutas de una manera muy eficiente.

Cuando se examinan tablas de enrutamiento reales, no es infrecuente que la ruta encontrada para una dirección IP dada no tenga una interfaz de salida válida y solo proporcione una dirección de salto siguiente. En estos casos, la siguiente dirección de salto se utiliza para una consulta posterior de la tabla de enrutamiento para tratar de obtener una interfaz de salida para esa siguiente dirección de salto. Esta reutilización de la siguiente dirección de salto se repite hasta que se obtiene una interfaz de salida.

De acuerdo con este enfoque, en una primera etapa, una consulta de ruta única de búsqueda utiliza una clave especulativa para encontrar una entrada de enrutamiento para la dirección IP especificada (IP_X) como se acaba de esbozar. Si el ipRouteType asociado es “directo”, IP_X (e $ipRouteIfIndex_X$) se devuelven en una respuesta al ordenador monitor como el salto siguiente. Es decir, está directamente conectado y por lo tanto no tiene capa 3 en el salto siguiente.

Si el ipRouteType asociado no es directo, se devuelven ipRouteNextHop e ipRouteIfIndex en respuesta al equipo del monitor.

En la expansión FindRecursiveRoute, la etapa FindSingleRoute se toma para la dirección IP requerida (IP_X). Si no se encuentra ninguna ruta, se devuelve un fallo. Si se encuentra una ruta, pero no hay una interfaz de salida, se devuelve ipRouteNextHop. Si se encuentra la ruta y ipRouteIfIndex es mayor que cero, se devuelven ipRouteNextHop e ipRouteIfIndex. Si se encuentra la ruta y el ipRouteIfIndex es igual a cero, se toma una etapa FindRecursiveRoute posterior para la dirección IP de ipRouteNextHop, con los mismos cuatro resultados posibles.

Mientras que la clave especulativa es una técnica particularmente buena para la consulta eficiente de grandes conjuntos de datos, su principal aplicación es cuando se consultan datos que están indexados con una clave derivada para la cual ya se conoce una clave parcial. Es por eso que es particularmente útil en el contexto del análisis de la tabla de rutas SNMP y la consulta de tablas ARP de SNMP. Sin embargo, también se puede determinar el comportamiento rápido de reenvío de dispositivos de red mediante otras técnicas de consulta, como el acceso a la CLI y la API XML.

Consulta de ARP

Se hará referencia a continuación a la figura 7 para describir una técnica eficiente para consultar una tabla ARP usando la manipulación especulativa. La generación de una consulta se analiza más detalladamente más adelante con referencia a la figura 7. Para el dispositivo que se está consultando, se obtiene una lista de índices de interfaz (IfIndex) a partir de la topología de red, o bien se puede recorrer IfIndex desde el propio dispositivo. Cada ifIndex para el dispositivo se combina con la dirección del NHL 3 para generar un conjunto de claves para incluir en la consulta al dispositivo. Por lo tanto, una consulta que contiene estas claves se formula y se transmite al dispositivo de enfoque. El dispositivo de enfoque produce cero o una respuesta satisfactoria. La figura 7 ilustra un formato de tabla ipNetToMediaEntry que, en principio, permitiría determinar la dirección MAC para cualquier dirección IP dada. Dado que no se puede encontrar una entrada única para una dirección IP específica, a menos que se sepa de qué interfaz se ha aprendido la entrada ARP, se utiliza la codificación especulativa combinando la dirección IP con todos y cada uno de los ifIndex del dispositivo. Es decir, cada clave de consulta se puede crear combinando la dirección IP con un ifIndex. De esta manera, el número de consultas SNMP es el número de interfaces en el dispositivo que es típicamente mucho menos que el número de entradas ARP en el dispositivo y por lo tanto es significativamente más eficiente.

En clave especulativa, varias claves de consulta pueden estar contenidas en un único mensaje de consulta.

Tecnologías/Protocolos adicionales

El algoritmo de identificación de ruta cuando se utiliza anteriormente proporciona una manera eficaz de identificar una ruta particular que es probable que un paquete o mensaje particular adopten a través de la red de dispositivos interconectados que operan de acuerdo con protocolos de red generalmente conocidos. Las situaciones surgen donde por una razón u otra, el algoritmo de identificación de ruta cumple con un desafío particular. Algunos de estos desafíos se analizan a continuación.

En algunos casos, la utilidad ejecutada en el algoritmo tiene que recorrer un segmento de red MPLS (conmutación por etiquetas multiprotocolo). Esto lo logra encontrando la asignación inicial de etiquetas (en el punto en que el tráfico entra en el segmento MPLS) y el seguimiento a través de la red MPLS saltando usando los detalles de salto por salto de etiqueta haciendo emerger, empujando y reenviando hasta que el tráfico tenga su etiqueta final emergida y deje el segmento MPLS.

Otro desafío es atravesar los límites de NAT que se pueden lograr encuestando tablas NAT del dispositivo NAT. Esto puede requerir sondeo especulativo en tiempo real para un NAT dinámico, pero puede ser posible usar el sondeo de fondo para un NAT estático.

Para los protocolos de túnel como IPSEC/GRE/SSL, etc., la utilidad comprueba una ruta directa de un extremo del túnel a otro (típicamente con un salto de capa desconocida 3 que representa todos los nodos intermedios). La utilidad comprueba además la información topológica específica del protocolo y comprueba en las tablas/interfaces de enrutamiento la presencia de saltos de encriptado/túnel.

Otro desafío es la virtualización. Es importante que el algoritmo identifique los puertos físicos de salida de modo que se pueda acceder a un dispositivo físico conectado al puerto de salida desde la topología. Muchas redes operan en diferentes capas de virtualización. Los conmutadores virtuales se pueden consultar usando API adicionales y para asegurar que el servidor de topología tenga información oportuna sobre la ubicación del host final, puede ser necesario que el servidor de topología se integre con plataformas de administración de virtualización para obtener actualizaciones sobre la reubicación de máquina virtual para permitir un sondeo proactivo De la ubicación del host final en los conmutadores virtuales afectados.

La utilidad negocia las tablas de enrutamiento y reenvío virtualizadas (VRF) consultando el reenvío IP apropiado (tabla de enrutamiento) requerido para un identificador de VRF específico. En SNMP, por ejemplo, esto puede hacerse utilizando cadenas de comunidad VRF contextualizadas.

REIVINDICACIONES

1. Un método implementado por ordenador para identificar en una red de dispositivos interconectados una ruta a través de la red desde un dispositivo fuente (X) hasta un dispositivo final de destino (Y), comprendiendo la ruta una secuencia conectada de dispositivos, comprendiendo el método en un ordenador monitor 16) conectados a la red:

identificar un primer dispositivo de conmutación en la ruta, teniendo el primer dispositivo una tabla de reenvío de tráfico con entradas divididas según la VLAN con la que están relacionadas;
establecer en qué redes de área local virtuales múltiples (VLAN) está participando el primer dispositivo;
transmitir una primera consulta al primer dispositivo, incluyendo la consulta un identificador de destino en forma de una dirección de conmutación y solicitar la identificación de un puerto de salida (Po) para los mensajes dirigidos al destino identificado por el identificador de destino cuando la consulta se recibe en el primer dispositivo, en el que la primera consulta se genera para la tabla de reenvío de tráfico utilizando un sondeo específico de VLAN para consultar la tabla de reenvío de tráfico en el contexto de cada VLAN en la que participa el dispositivo;
recibir un mensaje de resultado que identifica el puerto de salida (Po) e identificar un segundo dispositivo conectado al primer dispositivo basado en una topología de red (3) accesible por el ordenador monitor (16); y
direccionar una consulta siguiente al segundo dispositivo y recibir un siguiente mensaje de resultado identificando un puerto de salida (Po) del segundo dispositivo; e identificar desde la topología de red (3) un tercer dispositivo conectado al segundo dispositivo, en el que la ruta se identifica para incluir el primer, segundo y tercer dispositivos.

2. Un método de acuerdo con la reivindicación 1, donde la dirección de conmutación (L2) se deriva de una dirección de enrutamiento para un dispositivo de enrutamiento en la red.

3. Un método de acuerdo con cualquier reivindicación anterior, que comprende registrar un conjunto de dispositivos y puertos identificados para estar en la ruta en un almacén accesible al ordenador monitor (16) como un registro de ruta (81).

4. Un método de acuerdo con cualquier reivindicación anterior, donde el registro de ruta (81) incluye puertos de entrada (Pi) identificados desde la topología de red (3) para dispositivos identificados.

5. Un método de acuerdo con cualquier reivindicación anterior, donde la consulta se transmite desde el ordenador monitor (16) al dispositivo que se está consultando en un mensaje que se transmite a través de la red.

6. Un método de acuerdo con la reivindicación 5, donde el mensaje toma la forma de al menos un paquete dirigido al dispositivo que se está consultando.

7. Un método de acuerdo con cualquier reivindicación anterior, donde la consulta incluye una pluralidad de claves para consultar el dispositivo.

8. Un método de acuerdo con la reivindicación 7, donde una sola consulta que contiene dicha pluralidad de claves hace que se devuelva una respuesta con la identificación de un puerto de salida (Po).

9. Un método de acuerdo con cualquier reivindicación anterior, donde el primer dispositivo está conectado al dispositivo fuente (X) y se identifica basándose en la topología de red (3).

10. Un método de acuerdo con la reivindicación 9, donde el primer dispositivo conectado al dispositivo fuente (X) se identifica por la localización en la topología de red (3) de un puerto de red en el que se ha visto una dirección de conmutación del dispositivo fuente (X), y a su vez, al dispositivo que forma parte de la topología de red (3) a la que pertenece este puerto de red.

11. Un método de acuerdo con cualquier reivindicación anterior utilizando la consulta por lo menos una de un conjunto de claves que se han generado combinando el identificador de destino con una pluralidad de índices incrustados de la tabla de reenvío de tráfico.

12. Un sistema informático de monitorización (16) para identificar en una red de dispositivos interconectados una ruta a través de la red desde un dispositivo fuente (X) a un dispositivo destino (Y), comprendiendo el sistema informático:

una interfaz (86) conectada a la red para transmitir consultas y recibir respuestas;
estando el sistema informático de monitorización (16) caracterizado por un procesador (80) operable para ejecutar una herramienta de identificación de ruta que está adaptada para llevar a cabo el método de cualquiera de las reivindicaciones 1 a 11;
un primer medio de almacenamiento para almacenar el registro de ruta; y
un segundo medio de almacenamiento para almacenar la topología de red (3).

13. Un producto de programa informático que comprende instrucciones legibles por ordenador que cuando son ejecutadas por un procesador (80) ejecuta las etapas del método de cualquiera de las reivindicaciones 1 a 11.

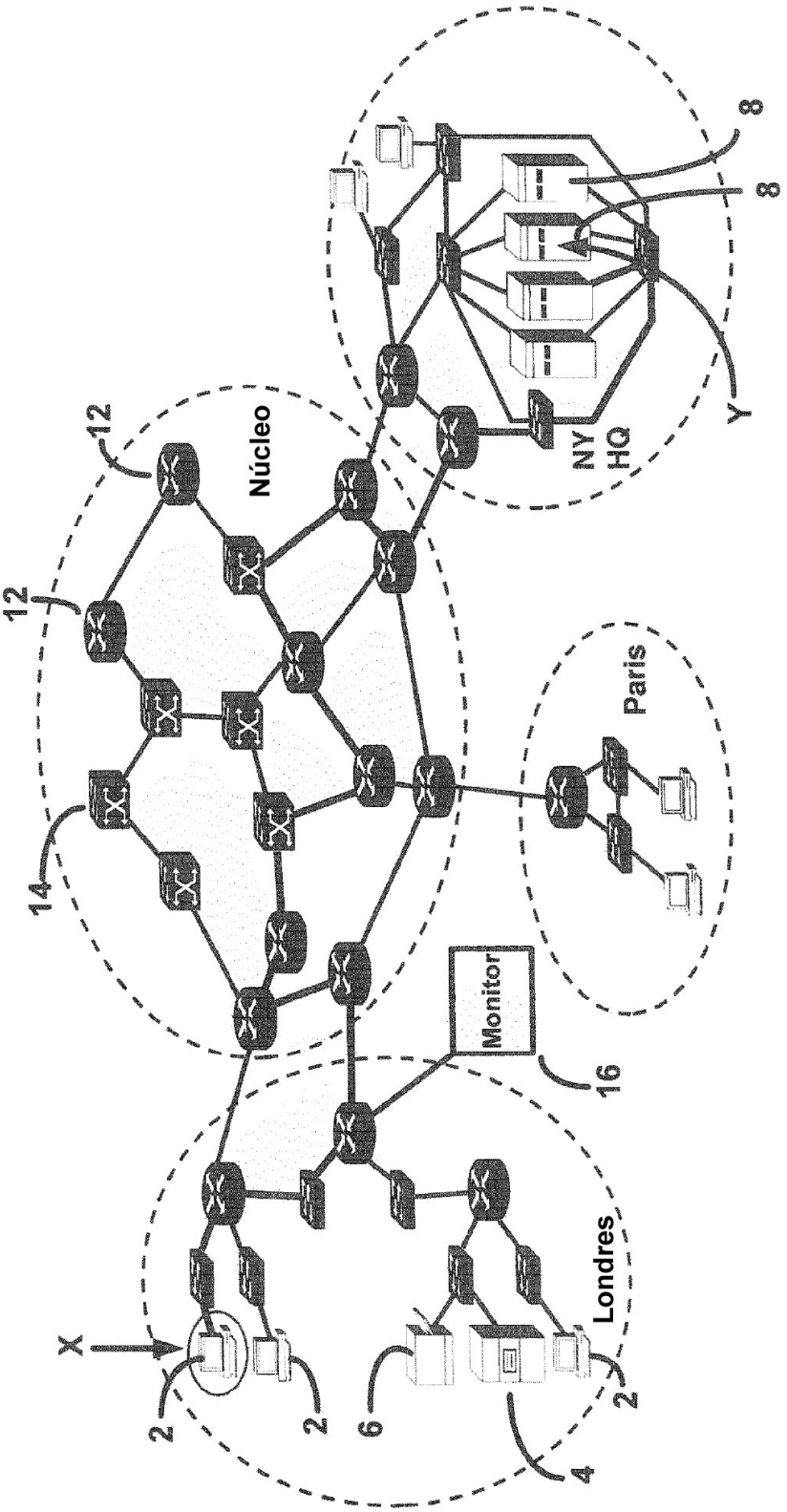


Fig. 1

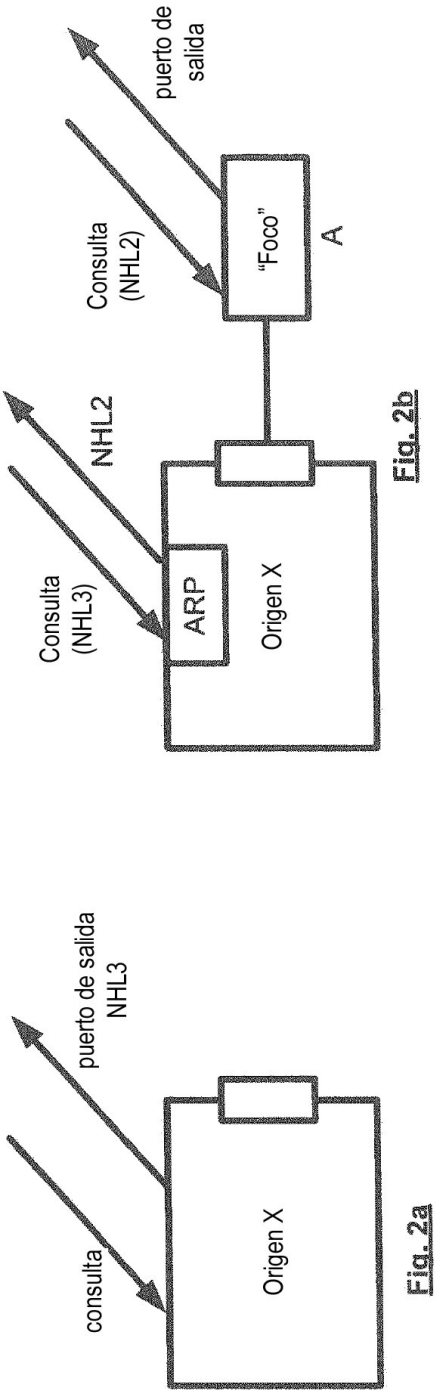


Fig. 2a

Fig. 2b

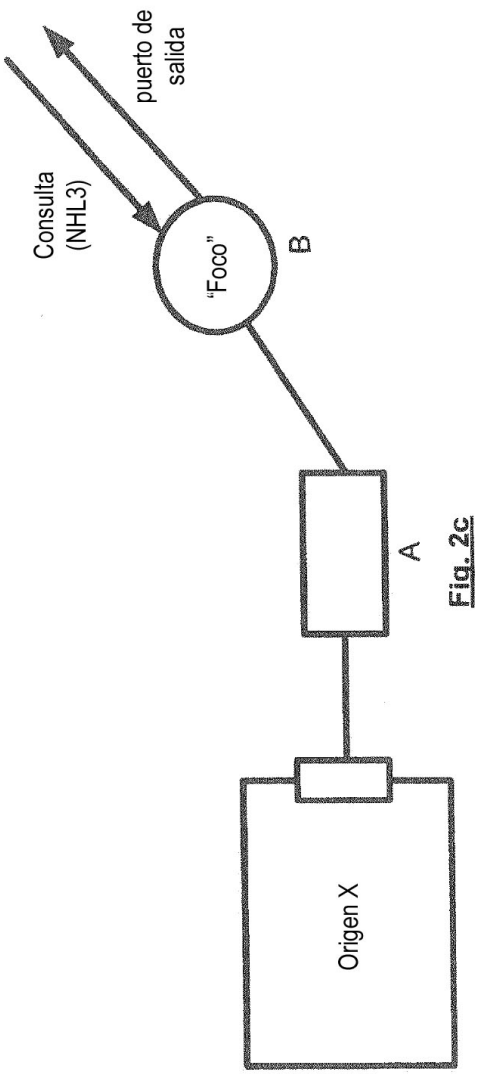


Fig. 2c

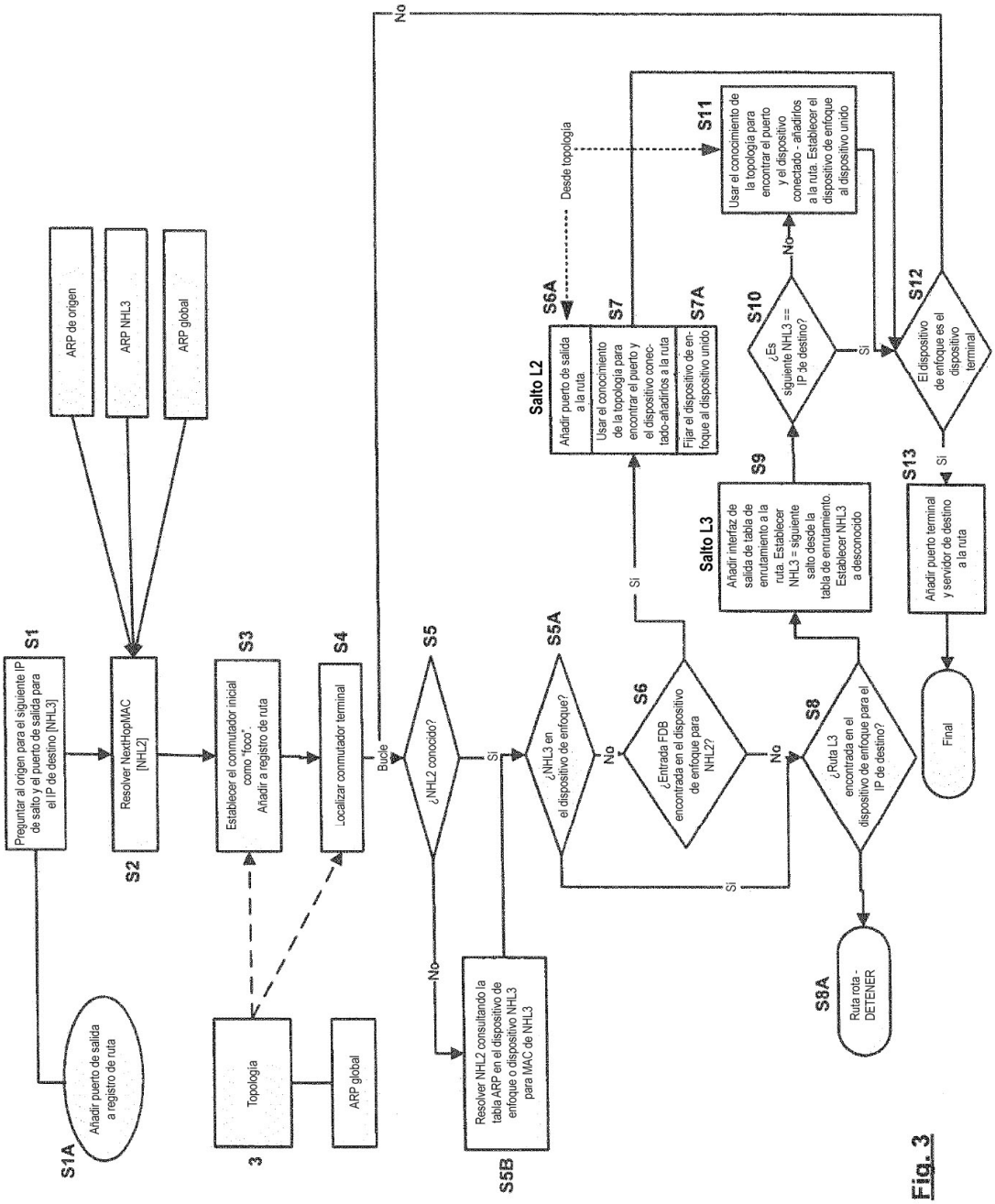

S1
Preguntar al origen para el siguiente IP de salto y el puerto de salida para el IP de destino [NHL3]
S1A
Añadir puerto de salida a registro de ruta
S2
Resolver NextHopMAC [NHL2]
ARP de origen
ARP NHL3
ARP global
S3
Establecer el conmutador inicial como 'foco'. Añadir a registro de ruta
Topología
ARP global
S4
Localizar conmutador terminal
Bucle
S5
¿NHL2 conocido?
No
Si
S5B
Resolver NHL2 consultando la tabla ARP en el dispositivo de enfoque o dispositivo NHL3 para MAC de NHL3
S5A
¿NHL3 en el dispositivo de enfoque?
No
Si
S6
¿Entrada FDB encontrada en el dispositivo de enfoque para NHL2?
No
Si
Salto L2
S6A
Desde topología
S7
Añadir puerto de salida a la ruta
Usar el conocimiento de la topología para encontrar el puerto y el dispositivo conectado-añadirlos a la ruta
S7A
Fijar el dispositivo de enfoque al dispositivo unido
S8
¿Ruta L3 encontrada en el dispositivo de enfoque para el IP de destino?
No
Si
S8A
Ruta rota - DETENER
Salto L3
S9
Añadir interfaz de salida de tabla de enrutamiento a la ruta. Establecer NHL3 = siguiente salto desde la tabla de enrutamiento. Establecer NHL3 a desconocido
S10
¿Es siguiente NHL3 == IP de destino?
No
Si
S11
Usar el conocimiento de la topología para encontrar el puerto y el dispositivo conectado - añadirlos a la ruta. Establecer el dispositivo de enfoque al dispositivo unido
S12
El dispositivo de enfoque es el dispositivo terminal
Si
S13
Añadir puerto terminal y servidor de destino a la ruta
Final

Fig. 3

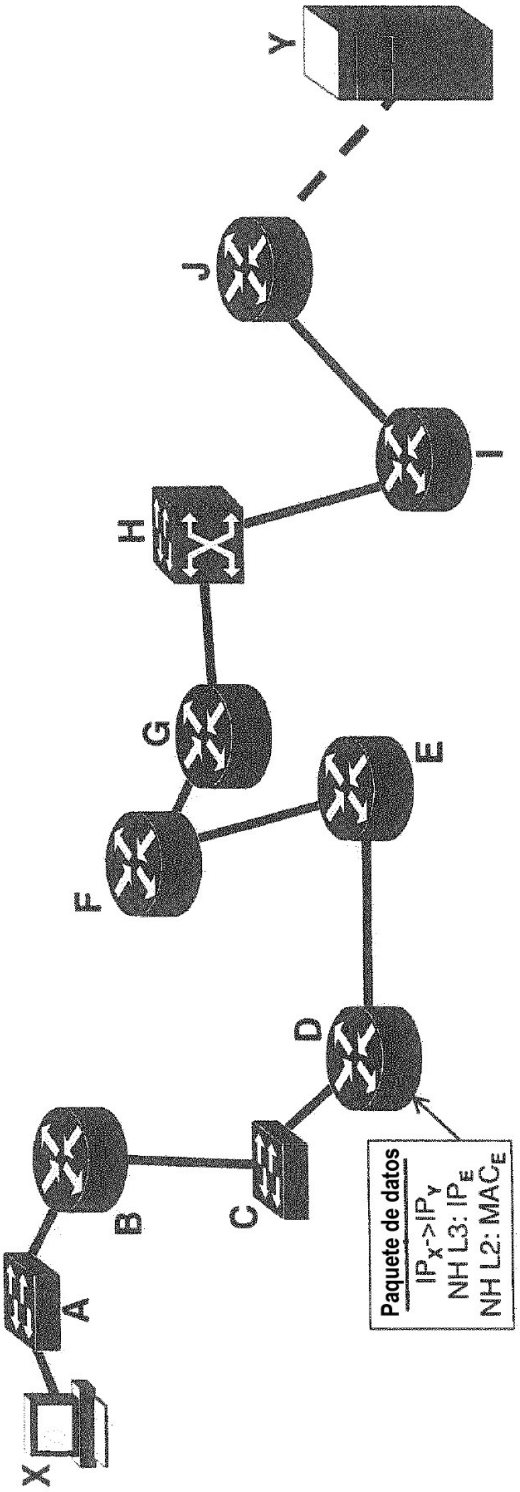


Fig. 4

ipRouteEntry TIPO DE 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 particular."

::= { **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 OBJETOS**

}

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
/31: 0000 1010 0010 1100 0000 0001 1101 0100
/30: 0000 1010 0010 1100 0000 0001 1101 0100
/29: 0000 1010 0010 1100 0000 0001 1101 0000
/28: 0000 1010 0010 1100 0000 0001 1101 0000
/27: 0000 1010 0010 1100 0000 0001 1100 0000
/26: 0000 1010 0010 1100 0000 0001 1100 0000
/25: 0000 1010 0010 1100 0000 0001 1000 0000
/24: 0000 1010 0010 1100 0000 0001 0000 0000
/23: 0000 1010 0010 1100 0000 0000 0000 0000
/22: 0000 1010 0010 1100 0000 0000 0000 0000
/21: 0000 1010 0010 1100 0000 0000 0000 0000
/20: 0000 1010 0010 1100 0000 0000 0000 0000
/19: 0000 1010 0010 1100 0000 0000 0000 0000
/18: 0000 1010 0010 1100 0000 0000 0000 0000
/17: 0000 1010 0010 1100 0000 0000 0000 0000
/16: 0000 1010 0010 1100 0000 0000 0000 0000
```

```
/15: 0000 1010 0010 1100 0000 0000 0000 0000
/14: 0000 1010 0010 1100 0000 0000 0000 0000
/13: 0000 1010 0010 1000 0000 0000 0000 0000
/12: 0000 1010 0010 0000 0000 0000 0000 0000
/11: 0000 1010 0010 0000 0000 0000 0000 0000
/10: 0000 1010 0000 0000 0000 0000 0000 0000
/09: 0000 1010 0000 0000 0000 0000 0000 0000
/08: 0000 1010 0000 0000 0000 0000 0000 0000
/07: 0000 1010 0000 0000 0000 0000 0000 0000
/06: 0000 1000 0000 0000 0000 0000 0000 0000
/05: 0000 1000 0000 0000 0000 0000 0000 0000
/04: 0000 0000 0000 0000 0000 0000 0000 0000
/03: 0000 0000 0000 0000 0000 0000 0000 0000
/02: 0000 0000 0000 0000 0000 0000 0000 0000
/01: 0000 0000 0000 0000 0000 0000 0000 0000
/00: 0000 0000 0000 0000 0000 0000 0000 0000
```

Fig. 6

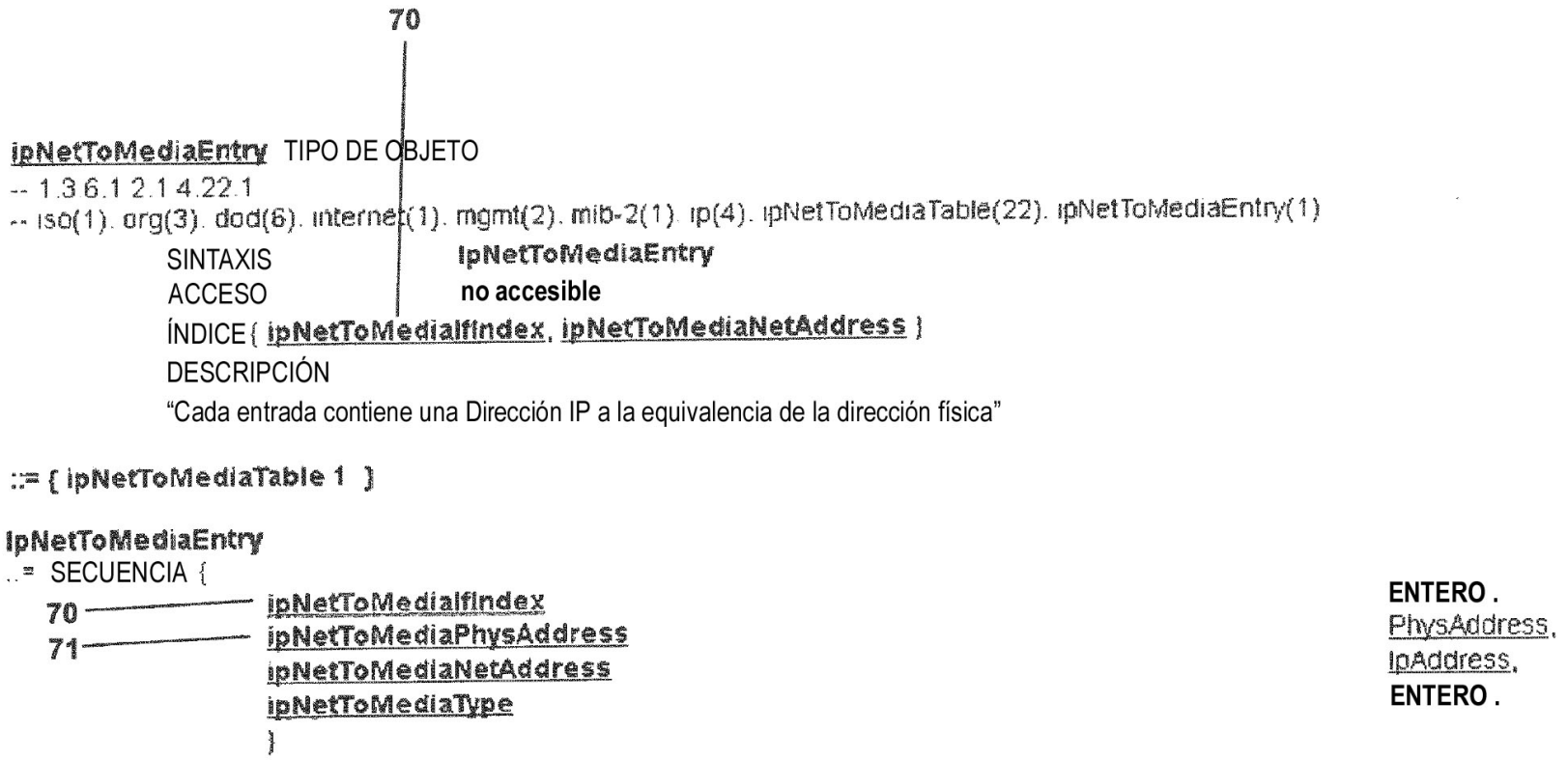

70
ipNetToMediaEntry TIPO DE OBJETO
-- 1.3.6.1.2.1.4.22.1
-- iso(1). org(3). dod(6). internet(1). mgmt(2). mib-2(1). ip(4). ipNetToMediaTable(22). ipNetToMediaEntry(1)
SINTAXIS IpNetToMediaEntry
ACCESO no accesible
ÍNDICE { ipNetToMediaIfIndex, ipNetToMediaNetAddress }
DESCRIPCIÓN
"Cada entrada contiene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

Fig. 7

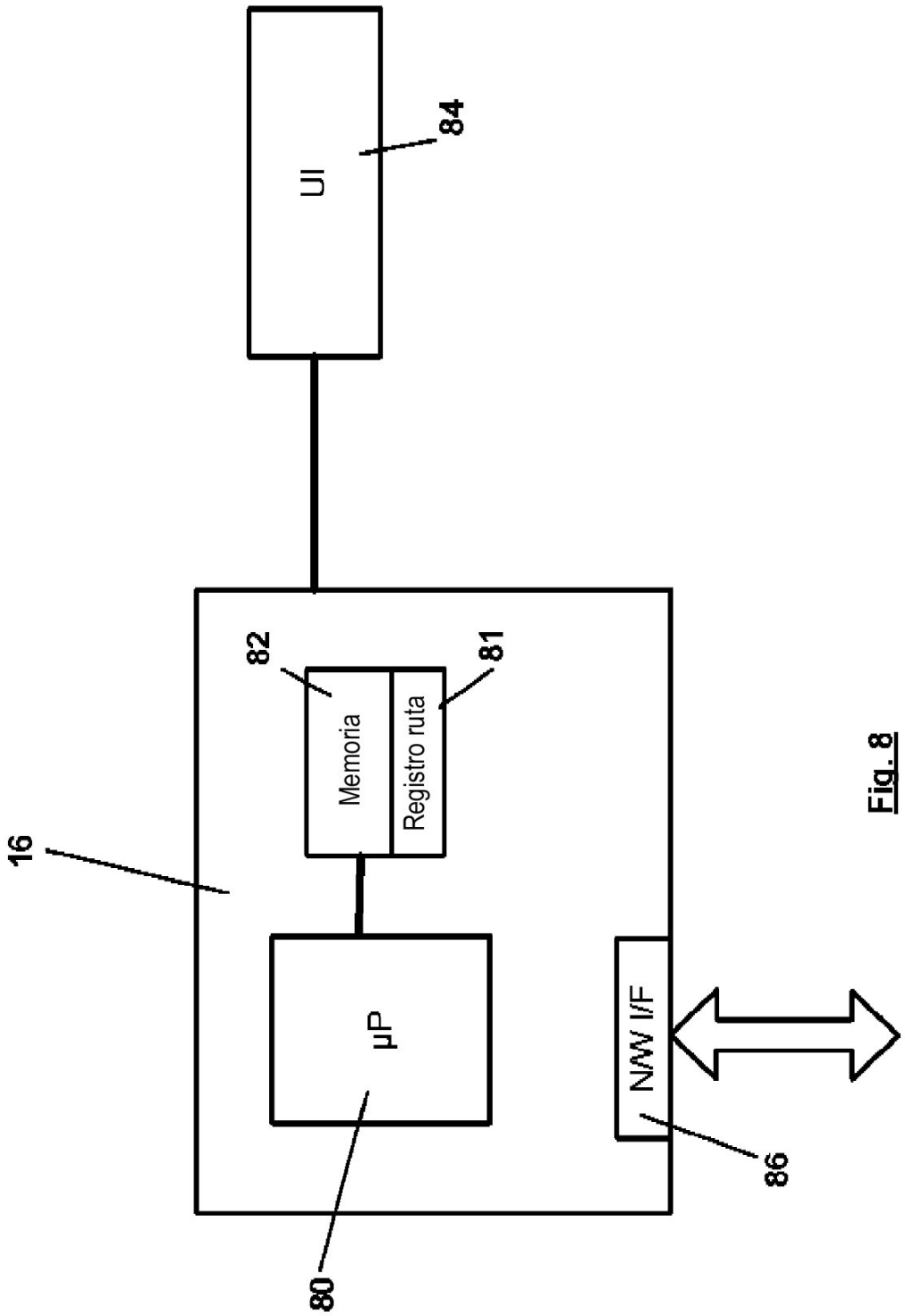


Fig. 8

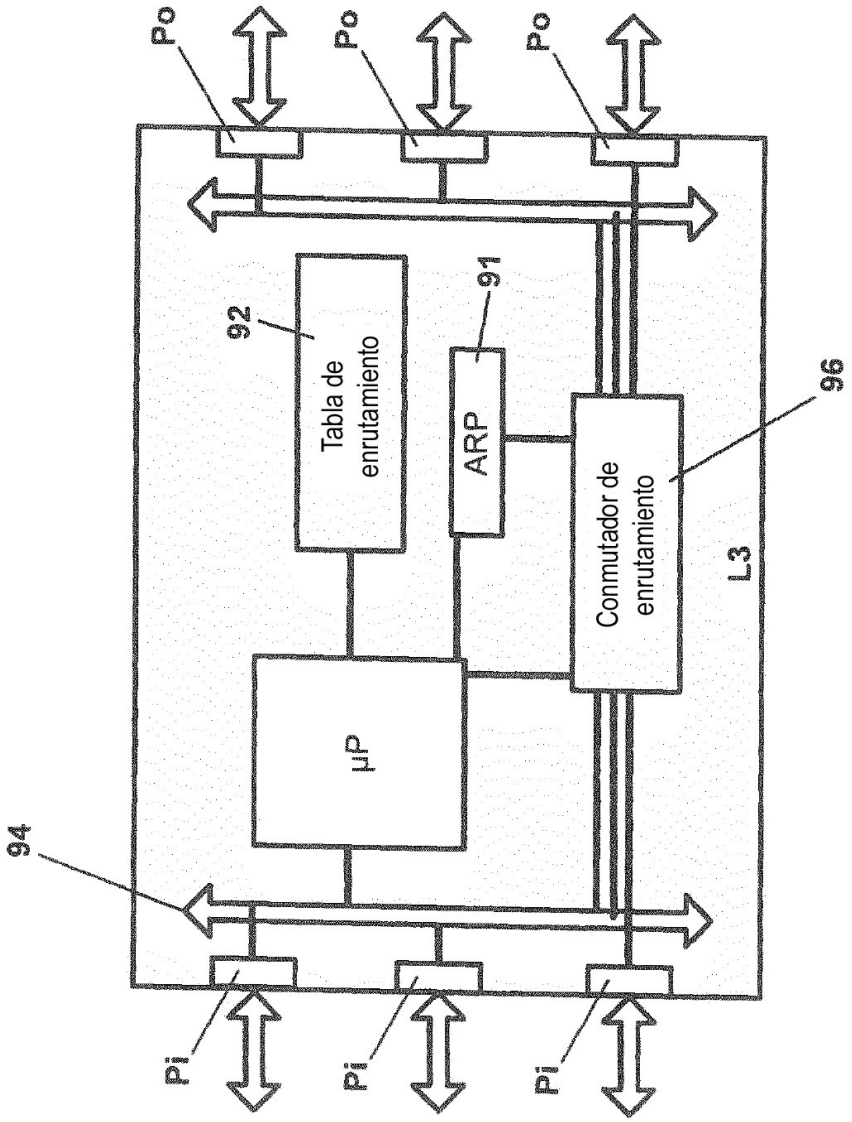

Po
Po
Po
92
Tabla de enrutamiento
91
ARP
96
Conmutador de enrutamiento
L3
μP
94
Pi
Pi
Pi

Fig. 9

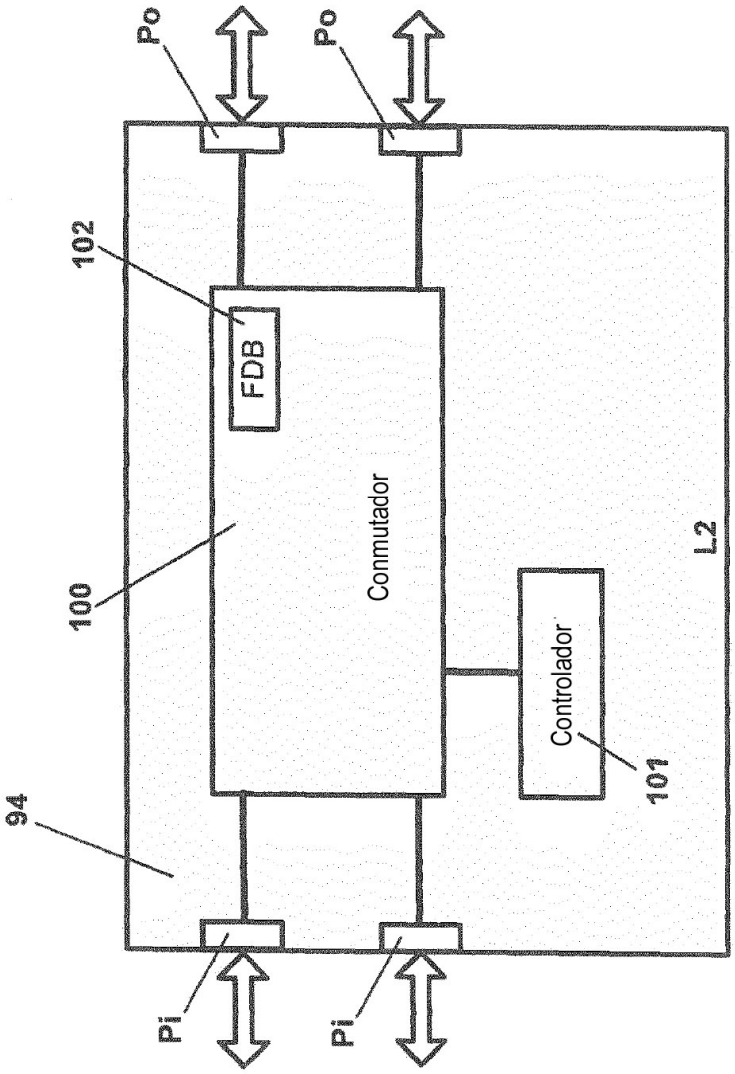
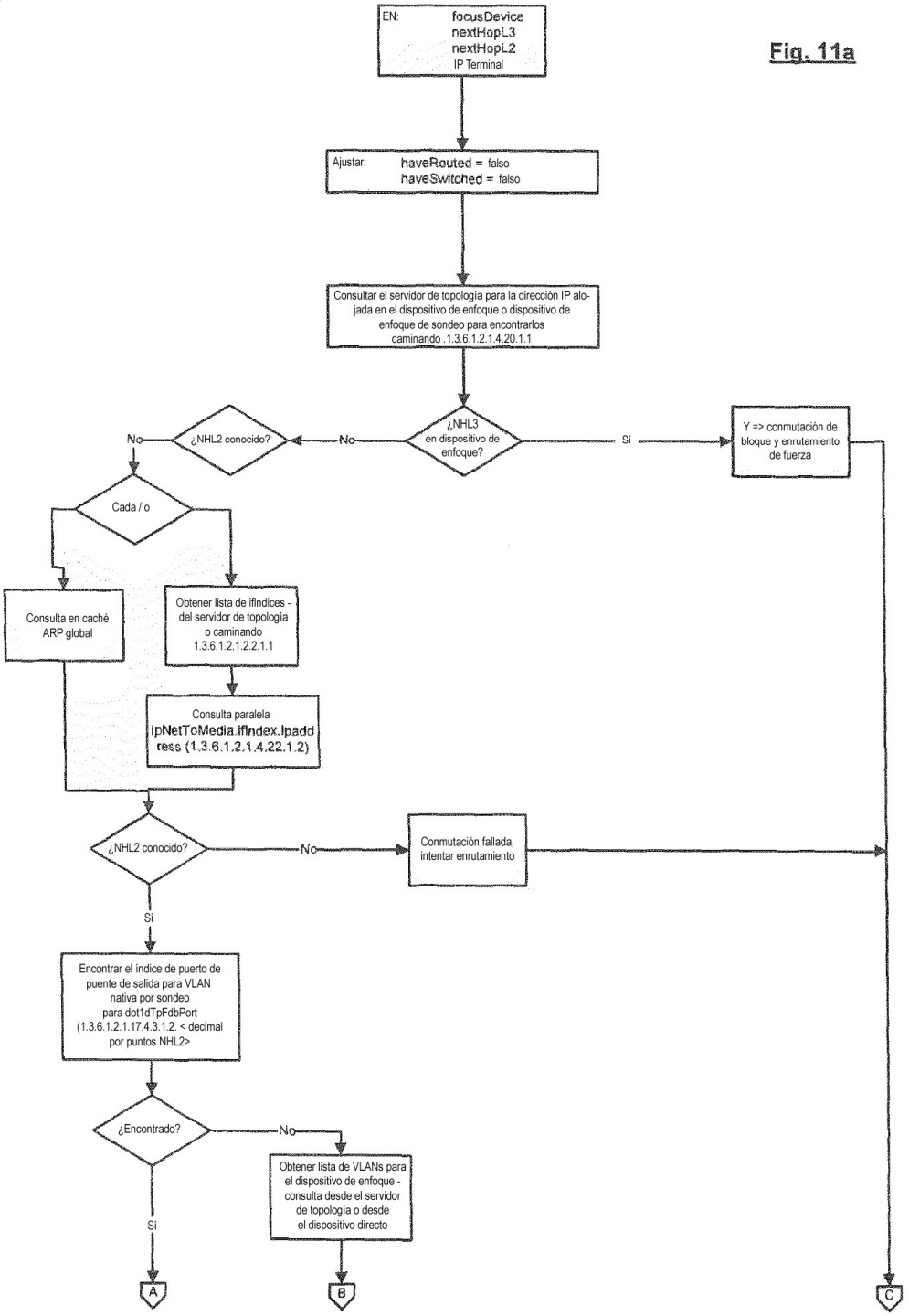


Fig. 10

Fig. 11a


EN: focusDevice
nextHopL3
nextHopL2
IP Terminal
Ajustar: haveRouted = falso
haveSwitched = falso
Consultar el servidor de topología para la dirección IP alojada en el dispositivo de enfoque o dispositivo de enfoque de sondeo para encontrarlos caminando .1.3.6.1.2.1.4.20.1.1
¿NHL3 en dispositivo de enfoque?
No
Sí
Y => conmutación de bloque y enrutamiento de fuerza
¿NHL2 conocido?
No
Cada / o
Consulta en caché ARP global
Obtener lista de ifIndices - del servidor de topología o caminando 1.3.6.1.2.1.2.2.1.1
Consulta paralela ipNetToMedia.ifIndex.Ipaddress (1.3.6.1.2.1.4.22.1.2)
¿NHL2 conocido?
No
Conmutación fallada, intentar enrutamiento
Sí
Encontrar el índice de puerto de puente de salida para VLAN nativa por sondeo para dot1dTpFdbPort (1.3.6.1.2.1.17.4.3.1.2. < decimal por puntos NHL2>
¿Encontrado?
No
Obtener lista de VLANs para el dispositivo de enfoque - consulta desde el servidor de topología o desde el dispositivo directo
Sí
A
B
C

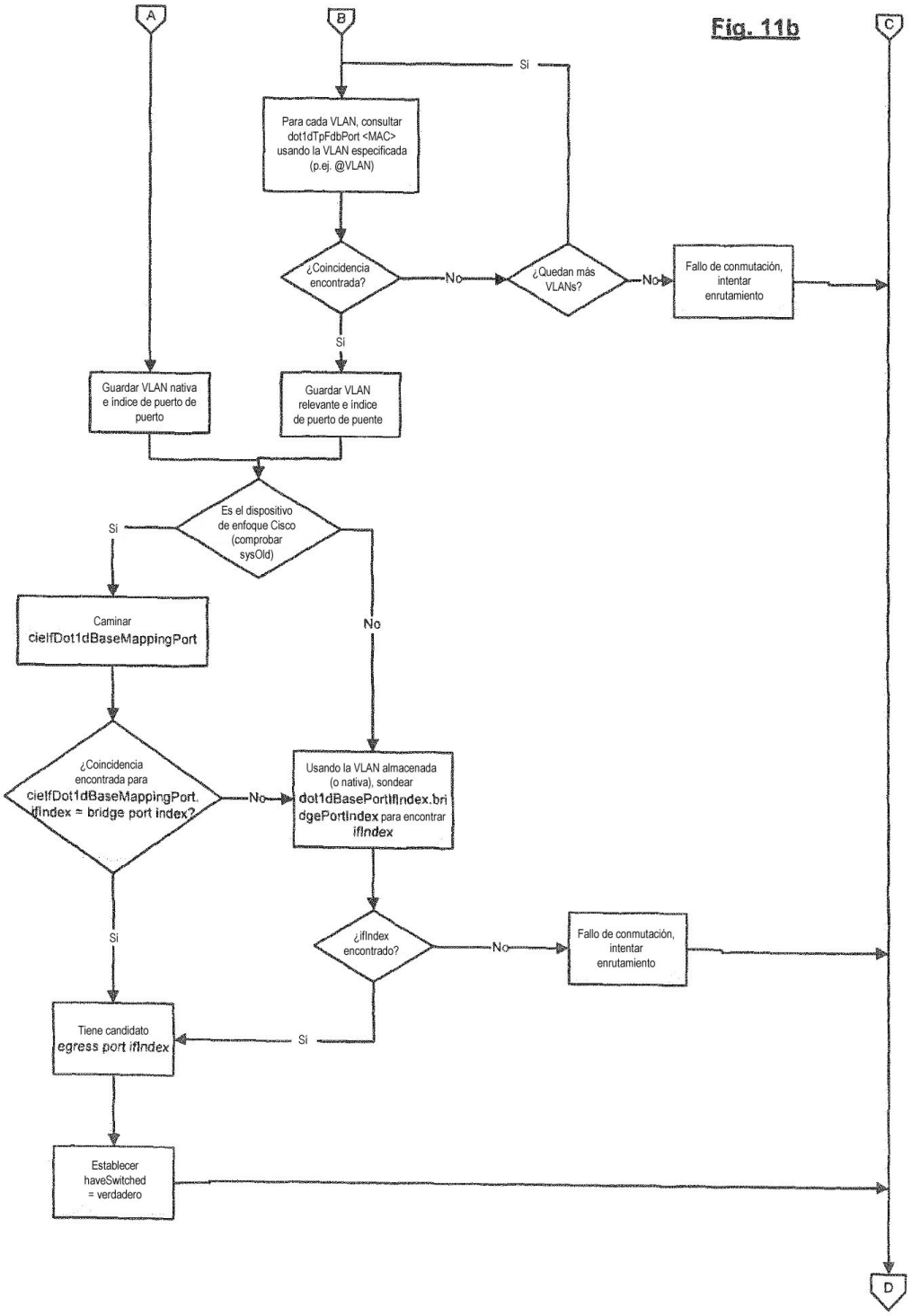
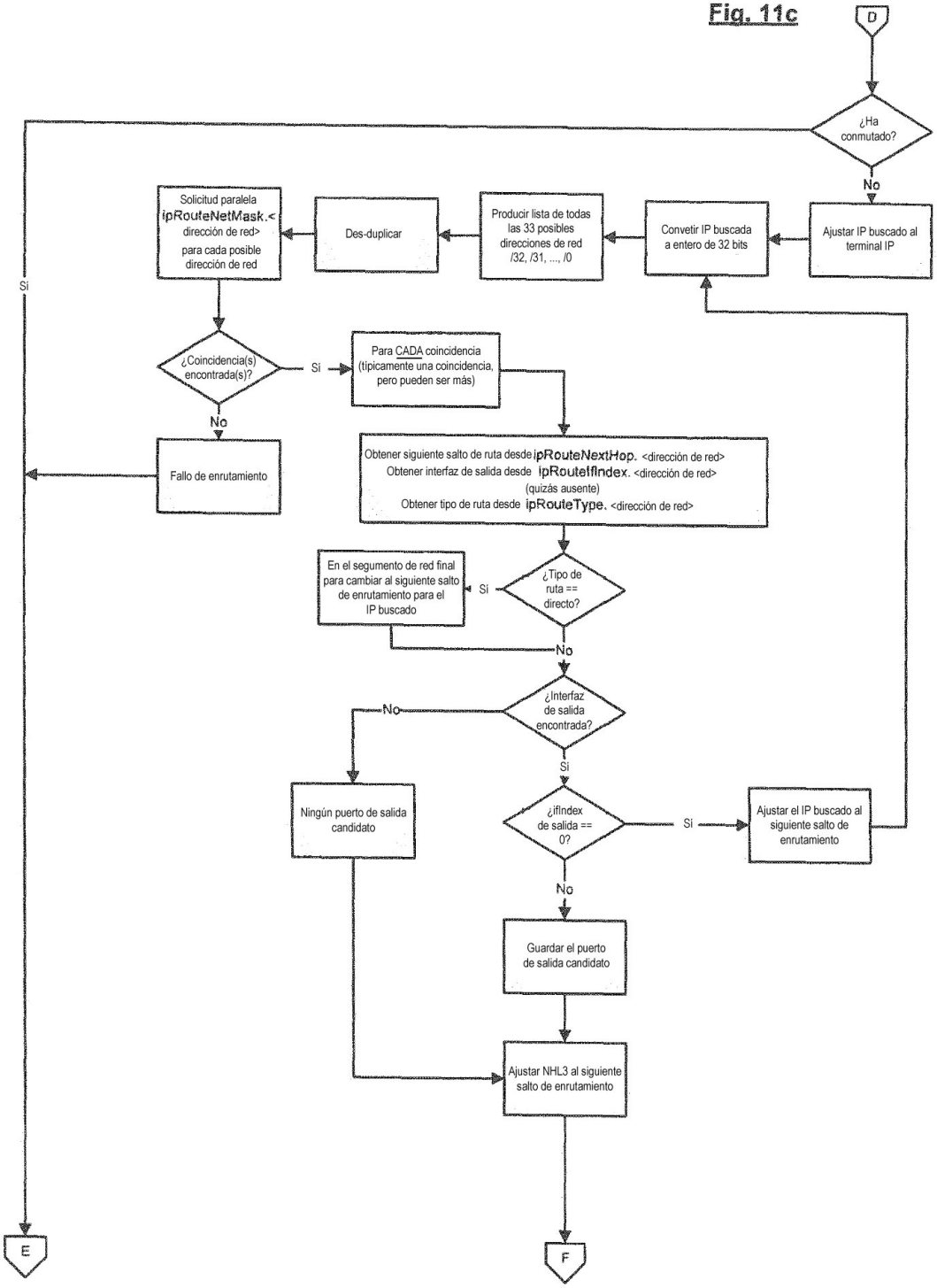

A
B
C
D
Si
Para cada VLAN, consultar dot1dTpFdbPort <MAC> usando la VLAN especificada (p.ej. @VLAN)
¿Coincidencia encontrada?
No
¿Quedan más VLANs?
No
Fallo de conmutación, intentar enrutamiento
Si
Guardar VLAN nativa e índice de puerto de puerto
Guardar VLAN relevante e índice de puerto de puente
Es el dispositivo de enfoque Cisco (comprobar sysOId)
Si
No
Caminar cieIfDot1dBaseMappingPort
¿Coincidencia encontrada para cieIfDot1dBaseMappingPort. ifIndex = bridge port index?
No
Usando la VLAN almacenada (o nativa), sondear dot1dBasePortIfIndex.bridgePortIndex para encontrar ifIndex
Si
¿ifIndex encontrado?
No
Fallo de conmutación, intentar enrutamiento
Tiene candidato egress port ifIndex
Si
Establecer haveSwitched = verdadero

Fig. 11b

Fig. 11c


D
¿Ha conmutado?
No
Si
Ajustar IP buscado al terminal IP
Convetir IP buscada a entero de 32 bits
Producir lista de todas las 33 posibles direcciones de red /32, /31, ..., /0
Des-duplicar
Solicitud paralela ipRouteNetMask.<dirección de red> para cada posible dirección de red
¿Coincidencia(s) encontrada(s)?
Si
Para CADA coincidencia (típicamente una coincidencia, pero pueden ser más)
No
Fallo de enrutamiento
Obtener siguiente salto de ruta desde ipRouteNextHop. <dirección de red>
Obtener interfaz de salida desde ipRouteIfIndex. <dirección de red>
(quizás ausente)
Obtener tipo de ruta desde ipRouteType. <dirección de red>
En el segumento de red final para cambiar al siguiente salto de enrutamiento para el IP buscado
Si
¿Tipo de ruta == directo?
No
¿Interfaz de salida encontrada?
No
Si
Ningún puerto de salida candidato
¿ifIndex de salida == 0?
Si
Ajustar el IP buscado al siguiente salto de enrutamiento
No
Guardar el puerto de salida candidato
Ajustar NHL3 al siguiente salto de enrutamiento
E
F

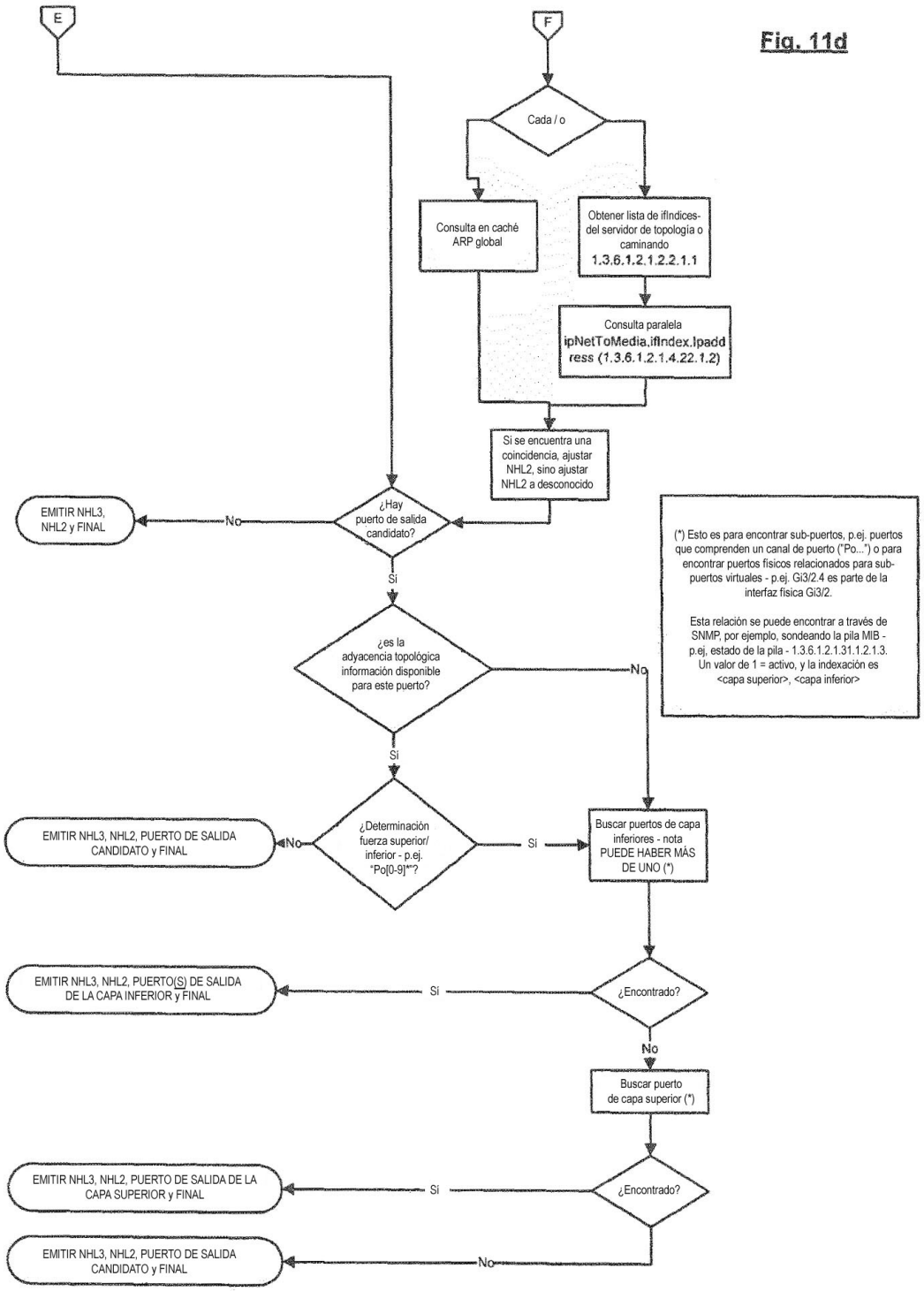

E
F
Cada / o
Consulta en caché ARP global
Obtener lista de ifIndices- del servidor de topología o caminando 1.3.6.1.2.1.2.2.1.1
Consulta paralela ipNetToMedia.ifIndex.Ipadd ress (1.3.6.1.2.1.4.22.1.2)
Si se encuentra una coincidencia, ajustar NHL2, sino ajustar NHL2 a desconocido
¿Hay puerto de salida candidato?
No
EMITIR NHL3, NHL2 y FINAL
Sí
¿es la adyacencia topológica información disponible para este puerto?
No
Sí
¿Determinación fuerza superior/ inferior - p.ej. "Po[0-9]*"?
No
EMITIR NHL3, NHL2, PUERTO DE SALIDA CANDIDATO y FINAL
Sí
Buscar puertos de capa inferiores - nota PUEDE HABER MÁS DE UNO (*)
¿Encontrado?
Sí
EMITIR NHL3, NHL2, PUERTO(S) DE SALIDA DE LA CAPA INFERIOR y FINAL
No
Buscar puerto de capa superior (*)
¿Encontrado?
Sí
EMITIR NHL3, NHL2, PUERTO DE SALIDA DE LA CAPA SUPERIOR y FINAL
No
EMITIR NHL3, NHL2, PUERTO DE SALIDA CANDIDATO y FINAL
(*) Esto es para encontrar sub-puertos, p.ej. puertos que comprenden un canal de puerto ("Po...") o para encontrar puertos físicos relacionados para sub-puertos virtuales - p.ej. Gi3/2.4 es parte de la interfaz física Gi3/2.
Esta relación se puede encontrar a través de SNMP, por ejemplo, sondeando la pila MIB - p.ej, estado de la 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>

Fig. 11d