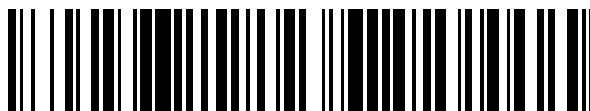


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 750 785**

51 Int. Cl.:

H04L 12/26 (2006.01)

H04L 12/24 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **21.10.2016** **E 16195009 (2)**

97 Fecha y número de publicación de la concesión europea: **07.08.2019** **EP 3160089**

54 Título: **Procedimiento y aparato para generar dependencias de red**

30 Prioridad:

22.10.2015 US 201514920557

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

27.03.2020

73 Titular/es:

**SOLARWINDS WORLDWIDE, LLC (100.0%)
7171 Southwest Parkway, Building 400
Austin, TX 78735 , US**

72 Inventor/es:

**LI, NA;
OSTASHENKOV, PAVLO y
ZATYLYN, KARLO MARTIN**

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 750 785 T3

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

DESCRIPCIÓN

Procedimiento y aparato para generar dependencias de red

5 Antecedentes:

Campo de la invención:

10 Las realizaciones de la presente invención se refieren a generar dependencias de red para una topología de red dada.

Descripción de la técnica relacionada:

15 Una topología de red generalmente describe una disposición de elementos de red dentro de una red. La topología de red de la red puede representar la ubicación de los elementos de la red, incluida la ubicación del dispositivo y/o la instalación de las conexiones entre los elementos de la red. Una topología de red también puede ilustrar cómo se transmiten los datos entre los diferentes elementos de la red dentro de la red.

20 Según Solarwinds, "Notas de lanzamiento de Orion NMP versión 11.5" 15 de abril de 2015, XP055345403, las dependencias de nodo a nodo se calculan automáticamente basándose en las conexiones de topología

El documento WO 01/86380 A2 se ocupa de detectar y suprimir fallas en elementos de red ubicados en diversos grupos lógicos. Se describe que se realizarían asociaciones entre los modelos de miembros del grupo lógico y los puertos que reenvían el tráfico para esos grupos lógicos.

25 Se puede encontrar más técnica anterior relevante en Anónimo "Automatic dependency calculation |THWACK", foro de la comunidad SolarWinds THWACK, 1 de abril de 2015, XP055345401, **y el** documento US 7 197 561 B1.

Resumen:

30 Según la presente invención, se proporcionan un procedimiento, un aparato y un producto de programa informático como se define en las reivindicaciones adjuntas.

Breve descripción de los dibujos:

35 Para una comprensión adecuada de la invención, se debe hacer referencia a los dibujos adjuntos, en los que:

La figura 1 ilustra una topología de red de ejemplo que incluye un motor de sondeo, según determinadas realizaciones de la presente invención.

40 La figura 2 ilustra rutas entre motores de sondeo y nodos de red, según determinadas realizaciones de la presente invención.

La figura 3(a) ilustra la determinación de un nodo/raíz central y una dependencia de un clúster, según determinadas realizaciones de la presente invención.

45 La figura 3(b) ilustra la determinación de raíces para cada clúster/grupo de nodos, según determinadas realizaciones.

La figura 3(c) ilustra un procedimiento de ejemplo para calcular una raíz para un clúster, según determinadas realizaciones.

50 La figura 3(d) ilustra la determinación de una raíz a partir de información local del sondeador, según determinadas realizaciones.

La figura 4 ilustra un procedimiento de generación automática de dependencias, evitando generar una dependencia que entre en conflicto con una dependencia definida por el usuario.

55 La figura 5 ilustra el almacenamiento de dependencias generadas dentro de una tabla, según determinadas realizaciones de la presente invención.

La figura 6 ilustra la eliminación de entradas correspondientes a dependencias, según determinadas realizaciones de la presente invención.

60 La figura 7 ilustra una interfaz que permite a un usuario ignorar una dependencia generada automáticamente, según determinadas realizaciones de la presente invención.

La figura 8 ilustra una interfaz que permite a un usuario habilitar o deshabilitar un procedimiento para generar dependencias automáticamente, según determinadas realizaciones de la presente invención.

65 La figura 9 ilustra una interfaz de ejemplo que permite a un usuario gestionar diferentes dependencias que han sido configuradas por el usuario y que se han generado automáticamente, según determinadas realizaciones de la presente invención.

La figura 10 ilustra un árbol de dependencia de ejemplo, según determinadas realizaciones de la presente invención.

La figura 11 ilustra un procedimiento para calcular una topología y determinar una autodependencia, según determinadas realizaciones de la invención.

La figura 12 ilustra el cálculo de dependencias de un clúster, para un sondeador dado, según determinadas

realizaciones de la invención.

La figura 13 ilustra un diagrama de flujo de un procedimiento según determinadas realizaciones de la invención.

La figura 14 ilustra un aparato según determinadas realizaciones de la invención.

La figura 15 ilustra un aparato según determinadas realizaciones de la invención.

5

Descripción detallada:

10 Determinadas realizaciones de la presente invención pueden dirigirse a un procedimiento y aparato para generar dependencias entre nodos de una topología de red dada. Generar una dependencia generalmente se refiere a determinar y/o establecer una relación entre al menos dos nodos de una red, en la que un nodo depende de al menos otro nodo. En otras palabras, una dependencia generada puede ser una conexión de nodo a nodo dentro de la topología de la red. Una topología de red generalmente se refiere a una representación de cómo los nodos de una red están conectados entre sí. Determinadas realizaciones pueden generar automáticamente dependencias para una topología de red dada, sin requerir ninguna entrada adicional del usuario. Determinadas realizaciones pueden generar automáticamente las dependencias aplicando un algoritmo sobre una topología de red dada. Con determinadas realizaciones, un usuario puede definir un clúster/grupo de nodos, en el que existen dependencias entre los nodos del clúster/grupo. Determinadas realizaciones también pueden estar dirigidas a determinar un nodo raíz de cada clúster/grupo de nodos, como se describe con más detalle a continuación. Determinadas realizaciones pueden generar automáticamente las dependencias dentro de cada clúster/grupo, y pueden garantizar que no haya conflictos ni inconsistencias dentro de las dependencias generadas.

20 Los dispositivos de red, tal como los motores de sondeo, pueden comunicarse con nodos dentro de la red. Un motor de sondeo puede comunicarse con el nodo para determinar el estado del nodo, para determinar las características operativas del nodo y/o para determinar la disponibilidad del nodo, por ejemplo. A medida que se generan dependencias, las dependencias generadas pueden corresponder a una ruta determinada/establecida entre un motor de sondeo y un nodo relevante. El motor de sondeo puede usar la ruta para llegar al nodo relevante.

25 La figura 1 ilustra una topología de red de ejemplo 100 que incluye un motor de sondeo 110. Como se describió anteriormente, un motor de sondeo generalmente realiza una comprobación del estado de los dispositivos de red para determinar en qué estado están los dispositivos de red, y/o realiza una comprobación de si los dispositivos de red todavía están en comunicación con la red. La topología de red 100 puede incluir una pluralidad de clústers (151, 152 y 153). Cada clúster puede corresponder a una parte diferente de la red general. Un clúster puede ser un clúster de oficinas centrales, mientras que otro clúster puede ser un clúster de sucursales. Determinadas conexiones/dispositivos pueden no ser evidentes de inmediato. Por ejemplo, la línea de puntos 140 indica que los dispositivos/conexiones de red entre la oficina central y las sucursales no son visibles en la topología 100. Los dispositivos de sondeo posiblemente no pueden supervisar los dispositivos de red entre la oficina principal y las sucursales, ya que un usuario posiblemente no pueda tener acceso a ellos. Posiblemente el usuario no pueda tener acceso a los dispositivos de red entre la oficina principal y las sucursales porque un proveedor de servicios puede tener el control de dichos dispositivos de red. Los nodos pueden estar conectados dentro de cada clúster, mientras que puede que no haya una conexión visible entre los propios clústers.

30 La figura 2 ilustra rutas entre motores de sondeo y nodos de red, según determinadas realizaciones. Como se describió anteriormente, cuando se generan dependencias automáticamente, el procedimiento de generación puede determinar una ruta que un motor de sondeo puede seguir para alcanzar un nodo de red. Por ejemplo, el motor de sondeo 210 puede seguir una ruta para alcanzar el nodo 250. La ruta que siguen los diferentes motores de sondeo (210, 220) para alcanzar un nodo dado puede ser diferente. Por ejemplo, la ruta desde el motor de sondeo 210 para alcanzar el nodo 250 es diferente de la ruta desde el motor de sondeo 220 para alcanzar el nodo 250. Como tal, determinadas realizaciones pueden determinar una ruta para que cada motor de sondeo llegue a cada nodo de red.

35 Como tal, al determinar una ruta entre un motor de sondeo y un nodo dado, cada dependencia generada (entre dos nodos, por ejemplo) puede ser una etapa direccional en la ruta. Podría haber múltiples rutas desde un motor de sondeo a un nodo de red determinado.

40 Un tipo de relación de dependencia entre dos nodos es una relación de nodo primario a nodo secundario, en la que el nodo secundario depende del nodo primario. En referencia nuevamente a la figura 2, un nodo primario 240 tiene una pluralidad de nodos secundarios (nodo 250 y nodo 260). En el caso de que el nodo primario 240 se caiga (de modo que la red no pueda alcanzar o comunicarse con el nodo primario 240), la red puede continuar sondeando un nodo secundario 250 del primario 240. Es decir, el motor de sondeo 210 todavía puede intentar comunicarse con el nodo secundario 250. Debido a que el nodo secundario 250 también parece estar inactivo (dado que el motor de sondeo 210 no puede alcanzar el nodo secundario 250 como resultado de que el nodo primario 240 está realmente inactivo), los motores de sondeo pueden activar alertas de red para indicar que el nodo secundario 250 está inactivo. Los motores de sondeo pueden generar las alertas, y una interfaz de usuario (IU) puede mostrar las alertas a un usuario. El usuario puede utilizar un navegador web o una herramienta especial (tal como una central de alertas, por ejemplo) para ver las alertas. En el caso de que un motor de sondeo no pueda alcanzar el nodo secundario 250, se examinan las dependencias asociadas del nodo secundario y los nodos o grupos primarios del nodo secundario se derivan de las dependencias examinadas. A continuación, si todos los nodos primarios del nodo secundario están inactivos,

55

60

65

determinadas realizaciones determinan que el estado del nodo 250 es inalcanzable. Si las alertas están configuradas para generarse cuando el estado de un nodo cambia a inactivo, el motor de sondeo no genera ninguna alerta si el estado del nodo 250 determina que no se puede alcanzar, como se describió anteriormente. Si no se determinan las dependencias, siempre que el motor de sondeo no pueda alcanzar el nodo 250, el estado del nodo 250 se considerará inactivo. Bajo esta condición, si las alertas se configuran para generarse cuando el estado de un nodo cambia a inactivo, entonces el motor de sondeo genera dichas alertas.

Sin embargo, en el caso de que un nodo secundario no sea accesible debido a que el nodo primario correspondiente está inactivo, las alertas activadas que se relacionan con el nodo secundario deben ignorarse porque el problema radica en el nodo primario. Con determinadas realizaciones, la generación/activación de alertas puede incluir dos etapas. La primera etapa es determinar el estado del nodo del nodo secundario. La segunda etapa es generar una alerta cuando se cumplen las condiciones para generar la alerta. Hay al menos dos formas de manejar las alertas. Con una primera forma de manejar las alertas, las únicas alertas que se generan son alertas que están destinadas a ser vistas por el usuario. Con esta estrategia, como se especificó anteriormente, cuando un motor de sondeo no puede alcanzar el nodo 250 y todos sus nodos primarios están inactivos, su estado de nodo es inalcanzable. Por lo tanto, no se genera ninguna alerta para este nodo inalcanzable, porque la condición para generar la alerta es un estado de nodo que está inactivo. Cuando el estado de un nodo cambia a inalcanzable, la condición para generar la alerta no se cumple y, por tanto, no se genera ninguna alerta. Con una segunda forma de manejar las alertas, la alerta se genera primero y luego se ignora si el usuario no debe ver la alerta generada. Es posible generar una alerta cuando el motor de sondeo no puede alcanzar el nodo 250, y luego ignorar la alerta cuando el procesamiento adicional indica que los nodos primarios del nodo 250 están todos inactivos. En dicho caso en el que algunas alertas generadas deben ignorarse, los motores de sondeo deben ignorar las alertas activadas que se relacionan con el nodo secundario para evitar inundar la red con alertas activadas. El motor de sondeo también debe evitar activar alertas de red para el nodo secundario.

Un nodo primario puede ser un nodo central (tal como, por ejemplo, un enrutador de borde). Como se describió anteriormente, cuando un nodo central de una red está inactivo, un motor de sondeo puede ser incapaz de alcanzar/comunicarse con los nodos que están configurados detrás (es decir, que están configurados para depender de) el nodo central caído.

Como se describió anteriormente, los motores de sondeo deben ignorar las alertas activadas que se relacionan con los nodos configurados detrás del nodo central caído, para evitar una avalancha de alertas. Como tal, determinadas realizaciones pueden primero identificar los nodos dependientes identificando los nodos que están configurados detrás del nodo central. Determinadas realizaciones pueden generar luego alertas solo para el nodo central, en lugar de generar alertas para todos los nodos que están configurados detrás del nodo central.

En vista de lo anterior, cuando un nodo es inalcanzable, determinadas realizaciones pueden determinar todos los elementos primarios de este nodo inalcanzable basándose en las dependencias que conoce el motor de sondeo (es decir, el motor de sondeo asignado al nodo inalcanzable). Si alguno de los elementos primarios (del nodo) está activo/accesible, entonces el estado del nodo inalcanzable se considera "inactivo". Por otro lado, si todos los elementos primarios están inactivos, el estado del nodo se considerará simplemente "inalcanzable" y las alertas activadas de dicho nodo pueden ignorarse. Con determinadas otras realizaciones, si todos los elementos primarios están inactivos para un nodo inalcanzable, entonces no se generan alertas para el nodo inalcanzable.

Determinadas realizaciones pueden realizar/ejecutar un algoritmo para generar automáticamente dependencias de red a pedido, o según un intervalo regular para que las dependencias generadas estén actualizadas. Determinadas realizaciones también pueden permitir al usuario añadir dependencias manualmente. Aunque la configuración manual de dependencias es típicamente tediosa, y aunque la configuración manual típicamente no puede adaptarse adecuadamente a una topología de red cambiante, determinadas realizaciones de la presente invención pueden proporcionar un procedimiento eficiente para implementar la configuración manual. Específicamente, determinadas realizaciones pueden integrar una funcionalidad que genera automáticamente dependencias junto con la funcionalidad en la que un usuario configura dependencias. Con determinadas realizaciones, las dependencias que se configuran/definen por el usuario pueden tener prioridad sobre las dependencias que se generan de forma automática.

Al generar dependencias, como se describió anteriormente, determinadas realizaciones pueden suprimir determinadas alertas, pueden derivar grupos basados en topología y pueden indicar la disponibilidad de nodos. Con respecto a la derivación de grupos basados en topología, un usuario puede querer definir un grupo que contenga todos los nodos que dependen directa o indirectamente (recursivamente) de un nodo específico. Con la determinación automática de dependencias (es decir, "autodependencia"), determinadas realizaciones pueden determinar todas las dependencias de las que este nodo es un elemento primario, directa o recursivamente, y luego todos los elementos secundarios de esas dependencias están en este grupo. Un usuario puede asignar permisos de acceso a este grupo a una cuenta, o definir condiciones de alerta basadas en el estado del grupo, por ejemplo. Esto permite al usuario gestionar los dispositivos de red en el nivel de grupo, en lugar de en el nivel de nodo. Como se describió anteriormente, las dependencias generadas se pueden usar para determinar un estado de nodo adecuado (es decir, respecto a si el estado del nodo esta "inactivo" o es simplemente "inalcanzable").

Los algoritmos de determinadas realizaciones pueden manejar un escenario con una pluralidad de motores de sondeo.

Cada nodo puede asignarse a diferentes motores de sondeo, y la asignación entre los nodos y los motores de sondeo puede cambiar con el tiempo. En cualquier momento, generalmente se asigna un nodo a un único motor de sondeo. En general, no es posible que el usuario asigne más de un motor de sondeo para sondear datos de un nodo dado. En otro momento, el usuario puede cambiar la configuración y dejar que otro motor de sondeo realice un sondeo del nodo dado.

La figura 3(a) ilustra la determinación de un nodo/raíz central y una dependencia de un clúster, según determinadas realizaciones. Las conexiones de topología pueden ser una entrada de los cálculos de determinadas realizaciones, y las dependencias autogeneradas pueden ser la salida. Determinadas realizaciones pueden determinar clúster de nodos, en los que los nodos dentro de un clúster están conectados. Determinadas realizaciones pueden, en la etapa 310, determinar el nodo raíz de cada clúster. Para cada motor de sondeo, determinadas realizaciones pueden determinar el nodo raíz de cada clúster. El nodo raíz puede corresponder a un nodo del que dependen otros nodos del clúster, tal como un nodo primario, por ejemplo. Los nodos raíz se pueden guardar dentro de una tabla autodependencia raíz. Para determinar el nodo raíz de un clúster, el procedimiento de generar dependencias automáticamente puede determinar primero un nodo que esté más cerca de un motor de sondeo particular, que luego se determina como la raíz del clúster. El procedimiento puede entonces construir/generar dependencias dentro de cada clúster, comenzando desde el nodo raíz determinado. En otras palabras, determinadas realizaciones pueden determinar una ruta desde el nodo raíz determinado a los otros nodos en el clúster. Las dependencias se pueden guardar en una tabla de dependencias, como se describe con más detalle a continuación.

Con respecto a la configuración manual de dependencias, determinadas realizaciones pueden permitir a los usuarios ajustar manualmente las dependencias añadiendo dependencias que faltan en la colección de dependencias generadas. Determinadas realizaciones también pueden corregir dependencias autogeneradas incorrectas. Por ejemplo, si las conexiones de topología son incorrectas por algún motivo (en las que, por ejemplo, el nodo 1 no está conectado al nodo 2, pero los datos de topología muestran que el nodo 1 está conectado al nodo 2), entonces las dependencias autogeneradas podrían ser incorrectas, o la dirección de la dependencia podría ser incorrecta. El usuario puede corregir esto añadiendo dependencias definidas por el usuario. Los usuarios también pueden ignorar las dependencias no deseadas. Con determinadas realizaciones, el usuario puede ajustar los cálculos realizados por el procedimiento de generación automática para: (1) permitir que las dependencias configuradas por el usuario tengan prioridad sobre las dependencias generadas automáticamente, y/o (2) evitar generar bucles de dependencia cíclicos entre las dependencias del usuario y las entradas generadas automáticamente. Un ciclo de dependencia cíclica corresponde a una serie de dependencias en las que cada nodo depende de otro nodo en la misma serie de dependencias.

La figura 3(b) ilustra la determinación de raíces para cada clúster/grupo de nodos, según determinadas realizaciones. Primero, determinadas realizaciones pueden determinar un grupo/clúster de nodos. Un clúster puede incluir nodos que están conectados entre sí basándose en una topología. Puede que no haya conexión (como lo refleja la topología) entre un nodo en un clúster y otro nodo en un clúster diferente. Segundo, determinadas realizaciones pueden determinar la raíz para cada clúster según el siguiente procedimiento (tan pronto como se determina una raíz, el procesamiento puede detenerse). Determinadas realizaciones pueden determinar una raíz definida por el usuario para un clúster dado. Por ejemplo, determinadas realizaciones pueden obtener una raíz de una tabla de protocolo de resolución de direcciones (ARP, por sus siglas en inglés). Determinadas realizaciones también pueden obtener la raíz de una puerta de enlace predeterminada de un motor de sondeo. Al determinar la raíz de un clúster particular, determinadas realizaciones pueden rastrear una ruta desde el motor de sondeo a un nodo en el clúster. Se puede determinar que el nodo más cercano al motor de sondeo es la raíz del clúster. Por ejemplo, en referencia a la figura 1, el sondeador 1 puede estar en la oficina principal, y determinadas realizaciones pueden determinar la raíz del sondeador 1 en el clúster "sucursal B". El sondeador 1 puede emitir una ruta de rastreo al nodo 10, y la ruta de la ruta de rastreo puede incluir "sondeador 1, nodo 1, nodo 2, nodo 8, nodo 9, nodo 10." Como tal, el nodo 8 puede ser la raíz del sondeador 1 en el clúster "sucursal B", porque el nodo 8 es el nodo más cercano al sondeador 1, desde dentro de la "sucursal B"

La figura 3(c) ilustra un procedimiento de ejemplo para calcular una raíz para un clúster, según determinadas realizaciones. Primero, determinadas realizaciones pueden determinar los sondeadores de una topología a partir de una lista de sondeadores. A continuación, determinadas realizaciones pueden determinar un sondeador para procesar a partir de la lista de sondeadores. Determinadas realizaciones pueden procesar un sondeador que sondea un clúster con más de un nodo dentro del clúster. Si hay una raíz definida por el usuario para el sondeador en el clúster entonces se puede determinar que la raíz definida por el usuario es la raíz de este sondeador en este clúster. Si no hay ninguna raíz definida por el usuario, determinadas realizaciones pueden determinar una raíz del clúster utilizando información local del sondeador. Como se describió anteriormente, determinadas realizaciones también pueden usar una ruta de rastreo para determinar una raíz del clúster.

La figura 3(d) ilustra la determinación de una raíz a partir de información local del sondeador, según determinadas realizaciones. Primero, los nodos adyacentes a un sondeador pueden determinarse a partir de la tabla ARP del sondeador. A continuación, si los sondeadores monitorean los nodos adyacentes, determinadas realizaciones

determinan el nodo adyacente que se monitorea y que tiene la mayor cantidad de conexiones a otros nodos monitoreados. Este nodo adyacente puede considerarse la raíz del sondeador. Como alternativa, si un sondeador monitorea una puerta de enlace predeterminada, entonces la puerta de enlace predeterminada puede considerarse la raíz del sondeador.

5 La figura 4 ilustra un procedimiento de generación automática de dependencias, evitando generar dependencias que entren en conflicto con una dependencia definida por el usuario. En el caso de que un usuario defina un nodo 420 como dependiente del nodo 410, entonces el procedimiento de generación automática de determinadas realizaciones evitaría duplicar esta dependencia definida por el usuario. En referencia a la figura 4, suponga que el nodo 3 se establece como un nodo dependiente del nodo 1 (se establece una dependencia entre el nodo 3 y el nodo 1), entonces determinadas realizaciones evitarán generar una dependencia entre un nodo 2 y un nodo 3. La dependencia entre el nodo 2 y el nodo 3 se evita porque el nodo 3 ya es accesible a través de la dependencia con el nodo 1. Además, el procedimiento de generación automática de determinadas realizaciones también evitaría generar una dependencia que entre en conflicto con la dependencia definida por el usuario. Por ejemplo, en referencia nuevamente a la figura 15 4, si se determina que el nodo 1 depende del nodo 2, entonces determinadas realizaciones evitarán generar una dependencia en la que el nodo 2 dependa del nodo 1.

Determinadas realizaciones pueden asegurar que no se forme un bucle de dependencia cíclica dentro de las dependencias. Determinadas realizaciones también pueden minimizar las dependencias numéricas que se generan al conservar solo aquellas dependencias que son útiles/relevantes para un motor de sondeo en una sola pasada. Una sola pasada generalmente puede significar que un nodo se procesa solo una vez para generar las dependencias y minimizar el número de dependencias. La alternativa es procesar primero el nodo dado para generar las dependencias. Una vez que se realiza el procesamiento para todos los nodos, entonces el mismo nodo se procesa nuevamente para eliminar las dependencias que no son útiles, para minimizar el número de dependencias. La alternativa puede requerir 25 dos pasadas y puede requerir mantener todas las dependencias generadas en la memoria (lo que puede resultar en un alto requisito de memoria) o guardar las dependencias generadas en una base de datos y luego eliminarlas, lo que puede causar actualizaciones sustanciales de la base de datos y problemas de rendimiento. Los detalles del procesamiento de una sola pasada se describen a continuación. Las dependencias pueden determinarse como útiles/relevantes, como se describe con más detalle a continuación.

30 Se puede determinar que una dependencia es útil/relevante para un motor de sondeo si la dependencia está dentro de una ruta que conecta el motor de sondeo a un nodo asignado a este motor de sondeo. La dependencia no es útil/relevante para un motor de sondeo, si ninguno de los elementos secundarios de la dependencia (y/o los elementos secundarios recursivos de la dependencia) es un nodo asignado al motor de sondeo. Si la dependencia no es útil/relevante para el motor de sondeo, entonces la dependencia se puede eliminar de forma segura. Cuando un nodo no responde a una solicitud de sondeo de su motor de sondeo asignado, la dependencia eliminada no se utilizará para determinar si el nodo está inactivo o inaccesible debido a un elemento primario inactivo.

40 Cuando se genera cada dependencia, la dependencia puede o no ser útil. Como tal, una dependencia generada puede necesitar ser almacenada hasta que todos los elementos secundarios de la dependencia hayan sido procesados, o hasta que uno de los elementos secundarios sea asignado al motor de sondeo relevante. Como tal, una dependencia generada puede necesitar mantenerse en la memoria. Se desconoce si una dependencia generada es útil o no hasta que se hayan procesado todos los elementos secundarios (que están asignados a este motor de sondeo y son elementos secundarios del nodo primario de la dependencia), o hasta que uno de los elementos secundarios se asigne al motor de sondeo relevante. Determinadas realizaciones pueden determinar cuándo se han procesado todos los elementos secundarios de un nodo primario porque determinadas realizaciones pueden rastrear cuántos nodos asignados a este motor de sondeo dentro de este clúster han sido procesados, y determinadas realizaciones pueden conocer el número total de nodos en este clúster que están asignados a este motor de sondeo. Después de que se hayan procesado todos los elementos secundarios de una dependencia, determinadas realizaciones pueden establecer entonces si alguno de los elementos secundarios está asignado a un motor de sondeo dado y, por tanto, pueden determinar si la dependencia es de por sí útil para el motor de sondeo dado. Cuando se determina que uno de los elementos secundarios está asignado al motor de sondeo relevante, se puede determinar que la dependencia es útil. Si se han procesado todos los elementos secundarios de la dependencia y no se determina que ninguno de los elementos secundarios esté asignado al motor de sondeo relevante, entonces se puede determinar que la 55 dependencia no es útil.

En vista de lo anterior, determinadas realizaciones pueden calcular/generar las dependencias dentro de un clúster para un motor de sondeo dado, y determinadas realizaciones pueden calcular/generar el nodo raíz en el clúster.

60 Determinadas realizaciones pueden realizar un filtrado basado en una entrada futura. Determinadas realizaciones pueden integrar una ruta de capa 2 y capa 3 para generar dependencias que se parecen mucho a las realidades de la red. Una conexión de capa 2 muestra más detalles sobre cómo se conectan los nodos, pero los nodos no son continuos. La capa 3 muestra menos detalles sobre cómo se conectan los nodos, pero son continuos en un clúster. Por ejemplo, en la figura 2, una conexión de capa 2 muestra que el nodo 3 está conectado al nodo 4 y el nodo 4 está conectado al nodo 5. La conexión de la capa 2 no muestra que el nodo 1 esté conectado al nodo 3, porque tanto el nodo 1 como el nodo 3 son enrutadores. La conexión de la capa 3 solo muestra que el nodo 3 está conectado al nodo

5 directamente porque el nodo 4 es un conmutador y no es visible en la conexión de la capa 3. La ruta real del nodo 3 al nodo 5 es del nodo 3 al nodo 4 y luego al nodo 5. La figura 12 muestra cómo construir la ruta completa usando las conexiones de la capa 2 y las conexiones de la capa 3

5 Determinadas realizaciones pueden manejar automáticamente una situación en la que la red es una red discontinua. Una red discontinua generalmente puede referirse a una red que no es completamente accesible/controlable por un usuario. Una parte de la red puede no ser completamente accesible /controlable para el usuario porque la parte puede ser controlada por un proveedor de servicios, por ejemplo. Para manejar una situación en la que la red es una red discontinua, determinadas realizaciones pueden encontrar una raíz para cada motor de sondeo y cada clúster, como se describió anteriormente. La figura 5 ilustra el almacenamiento de dependencias generadas dentro de una tabla 510. Al habilitar la generación automática de dependencias, las dependencias (511, 512, 513, 514, etc.) pueden generarse rápidamente. Las dependencias recién generadas pueden aparecer en una tabla 510 (tal como, por ejemplo, una tabla Orion.Dependencias). Al deshabilitar la generación automática de dependencias, las dependencias que se generaron previamente pueden eliminarse rápidamente de la tabla Orion.Dependencias. Cuando se cambia la configuración para deshabilitar o habilitar la generación automática de dependencias, el cambio de la configuración puede desencadenar un evento de auditoría.

20 La figura 6 ilustra la eliminación de entradas de dependencias, según determinadas realizaciones de la presente invención. Determinadas realizaciones pueden permitir a un usuario gestionar las dependencias que se generan automáticamente. Determinadas realizaciones pueden no permitir que el usuario edite ni permitir que el usuario elimine las dependencias que se generan automáticamente. Sin embargo, el usuario puede ignorar determinadas dependencias. Las entradas ignoradas (tal como la entrada 611) pueden eliminarse de la tabla Orion. Dependencias 610. El usuario puede decidir si una entrada generada automáticamente debe ignorarse (no se muestra en la tabla de dependencias). Por ejemplo, el usuario puede decidir ignorar una entrada generada automáticamente si sabe que dicha dependencia no existe. Por ejemplo, el usuario puede saber que el nodo 4 no está conectado al nodo 5, pero la topología puede ser incorrecta, y la topología incorrecta puede incluir una conexión entre el nodo 4 y el nodo 5. A continuación, cuando la dependencia automática genera la entrada 611, el usuario puede corregir este problema ignorando la entrada 611. Si hay una conexión entre el nodo 4 y el nodo 5 que indica que el nodo 5 depende del nodo 4, pero el nodo 4 realmente depende del nodo 5, entonces el usuario puede corregir este problema añadiendo una dependencia configurada por el usuario (en la que el elemento primario es el nodo 5 y el elemento secundario es el nodo 4). La entrada 611 puede ignorarse porque la entrada puede corresponder a una dependencia que es duplicada de la dependencia 612. Una dependencia que se ignora no afectará los estados de los nodos que se relacionan con la dependencia ignorada. Las dependencias ignoradas se pueden mostrar en la pestaña "gestionar dependencias ignoradas" 730 de la figura 7, por ejemplo.

35 La figura 7 ilustra una interfaz 710 que permite a un usuario ignorar una dependencia generada automáticamente. La interfaz 710 permite al usuario gestionar dependencias. La interfaz 710 puede incluir una ventana 720 que permite a un usuario ignorar una o más dependencias generadas automáticamente.

40 La figura 8 ilustra una interfaz 810 que permite a un usuario habilitar o deshabilitar un procedimiento para generar dependencias automáticamente, según determinadas realizaciones. Con determinadas realizaciones, el procedimiento de generación automática de dependencias puede ser habilitado o deshabilitado por un usuario con derechos de administrador. El usuario puede habilitar/deshabilitar la funcionalidad a través de una página de configuración de sondeo/página de gestión de dependencias, por ejemplo.

45 Con determinadas realizaciones, la generación automática de dependencias puede habilitarse si la topología de red permite la generación automática de dependencias. Determinadas realizaciones pueden permitir a un usuario eliminar las dependencias autogeneradas ignoradas. Sin embargo, como resultado de eliminar las dependencias autogeneradas ignoradas, estas dependencias autogeneradas se pueden volver a añadir/enumerar en la tabla Orion.Dependencias al realizar un procedimiento de cálculo/generación posterior, y las dependencias autogeneradas ignoradas entonces pueden volverse activas nuevamente.

50 Con determinadas realizaciones, se puede generar un evento de auditoría cuando el usuario ignora las dependencias generadas automáticamente o cuando el usuario elimina una dependencia ignorada generada automáticamente. Un ejemplo del evento de auditoría se relaciona con la creación de un evento para la acción y añadirlo al registro de eventos. Así, el usuario puede ver y registrar eventos y notar las acciones que cambiaron las dependencias. Por lo tanto, la acción puede registrarse en el historial y puede auditarse más adelante.

60 Con respecto a la funcionalidad de las dependencias de generación automática, la siguiente información de configuración puede almacenarse en una base de datos de la siguiente manera. Primero, una entrada puede indicar si una funcionalidad de generación automática está habilitada o no. La entrada (tal como "SWNetPerfMon-Settings-AutoDependency") puede habilitarse o deshabilitarse.

65 Cada una de las dependencias puede almacenarse con algunos o todos los siguientes parámetros. Un parámetro de cada dependencia puede ser un parámetro "autogestionado". Este parámetro puede indicar si la dependencia/entrada correspondiente se generó automáticamente (es decir, en la que el parámetro se establece en "verdadero") o si la

dependencia/entrada correspondiente se definió manualmente/por el usuario (es decir, en la que el parámetro se establece en "falso").

5 Otro parámetro de cada dependencia puede ser un parámetro "EngineID". Este parámetro puede indicar el motor de sondeo al que está asociada la dependencia. Con determinadas realizaciones, las dependencias definidas por el usuario pueden estar asociadas a todos los motores de sondeo. Una dependencia generada automáticamente generalmente puede estar asociada con un motor de sondeo.

10 Otro parámetro para cada dependencia puede ser un parámetro de "categoría". Este parámetro puede indicar el tipo de dependencia.

15 Se pueden añadir otros parámetros para cada dependencia, por razones de rendimiento. Estos otros parámetros pueden incluir ParentEntityType, que indica un tipo de entidad de un nodo primario de la dependencia. Otro parámetro puede incluir ParentNetObjectID, que indica un identificador del objeto primario de la dependencia. Otro parámetro puede incluir ChildEntityType, que indica un tipo de entidad de un nodo secundario de la dependencia. Otro parámetro puede incluir ChildNetObjectID, que indica un identificador del objeto secundario de la dependencia.

20 Con determinadas realizaciones, una nueva tabla puede ser una tabla "autodependencia raíz". Esta tabla se puede usar al calcular dependencias. Esta tabla puede contener el nodo raíz que se calcula para cada motor de sondeo y clúster. Determinadas realizaciones también pueden incluir una tabla "autodependencias eliminadas". Esta tabla puede contener las dependencias autogeneradas ignoradas.

25 La figura 9 ilustra una interfaz de ejemplo 910 que permite a un usuario gestionar las diferentes dependencias que se han configurado por el usuario y que se han generado automáticamente. Cada entrada 911, 912, 913 de la interfaz 910 puede corresponder a una dependencia diferente. Una ventana emergente 915 puede ilustrar las características operativas/disponibilidad de cada nodo dependiente de cada dependencia.

30 La figura 10 ilustra un árbol de dependencia de ejemplo. Con determinadas realizaciones, un usuario puede ver las dependencias que se han generado utilizando el árbol de dependencias visualizado. El usuario puede entonces determinar si algunas de las dependencias generadas automáticamente deben ser ignoradas. Cada entrada dentro del árbol de dependencias de ejemplo puede corresponder a una dependencia, y cada entrada puede incluir un nombre/identificador de la dependencia, el nodo primario de la dependencia y el nodo secundario de la dependencia.

35 La figura 11 ilustra un procedimiento para calcular una topología y determinar una autodependencia, según determinadas realizaciones de la invención. En la etapa 1, el usuario puede activar el cálculo (tal como un cálculo a pedido, por ejemplo) o un temporizador puede activar el cálculo (tal como un cálculo periódico, por ejemplo). En la etapa 2, se puede calcular la topología y las conexiones de la topología se pueden almacenar en la tabla conexiones de topología. En la etapa 3, las dependencias pueden calcularse automáticamente basándose las conexiones de topología generadas en la etapa 2.

40 La figura 12 ilustra el cálculo de dependencias de un clúster, para un sondeador dado, según determinadas realizaciones de la invención. Como se describió anteriormente, determinadas realizaciones pueden integrar una ruta de capa 2 y de capa 3 para generar dependencias que se parecen mucho a las realidades de la red. La figura 12 muestra el flujo de procesamiento para un motor de sondeo y clúster determinados, y muestra cómo combinar conexiones de capa 2 y capa 3 para generar dependencias detalladas para formar una ruta desde el motor de sondeo a todos los nodos en un clúster. El cálculo comienza desde la raíz del clúster para un motor de sondeo dado. Todos los nodos que están directamente conectados a la raíz son sus elementos secundarios. Las dependencias que usan conexiones de capa 2 se calculan primero de forma recursiva para N etapas (el valor predeterminado de N puede ser 3 etapas, por ejemplo). Luego se calculan las dependencias que usan conexiones de capa 3. Si se puede alcanzar un nodo mediante dependencias derivadas de la capa 2, no se derivará ninguna dependencia de la capa 3 para alcanzar dicho nodo. Por ejemplo, en la figura 2, basándose en la conexión de capa 2, el nodo 3 tiene el nodo secundario 4 (AutoDep-1-3-4-2) y el nodo 4 tiene el nodo secundario 5 (AutoDep-1-4-5-3). A continuación, según la conexión de la capa 3, el nodo 3 tiene el nodo secundario 5 (AutoDep-1-3-5-2, esta dependencia puede ignorarse porque el motor de sondeo puede alcanzar del nodo 5 al nodo 4 usando una dependencia derivada de la capa 2). Las dependencias de capa 2 pueden tener prioridad sobre las dependencias de capa 3 porque la capa 2 proporciona más detalles de la conectividad.

60 La condición que verifica el número de nodos procesados para este sondeador busca minimizar el número de dependencias cuando se procesan todos los nodos asignados a este motor de sondeo, y no se pueden generar más dependencias útiles.

65 La figura 13 ilustra un diagrama de flujo de un procedimiento según determinadas realizaciones de la invención. El procedimiento ilustrado en la figura 13 incluye, en 1310, determinar, mediante un motor de sondeo, una raíz. La raíz comprende un primer nodo dentro de un clúster de nodos de una red. El procedimiento también puede incluir, en 1320, generar al menos una dependencia de red entre la raíz y un segundo nodo. La al menos una dependencia de red generada corresponde a una conexión entre nodos de la red. La al menos una dependencia de red generada

corresponde a una ruta direccional desde el motor de sondeo al segundo nodo. El procedimiento también puede incluir, en 1330, sondear el segundo nodo. El sondeo se realiza a través de la al menos una dependencia de red que se ha generado entre el motor de sondeo y el segundo nodo. El procedimiento también puede incluir, en 1340, determinar que el segundo nodo es inalcanzable. Con determinadas realizaciones, se puede determinar que el segundo nodo sondeado es inalcanzable después de que el motor de sondeo envía una solicitud y no recibe una respuesta antes de que se agote el tiempo de espera. El procedimiento también puede incluir, en 1350, generar alertas activadas relacionadas con el segundo nodo inalcanzable, si el motor de sondeo determina que cualquiera de los nodos primarios del segundo nodo es accesible. Determinadas realizaciones no generan alertas relacionadas con el nodo sondeado inalcanzable, si ninguno de los nodos primarios del nodo primario es accesible.

La figura 14 ilustra un aparato según determinadas realizaciones de la invención. En una realización, el aparato puede ser un nodo de red configurado para realizar las funciones de un motor de sondeo, por ejemplo. El motor de sondeo puede estar en un servidor. El motor de sondeo puede o no ser un dispositivo de red. El motor de sondeo puede ser un host final. El motor de sondeo puede realizar la función de determinar las dependencias de la red. Si hay varios motores de sondeo en el sistema, solo un motor de sondeo puede necesitar calcular las dependencias para todo el sistema, una vez que el motor de sondeo recibe la topología de todo el sistema.

El aparato de la figura 14 puede realizar, al menos, el procedimiento de la figura 13. El aparato 10 puede incluir un procesador 22 para procesar información y ejecutar instrucciones u operaciones. El procesador 22 puede ser cualquier tipo de procesador de propósito general o específico. Mientras se muestra un único procesador 22 en la figura 14, se pueden utilizar múltiples procesadores según otras realizaciones. El procesador 22 también puede incluir uno o más ordenadores de propósito general, ordenadores de propósito especial, microprocesadores, procesadores de señal digital (DSP), matrices de compuertas programables en campo (FPGA), circuitos integrados de aplicación específica (ASIC) y procesadores basados en una arquitectura de procesador multinúcleo, como ejemplos.

El aparato 10 puede incluir además una memoria 14, acoplada al procesador 22, para almacenar información e instrucciones que puede ejecutar el procesador 22. La memoria 14 puede ser una o más memorias y de cualquier tipo adecuado para el entorno de aplicación local, y puede implementarse utilizando cualquier tecnología de almacenamiento de datos volátil o no volátil adecuada, tal como un dispositivo de memoria basado en semiconductores, un dispositivo y sistema de memoria magnética, un dispositivo y sistema de memoria óptica, memoria fija y memoria extraíble. Por ejemplo, la memoria 14 incluye cualquier combinación de memoria de acceso aleatorio (RAM), memoria de solo lectura (ROM), almacenamiento estático, tal como un disco magnético u óptico, o cualquier otro tipo de máquina no transitoria o medio legible por ordenador. Las instrucciones almacenadas en la memoria 14 pueden incluir instrucciones de programa o código de programa de ordenador que, cuando es ejecutado por el procesador 22, permiten que el aparato 10 realice las tareas que se describen en esta invención.

El aparato 10 también puede incluir una o más antenas (no mostradas) para transmitir y recibir señales y/o datos hacia y desde el aparato 10. El aparato 10 puede incluir además un transceptor 28 que modula la información en una forma de onda portadora para la transmisión por la(s) antena(s) y demodula la información recibida a través de la(s) antena(s) para su posterior procesamiento por otros elementos del aparato 10. En otras realizaciones, el transceptor 28 puede ser capaz de transmitir y recibir señales o datos directamente.

El procesador 22 puede realizar funciones asociadas con el funcionamiento del aparato 10, incluyendo, sin limitación, precodificación de parámetros de ganancia/fase de antena, codificación y decodificación de bits individuales que forman un mensaje de comunicación, formateo de información y control general del aparato 10, incluidos los procedimientos relacionados con la gestión de recursos de comunicación. El aparato 10 también puede funcionar como un transceptor en forma de una tarjeta de red que se conecta a una red.

En una realización, la memoria 14 puede almacenar módulos de software que proporcionan funcionalidad cuando son ejecutados por el procesador 22. Los módulos pueden incluir un sistema operativo 15 que proporciona funcionalidad del sistema operativo para el aparato 10. La memoria también puede almacenar uno o más módulos funcionales 18, tales como una aplicación o programa, para proporcionar funcionalidad adicional para el aparato 10. Los componentes del aparato 10 pueden implementarse en hardware, o como cualquier combinación adecuada de hardware y software.

La figura 15 ilustra un aparato según determinadas realizaciones de la invención. El aparato 1500 puede ser un elemento/entidad de red tal como un nodo de red que está configurado para funcionar como un motor de sondeo, por ejemplo. El aparato 1500 puede incluir una primera unidad de determinación 1510 que determina una raíz. La raíz incluye un primer nodo dentro de un clúster de nodos de una red. El aparato 1500 también puede incluir una primera unidad generadora 1520 que genera al menos una dependencia de red entre la raíz y un segundo nodo. La al menos una dependencia de red generada corresponde a una conexión entre nodos de la red. La al menos una dependencia de red generada corresponde a una ruta direccional desde el motor de sondeo al segundo nodo. El aparato 1500 también puede incluir una unidad de sondeo 1530 que sondea el segundo nodo. El sondeo se realiza a través de la al menos una dependencia de red que se ha generado entre el motor de sondeo y el segundo nodo. El aparato 1500 también puede incluir una segunda unidad de determinación 1540 que determina que el segundo nodo es inalcanzable. El aparato 1500 también puede incluir una segunda unidad generadora 1550 que genera alertas activadas relacionadas con el segundo nodo inalcanzable, si se determina que cualquiera de los nodos primarios del segundo

nodo es accesible por el aparato 1500.

5 Los rasgos, ventajas y características descritas de la invención se pueden combinar de cualquier manera adecuada en una o más realizaciones. Un experto en la materia relevante reconocerá que la invención se puede practicar sin una o más de las características o ventajas específicas de una realización particular. En otros casos, se pueden reconocer características y ventajas adicionales en determinadas realizaciones que pueden no estar presentes en todas las realizaciones de la invención. Un experto en la materia comprenderá fácilmente que la invención como se indicó anteriormente se puede practicar con etapas en un orden diferente, y/o con elementos de hardware en configuraciones que son diferentes a las que se describen. Por lo tanto, aunque la invención se ha descrito basándose en estas realizaciones preferidas, será evidente para los expertos en la materia que determinadas modificaciones, 10 variaciones y construcciones alternativas serían evidentes, mientras se mantengan dentro del alcance de la invención como se define en las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un procedimiento, que comprende:

5 determinar (1310), mediante un motor de sondeo (210, 220), una raíz conectada al motor de sondeo, en el que la raíz comprende un primer nodo dentro de un clúster de nodos de una red;
 generar (1320), mediante el motor de sondeo (210, 220), una dependencia de red entre la raíz y un segundo nodo (250, 260) en el clúster, en el que la dependencia de red generada corresponde a una conexión desde la raíz al
 10 segundo nodo;
 determinar, mediante el motor de sondeo (210, 220), una ruta desde el motor de sondeo (210, 220) hasta el segundo nodo (250, 260) a través de la raíz basándose en la dependencia de red generada entre la raíz y el segundo nodo (250, 260); sondear (1330), mediante el motor de sondeo (210, 220), el segundo nodo (250, 260) a través de la ruta determinada desde el motor de sondeo (210, 220) hasta el segundo nodo (250, 260);
 15 determinar, mediante el motor de sondeo (210, 220), si se puede alcanzar un nodo primario (240) del segundo nodo (250, 260), en el que el nodo primario (240) tiene una pluralidad de nodos secundarios (250, 260) que dependen del nodo primario (240) y el segundo nodo (250, 260) es uno de la pluralidad de nodos secundarios (250, 260) del nodo primario (240);
 determinar (1340), mediante el motor de sondeo (210, 220), que el segundo nodo (250, 260) es inalcanzable; y
 20 generar (1350), mediante el motor de sondeo (210, 220), alertas activadas relacionadas con el segundo nodo inalcanzable (250, 260), si se determina que el nodo primario (240) del segundo nodo (250, 260) es accesible por el motor de sondeo (210, 220).

2. El procedimiento según la reivindicación 1, en el que la determinación de la raíz comprende determinar la raíz basándose en al menos una de una definición de usuario y una información local del motor de sondeo.

25

3. Un aparato (10), que comprende:

al menos un procesador (22); y
 al menos una memoria (14) incluyendo el código del programa informático,
 30 la al menos una memoria (14) y el código del programa informático configurados, con el al menos un procesador (22), para hacer que el aparato (10) al menos determine una raíz conectada al aparato (10), en el que la raíz comprende un primer nodo dentro de un clúster de nodos de una red; generar una dependencia de red entre la raíz y un segundo nodo (250, 260) en el clúster, en el que la dependencia de red generada corresponde a una conexión desde la raíz al segundo nodo;
 35 determinar una ruta desde el aparato (10) al segundo nodo (250, 260) a través de la raíz basándose en la dependencia de red generada entre la raíz y el segundo nodo (250, 260); sondear el segundo nodo (250, 260) a través de la ruta determinada desde el aparato hasta el segundo nodo (250, 260);
 determinar si se puede alcanzar un nodo primario (240) del segundo nodo (250, 260), en el que el nodo primario (240) tiene una pluralidad de nodos secundarios (250, 260) que dependen del nodo primario (240) y el segundo
 40 nodo (250, 260) es uno de la pluralidad de nodos secundarios (250, 260) del nodo primario (240);
 determinar que el segundo nodo (250, 260) es inalcanzable; y
 generar alertas activadas relacionadas con el segundo nodo inalcanzable (250, 260), si se determina que el nodo primario (240) del segundo nodo (250, 260) es accesible.

45 4. El aparato según la reivindicación 3, en el que la determinación de la raíz comprende determinar la raíz basándose en al menos una de una definición de usuario y una información local del aparato.

50 5. Un producto de programa informático, incorporado en un medio legible por ordenador no transitorio, configurado el producto de programa informático para controlar un procesador para realizar el procedimiento según la reivindicación 1 o 2.

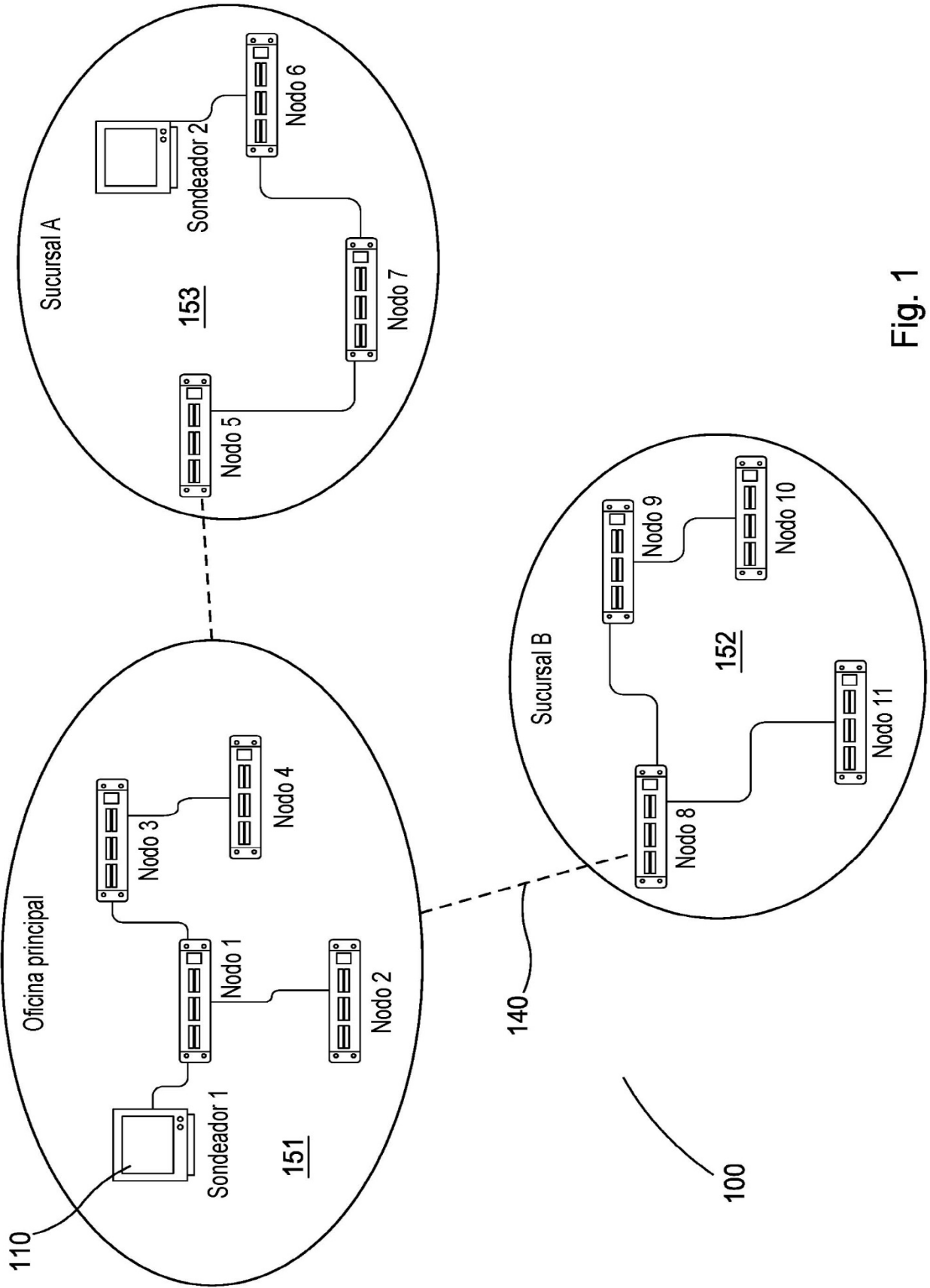


Fig. 1

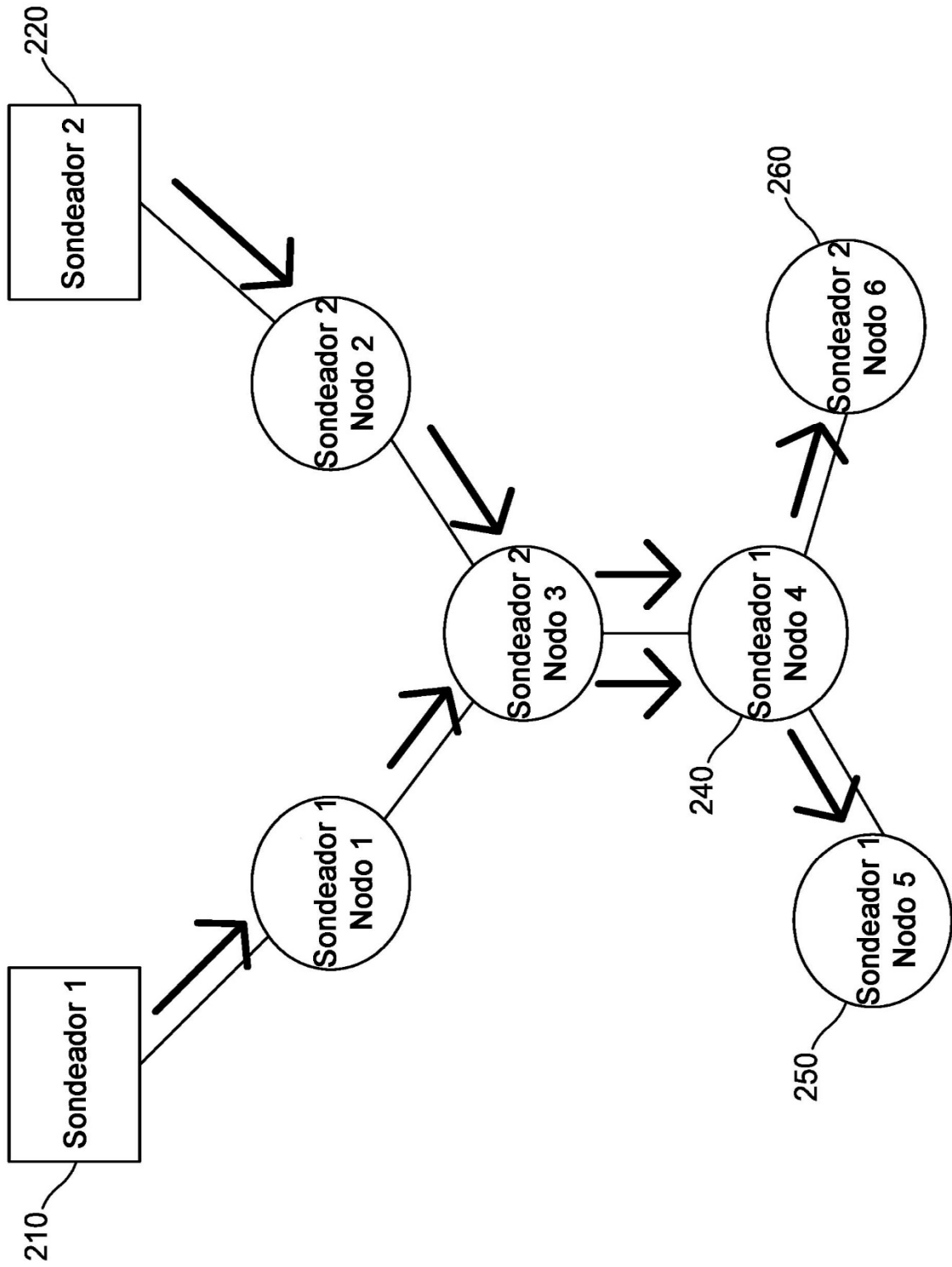


Fig. 2

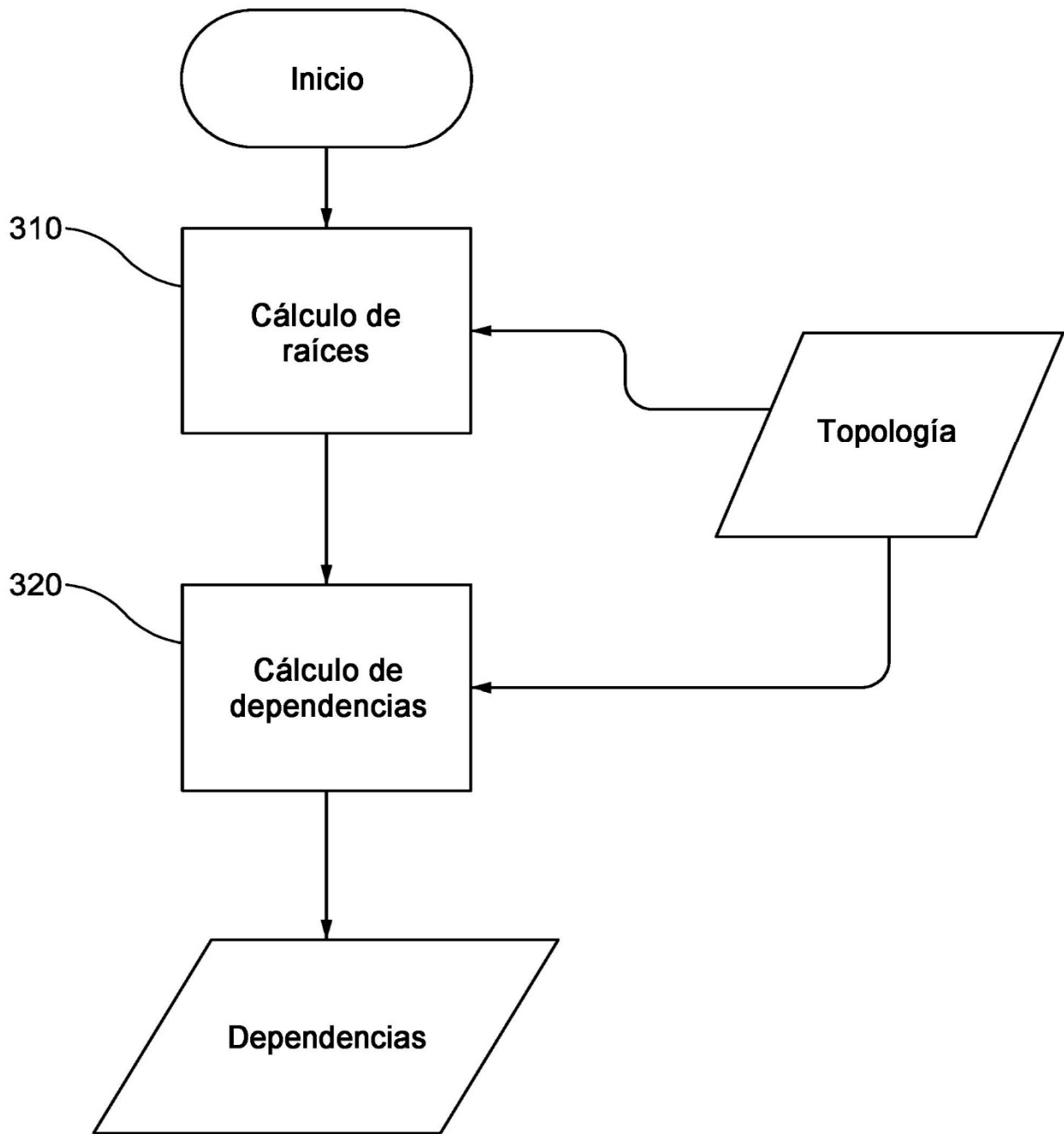


Fig. 3(a)

Cálculo de raíces

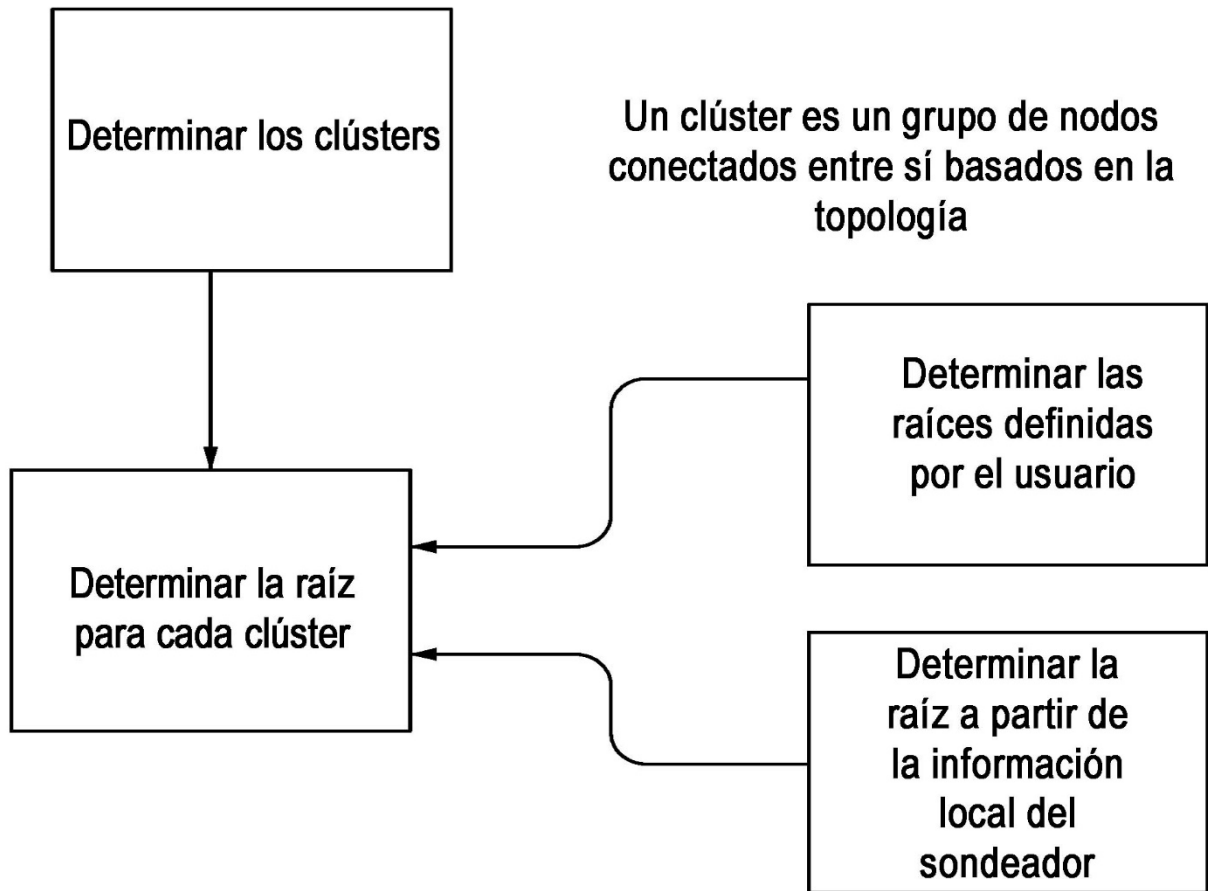


Fig. 3(b)

Calcular la raíz para el clúster

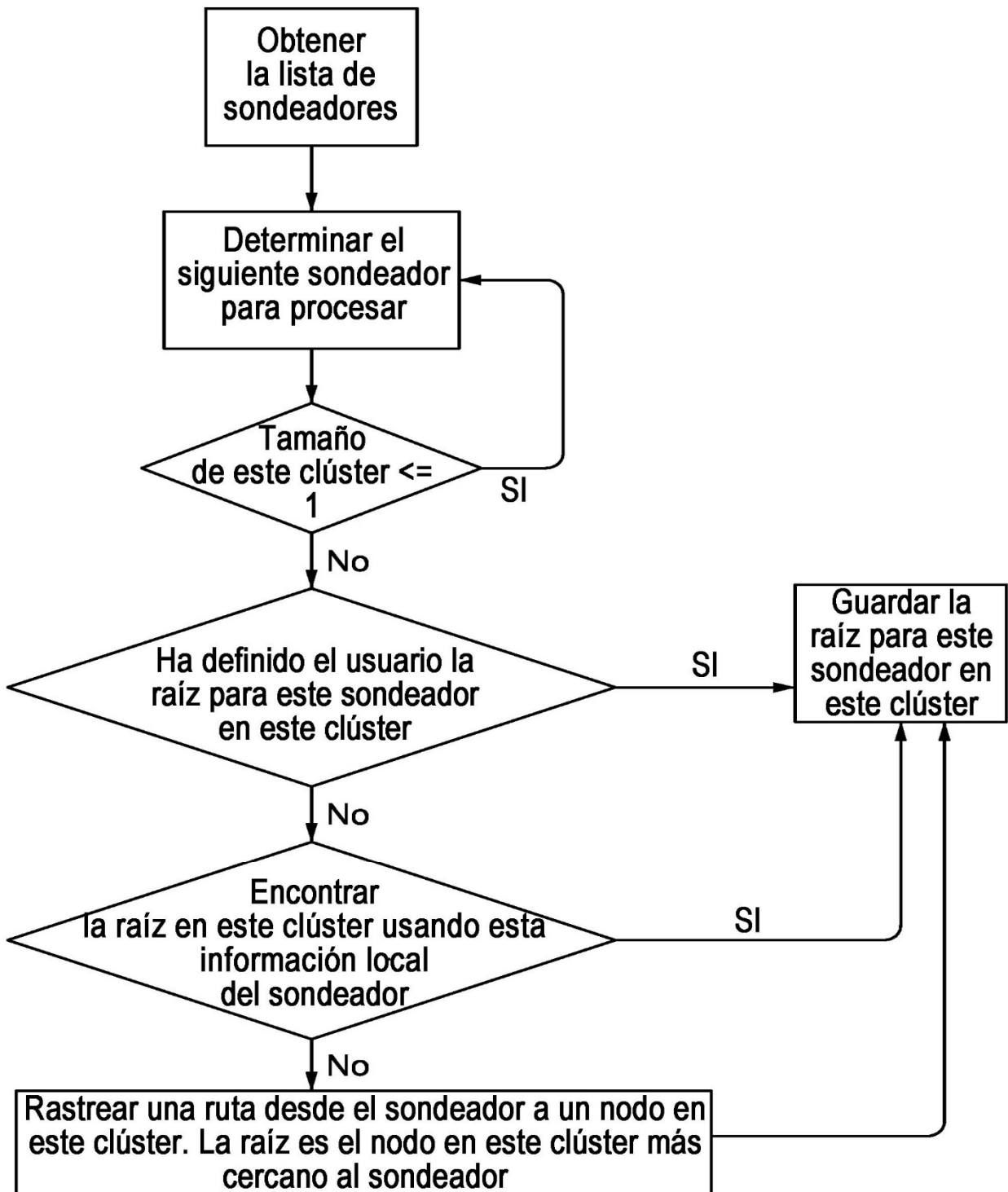


Fig. 3(c)

Determinar la raíz a partir de la información local del sondeador

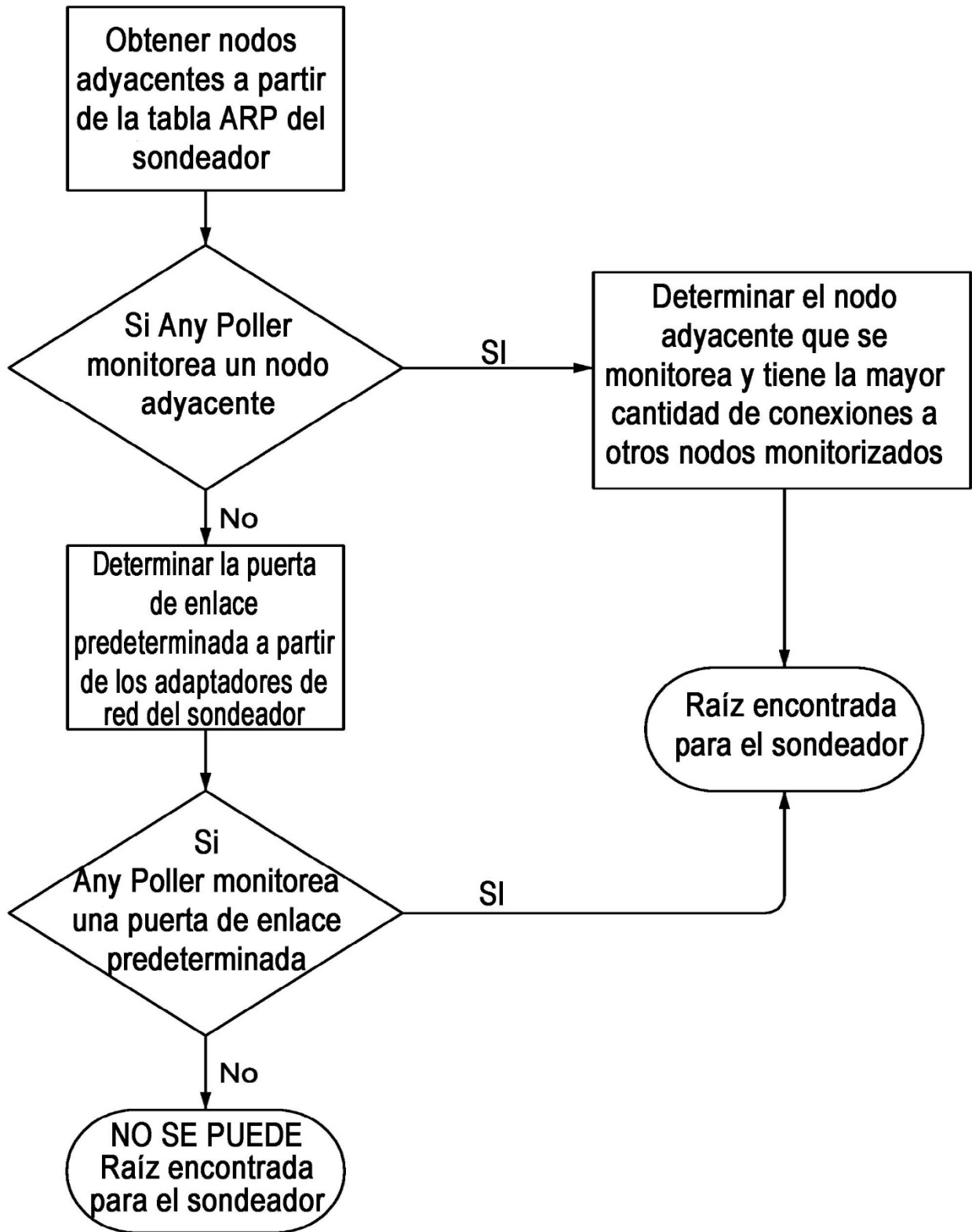


Fig. 3(d)

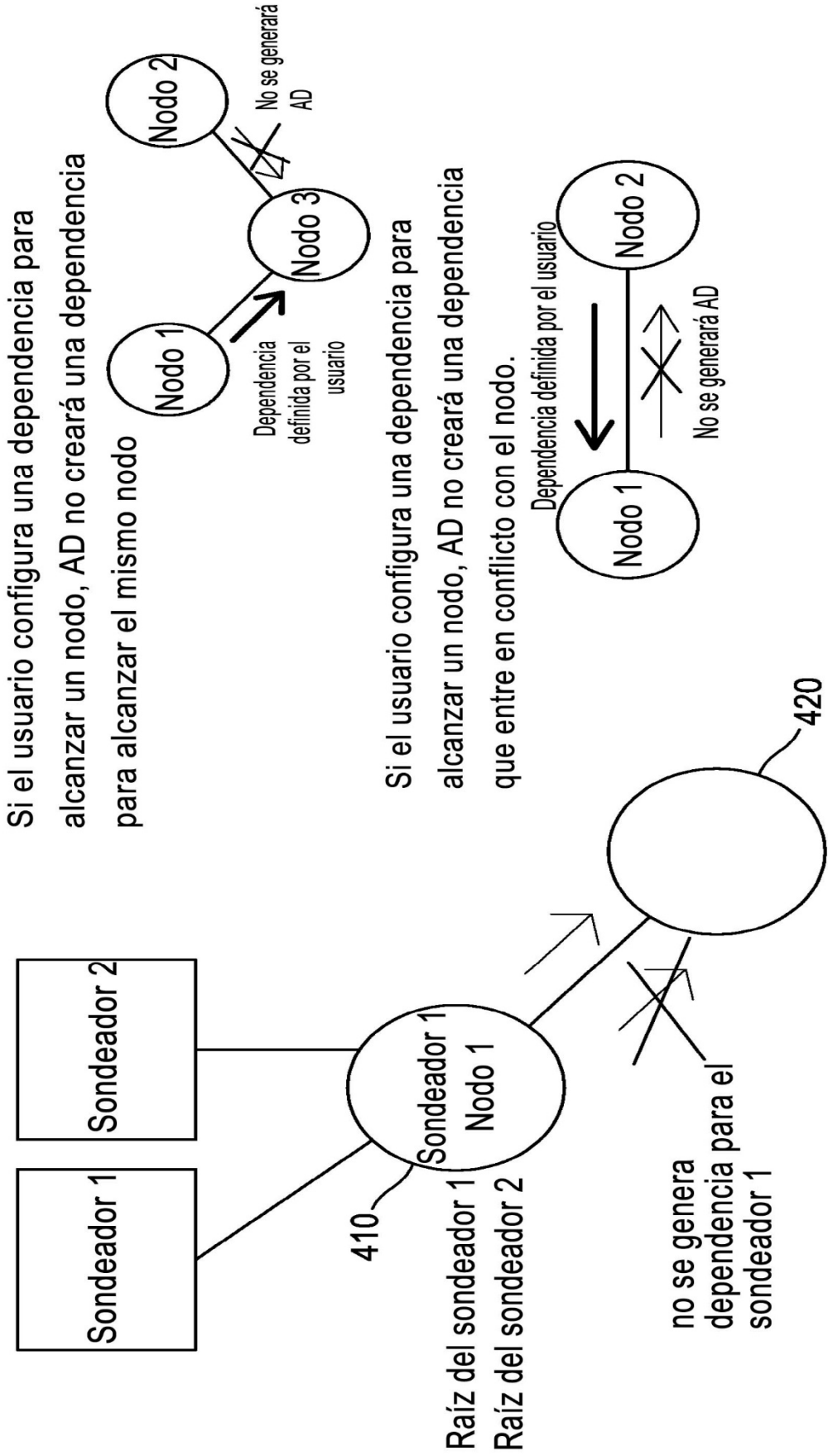


Fig. 4

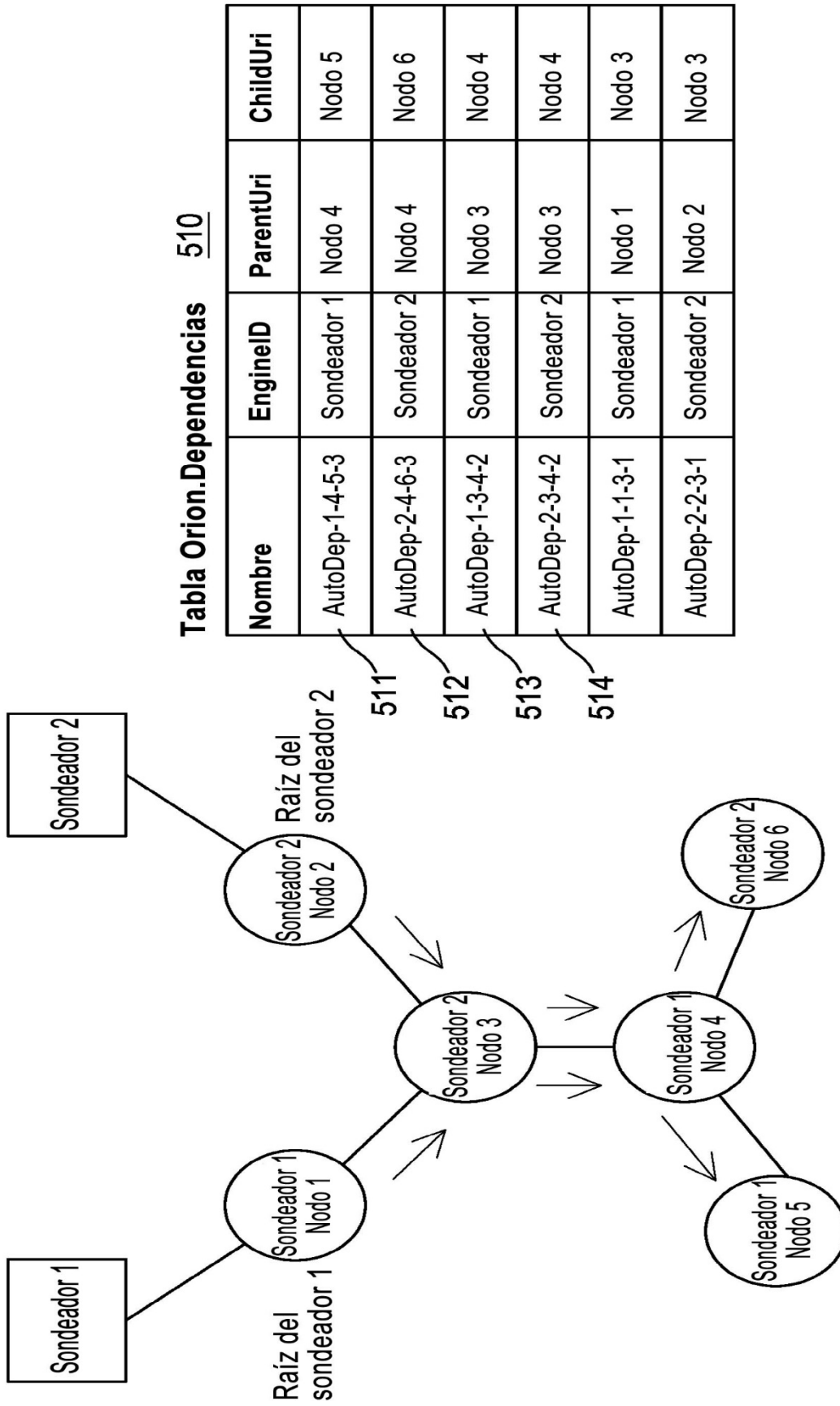


Tabla Orion.Dependencias 510

Nombre	EngineID	ParentUri	ChildUri
AutoDep-1-4-5-3	Sondeador 1	Nodo 4	Nodo 5
AutoDep-2-4-6-3	Sondeador 2	Nodo 4	Nodo 6
AutoDep-1-3-4-2	Sondeador 1	Nodo 3	Nodo 4
AutoDep-2-3-4-2	Sondeador 2	Nodo 3	Nodo 4
AutoDep-1-1-3-1	Sondeador 1	Nodo 1	Nodo 3
AutoDep-2-2-3-1	Sondeador 2	Nodo 2	Nodo 3

Fig. 5

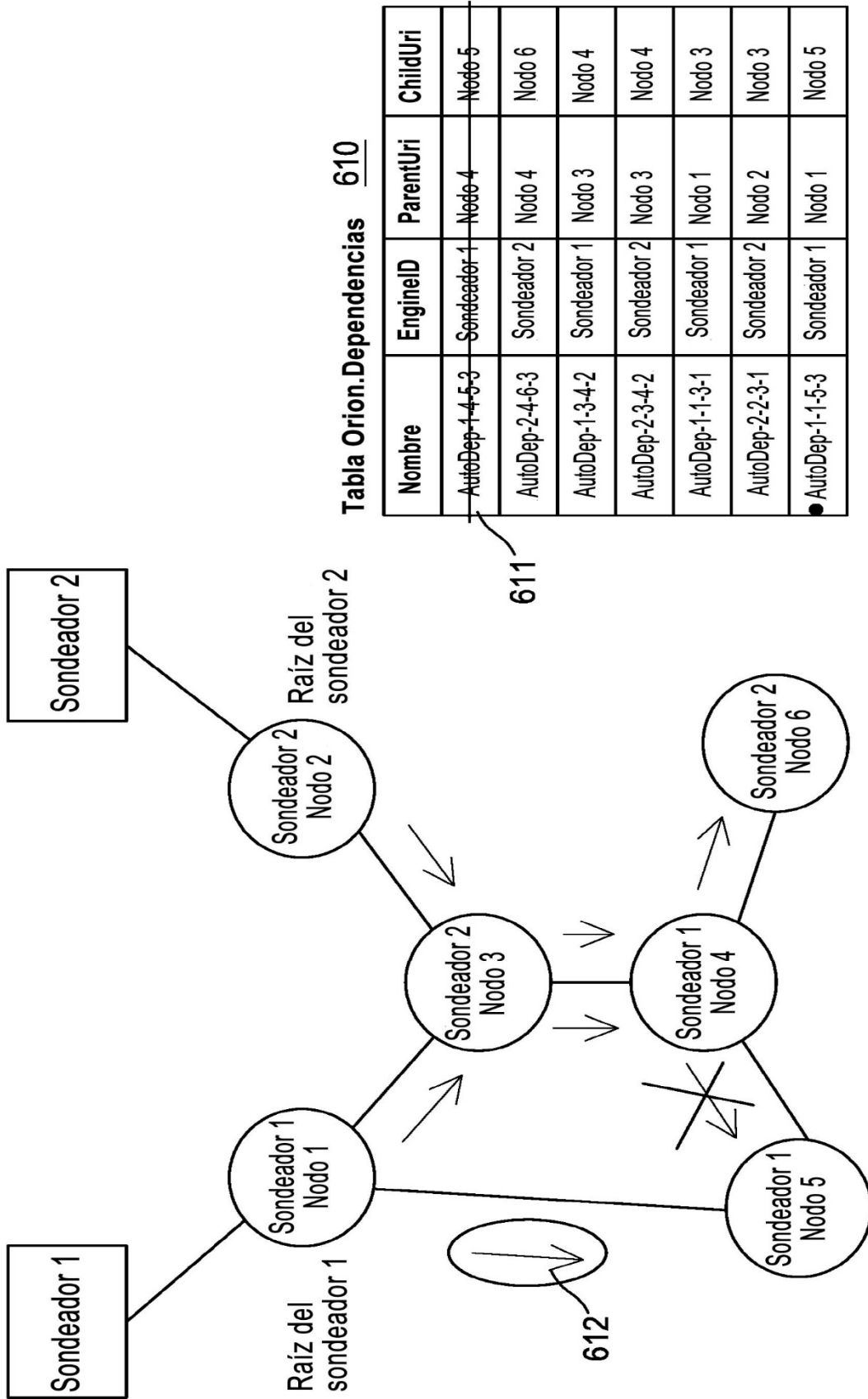


Fig. 6

720

Admin > [Gestionar dependencias](#)

Gestionar dependencias 710

¿Qué son las dependencias? Un elemento secundario depende de un elemento primario. El elemento secundario aparecerá "inalcanzable" en lugar de "inactivo" cuando el elemento primario esté inactivo.

¿Por qué usar dependencias? Evitar una avalancha de alertas inactivas cuando una interfaz/nodo central se cae, realizar informes más precisos. >Más información sobre dependencias

¿Cómo crear las dependencias? Puede crearlas manualmente o usar un cálculo periódico automático.

Calcular dependencias automáticamente:

Ignorar la autodependencia

Las dependencias automáticas seleccionadas se añadirán a la lista ignorar y no serán incluidas en el cálculo siguiente

Gestionar dependencias Gestionar las dependencias ignoradas

Grupo de: ▼

	Nombre de la dependencia	Elemento primario	Elemento secundario	Origen
<input type="checkbox"/>	dependencia FL-SNR-0300270-5	<input type="radio"/> FL-SNR-0300270-5	<input type="radio"/> Cisco 7500	Usuario definido
<input checked="" type="checkbox"/>	AutoDep-1-6-8-1	<input type="radio"/> stp-1841.lab.stp	<input type="radio"/> stp-3750stack.lab.ew	Automáticamente calculado


730

Fig. 7

Cálculos y umbrales		<u>810</u>
Cálculo de la disponibilidad (avanzado)	Estado del nodo	<input type="button" value="v"/>
Cálculo inicial (avanzado)		<input checked="" type="checkbox"/>
Habilitar autodependencias		<input checked="" type="checkbox"/>

Admin ▶ Gestionar dependencias ▶

Gestionar dependencias

 **¿Qué son las dependencias?** Un elemento secundario depende de un elemento primario.
¿Por qué usar dependencias? Evitar una avalancha de alertas inactivas cuando

¿Cómo crear las dependencias? Puede crearlas manualmente o

Calcular dependencias automáticamente

Fig. 8

Gestionar dependencias

¿Qué son las dependencias? Un elemento secundario depende de un elemento primario. El elemento secundario aparecerá "inalcanzable" en lugar de "inactivo" cuando el elemento primario esté inactivo.
 ¿Por qué usar dependencias? Evitar una avalancha de alertas inactivas cuando una interfaz/nodo central se cae, realizar informes más precisos. >Más información sobre dependencias

¿Cómo crear las dependencias? Puede crearlas manualmente o usar un cálculo periódico automático de la topología de red.
 Calcular dependencias automáticamente

Gestionar dependencias | Gestionar las dependencias ignoradas

Grupo de: [Sin agrupación]

Nombre de la dependencia	Elemento primario	Elemento secundario	Origen
<input checked="" type="checkbox"/> AutoDep-1-544-584-3	<input type="radio"/> 10.199.6.88	<input type="radio"/> stp-6009	Calculado automáticamente
<input checked="" type="checkbox"/> AutoDep-1-587-687-2	<input type="radio"/> SIP-HP4202	<input type="radio"/> 10.199.6.194	Calculado automáticamente
<input checked="" type="checkbox"/> AutoDep-1-587-670-2	<input type="radio"/> SIP-HP4202	<input type="radio"/> 10.199.6.140	Calculado automáticamente
<input checked="" type="checkbox"/> AutoDep-1-587-669-2	<input type="radio"/> SIP-HP4202	<input type="radio"/> 10.199.6.139	Calculado automáticamente
<input checked="" type="checkbox"/> AutoDep-1-585-606-2	<input type="radio"/> SIP-Foundry-624	<input type="radio"/> stp-hpux11-itan	Calculado automáticamente
<input checked="" type="checkbox"/> AutoDep-1-584-644-1	<input type="radio"/> stp-6009	<input type="radio"/> 10.199.6.88	Calculado automáticamente
<input checked="" type="checkbox"/> AutoDep-1-580-728-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> RMC	Calculado automáticamente
<input checked="" type="checkbox"/> AutoDep-1-580-722-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> NABLE-2K8-64	Calculado automáticamente
<input checked="" type="checkbox"/> AutoDep-1-580-721-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> NABLE-2K8-32	mente
<input checked="" type="checkbox"/> AutoDep-1-580-719-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> NABLE-2K8-32	mente
<input checked="" type="checkbox"/> AutoDep-1-580-717-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> NABLE-2K8-32	mente
<input checked="" type="checkbox"/> AutoDep-1-580-716-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> LAB-2K3-HP	mente
<input checked="" type="checkbox"/> AutoDep-1-580-714-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> LAB-NABLE-2K	mente
<input checked="" type="checkbox"/> AutoDep-1-580-699-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> 10.199.6.230	mente
<input checked="" type="checkbox"/> AutoDep-1-580-698-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> NuDesign SNM	mente
<input checked="" type="checkbox"/> AutoDep-1-580-661-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> NuDesign SNM	mente
<input checked="" type="checkbox"/> AutoDep-1-580-661-1	<input type="radio"/> stp-1841.lab-stp	<input type="radio"/> LAB-AUS-WOLTST.lab.exe	Calculado automáticamente

El nodo está activo.
 Dirección IP de sondeo: 10.199.6.238
 Tipo de máquina: Servidor Windows 2003
 Tiempo de resp. promedio: 2 ms
 Pérdida de paquetes: 0 %
 Carga CPU: 1 %
 Memoria usada: 71 %

1 de 7 | |

Visualización de dependencias 1-20 de 135. Número de elementos por página: 20

Fig. 9 915

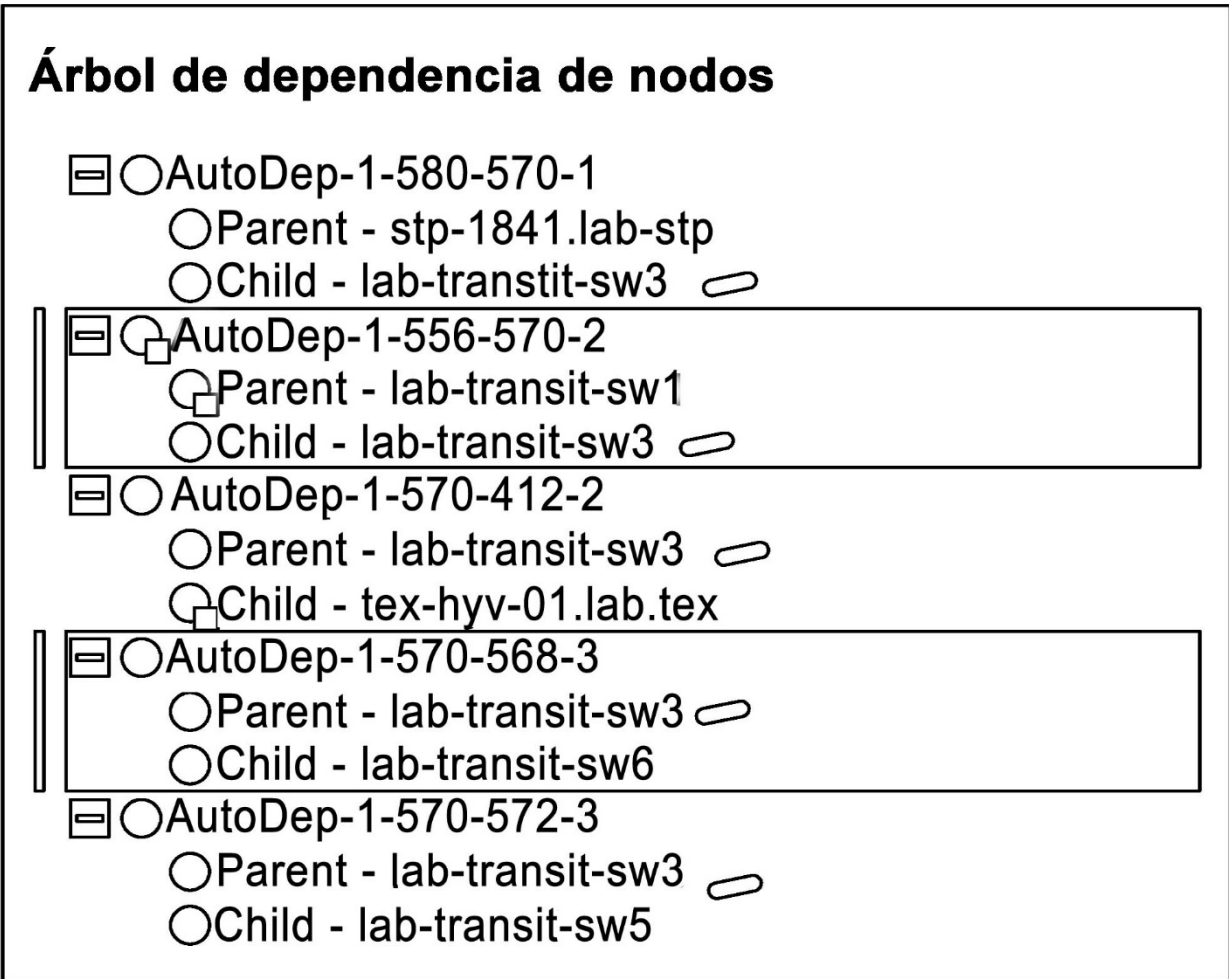


Fig. 10

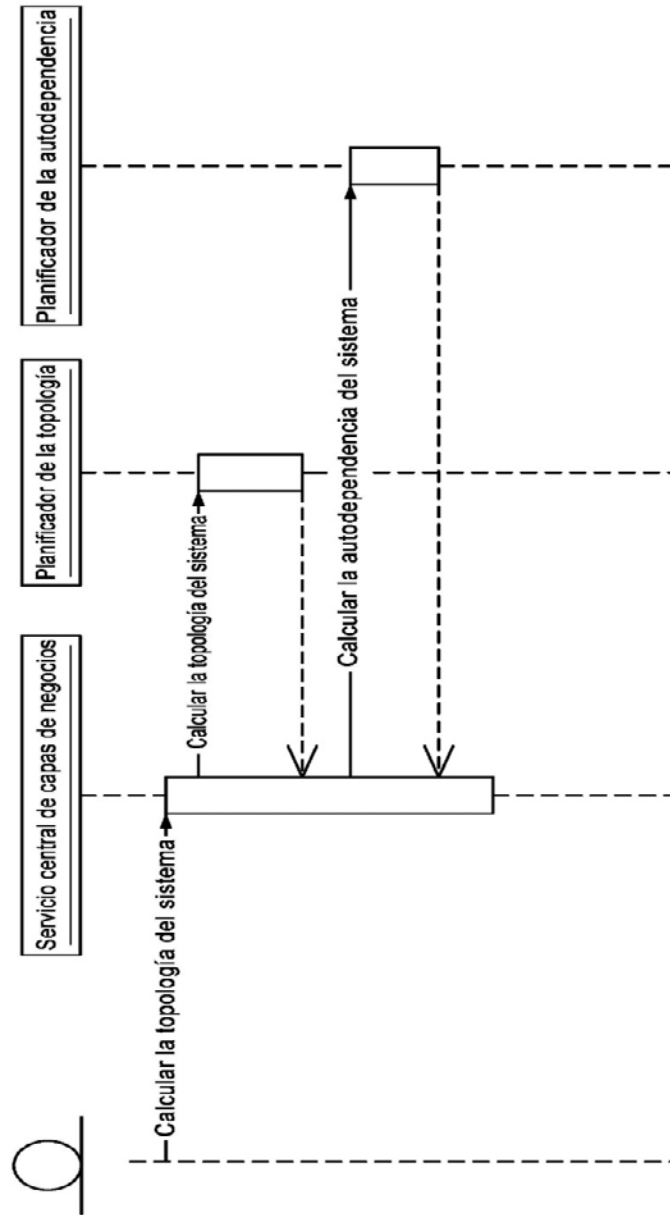


Fig. 11

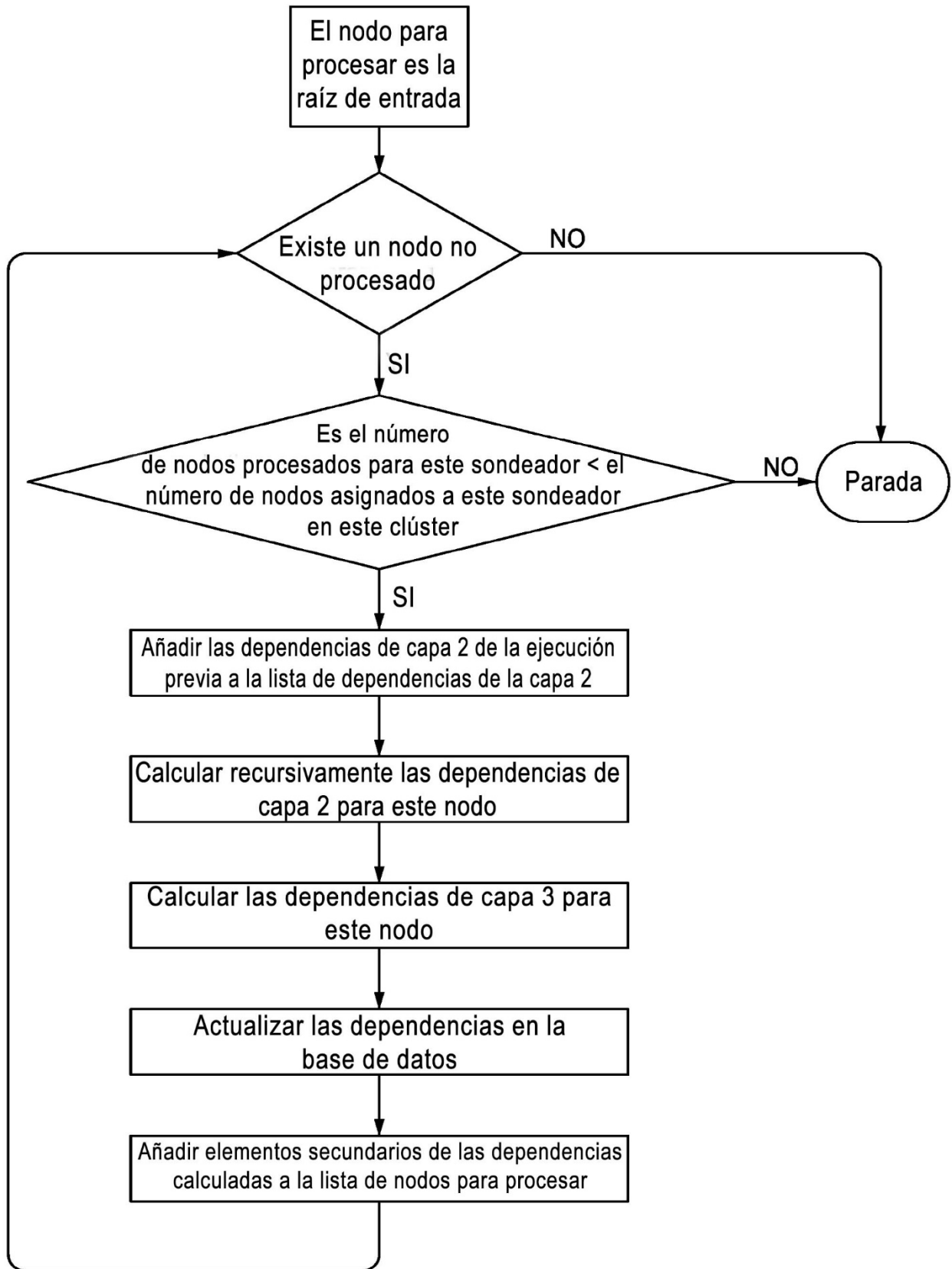


Fig. 12

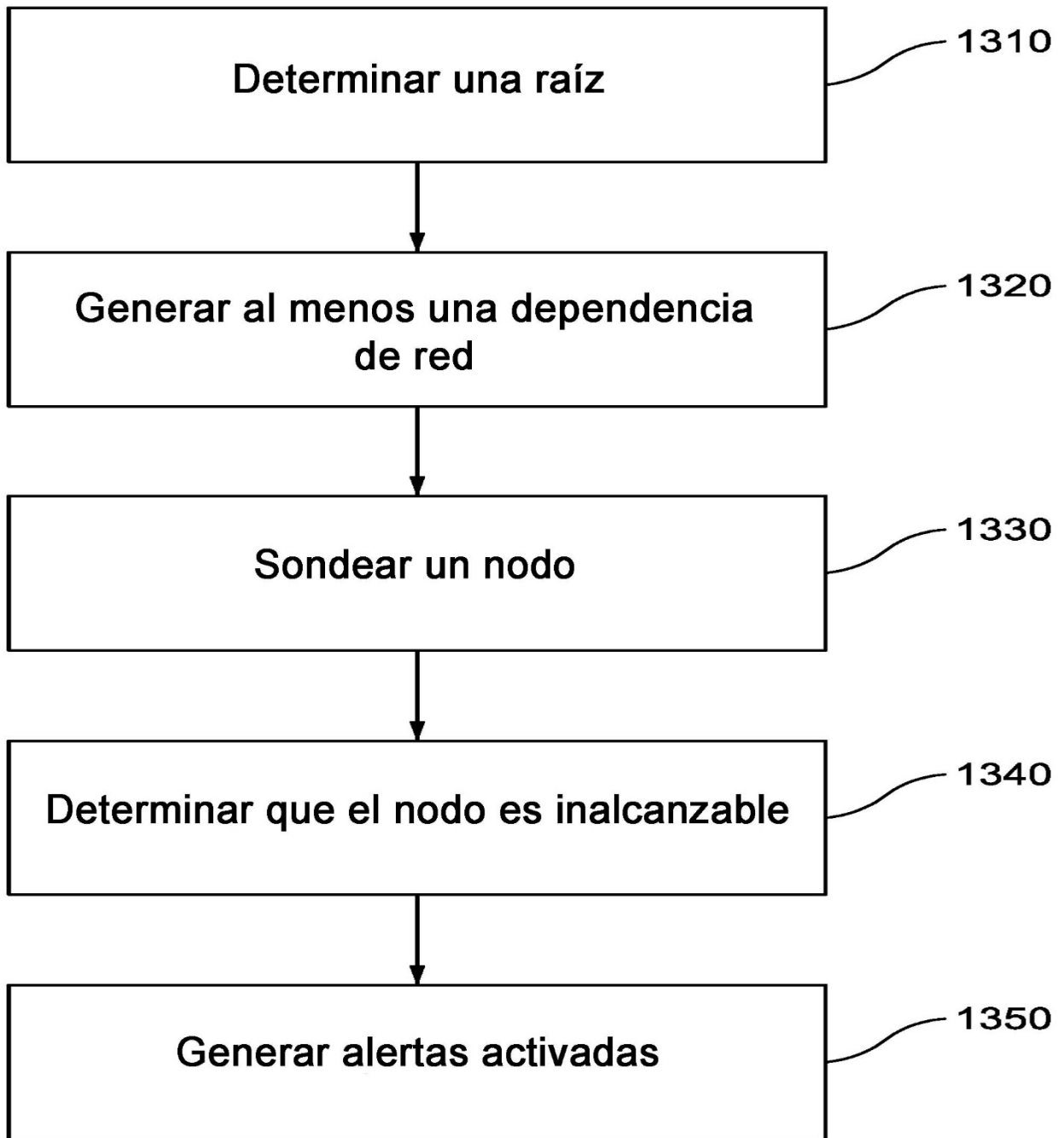


Fig. 13

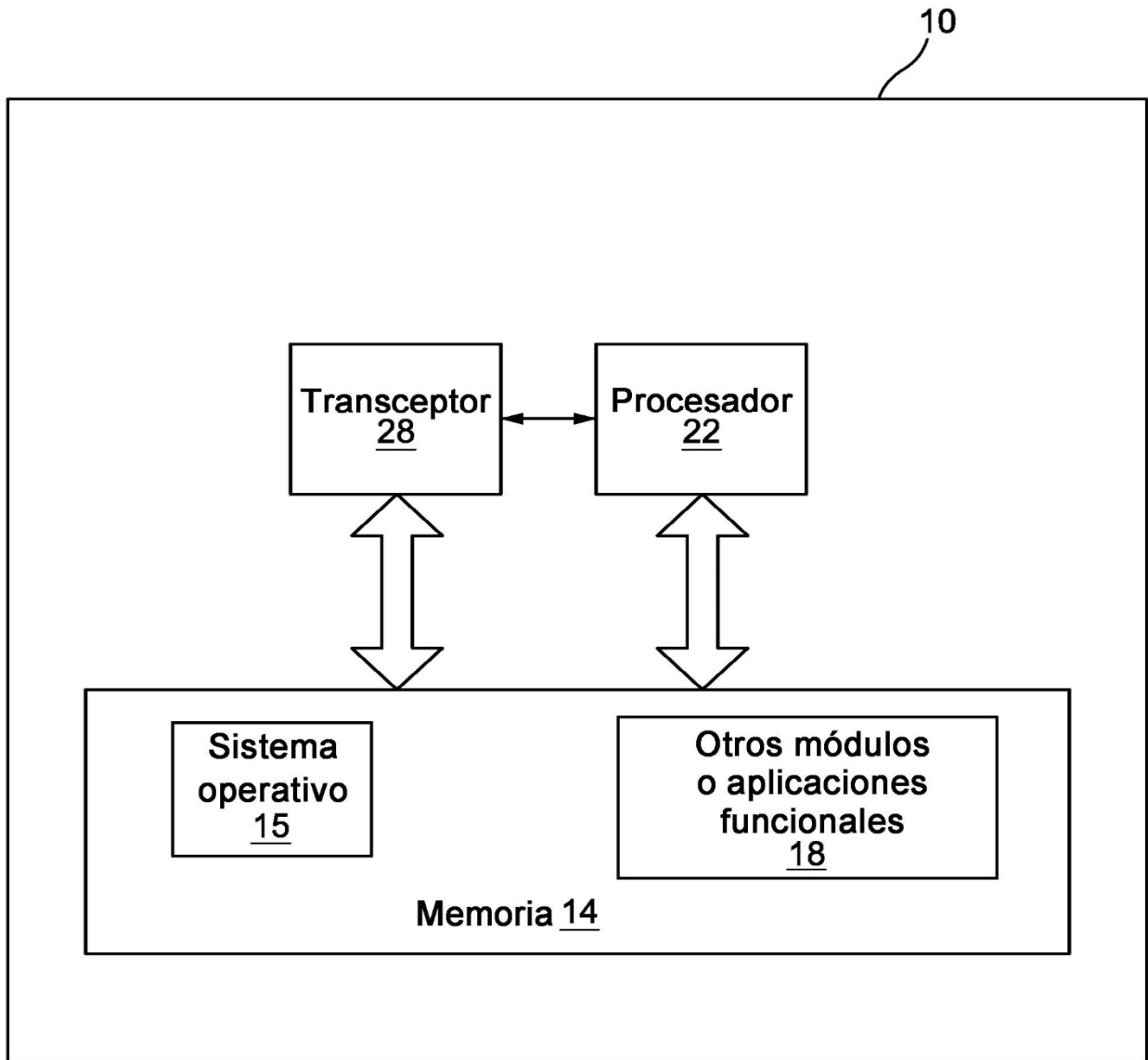


Fig. 14

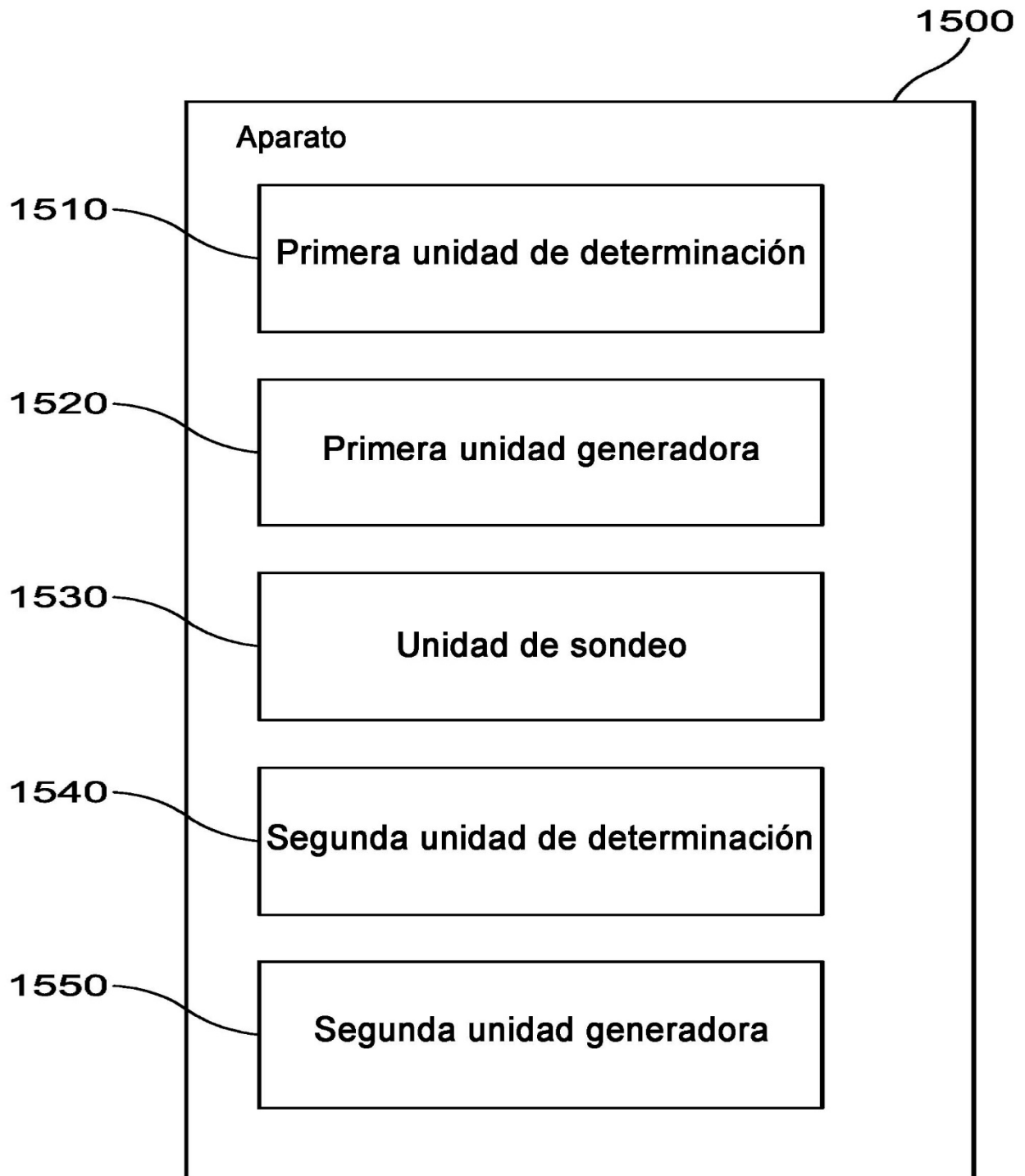


Fig. 15