(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 432 793**

(51) Int. CI.:

H04L 12/24 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **06.06.2008 E 08757628 (6)**

(97) Fecha y número de publicación de la concesión europea: **14.08.2013 EP 2166699**

(54) Título: **Un método y un sistema para enviar las notificaciones de eventos basadas en NETCONF**

(30) Prioridad:

06.07.2007 CN 200710126058

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
05.12.2013

(73) Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building Bantian
Longgang District, Shenzhen
Guangdong 518129 , CN

(72) Inventor/es:

JI, XIAOFENG

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

DESCRIPCIÓN

Un método y un sistema para enviar las notificaciones de eventos basadas en NETCONF.

Campo de la invención

La presente invención está relacionada con una tecnología de las comunicaciones móviles, y en particular, con un método y un sistema para enviar una notificación basada en NETCONF.

Antecedentes de la invención

Teniendo en cuenta los muchos defectos del Protocolo de Gestión de Red Simplificado (SNMP), en la técnica anterior se ha propuesto un Protocolo de Configuración de Red (NETCONF). NETCONF básicamente supera los defectos del SNMP y es aplicable a la gestión de un gran número de dispositivos complejos. NETCONF es un protocolo de configuración para gestión mediante capas. Proporciona mecanismos como, por ejemplo, configuración, instalación, mantenimiento y eliminación de dispositivos de red. NETCONF utiliza un Lenguaje Extensible mediante Etiquetas (XML) para encapsular datos de configuración. Las operaciones del protocolo tienen lugar en la capa de la Llamada a Procedimiento Remoto (RPC) simple.

La FIG. 1 muestra una estructura por capas de NETCONF en la técnica anterior. Tal y como se muestra en la FIG. 1, el protocolo NETCONF se divide conceptualmente en cuatro capas. La primera capa es la capa del protocolo de transporte. En la capa del protocolo de transporte, los protocolos comunes incluyen el Protocolo Extensible de Intercambio de Bloques (BEEP), el Terminal Seguro (SSH), la Capa de Conexión Segura (SSL), y la consola. La segunda capa es la capa RPC, la cual incluye una RPC y una respuesta a la RPC. La tercera capa es la capa de operaciones. En la capa de operaciones, las operaciones comunes incluyen Get-Config (Obtener configuración), Edit-config (Editar configuración), y notificación. La cuarta capa es la capa de contenidos (también denominada la capa de gestión de configuración), la cual se utiliza para gestionar los datos de configuración.

En el protocolo NETCONF, el sistema necesita proporcionar notificaciones de modo que los usuarios conozcan a tiempo los eventos que suceden en el sistema. Los eventos se refieren a eventos que suceden en ciertos dispositivos y deberían detectarse como, por ejemplo, cambio de configuración, fallo, cambio de estado, traspaso de umbral, e intrusión anómala. Por lo tanto, un cliente NETCONF tiene que suscribirse a las notificaciones procedentes de un servidor NETCONF. Si la suscripción se realiza con éxito, el servidor NETCONF registra en una lista de suscripciones el mensaje de suscripción del cliente NETCONF. Cuando se produce un evento en el sistema, el servidor NETCONF envía al cliente NETCONF una notificación de acuerdo con el mensaje de suscripción registrado para informar al cliente NETCONF de la ocurrencia. Si sucede alguno de los siguientes casos, el servidor NETCONF no envía una notificación al cliente NETCONF: se termina una sesión NETCONF; el evento se encuentra fuera del ámbito de la suscripción; la suscripción al evento cambia. Se debe tener en cuenta que las peticiones RPC nunca se procesan en una sesión utilizada para la notificación de eventos en el servidor NETCONF.

La notificación anterior está definida únicamente de acuerdo con las tres capas inferiores de NETCONF, y en la técnica anterior no se ha definido ningún modo de notificación de acuerdo con la estructura de cuatro capas completa basada en NETCONF. Por lo tanto, se espera que en otros entornos de aplicación de la técnica anterior se adoptará un modo de notificación parecido para implementar notificaciones basadas en NETCONF.

El documento del IETF draft-ietf-netconf-notification-02 (Notificaciones de Eventos NETCONF) describe un esquema para enviar mensajes asíncronos, o notificaciones de eventos en NETCONF. Define tanto las operaciones necesarias para soportar este concepto como también trata las implicaciones para la asociación con protocolos de transporte.

Resumen de la invención

La presente invención proporciona un método para enviar una notificación basada en NETCONF, y el método es compatible con varios modos de notificación para implementar notificaciones basadas en NETCONF.

La presente invención también proporciona un sistema para enviar una notificación basada en NETCONF, y el sistema es compatible con varios modos de notificación para implementar notificaciones basadas en NETCONF.

La presente invención proporciona, además, un dispositivo para enviar una notificación basada en NETCONF, y el dispositivo es compatible con varios modos de notificación para implementar notificaciones basadas en NETCONF.

La presente invención proporciona, además, un dispositivo para recibir una notificación basada en NETCONF, y el dispositivo es compatible con varios modos de notificación para implementar notificaciones basadas en NETCONF.

La solución técnica de la presente invención es del siguiente modo:

De acuerdo con un primer aspecto de la invención, el método para enviar una notificación basada en NETCONF incluye: establecer una sesión basada en el Protocolo de Control de Transporte, TCP, entre un cliente NETCONF y un servidor NETCONF; aceptar, por parte del servidor NETCONF, una suscripción de notificaciones del cliente

NETCONF, en donde la aceptación de la suscripción de notificaciones del cliente NETCONF comprende: recibir, por parte del servidor NETCONF, una petición de suscripción enviada por el cliente NETCONF, en donde la petición de suscripción incluye información de requisitos de formato y un tipo de notificaciones de suscripción a las que el cliente NETCONF se quiere suscribir; y añadir, por parte del servidor NETCONF, la petición de suscripción a una lista de suscripciones configurada previamente después de que la suscripción se haya realizado con éxito; generar, por parte del servidor NETCONF, una notificación cuando suceda un evento en el sistema, y convertir la notificación en un formato que pueda ser identificado por el cliente NETCONF de acuerdo con la suscripción de notificaciones del cliente NETCONF; y enviar, por parte del servidor NETCONF, la notificación convertida al cliente NETCONF; en donde la conversión de la notificación en el formato que puede ser identificado por el cliente NETCONF comprende: encapsular, por parte del servidor NETCONF, la notificación en un formato de Lenguaje de Etiquetas Extensible, XML, si la petición de suscripción no incluye información de requisitos de formato; o si la petición de suscripción incluye, además, la información de requisitos de formato, encapsular, por parte del servidor NETCONF, la notificación en el formato XML y convertir la notificación encapsulada en un formato solicitado en la información de requisitos de formato del cliente NETCONF. En donde el tipo de las notificaciones suscritas es una Captura del Protocolo Simple de Gestión de Redes, SnmpTrap, o syslog (registro del sistema); en donde si el tipo de la notificación es SnmpTrap, la encapsulación de la notificación en el formato XML comprende: encapsular, por parte del servidor NETCONF, la información incluida en la notificación SnmpTrap en un contenido de una etiqueta de acuerdo con un modelo de notificación basada en NETCONF, en donde el contenido encapsulado responde a una estructura jerárquica del modelo de notificación basada en NETCONF; el modelo de notificación tiene una estructura jerárquica que comprende una etiqueta, una subetiqueta y un contenido de la etiqueta; y la subetiqueta es una etiqueta de datos; si el tipo de la notificación es syslog, la encapsulación de la notificación en el formato XML comprende: encapsular, por parte del servidor NETCONF, la información incluida en el syslog en un contenido de una etiqueta de acuerdo con un modelo de notificación basada en NETCONF, en donde: el contenido encapsulado responde a una estructura jerárquica del modelo de notificación basada en NETCONF; el modelo de notificación tiene una estructura jerárquica que comprende una etiqueta, una subetiqueta y un contenido de la etiqueta; y la subetiqueta es una etiqueta de datos; en donde el syslog comprende un “PRI” (prioridad), un “HEADER” (cabecera), y un “MSG” (mensaje), en donde: el “PRI” comprende “MsgSource” (fuente del mensaje), fuente del registro e información “Severity” (gravedad); el “HEADER” comprende información de “timestamp” (marca de tiempo) y “agentaddr” (dirección del agente); y la información en el “MSG” se configura en función de los requisitos de los usuarios; y el método para que el servidor NETCONF encapsule en el contenido de la etiqueta la información incluida en el syslog comprende: asociar, por parte del servidor NETCONF, el syslog a una etiqueta y asociar la información incluida en el syslog a diferentes niveles de subetiquetas de la etiqueta del syslog de acuerdo con una relación jerárquica; y establecer, por parte del servidor NETCONF, una etiqueta como syslogs y establecer una o más etiquetas de syslog como subetiquetas de los syslogs.

De acuerdo con un segundo aspecto de la invención, el sistema para enviar una notificación basada en NETCONF incluye: un cliente NETCONF y un servidor NETCONF. El servidor NETCONF incluye: un manejador de sesiones, un gestor de notificaciones y un centro de eventos.

El manejador de sesiones está configurado para establecer una sesión basada en el Protocolo de Control de Transporte, TCP, entre un cliente NETCONF y el servidor NETCONF; aceptar una suscripción de notificaciones del cliente NETCONF.

El centro de eventos está configurado para generar una notificación cuando se produce un evento en el sistema y para enviar la notificación al gestor de notificaciones.

El gestor de notificaciones está configurado para almacenar en una lista de suscripción configurada previamente una suscripción de notificaciones recibida desde el manejador de sesiones; reenviar al manejador de sesiones, de acuerdo con la lista de suscripciones, la notificación enviada por el centro de eventos.

El manejador de sesiones está configurado, además, para convertir la notificación enviada por el gestor de notificaciones a un formato que pueda ser identificado por el cliente NETCONF, y para enviar al cliente NETCONF la notificación convertida, en donde la conversión del manejador de sesiones de la notificación al formato que pueda ser identificado por el cliente NETCONF comprende: si la petición de suscripción no incluye la información de requisitos de formato encapsular la notificación en el formato del Lenguaje de Etiquetas Extensible, XML; o si la petición de suscripción incluye la información de requisitos de formato encapsular la notificación en el formato XML y convertir la notificación encapsulada a un formato solicitado en la información de requisitos de formato del cliente NETCONF. En donde el tipo de notificaciones suscritas es una Captura del Protocolo Simple de Gestión de Redes, SnmpTrap, o syslog; en donde si el tipo de la notificación es SnmpTrap, la encapsulación de la notificación en el formato XML comprende: encapsular, por parte del servidor NETCONF, la información incluida en la notificación SnmpTrap en un contenido de una etiqueta de acuerdo con un modelo de notificación basada en NETCONF, en donde: el contenido encapsulado responde a una estructura jerárquica del modelo de notificación basada en NETCONF; el modelo de notificación tiene una estructura jerárquica que comprende una etiqueta, una subetiqueta y un contenido de la etiqueta; y la subetiqueta es una etiqueta de datos; si el tipo de la notificación es syslog, la encapsulación de la notificación en el formato XML comprende: encapsular, por parte del servidor NETCONF, la información incluida en el syslog en un contenido de una etiqueta de acuerdo con un modelo de notificación basada en NETCONF, en donde: el contenido encapsulado responde a una estructura jerárquica del modelo de notificación

basada en NETCONF; el modelo de notificación tiene una estructura jerárquica que comprende una etiqueta, una subetiqueta y un contenido de la etiqueta; y la subetiqueta es una etiqueta de datos; en donde el syslog comprende un “PRI”, un “HEADER”, y un “MSG”, en donde: el “PRI” comprende “MsgSource”, fuente del registro e información “Severity”; el “HEADER” comprende información de “timestamp” y “agentaddr”; y la información en el “MSG” está configurada en función de los requisitos de los usuarios; y el método para que el servidor NETCONF encapsule en el contenido de la etiqueta la información incluida en el syslog comprende: asociar, por parte del servidor NETCONF, el syslog a una etiqueta y asociar la información incluida en el syslog a diferentes niveles de subetiquetas de la etiqueta del syslog de acuerdo con una relación jerárquica; y establecer, por parte del servidor NETCONF, una etiqueta como syslogs y establecer una o más etiquetas de syslog como subetiquetas de los syslogs.

Con la solución técnica anterior, el servidor NETCONF puede generar una notificación del modo correspondiente según sea necesario cuando se produzca un evento en el sistema, convertir la notificación generada a un formato requerido por el cliente, y enviar al cliente NETCONF la notificación convertida. En el sistema completo se pueden convertir formatos de notificaciones. De este modo no existen restricciones estrictas sobre el formato de una notificación generada por el servidor NETCONF. Por lo tanto, las notificaciones basadas en NETCONF pueden ser compatibles con varios modos de notificaciones.

Las solución técnica de la presente invención se describe de aquí en adelante en detalle haciendo referencia a los dibujos adjuntos y a ejemplos de modos de realización.

Breve descripción de los dibujos

La FIG. 1 muestra una estructura de capas de NETCONF en la técnica anterior;

la FIG. 2 es un diagrama de flujo general de un método proporcionado de acuerdo con un modo de realización de la presente invención;

la FIG. 3 es un diagrama de flujo de un método proporcionado de acuerdo con un primer ejemplo de modo de realización de la presente invención;

la FIG. 4 muestra un modelo de notificación basada en NETCONF utilizado en los modos de realización de la presente invención;

la FIG. 5 muestra un modelo de encapsulación de SnmpTrap en el formato XML de acuerdo con un modo de realización de la presente invención;

la FIG. 6 muestra una regla de conversión entre XML-Snmpv2trap y XML-Snmpv1trap a través de la Transformación de Lenguaje de Hoja de Estilo Extensible (XSLT) de acuerdo con un modo de realización de la presente invención;

la FIG. 7 es un diagrama de flujo de un método proporcionado de acuerdo con un segundo ejemplo de modo de realización de la presente invención;

la FIG. 8 muestra un modelo de encapsulación de syslog en el formato XML de acuerdo con un modo de realización de la presente invención; y

la FIG. 9 muestra una estructura de un sistema proporcionado de acuerdo con un modo de realización de la presente invención.

Descripción detallada de la invención

Para explicar mejor el objetivo, la solución técnica y las ventajas de la presente invención, a continuación se describen en detalle modos de realización de la presente invención haciendo referencia a los dibujos adjuntos.

En la actualidad, los modos de notificación ampliamente utilizados incluyen los modos SnmpTrap y syslog. SnmpTrap se clasifica en tres versiones, esto es, Snmpv1Trap, Snmpv2Trap y Snmpv3Trap.

La RFC 1157 describe en detalle cómo enviar mensajes de alarma a una aplicación de la capa superior utilizando Snmpv1Trap. Además, la RFC 1157 define el formato de la Snmpv1Trappdu, tal como se describe en la Tabla 1.

Tabla 1 Formato de la Snmpv1Trappdu

version-ver1	community	PduData (datos de la Pdu)							
		eid	agentaddr	gid (generic-trap)	sid (specific-trap)	timestamp	vb[n]		
							name	value	…

donde:

“version-ver1” es utilizado para identificar la versión de la Snmpv1Trappdu.

“community” (comunidad) es utilizado por una aplicación de la capa superior para autenticar el ID del remitente de un mensaje de alarma.

“eid” es un código de empresa como, por ejemplo, 1011.

“agentaddr” es utilizado para identificar la dirección del Protocolo de Internet (IP) del remitente de un mensaje de alarma.

“gid” es utilizado para identificar seis casos comunes en los que es necesario notificar una alarma, incluyendo coldStart (inicio en frío), warmStart (inicio en caliente), linkDown (fallo de conexión), linkUp (establecimiento de conexión), authenticationFailure (fallo de autenticación), y egpNeighborLoss (fallo o apagado del router externo adyacente). En una aplicación real, se utilizan los enteros 0-5 para representar los seis casos anteriores.

“sid” es utilizado para identificar un caso especial en el que es necesario notificar una alarma como, por ejemplo, un caso definido por los usuarios.

“timestamp” es utilizado para identificar el instante ocurrencia de la alarma.

“vb[n]”, el cual es la abreviatura de enlace variable, es utilizado para identificar el contenido específico de la alarma. Un mensaje de alarma puede incluir múltiples vb.

“name” (nombre) es utilizado para identificar un objeto de alarma, como, por ejemplo, una interfaz o una Unidad de Procesamiento Central (CPU).

“value” (valor) es utilizado para identificar el contenido del evento como, por ejemplo, interfaz caído o sobrecargado.

La RFC 1905 describe cómo enviar mensajes de alarma a una aplicación de la capa superior utilizando Snmpv2Trap. Además, la RFC 1905 define el formato específico de la Snmpv2Trappdu, tal como se describe en la Tabla 2.

Tabla 2 Formato de la Snmpv2Trappdu

		PduData						
		vb[0]		vb[1]		…	vb[n] (opcional)	
version-ver2c	community	sysupTime (utilizado para identificar el momento de inicio del sistema)	valor de sysuptime (momento de inicio del sistema)	trapoid (utilizado para identificar un objeto de alarma)	valor de trapoid (valor de una variable de alarma)	…	Eid	value

La mayor parte de los campos de la Tabla 2 son parecidos a los campos de la Tabla 1 y no se vuelven a describir.

El formato de la Snmpv3Trappdu es parecido al formato de la Snmpv2Trappdu. La diferencia es que, comparado con el formato de la Snmpv2Trappdu, en el formato de la Snmpv3Trappdu se añaden “contextname” (nombre del contexto) y “contextID” (ID del contexto).

Las alarmas se pueden notificar en el modo SnmpTrap anterior, pero el modo SnmpTrap está basado en el Protocolo de Datagramas de Usuario (UDP), y UDP no es un protocolo fiable. Por lo tanto, cuando se notifica una alarma en el modo SnmpTrap, los mensajes pueden no llegar de forma fiable. El modo Snmptrap utiliza mensajes PDU binarios para su encapsulación, y durante el análisis es necesario utilizar una biblioteca de funciones especiales. Por lo tanto, repercute en la universalidad del SnmpTrap. Además, el modo SnmpTrap es incompatible con otros modos de notificación.

La RFC 3164 describe cómo enviar registros del sistema a una aplicación de la capa superior utilizando un mensaje syslog. Además, la RFC 3164 define el formato específico de un mensaje syslog, tal como se describe en la Tabla 3.

Tabla 3 Formato de un Mensaje Syslog

PRI (Facility x 8 + Severity = PRI)		HEADER		MSG			
Facility	Severity	timestamp	hostname				

Como se ha descrito en la Tabla 3, un mensaje de syslog completo está formado por tres partes, esto es, “PRI” (prioridad), “HEADER” (cabecera del mensaje) y “MSG” (contenido del mensaje).

“Facility” (origen) es utilizado para identificar la fuente del registro. En una aplicación real, se utilizan códigos numéricos diferentes para representar diferentes fuentes del registro, tal como se describe en la Tabla 4.

Tabla 4 Asociación Entre Códigos Numéricos y Facility

Código numérico	Facility
0	Mensajes del núcleo
1	Mensajes de nivel de usuario
2	Sistema de correo
3	Demonios del sistema
4	Mensajes de Seguridad/Autorización (nota 1)
5	Mensajes generados internamente por syslogd
6	Subsistema de impresora de línea
7	Subsistema de noticias de red
8	Subsistema UUCP
9	Demonio de reloj (nota 2)
10	Mensajes de Seguridad/Autorización (nota 1)
11	Demonio FTP
12	Subsistema NTP
13	Auditoría de registro (nota 1)
14	Alerta de registro (nota 1)
15	Demonio de reloj (nota 2)
16	Uso local 0 (local0)
. . .	. . .
23	Uso local 7 (local7)

“Severity” es utilizado para identificar la gravedad de un mensaje. En una aplicación real, diferentes códigos numéricos representan diferentes niveles de gravedad, como se describe en la Tabla 5

Tabla 5 Asociación Entre Códigos Numéricos y Severity

Código numérico	Severity
0	Emergencia: el sistema está inutilizado
1	Alerta: la acción se debe realizar de forma inmediata

2	Crítico: condiciones críticas
3	Error: condiciones de error
4	Advertencia: condiciones de advertencia
5	Aviso: normal pero situación importante
6	Información: mensajes de información
7	Depuración: mensajes de nivel de depuración

Después de haber especificado “Facility” y “Severity”, se puede calcular la prioridad del mensaje de syslog utilizando la fórmula “Facility x 8 + Severity = PRI”. En la fórmula, 8 es un coeficiente con un valor fijo.

“HEADER” incluye “timestamp” y “hostname” (nombre del equipo). El “timestamp” se utiliza para identificar el momento de generación del registro. El “hostname” se utiliza para identificar el sistema que genera el registro.

El contenido de “MSG” tiene que determinarse de acuerdo con la situación real y, por lo tanto, no está restringido por el formato del mensaje de syslog.

Por ejemplo:

<34>Oct 11 22:14:15 mymachine su: ‘su root’ failed for lonvick on /dev/pts/8

Facility=4 (mensajes de seguridad/autorización (nota 1)), Severity=2 (Crítico: condiciones críticas)

El significado del ejemplo anterior es como sigue: un dispositivo notifica al servidor de registro que un usuario se ha autenticado como superusuario pero ha fallado la autenticación; eso es un evento de autenticación de seguridad y, por lo tanto, el valor de “Facility” es 4; este evento es muy importante pero no afecta al funcionamiento normal del dispositivo y, por lo tanto, el valor de “Severity” es 2.

De forma parecida al modo SnmpTrap, el modo syslog puede reportar registros del sistema, pero los mensajes se transportan sobre UDP. Como UDP no es un protocolo fiable, los mensajes pueden no llegar de forma fiable. El modo syslog es incompatible con otros modos de notificación. Además, el modo syslog adopta encapsulación de texto, pero el texto se encuentra en un formato simple de cadena de caracteres y no se puede analizar. Por lo tanto se ve afectada la universalidad del syslog.

Como se ha mencionado más arriba, en la técnica anterior no se ha definido ningún modo de notificación de acuerdo con la estructura completa de cuatro capas basada en NETCONF. Por lo tanto, en la técnica anterior se espera que para implementar notificaciones basadas en NETCONF en otros entornos de aplicación se adopte un modo de notificación parecido. Los modos SnmpTrap y syslog tienen sus propios defectos y el protocolo adoptado y el formato del mensaje son diferentes de los de NETCONF. Por lo tanto, en la técnica anterior los modos SnmpTrap y syslog no se pueden aplicar de forma sencilla a NETCONF.

Los modos de realización de la presente invención se basan en NETCONF. En la presente solicitud, un cliente NETCONF (denominado de aquí en adelante como cliente) se suscribe a notificaciones desde un servidor NETCONF (denominado de aquí en adelante como servidor); cuando en el sistema se produce un evento el servidor genera una notificación, convierte la notificación generada a un formato que pueda ser identificado por el cliente y envía al cliente la notificación convertida.

La FIG. 2 es un diagrama de flujo general de un método proporcionado en un modo de realización de la presente invención. La FIG. 2 incluye los siguientes pasos:

Paso 201: un cliente se suscribe para recibir notificaciones desde un servidor.

En este paso, un método para que el cliente se suscriba para recibir notificaciones desde el servidor es como sigue: el cliente envía una petición de suscripción que incluye el tipo de notificaciones a las que el cliente se quiere suscribir. Después de que la suscripción se haya realizado con éxito, el servidor añade la petición de suscripción a una lista de suscripciones configurada previamente. La lista de suscripciones almacena información sobre clientes que se suscriben a notificaciones tal como el tipo de notificaciones suscritas. La lista de suscripciones también puede almacenar otra información, si dicha información existe. Además, el servidor genera una respuesta que indica si la suscripción se ha realizado con éxito, convierte la respuesta al formato XML y envía al cliente la respuesta convertida.

Antes de este paso, el método incluye, además: establecer entre el cliente y el servidor una sesión basada en TCP (tal como una sesión SSH).

Paso 202: cuando en el sistema se produce un evento, el servidor genera una notificación, convierte la notificación a un formato que pueda ser identificado por el cliente y envía al cliente la notificación convertida.

Cuando en el sistema se produce un evento, el servidor genera una notificación del tipo correspondiente de acuerdo con el tipo del evento, localiza al cliente que se ha suscrito a notificaciones de ese tipo de acuerdo con el tipo de notificaciones suscritas almacenadas en la lista de suscripciones, y envía la notificación al cliente.

En general, si el cliente no tiene un requisito especial, el servidor encapsula la notificación en el formato XML de acuerdo con el modo por defecto, y a continuación envía la notificación encapsulada al cliente.

Si el cliente quiere recibir una notificación en cualquier otro formato, en el paso 201, la solicitud de suscripción enviada incluye, además, información de requisitos de formato. En consecuencia, en este paso el servidor convierte, además, la notificación encapsulada en el formato XML a un formato solicitado por el cliente.

Las notificaciones anteriores se pueden realizar en varios modos de notificación en la técnica anterior como, por ejemplo, SnmpTrap o syslog. En este caso, la petición de suscripción del paso 201 puede incluir, además, un tipo de filtro y un resultado de filtro requerido. El tipo de filtro incluye un filtro SUBTREE (subárbol) y un filtro del Lenguaje de la Ruta XML (Xpath). El tipo de filtro indica un modo de filtrado adoptado por el servidor para obtener un resultado del filtrado requerido por el cliente. El resultado del filtrado requerido depende del tipo de notificaciones a las que se suscribe el cliente. Si el tipo de las notificaciones suscritas es SnmpTrap, el resultado del filtrado requerido describe el tipo de SnmpTrap específico de las notificaciones suscritas, esto es, Snmpv1Trap, Snmpv2Trap o Snmpv3Trap. Si el tipo de las notificaciones suscritas es syslog, el resultado del filtrado requerido describe la prioridad o el rango de prioridades de las notificaciones suscritas.

En consecuencia, en el paso 202, el servidor obtiene una notificación a la que el cliente se ha suscrito realmente filtrando en función del tipo de filtro almacenado en la lista de suscripciones y el resultado del filtrado requerido.

A continuación, además, se describe en detalle un método proporcionado en la presente invención a través de un ejemplo de modo de realización.

La FIG. 3 es un diagrama de flujo de un método proporcionado en el primer ejemplo de modo de realización de la presente invención. Este modo de realización está basado en NETCONF. Se asume que el tipo de notificación generada por un servidor es SnmpTrap. El servidor incluye un manejador de sesiones, un gestor de notificaciones y un centro de eventos.

Tal como se muestra en la FIG. 3, A-G son distintos hilos de procesamiento. A es un hilo de procesamiento del cliente; B es un hilo de recepción y procesamiento de mensajes de sesiones del servidor; C es un hilo de realimentación y procesamiento de mensajes de sesiones del servidor; E es un hilo de suscripción y procesamiento del servidor; F es un hilo de recepción de Capturas de sesiones del servidor; y G es una fuente de eventos de un dispositivo.

Como se muestra en la FIG. 3, este modo de realización incluye los siguientes pasos:

Un cliente establece una sesión SSH con el manejador de sesiones para el intercambio posterior de información entre el cliente y el manejador de sesiones. La sesión SSH se puede establecer de acuerdo con el protocolo asociado y no se describe aquí.

Paso 301: el cliente envía una petición de suscripción al manejador de sesiones.

La petición de suscripción incluye el tipo de notificaciones suscritas, un tipo de filtro y un resultado del filtrado requerido. En este modo de realización, si el tipo de las notificaciones a las que se suscribe el cliente es todos los eventos con la cadena “Snmptrap over netconf” (“Snmptrap sobre netconf”), y el tipo de filtro es el filtro SUBTREE, el resultado del filtrado requerido es un conjunto de eventos en el formato Snmpv1trap.

En la técnica anterior, se encuentra claramente definido el formato de una petición de suscripción para una notificación basada en NETCONF. La petición de suscripción enviada en este paso aplica únicamente a la técnica anterior. Por ejemplo:

<?xml version="1.0" encoding="UTF-8"?>

<rpc xmlns="urn: ietf:params:xml:ns:netconf:base:1.0"

xmlns:notify="urn:ietfparams:xml:ns:netconf:notification:1.0"

xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0"

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:chemaLocation="urn:ietf:params:xml:ns:netconf:notification:1.0
D:\project\XML\netconf\xsd\NC-Notification.xsd" message-id="101">
  <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <streams>
      <stream>snmpTrapPdu over netconf</stream>
    </streams>//Suscribir a todos los eventos con la cadena “snmptrap over netconf”
    <filter type="subtree">// El tipo de filtro es el filtro SUBTREE
      <top xmlns="http://huawei.com/comm/trap">
        <snmpTrapPdu>
          <snmpv1trap/>
        </snmpTrapPdu>//El resultado del filtro es un conjunto de eventos en el formato snmpv1trap
      </top>
    </filter>
    <named-profile>notification.profile</named-profile>
    <startTime>2001-12-24T12:23:45</startTime>
  </create-subscription>
</rpc>
```

Paso 302: el manejador de sesiones registra la petición de suscripción recibida en la cola de datos para ser procesada por el gestor de notificaciones.

Paso 303: el gestor de notificaciones procesa la petición de suscripción. Después de que la suscripción se haya realizado con éxito, el gestor de notificaciones añade la petición de suscripción a una lista de suscripciones configurada previamente.

Paso 304: el gestor de notificaciones envía una respuesta al manejador de sesiones.

En este paso, el gestor de notificaciones genera una respuesta que indica si la suscripción se ha realizado con éxito, y registra la respuesta en la cola de datos para ser procesada por el manejador de sesiones.

Paso 305: el manejador de sesiones convierte la respuesta al formato XML.

La respuesta generada en el paso 304 puede encontrarse en un formato de lenguaje de un dispositivo como, por ejemplo, el formato de lenguaje C. Por lo tanto, en este paso tiene que convertirse el formato de la respuesta.

Paso 306: el manejador de sesiones envía la respuesta al cliente.

El manejador de sesiones envía al cliente en el formato XML la respuesta generada que indica si la suscripción se ha realizado con éxito.

Paso 307: el centro de eventos envía una información de alarma SnmpTrap al gestor de notificaciones.

Cuando se produce un evento en el sistema, el centro de eventos genera una notificación del tipo correspondiente de acuerdo con el tipo del evento. Por ejemplo, si el evento falla al ser autenticado, el centro de eventos genera una información de alarma Snmpv1Trap.

Se supone que la información de alarma SnmpTrap enviada desde el centro de eventos al gestor de notificaciones incluye Snmpv1Trap y Snmpv2Trap.

Paso 308: el gestor de notificaciones lleva a cabo el filtrado de acuerdo con la lista de suscripciones.

El gestor de notificaciones localiza al cliente que se ha suscrito a las notificaciones del tipo SnmpTrap de acuerdo con el tipo de las notificaciones suscritas almacenadas en la lista de suscripciones, y obtiene una notificación en el

formato Snmpv1Trap por filtrado SUBTREE de acuerdo con el tipo de filtro almacenado y el resultado del filtrado requerido.

El gestor de notificaciones también puede adoptar el modo de filtrado Xpath para obtener una notificación en el formato Snmpv1Trap. Los dos modos de filtrado se basan en la técnica anterior y, por lo tanto, no se describen en detalle en la presente solicitud. Se debe observar que independientemente del modo de filtrado que se adopte se toma una PDU completa de una notificación como la unidad básica para el filtrado; esto es, para asegurar la integridad de la información no se filtra la información interna.

Paso 309: el gestor de notificaciones envía la notificación filtrada al manejador de sesiones.

Mediante una respuesta, el gestor de notificaciones registra la notificación en la cola de datos a ser procesada por el manejador de sesiones.

Paso 310: el manejador de sesiones encapsula la notificación en el formato XML.

En este modo de realización, un método para encapsular la notificación en el formato SnmpTrap dentro del formato XML incluye:

encapsular la información incluida en la notificación SnmpTrap en un contenido de etiqueta de un modelo de notificación basada en NETCONF de la técnica anterior. El contenido encapsulado también necesita ajustarse a la estructura jerárquica del modelo de notificación basada en NETCONF.

La FIG. 4 muestra un modelo de notificación basada en NETCONF utilizado en los modos de realización de la presente invención. Tal como se muestra en la FIG. 4, “notification” (notificación) y “data” (datos) indican, respectivamente, una etiqueta y una subetiqueta del modelo; una caja dibujada con línea discontinua indica el contenido de una etiqueta. “any” (cualquiera) en el contenido de la etiqueta de la subetiqueta “data” indica que en esta parte del contenido se puede añadir cualquier información que se requiera; el símbolo entre “notification” y “data” y el símbolo que aparece después de “data” indican que las dos partes anteriores y posteriores al símbolo se encuentran en una estructura secuencial.

En la técnica anterior, SnmpTrap se clasifica en tres tipos, esto es, Snmpv1Trap, Snmpv2Trap y Snmpv3Trap, y los componentes de Snmpv1Trap, Snmpv2Trap y Snmpv3Trap son conocidos. Por lo tanto, en este modo de realización, SnmpTrap se asocia a una etiqueta (en realidad, es una subetiqueta de “data”); Snmpv1Trap, Snmpv2Trap y Snmpv3Trap se asocian a subetiquetas de SnmpTrap; y la información incluida en Snmpv1Trap, Snmpv2Trap y Snmpv3Trap se asocia, respectivamente, a una subetiqueta de primer nivel, una subetiqueta de segundo nivel y una subetiqueta de tercer nivel, en función de la relación que se muestra en la Tabla 1 o la Tabla 2.

La FIG. 5 muestra un modelo para encapsular SnmpTrap en el formato XML en un modo de realización de la presente invención. Se toma como ejemplo Snmpv1Trap. Tal como se describe en la Tabla 1, “PduData” incluye “timestamp”, “agentaddr”, “eid”, “sid” y “vbs”, y estas partes se encuentran en una relación semejante. Por lo tanto, en el modo de encapsulación de Snmpv1Trap, “timestamp”, “agentaddr”, “eid”, “sid” y “vbs” de la Tabla 1 se asocian, respectivamente, a una subetiqueta del mismo nivel de Snmpv1Trap. En este modo de realización “community” en la Tabla 1 se asocia a un atributo. El modo de encapsulación de Snmpv2Trap es parecido al de Snmpv1Trap. Esto es, los contenidos en “PduData” (como “vb”) descritos en la Tabla 2 se asocian a subetiquetas de Snmpv2Trap, y los contenidos en “vb” como, por ejemplo, “oid” y “value” se asocian a subetiquetas del nivel siguiente a la subetiqueta “vb”. Basándose en el modo de encapsulación de Snmpv2Trap, el modo de encapsulación de Snmpv3Trap incluye, además, “contextname” y “contextid”.

Se debe observar que detrás de la subetiqueta Snmpv2Trap que se muestra en la FIG. 5, también se incluye una subetiqueta Snmpv2TrapGroup (grupo de Snmpv2Trap). La subetiqueta se establece únicamente para facilitar las descripciones del modo de encapsulación de Snmpv3Trap.

Para resumir, de acuerdo con la FIG. 4 y la FIG. 5, el modo de encapsulación de SnmpTrap en algunos modos de realización de la presente invención se utiliza para encapsular todos los contenidos que se muestran en la FIG. 5 en el campo “any” que se muestra en la FIG. 4, y los contenidos encapsulados en “any” también necesitan ajustarse a la estructura jerárquica del modelo que se muestra en la FIG. 4.

Por ejemplo:

Se notifica al cliente que en el sistema se produce un evento de interfaz caída, y la notificación generada es una alarma Snmpv1Trap, donde “community” es “public” (público); “timestamp” es “3:40”; “agentaddr” es “10.111.64.12”; “eid” es “1.3.6.1.4.1.2011”; “gid” es “2”; y “sid” es “0”. La alarma tiene un “vb”; el “oid” del “vb” es “1.3.6.1.4.1.2011.2.1”; y “value” es “2”.

El texto de la alarma Snmpv1Trap encapsulada en un fichero XSD en el formato XML es como sigue:

<?xml version="1.0" encoding="UTF-8"?>

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:hwtrap="http://huawei.com/common/trap"
xmlns:notify="urn:ietfparams:xml:ns:netconfnotification:1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi: schemaLocation="http://huawei. com/common/trap
D:\project\XML\netconf\xsd\snmptrap.xsd">
  <data>
    <snmpTrapPdu xmlns="http://huawei.com/common/trap">
      <snmpv1trap community="public">
        <timestamp>3:40</timestamp>
        <agentaddr>10.111.64.12</agentaddr>
        <eid>1.3.6.1.4.1.2011</eid>
        <gid>2</gid>
        <sid>0</sid>
        <vbs>
          <vb>
            <oid>1.3.6.1.4.1.2011.2.1</oid>
            <value>2(enlace caído)</value>
          </vb>
        </vbs>
      </snmpv1trap>
    </snmpTrapPdu>
  </data>
</notification>
```

Paso 311: el manejador de sesiones envía al cliente la notificación encapsulada en el formato XML.

Se debe observar que en este modo de realización, la petición de suscripción enviada por el cliente puede incluir, además, información de requisitos de formato, la cual describe el formato de notificación requerido por el cliente. Por ejemplo, la notificación generada por el servidor se encuentra en el formato XML-Snmpv1Trap, pero el cliente quiere recibir una notificación en el formato XML-Snmpv2Trap. En este caso, el paso 310 incluye, además: el manejador de sesiones convierte una notificación en el formato XML-Snmpv1Trap al formato XML-Snmpv2Trap, y el modo de conversión adoptado puede ser el modo de Transformación del Lenguaje de Hoja de Estilo Extensible (XSLT).

El XSLT es un lenguaje que describe cómo realizar la conversión entre documentos XML. El XSLT y el estándar XPath están sometidos conjuntamente a desarrollo para la gestión de datos de red. El estándar XPath especifica el contenido a convertir. El XSLT proporciona el lenguaje complementario correspondiente que describe cómo implementar una conversión. El XSLT especifica una serie de reglas para convertir documentos XML fuente en documentos XML destino. En algunos modos de realización de la presente invención, los documentos XML destino pueden ser ficheros XML-Snmpv1Trap, XML-Snmpv2Trap, XML-Snmpv3Trap y XML-syslog. La conversión basada en estas reglas se puede implementar mediante un procesador de XSLT. Además de la conversión desde documentos XML fuente a documentos XML destino, el XSLT está configurado, además, para convertir documentos XML en ficheros en otros formatos como, por ejemplo, ficheros del Lenguaje de Hipertexto mediante Etiquetas (HTML).

La FIG. 6 muestra una regla de conversión entre XML-Snmpv2Trap y XML-Snmpv1Trap utilizando el XSLT en un modo de realización de la presente invención. Tal como se muestra en la FIG. 6, “timestamp”, “community” y “vbs” se convierten mutuamente en los dos formatos. “eid” en XML-Snmpv1Trap es fijo y es un parámetro introducido antes

de ejecutar el programa de conversión. “trapoid” en XML-Snmpv2Trap y “gid” y “sid” en XML Snmpv1Trap se convierten de acuerdo con las reglas descritas en el programa de la FIG. 6. Esto es, cuando “trapoid” es “1.3.6.1.6.3.1.1.5.1”, los “gid” y “sid” convertidos son ambos “0”. Cuando “trapoid” tiene un valor distinto de “1.3.6.1.6.3.1.1.5.1”, “gid” y “sid” se pueden convertir de acuerdo con el siguiente ejemplo.

La FIG. 6 muestra la conversión entre XML-Snmpv2Trap y XML-Snmpv1Trap. Por ejemplo, el script (programa) XSLT para la conversión desde XML-Snmpv2Trap a XML-Snmpv1Trap es como sigue:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:style sheet version-"1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
    <xsl:variable name="eid">1.3.6.1.4.1.2011</xsl:variable>
    <xsl:for-each select="snmpTrapPdu">
      <xsl:element name="snmpTrapPdu">
        <xsl:for-each select="snmpv2trap">
          <xsl:element name="snmpv1trap">
            <xsl:attribute name="community">
              <xsl:value-of select="@community"/>
            </xsl: attribute>
            <timestamp>
              <xsl:value-of select="timestamp"/>
            </timestamp>
            <agentaddr>0.0.0.0</agentaddr>
            <xsl:choose>
              <xsl:when test="trapoid=&quot;1.3.6.1.6.3.1.1.5.1 &quot;">
                <eid>
                  <xsl:value-of select="$eid"/>
                </eid>
                <gid>0</gid>
                <sid>0</sid>
              </xsl:when>//Cuando "trapoid" es "1.3.6.1.6.3.1.1.5.1 ", los "gid" y "sid" convertidos son ambos "0".
              <xsl:when test="trapoid=&quot;1.3.6.1.6.3.1.1.5.2&quot;">
                <eid>
                  <xsl:value-of select="$eid"/>
                </eid>
                <gid>1</gid>
                <sid>0</sid>
              </xsl:when>//Cuando "trapoid" es "1.3.6.1.6.3.1.1.5.2", el "gid" convertido es "1" y el "sid" convertido es "0".
              <xsl:when test="trapoid=&quot;1.3.6.1.6.3.1.1.5.3&quot;">
```

```
<eid>
<xsl:value-of select="$eid"/>
</eid>
<gid>2</gid>
<sid>0</sid>
</xsl:when>//Cuando "trapoid" es "1.3.6.1.6.3.1.1.5.3", el "gid" convertido es "2" y el "sid" convertido es "0".
<xsl:when test="trapoid=&quot;1.3.6.1.6.3.1.1.5.4&quot;">
<eid>
<xsl:value-of select="$eid"/>
</eid>
<gid>3</gid>
<sid>0</sid>
</xsl:when>//Cuando "trapoid" es "1.3.6.1.6.3.1.1.5.4", el "gid" convertido es "3" y el "sid" convertido es "0".
<xsl:when test="trapoid=&quot;1.3.6.1.6.3.1.1.5.5&quot;">
<eid>
<xsl:value-of select="$eid"/>
</eid>
<gid>4</gid>
<sid>0</sid>
</xsl:when>//Cuando "trapoid" es "1.3.6.1.6.3.1.1.5.5", el "gid" convertido es "4" y el "sid" convertido es "0".
<xsl:when test="trapoid=&quot;1.3.6.1.6.3.1.1.5.6&quot;">
<eid>
<xsl:value-of select="$eid"/>
</eid>
<gid>5</gid>
<sid>0</sid>
</xsl:when>//Cuando "trapoid" es "1.3.6.1.6.3.1.1.5.6", el "gid" convertido es "5" y el "sid" convertido es "0".
<xsl:otherwise>
<eid>
<xsl:value-of select="$eid"/>
</eid>
<gid>6</gid>
<sid>
<xsl:value-of select="substring(trapoid,string-length(trapoid)-2,string-length(trapoid)"/>
```

</sid>

</xsl:otherwise>//Cuando "trapoid" es un valor distinto de los valores precedentes, el "gid" convertido es "6" y el "sid" convertido es especificado de acuerdo con el modo de "select=.....".

</xsl:choose>

<xsl:copy-of select="vbs"/>

</xsl:element>

</xsl:for-each>

</xsl:element>

</xsl:for-each>

</xsl:template>

</xsl:stylesheet>

La FIG. 7 es un diagrama de flujo de un método proporcionado en el segundo ejemplo de modo de realización de la presente invención. Comparado con el modo de realización que se muestra en la FIG. 3, la notificación generada por el servidor en este modo de realización es en el formato syslog.

Tal como se muestra en la FIG. 7, este modo de realización incluye los siguientes pasos:

Un cliente establece una sesión SSH con un manejador de sesiones. El proceso de establecimiento de una sesión SSH está basado en la técnica anterior.

Paso 701: el cliente envía una petición de suscripción al manejador de sesiones.

La petición de suscripción incluye el tipo de notificaciones suscritas, un tipo de filtro y un resultado del filtrado requerido. En este modo de realización, si el tipo de notificaciones a las que se suscribe el cliente es todos los eventos con la cadena “syslog over netconf”, y el tipo de filtro es filtro Xpath, el resultado del filtrado requerido es un conjunto de eventos con prioridades 8-15. Además, la petición de suscripción puede especificar que una notificación se ha almacenada en el archivo “syslog-subscription.profile”, y que el tiempo de inicio de la suscripción es 2001-12-24T12:23:45.

En la técnica anterior está claramente definido el formato de una petición de suscripción para una notificación basada en NETCONF. La petición de suscripción enviada en este paso aplica únicamente a la técnica anterior. Por ejemplo:

<?xml version="1.0" encoding="UTF-8"?>

<rpc xmlns="urn:ietf:params:xml:ns:netconfbase:1.0"

xmlns:notify="urn:ietfparams:xml:ns:netconf:notification:1.0"

xmlns:netconf"urn:ietf:params:xml:ns:netconfbase:1.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="urn:ietf:params:xml:ns:netconf:notification:1.0

D:\project\xML\netconf\sd\NC-Notification.xsd" message-id=" 101 ">

<create-subscription xmlns="um:ietf:params:xml:ns:netconf:notification: 1.0">

<streams>

<stream>syslog over netconf</stream>

</streams>//Suscribir a todos los eventos con al cadena "syslog over netconf’.

<filter type="xpath" select="/syslogs/syslog/PRI[MsgSource=1]"/>

//Utilizar el modo de filtro Xpath para obtener los eventos con prioridades 8-15.

<named-profile>syslog-subscription.profile</named-profile>//Almacenar una notificación en un fichero especificado.

<startTime>2001-12-24T12:23:45</startTime>//Tiempo de inicio de la suscripción.

</create-subscription>

</rpc>

Los pasos 702-706 son parecidos a los pasos 302-306 y no se describen de nuevo.

Paso 707: el centro de eventos envía información de syslog al gestor de notificaciones.

Cuando se produce un evento en el sistema, el centro de eventos genera una notificación del tipo correspondiente de acuerdo con el tipo del evento. Se supone que la información de syslog enviada desde el centro de eventos al gestor de notificaciones incluye eventos de cada prioridad.

Paso 708: el gestor de notificaciones lleva a cabo el filtrado en función de la lista de suscripciones.

En este paso, el gestor de notificaciones localiza al cliente que se ha suscrito a notificaciones de tipo syslog de acuerdo con el tipo de notificaciones suscritas almacenadas en la lista de suscripciones, y obtiene las notificaciones de los eventos con prioridades 8-15 mediante el filtrado Xpath de acuerdo con el tipo de filtrado almacenado y el resultado del filtrado requerido.

Para obtener notificaciones de eventos con prioridades 8-15 también se puede utilizar el modo de filtrado SUBTREE. Los dos modos de filtrado están basados en la técnica anterior y por lo tanto no se describen detalladamente en la presente solicitud. Se debe observar que se toma una notificación de syslog completa como la unidad básica para el filtrado independientemente del modo de filtrado que se adopte; esto es, para asegurar la integridad de la información no se filtra la información interna.

Paso 709: el gestor de notificaciones envía una notificación filtrada al manejador de sesiones.

Paso 710: el manejador de sesiones encapsula la notificación en el formato XML.

En este modo de realización, un método para encapsular la notificación del formato syslog en el formato XML es del siguiente modo:

encapsular la información incluida en la notificación syslog dentro de un contenido de etiqueta de un modelo de notificación basada en NETCONF de la técnica anterior. El contenido encapsulado también necesita ajustarse a la estructura jerárquica del modelo de notificación basada en NETCONF.

Tal como se describe en la Tabla 3, una notificación syslog consta de tres partes: “PRI”, “HEADER” y “MSG”. El “PRI” incluye la información “MsgSource” y “Severity”. El “HEADER” incluye información “timestamp” y “agentaddr”. La información en el “MSG” se puede configurar en función de los requisitos reales. De este modo, en este modo de realización, para encapsular una notificación del formato syslog en el formato XML, syslog debe asociarse a una etiqueta, y la información incluida en syslog se asocia a la subetiqueta de primer nivel o a la subetiqueta de segundo nivel de syslog de acuerdo con la relación jerárquica anterior.

La FIG. 8 muestra un modelo para encapsular syslog en el formato XML en un modo de realización de la presente invención. Tal como se muestra en la FIG. 8, “PRI”, “HEADER” y “MSG” se asocian a tres subetiquetas de la etiqueta syslog, y la información incluida en “PRI”, “HEADER” y “MSG” se asocia respectivamente, además, a una subetiqueta de “PRI”, “HEADER” y “MSG”. Delante de la etiqueta syslog que se muestra en la FIG. 8, también se incluye una etiqueta de syslog. La etiqueta de syslog se utiliza para encapsular el envío múltiples notificaciones syslog en la misma notificación syslogs para su envío.

Para resumir, de acuerdo con la FIG. 4 y la FIG. 8, el modo de encapsulación syslog en algunos modos de realización de la presente invención se utiliza para encapsular todos los contenidos que se muestran en la FIG. 8 en el campo “any” que se muestra en la FIG. 4, y los contenidos encapsulados en el campo “any” también se debe ajustar a la estructura jerárquica del modelo que se muestra en la FIG. 4.

Por ejemplo:

Se notifica al cliente que en el sistema se produce un evento de interfaz caída, y la notificación generada es una notificación syslog, donde “PRI” es “8”; “timestamp” es “2001-12-24T12:23:45”; “agentaddr” es “10.111.64.12”; el índice de la interfaz de alarma es “1”; y la información de la alarma es “linkDown”. El texto de la notificación syslog encapsulada en un fichero XSD en el formato XML es como sigue:

<?xml version="1.0" encoding="UTF-8"?>

<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"

xmlns:hwsyslog="http://huawei.com/common/syslog"

```
xmlns:notify="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:chemaLocation="http://huawei.com/common/syslog syslog.xsd">
  <data>
    <syslogs xmlns="http://huawei.com/common/syslog">
      <syslog>
        <PRI>
          <MsgSource>1</MsgSource>
          <Severity>0</Severity>
        </PRI>
        <HEADER>
          <timestamp>2001-12-24T12:23:45</timestamp>
          <agentaddress>10.111.64.12</agentaddress>
        </HEADER>
        <MSG>
          <top xmlns="http://huawei.com/syslogmsg">
            <code>2011</code>
            <ifindex> 1 </ifindex>
            <alarminfo>linkdown</alarminfo>
          </top>
        </MSG>
      </syslog>
    </syslogs>
  </data>
</notification>
```

Paso 711: el manejador de sesiones envía al cliente la notificación encapsulada en el formato XML.

El procedimiento posterior es parecido al modo de realización que se muestra en la FIG. 3 y no se vuelve a describir aquí.

De acuerdo con el método anterior, la FIG. 9 muestra una estructura de un sistema proporcionado en un modo de realización de la presente invención. Tal como se muestra en la FIG. 9, el sistema incluye un cliente 901 y un servidor 902.

El cliente 901 está configurado para suscribirse a notificaciones del servidor 902 y para recibir notificaciones enviadas por el servidor 902.

El servidor 902 está configurado para generar una notificación cuando se produce un evento en el sistema, convertir la notificación a un formato que pueda ser identificado por el cliente 901, y enviar al cliente 901 la notificación convertida.

El servidor 902 incluye un manejador 9021 de sesiones, un gestor 9022 de notificaciones, y un centro 9023 de eventos.

El manejador 9021 de sesiones está configurado para: enviar al gestor 9022 de notificaciones una petición de suscripción recibida desde el cliente 901, incluyendo la petición de suscripción el tipo de notificaciones a las que se quiere suscribir el cliente, convertir al formato XML una respuesta recibida desde el gestor 9022 de notificaciones,

enviar al cliente 901 la respuesta convertida, recibir una notificación enviada desde el gestor 9022 de notificaciones, convertir la notificación al formato XML, y enviar al cliente 901 la notificación convertida.

El gestor 9022 de notificaciones está configurado para: añadir a una lista de suscripciones configurada previamente la petición de suscripción recibida desde el manejador 9021 de sesiones, devolver al manejador 9021 de sesiones una respuesta que indica si se ha realizado con éxito la suscripción, recibir una notificación enviada desde el centro 9023 de eventos, obtener el cliente que se ha suscrito a las notificaciones de ese tipo filtrando de acuerdo con la lista de suscripciones, y enviar la notificación al manejador 9021 de sesiones.

El centro 9023 de eventos está configurado para generar una notificación de un tipo correspondiente a un evento que se produce en el sistema y enviar la notificación al gestor 9022 de notificaciones.

La petición de suscripción anterior puede incluir, además, un tipo de filtro y un resultado del filtrado requerido. En consecuencia, el gestor 9022 de notificaciones está configurado, además, para obtener una notificación a la que se ha suscrito realmente el cliente 901 en función del tipo de filtro y el resultado del filtrado requerido.

Si es necesario, la petición de suscripción anterior puede incluir, además, información de requisitos de formato. En consecuencia, el manejador 9021 de sesiones está configurado, además, para convertir una notificación en formato XML a un formato solicitado en la información de requisitos de formato del cliente 901.

De este modo, con la solución técnica de la presente invención, varios modos de notificación resultan compatibles de forma efectiva. El cliente y el servidor adoptan el protocolo SSH como portador de la conexión, y el SSH utiliza una conexión TCP para cifrar el contenido, asegurando de este modo la llegada fiable de una notificación y la seguridad del contenido de la información. Las notificaciones se envían mediante flujos de texto XML; por lo tanto, las notificaciones se pueden convertir fácilmente utilizando el XSLT en cualquier capa, facilitando de este modo el procesamiento posterior del cliente. En algunos modos de realización de la presente invención, en el mismo fichero XML se colocan muchas notificaciones para su transmisión, reduciendo de este modo las interacciones y el tráfico de datos. Además, en algunos modos de realización de la presente invención, es muy conveniente implementar la suscripción mediante filtros. Los modos de filtrado no son solo flexibles, sino que también implementan un filtro personalizado para cada objetivo.

El cliente incluye un módulo de suscripción y un módulo de recepción.

El módulo de suscripción está configurado para suscribirse a notificaciones procedentes de un servidor NETCONF. El módulo de recepción está configurado para recibir una notificación enviada por un servidor NETCONF. La notificación se genera cuando se produce un evento en el sistema y el servidor NETCONF la convierte a un formato que pueda ser identificado por el cliente NETCONF de acuerdo con la suscripción a notificaciones del módulo de suscripción.

Aquellos experimentados en la técnica pueden entender que todos o parte de los pasos de los modos de realización anteriores se pueden implementar mediante hardware controlado por un programa. El programa puede encontrarse almacenado en un medio de almacenamiento legible por un ordenador. Cuando es ejecutado, el programa lleva a cabo los pasos de los modos de realización de los métodos anteriores. El medio de almacenamiento puede ser cualquier medio capaz de almacenar códigos de programa, por ejemplo, una Memoria de Solo Lectura (ROM), una Memoria de Acceso Aleatorio (RAM), un disco magnético y un disco compacto.

Aunque la solución técnica de la presente invención se ha descrito a través de varios ejemplos de modos de realización, la invención no se encuentra limitada a dichos modos de realización.

REIVINDICACIONES

1. Un método para enviar una notificación basada en el Protocolo de Configuración de Red, NETCONF, que comprende:

establecer una sesión basada en el Protocolo de Control de Internet, TCP, entre un cliente NETCONF y un servidor NETCONF;

aceptar, por parte del servidor NETCONF, una suscripción de notificación del cliente NETCONF en donde la aceptación de la suscripción de notificación del cliente NETCONF comprende: recibir, por parte del servidor NETCONF, una petición de suscripción enviada por el cliente NETCONF, en donde la petición de suscripción incluye información de requisitos de formato y un tipo de notificación suscrita a la que el cliente NETCONF quiere suscribirse; y añadir, por parte del servidor NETCONF, la petición de suscripción a una lista de suscripciones configurada previamente después de que la suscripción se haya realizado con éxito;

generar, por parte del servidor NETCONF, una notificación cuando se produzca un evento en un sistema, convertir la notificación a un formato que pueda ser identificado por el cliente NETCONF de acuerdo con la suscripción de notificación del cliente NETCONF; y

enviar, por parte del servidor NETCONF, la notificación convertida al cliente NETCONF;

en donde la conversión de la notificación al formato que puede ser identificado por el cliente NETCONF comprende:

encapsular, por parte del servidor NETCONF, la notificación en un formato del Lenguaje Extensible mediante Etiquetas, XML, si la petición de suscripción no incluye información de requisitos de formato; o

encapsular, por parte del servidor NETCONF, la notificación en el formato XML y convertir la notificación encapsulada a un formato solicitado en la información de requisitos de formato del cliente NETCONF si la petición de suscripción incluye, además, información de requisitos de formato;

en donde el tipo de notificaciones suscritas es Captura del Protocolo Simple de Gestión de Redes, SnmpTrap, o syslog;

en donde si el tipo de la notificación es SnmpTrap, la encapsulación de la notificación en el formato XML comprende:

encapsular, por parte del servidor NETCONF, la información incluida en la notificación SnmpTrap dentro de un contenido de etiqueta de acuerdo con un modelo de notificación basada en NETCONF, en donde el contenido encapsulado responde a una estructura jerárquica del modelo de notificación basada en NETCONF; el modelo de notificación tiene una estructura jerárquica que comprende una etiqueta, una subetiqueta y un contenido de etiqueta; y la subetiqueta es una etiqueta de datos que indica que, si es necesario, se puede añadir cualquier información a esta parte del contenido,

si el tipo de la notificación es syslog, la encapsulación de la notificación en el formato XML comprende:

encapsular, por parte del servidor NETCONF, la información incluida en el syslog dentro de un contenido de etiqueta de acuerdo con un modelo de notificación basada en NETCONF, en donde: el contenido encapsulado responde a una estructura jerárquica del modelo de notificación basada en NETCONF; el modelo de notificación tiene una estructura jerárquica que comprende una etiqueta, una subetiqueta y un contenido de etiqueta; y la subetiqueta es una etiqueta de datos;

en donde el syslog comprende un “PRI”, un “HEADER” y un “MSG”, en donde: el “PRI” comprende “MsgSource”, fuente del registro, e información de “Severity”; el “HEADER” comprende información de “timestamp” y “agentaddr”; y la información en el “MSG” está configurada en función de los requisitos de los usuarios; y

el método para que el servidor NETCONF encapsule la información incluida en el syslog dentro del contenido de la etiqueta comprende:

asociar, por parte del servidor NETCONF, el syslog a una etiqueta y asociar la información incluida en el syslog a diferentes niveles de subetiquetas de la etiqueta syslog de acuerdo con la relación jerárquica; y

asociar, por parte del servidor NETCONF, una etiqueta como syslogs y establecer una o más etiquetas syslog como subetiquetas de syslogs.

2. El método de la reivindicación 1, en donde la generación de la notificación comprende:

generar, por parte del servidor NETCONF, la notificación de un tipo correspondiente al tipo del evento que se ha producido.

3. El método de la reivindicación 2, en donde antes de que el servidor NETCONF convierta la notificación encapsulada al formato solicitado que pueda ser identificado por el cliente NETCONF, el método comprende, además:

localizar, por parte del servidor NETCONF, el cliente NETCONF que se ha suscrito a las notificaciones del tipo de acuerdo con el tipo de las notificaciones suscritas almacenadas en la lista de suscripciones.

4. El método de una cualquiera de las reivindicaciones 1-3, en donde la petición de suscripción incluye, además, un tipo de filtro y un resultado del filtrado requerido; y el tipo de filtro es un filtro SUBTREE o de Lenguaje de Ruta XML, Xpath.

5. El método de la reivindicación 4, en donde antes de que el servidor NETCONF convierta la notificación al formato solicitado que pueda ser identificado por el cliente NETCONF, el método comprende, además:

obtener, por parte del servidor NETCONF, una notificación a la que el cliente NETCONF se ha suscrito realmente, filtrando en función del tipo de filtro almacenado en la lista de suscripciones y el resultado del filtrado requerido.

6. El método de la reivindicación 1, en donde el método para que el servidor NETCONF convierta la notificación que se encuentra encapsulada en el formato XML a un formato solicitado en la información de requisitos de formato del cliente NETCONF comprende:

convertir, por parte del servidor NETCONF, la notificación en el formato XML al formato solicitado por el cliente NETCONF mediante Transformación del Lenguaje de Hoja de Estilo Extensible, XSLT.

7. Un dispositivo para enviar una notificación basada en el Protocolo de Configuración de Red, NETCONF, en donde el dispositivo es un servidor NETCONF, comprendiendo el servidor NETCONF un manejador de sesiones (9021), un gestor de notificaciones (9022) y un centro de eventos (9023), en donde:

el manejador de sesiones (9021) está configurado para establecer una sesión basada en el Protocolo de Control de Transporte, TCP, entre un cliente NETCONF y el servidor NETCONF, y aceptar una suscripción de notificación del cliente NETCONF;

el centro de eventos (9023) está configurado para generar una notificación cuando se produce un evento en un sistema y enviar la notificación al gestor de notificaciones;

el gestor de notificaciones (9022) está configurado para almacenar en una lista de suscripciones configurada previamente una suscripción de notificación recibida desde el manejador de sesiones, y reenviar la notificación enviada por el centro de eventos al manejador de sesiones en función de la lista de suscripciones; y

el manejador de sesiones (9021) está configurado, además, para convertir la notificación enviada por el gestor de notificaciones a un formato que pueda ser identificado por el cliente NETCONF, y enviar la notificación convertida al cliente NETCONF;

en donde la conversión, por parte del manejador de sesiones, de la notificación al formato que puede ser identificado por el cliente NETCONF comprende:

encapsular la notificación en un formato del Lenguaje Extensible mediante Etiquetas, XML, si la petición de suscripción no incluye información de requisitos de formato; o

encapsular la notificación en el formato XML y convertir la notificación encapsulada a un formato solicitado en la información de requisitos de formato del cliente NETCONF si la petición de suscripción incluye, además, información de requisitos de formato;

en donde el tipo de notificaciones suscritas es Captura del Protocolo Simple de Gestión de Redes, SnmpTrap, o syslog;

en donde si el tipo de la notificación es SnmpTrap, la encapsulación de la notificación en el formato XML comprende:

encapsular, por parte del servidor NETCONF, la información incluida en la notificación SnmpTrap dentro de un contenido de etiqueta de acuerdo con un modelo de notificación basada en NETCONF, en donde el contenido encapsulado responde a una estructura jerárquica del modelo de notificación basada en NETCONF; el modelo de notificación tiene una estructura jerárquica que comprende una etiqueta, una subetiqueta y un contenido de etiqueta; y la subetiqueta es una etiqueta de datos;

si el tipo de la notificación es syslog, la encapsulación de la notificación en el formato XML comprende:

encapsular, por parte del servidor NETCONF, la información incluida en el syslog dentro de un contenido de

etiqueta de acuerdo con un modelo de notificación basada en NETCONF, en donde: el contenido encapsulado responde a una estructura jerárquica del modelo de notificación basada en NETCONF; el modelo de notificación tiene una estructura jerárquica que comprende una etiqueta, una subetiqueta y un contenido de etiqueta; y la subetiqueta es una etiqueta de datos que indica que si es necesario se puede añadir cualquier información a esta parte del contenido;

en donde el syslog comprende un “PRI”, un “HEADER” y un “MSG”, en donde: el “PRI” comprende información de “MsgSource”, fuente del registro y “Severity”; el “HEADER” comprende información de “timestamp” y “agentaddr”; y la información en el “MSG” está configurada en función de los requisitos de los usuarios; y

el método para que el servidor NETCONF encapsule la información incluida en el syslog dentro del contenido de la etiqueta comprende:

asociar, por parte del servidor NETCONF, el syslog a una etiqueta y asociar la información incluida en el syslog a diferentes niveles de subetiquetas de la etiqueta syslog de acuerdo con la relación jerárquica; y

asociar, por parte del servidor NETCONF, una etiqueta como syslogs y establecer una o más etiquetas syslog como subetiquetas de los syslogs.

8. El dispositivo de la reivindicación 7, en donde:

la petición de suscripción incluye, además, un tipo de filtro y un resultado del filtrado requerido; y

el gestor de notificaciones está configurado, además, para obtener una notificación a la que se ha suscrito realmente el cliente NETCONF mediante filtrado de acuerdo con el tipo de filtro y el resultado del filtrado requerido.

9. Un sistema para enviar notificaciones basadas en el Protocolo de Configuración de Red, NETCONF, utilizando un dispositivo (902) como el reivindicado en la reivindicación 7 y un cliente NETCONF (901).

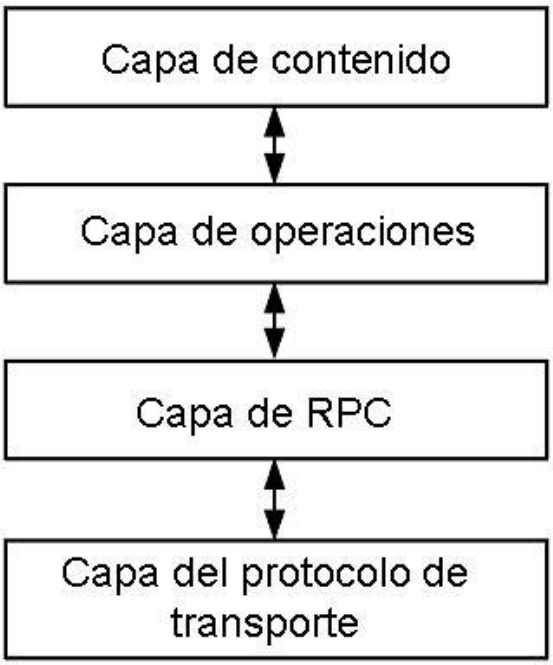
Capa de contenido
Capa de operaciones
Capa de RPC
Capa del protocolo de transporte

FIG. 1

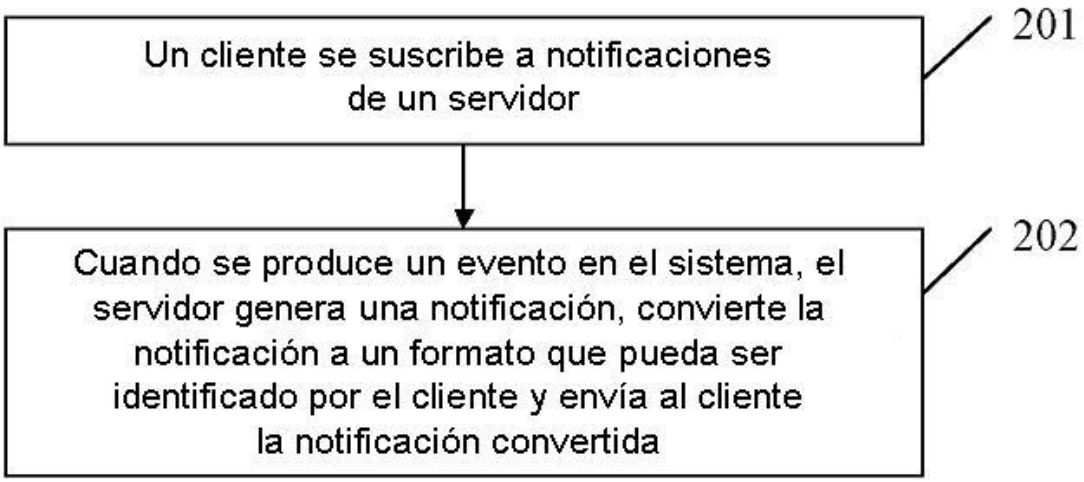
Un cliente se suscribe a notificaciones de un servidor
201
Cuando se produce un evento en el sistema, el servidor genera una notificación, convierte la notificación a un formato que pueda ser identificado por el cliente y envía al cliente la notificación convertida
202

FIG. 2

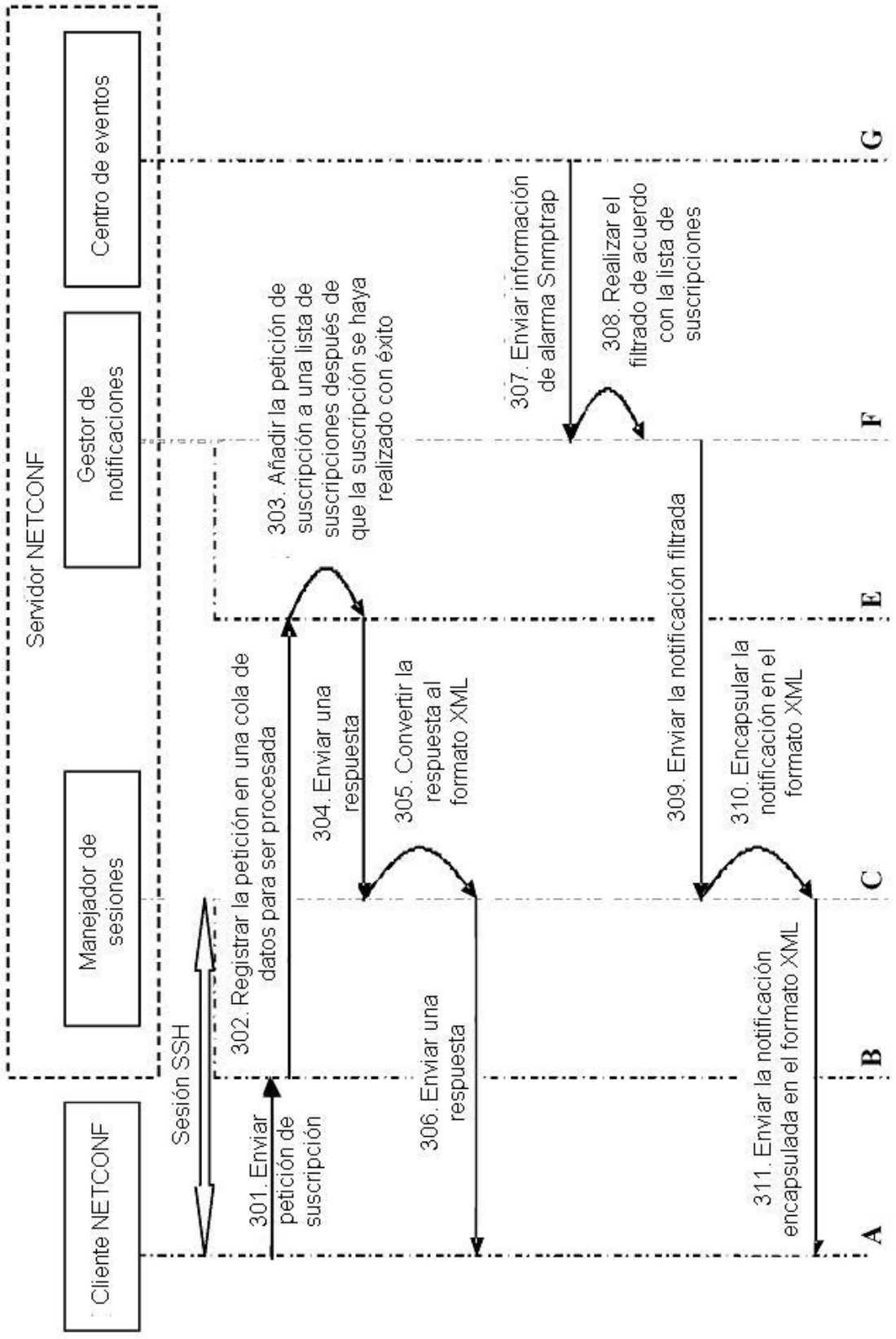
Cliente NETCONF
Servidor NETCONF
Manejador de sesiones
Gestor de notificaciones
Centro de eventos
Sesión SSH
301. Enviar petición de suscripción
302. Registrar la petición en una cola de datos para ser procesada
303. Añadir la petición de suscripción a una lista de suscripciones después de que la suscripción se haya realizado con éxito
304. Enviar una respuesta
305. Convertir la respuesta al formato XML
306. Enviar una respuesta
307. Enviar información de alarma Snmptrap
308. Realizar el filtrado de acuerdo con la lista de suscripciones
309. Enviar la notificación filtrada
310. Encapsular la notificación en el formato XML
311. Enviar la notificación encapsulada en el formato XML
A
B
C
E
F
G

FIG. 3

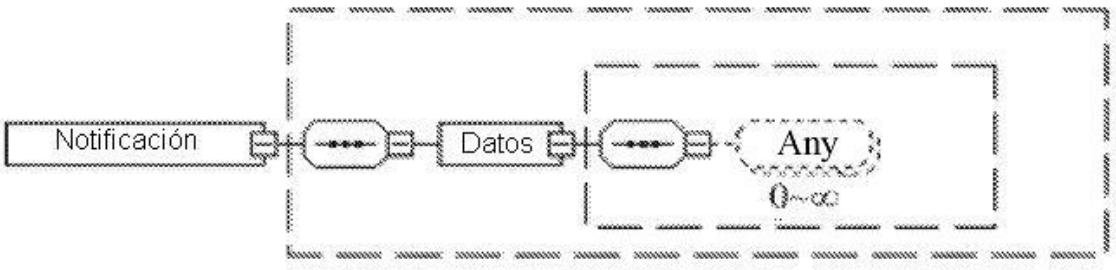

FIG. 4

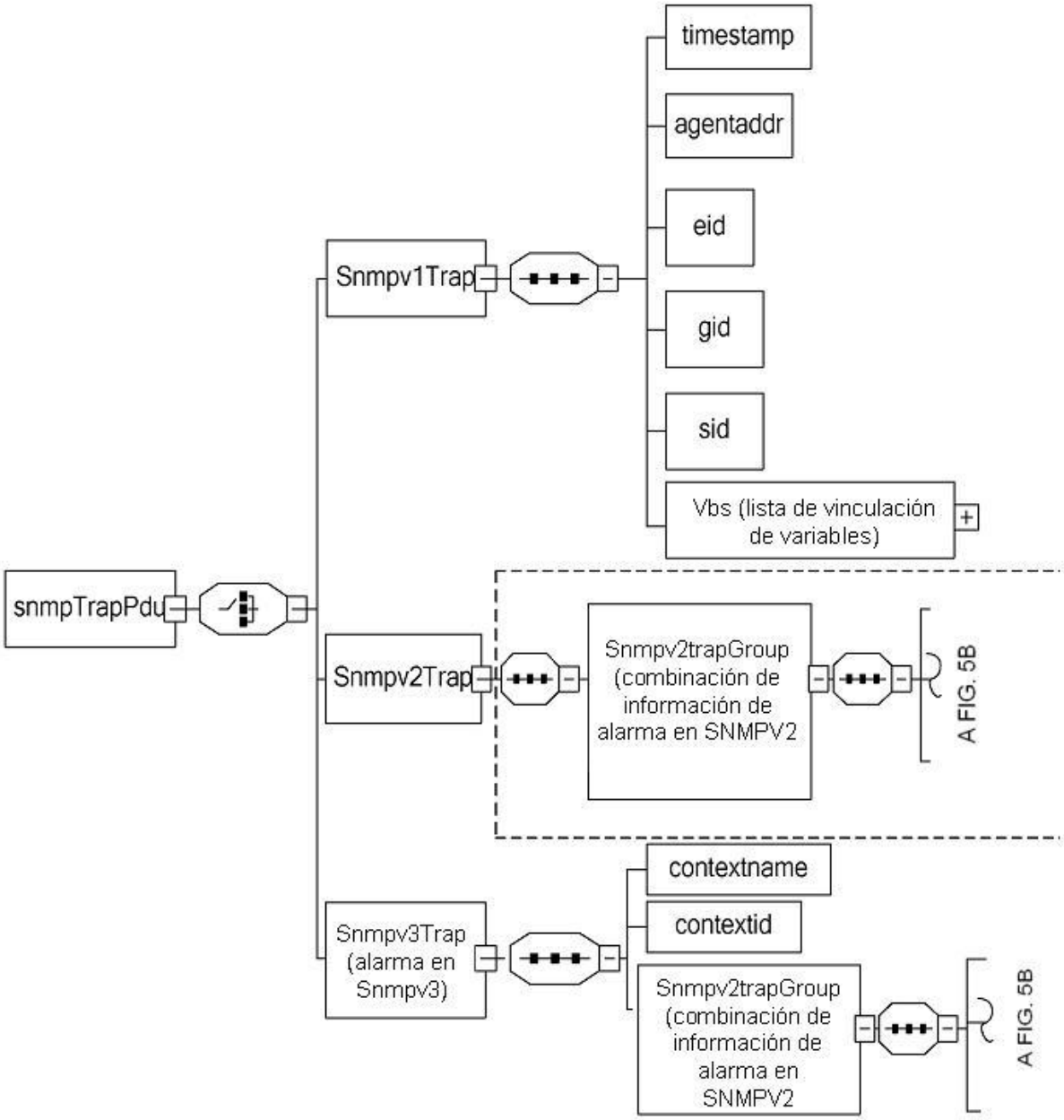
timestamp
agentaddr
eid
Snmpv1Trap
gid
sid
Vbs (lista de vinculación de variables)
snmpTrapPdu
Snmpv2Trap
Snmpv2trapGroup (combinación de información de alarma en SNMPV2
A FIG. 5B
contextname
contextid
Snmpv3Trap (alarma en Snmpv3)
Snmpv2trapGroup (combinación de información de alarma en SNMPV2
A FIG. 5B

FIG. 5A

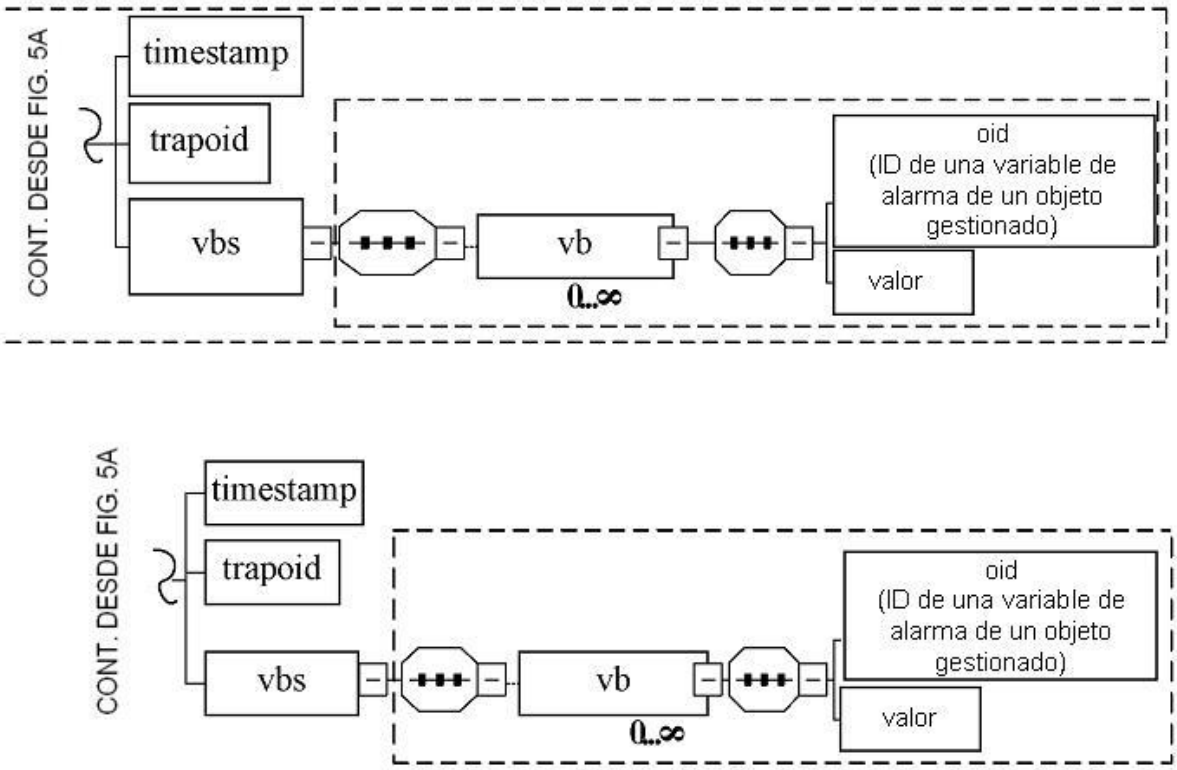
CONT. DESDE FIG. 5A
timestamp
trapoid
vbs
vb
0..∞
oid
(ID de una variable de alarma de un objeto gestionado)
valor
CONT. DESDE FIG. 5A
timestamp
trapoid
vbs
vb
0..∞
oid
(ID de una variable de alarma de un objeto gestionado)
valor

FIG. 5B

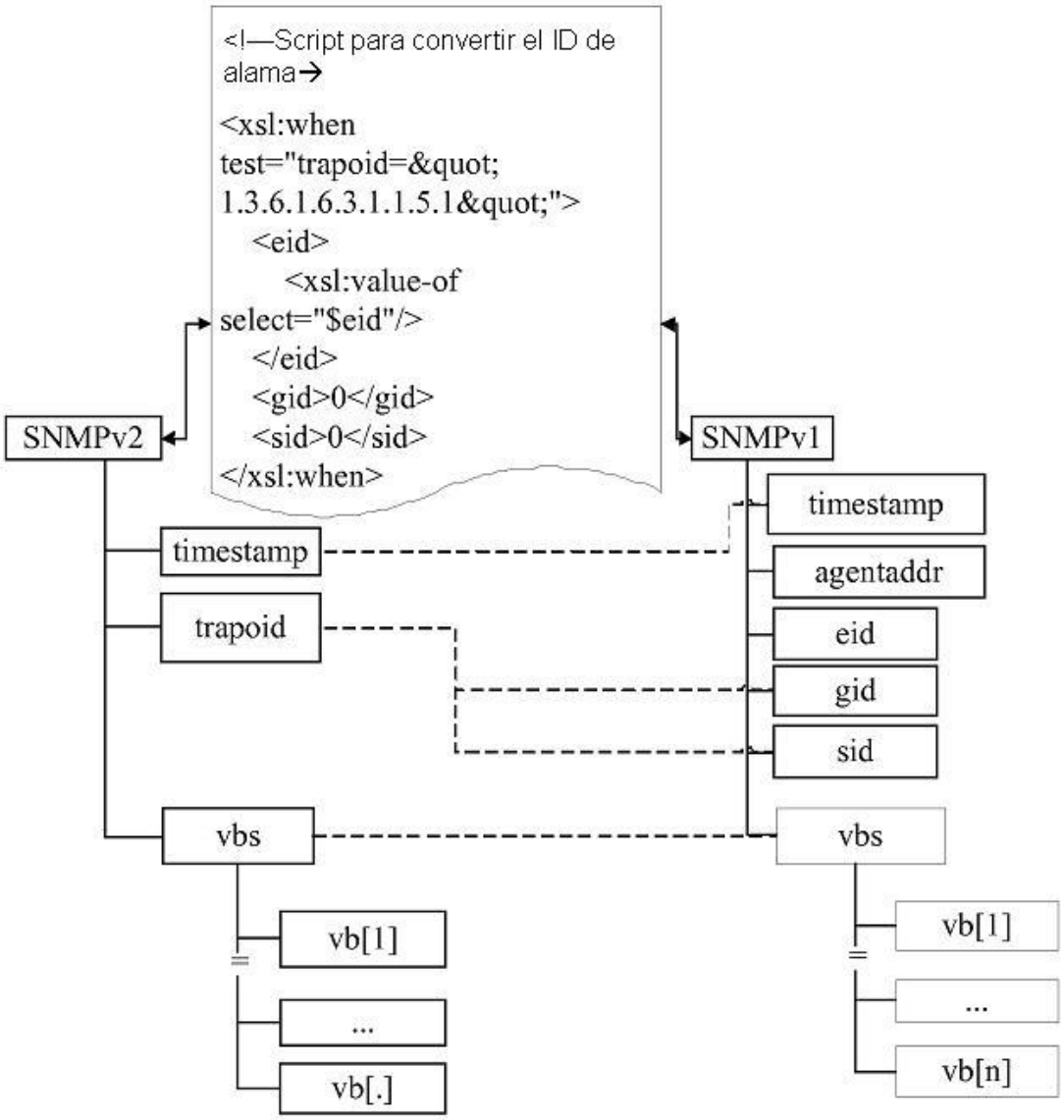
<!—Script para convertir el ID de alama→
<xsl:when test="trapoid=" 1.3.6.1.6.3.1.1.5.1"">
<eid>
<xsl:value-of select="$eid"/>
</eid>
<gid>0</gid>
<sid>0</sid>
</xsl:when>
SNMPv2
SNMPv1
timestamp
trapoid
vbs
vb[1]
...
vb[.]
timestamp
agentaddr
eid
gid
sid
vbs
vb[1]
...
vb[n]

FIG. 6

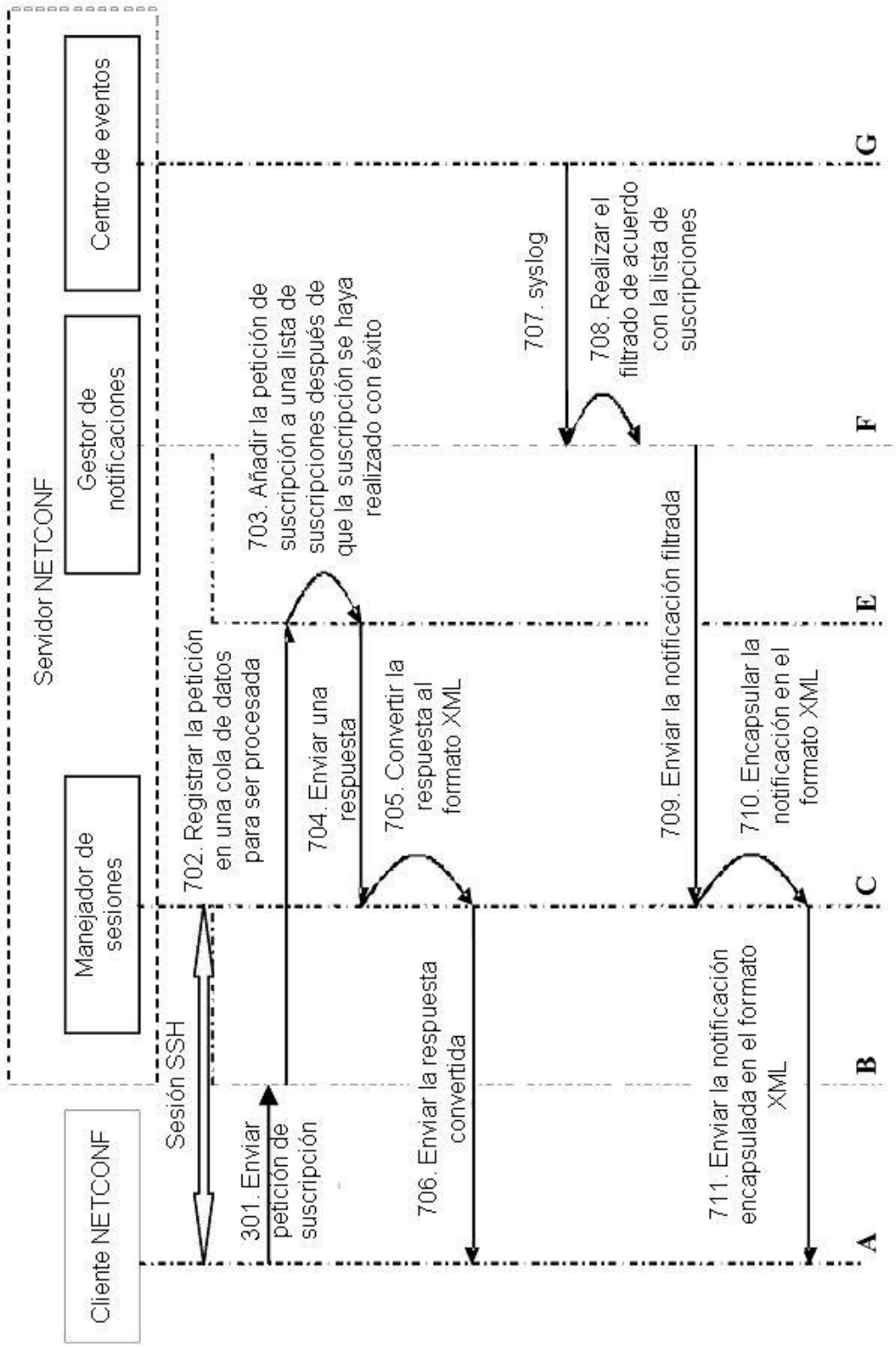
Cliente NETCONF
Servidor NETCONF
Manejador de sesiones
Gestor de notificaciones
Centro de eventos
A
B
C
E
F
G
Sesión SSH
301. Enviar petición de suscripción
702. Registrar la petición en una cola de datos para ser procesada
703. Añadir la petición de suscripción a una lista de suscripciones después de que la suscripción se haya realizado con éxito
704. Enviar una respuesta
705. Convertir la respuesta al formato XML
706. Enviar la respuesta convertida
707. syslog
708. Realizar el filtrado de acuerdo con la lista de suscripciones
709. Enviar la notificación filtrada
710. Encapsular la notificación en el formato XML
711. Enviar la notificación encapsulada en el formato XML

FIG. 7

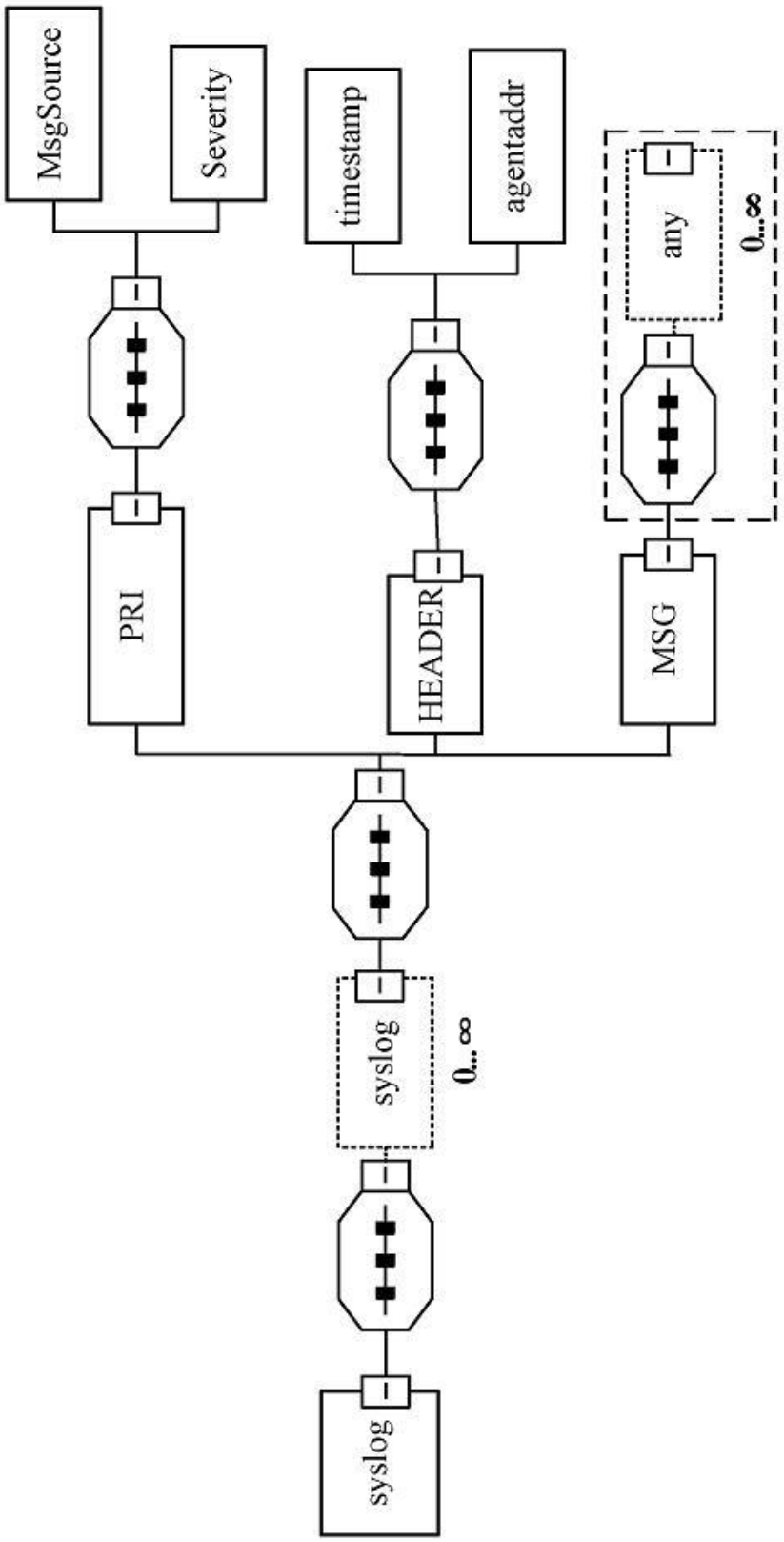
MsgSource
Severity
timestamp
agentaddr
any
0..∞
PRI
HEADER
MSG
syslog
0..∞
syslog

FIG. 8

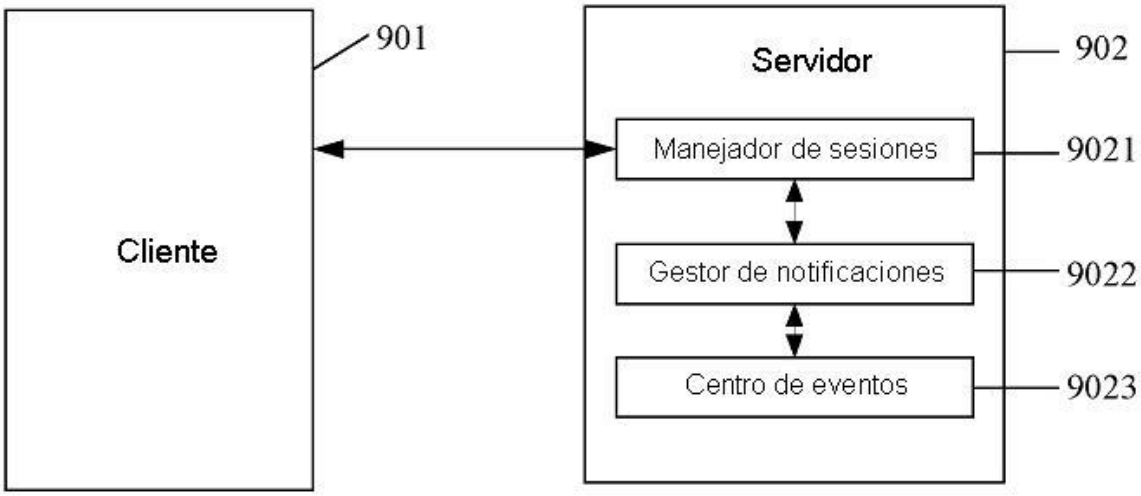
901
Cliente
Servidor
902
Manejador de sesiones
9021
Gestor de notificaciones
9022
Centro de eventos
9023

FIG. 9