(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 429 396**

(21) Número de solicitud: 201230419

(51) Int. CI.:

H04L 12/26 (2006.01)

(12) PATENTE DE INVENCIÓN B1

(22) Fecha de presentación:

20.03.2012

(43) Fecha de publicación de la solicitud:

14.11.2013

(88) Fecha de publicación diferida del informe sobre el estado de la técnica:
12.02.2014

Fecha de la concesión:

11.11.2014

(45) Fecha de publicación de la concesión:

18.11.2014

(73) Titular/es:

TELEFÓNICA, S.A. (100.0%)
Gran Vía, 28
28013 Madrid (Madrid) ES

(72) Inventor/es:

GARCÍA DE BLAS, Gerardo;
MONTES MORENO, Pablo;
RAMÓN SALGUERO, Francisco Javier y
TIERNO SEPÚLVEDA, Alfonso

(74) Agente/Representante:

ARIZTI ACHA, Monica

(54) Título: **MÉTODO Y SISTEMA PARA MONITORIZACIÓN DE TRÁFICO DE RED**

(57) Resumen:
Método y sistema para monitorización de tráfico de red.
El método comprende las etapas de:
a) adquirir con un módulo de captura, datos de tráfico de una línea de entrada y reenviar dichos datos de tráfico a un módulo de detección; y
b) recibir, dicho módulo de detección, dichos datos de tráfico para realizar un análisis de inspección profunda de datos de dichos datos de tráfico recibidos para realizar una detección de los mismos,
en el que dicha etapa a) comprende, adquirir dichos datos de tráfico de al menos dos de dichas líneas de entrada y clasificarlos de modo que dicho reenvío a dicho módulo de detección se realice en un orden cronológico para aquellos paquetes pertenecientes a un flujo específico de dichos datos de tráfico procedentes de dichas al menos dos líneas de entrada.
El sistema de la invención está previsto para implementar el método de la invención.

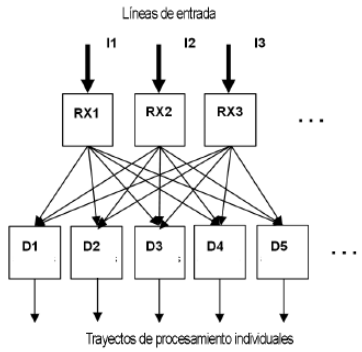

Figura 7

ES 2 429 396 B1

MÉTODO Y SISTEMA PARA MONITORIZACIÓN DE TRÁFICO DE RED

DESCRIPCIÓN

Campo de la técnica

La presente invención se refiere a la monitorización de redes y en general se refiere a un método y un sistema para monitorización de tráfico de red.

Estado de la técnica anterior

La monitorización de redes se ha convertido en una tarea importante en las redes IP modernas. Permite mantener la estabilidad, disponibilidad y seguridad del sistema de red y permite tomar decisiones acertadas con respecto a planificación de red y capacidad.

En la monitorización de redes, además de conocer la cantidad tráfico, se requiere clasificarlo por tipo, es decir determinar el protocolo/servicio que lo provoca. Esta clasificación de tráfico se deduce usando patrones de tráfico, siendo estos patrones diferentes para vídeo, voz sobre IP, navegación web, reproducción en tiempo real, etc.

Existen muchos productos comerciales tales como Sandvine [1], iPoque [2] o Cisco SCE [3] que proporcionan una solución basándose en análisis de inspección profunda de paquetes (DPI) y la detección de patrones de tráfico. Estos sistemas inspeccionan el tráfico que atraviesa un enlace y lo clasifican como perteneciente a una clase específica de aplicación o lo clasifican como desconocido. Esta información se usa para proporcionar informes de tráfico que son el resultado final del sistema.

Las soluciones DPI actuales realizan análisis de tráfico en tiempo real basándose en firmas de tráfico, es decir filtros y patrones predefinidos en el nivel de aplicación de los paquetes. Una vez aplicada esta detección al tráfico, estos dispositivos realizan una interpretación en tiempo real cuyo propósito es obtener una interpretación completa del tráfico.

La figura 1 ilustra la arquitectura de las soluciones DPI actuales. Como puede observarse, están implementadas como un sistema monolítico que detecta patrones, realiza una interpretación en tiempo real y correlaciona la información obtenida.

Una alternativa para estos sistemas DPI es la patente WO 2011051750 (PCT/EP2011/070875), este método consiste en separar la captura de tráfico del procesamiento de tráfico, permitiendo el análisis y la interpretación de la captura en una fase posterior. La captura de tráfico comprende dos tipos de información:

1. Información útil para la contabilidad de tráfico (ACC) como el volumen de *bytes* y paquetes por flujo.

2. Información útil para la clasificación de tráfico, o información META. Los paquetes clave se detectan y se guardan los campos literales de estos paquetes que contienen la información más relevante. Por ejemplo, un paquete relevante podría ser una petición HTTP y uno de sus campos clave, el nombre del *host*. Esta información META contiene los datos comprendidos necesarios para clasificar el flujo en una aplicación o servicio específico y conserva toda la información interesante que puede usarse para análisis y estudios adicionales en la fase de posprocesamiento posterior.

La información META es lo suficientemente general para permitir el uso de firmas en el posprocesamiento, pero al mismo tiempo, su volumen es significativamente inferior al tráfico original. Debido al tamaño reducido de la captura ésta puede procesarse en línea o almacenarse para su análisis en un momento diferente. Como las firmas no se aplican en el momento de la captura sino en el posprocesamiento, es posible añadir firmas según se requiera para analizar múltiples veces una misma captura, garantizando de esta manera que no se pierda información por no tener una firma específica instalada en el momento de la captura.

Problemas con las soluciones existentes

Como las soluciones DPI comerciales actuales usan un enfoque de sistema monolítico, cualquier clase de tráfico que no se interpreta con las firmas instaladas se pierde para siempre, puesto que la interpretación DPI generada no proporciona información suficiente para realizar un análisis posterior. Por tanto existe una fuerte dependencia de la actualización de las firmas. Un problema adicional de esta DPI es que la correlación con fuentes de datos externos se realiza en el momento de captura, de modo que estos datos externos deben estar disponibles para la DPI en tiempo real. Sin embargo, muchas veces estos datos externos no están disponibles en el momento de captar el tráfico.

El método descrito en el documento WO 2011051750 evita la limitación mencionada de la DPI comercial pero tiene una desventaja común con la misma. Todos los sistemas DPI presentados incluyendo el del documento WO 2011051750, con el fin de analizar el tráfico de manera apropiada, necesitan procesar por separado la información por flujo analizando paquetes en un estricto orden cronológico. Mantener el orden de todos los paquetes dentro del mismo flujo es esencial para interpretar el tráfico correctamente, por ejemplo, no tendría sentido analizar un paquete que contuviera una respuesta de un servidor antes de saber lo que se solicitó. Si sólo están presentes algunos paquetes del flujo, las firmas no clasificarán el tráfico de manera apropiada (en DPI [1] [2] [3]) o no extraerán toda la información META (documento WO 2011051750).

La desventaja mencionada en último lugar tiene un gran impacto en la aplicación práctica del análisis DPI para clasificar tráfico puesto que es bastante común que, debido a las condiciones de encaminamiento normales, los paquetes del mismo flujo viajen por diferentes líneas de transmisión. La figura 3 ilustra un punto de presencia (PoP) con dos trayectos por dirección, donde los paquetes pueden encaminarse por una línea diferente

dependiendo de la carga de tráfico de la red. En esa situación no puede analizarse cada línea por separado por DPI independientes (figura 4) porque cada DPI capturará flujos incompletos.

La única manera de analizar apropiadamente el tráfico dispersado en varias líneas es concentrar el tráfico con el fin de tener flujos completos (solución de la figura 5). Sin embargo esto tiene la desventaja de que es necesario que la DPI procese una única línea con una tasa de transmisión de datos enorme. La solución no puede escalarse cuando el número de líneas para concentrar es grande y/o son líneas de alta capacidad. Un problema adicional de este enfoque es la necesidad de un concentrador *hardware* externo. Ejemplos de estos concentradores pueden encontrarse en las soluciones de Gigamon [5] o las tarjetas de Endace [6].

Para poder analizar elevadas tasas de transmisión de tráfico es necesario dividir este tráfico en diferentes líneas de procesamiento. Esta división del tráfico no es trivial porque, tal como se introdujo anteriormente, para una detección e interpretación correctas del tráfico es necesario que todos los paquetes de un mismo flujo lleguen al mismo punto de análisis y garantizar su orden cronológico. Este tipo de clasificación puede implementarse como un equilibrio de carga basado en las cabeceras de los paquetes, aunque debe considerarse que para tasas de transmisión de tráfico elevadas esta simple operación se convierte en el cuello de botella de todo el sistema puesto que todo el tráfico pasa a través de este único punto.

La figura 6 representa la implementación deducida de la solución presentada por el documento WO2011051750. Su esquema se basa en un único trayecto de procesamiento, es decir el tráfico se captura en un único punto y todo el tráfico se procesa en un único módulo de detección. Como se indicó anteriormente, el tráfico no puede dividirse fácilmente con el fin de replicar el módulo de “detección de tráfico” y hacer que este sistema sea escalable. Esta arquitectura sólo podría analizar más de una línea de tráfico añadiendo a la misma un concentrador, aunque de nuevo, hacer que todo el tráfico pase a través de un único punto de análisis implica problemas de escalabilidad.

Sumario de la invención

Es necesario ofrecer una alternativa al estado de la técnica que cubra los huecos hallados en el mismo, particularmente los relacionados con la falta de propuestas que permitan la monitorización de tráfico solucionando el problema de la escalabilidad de los sistemas DPI actuales mientras que se permita el análisis de múltiples trayectos de tráfico.

A diferencia de las propuestas conocidas, la presente invención se refiere, en un primer aspecto, a un método para monitorización de tráfico de red, comprendiendo el método las etapas de:

a) adquirir con un módulo de captura, datos de tráfico de una línea de entrada y reenviar dichos datos de tráfico a un módulo de detección; y

b) recibir, dicho módulo de detección, dichos datos de tráfico para realizar un análisis de inspección profunda de datos de dichos datos de tráfico recibidos para realizar una detección de los mismos,

en el que dicha etapa a) comprende, adquirir dichos datos de tráfico de al menos dos de dichas líneas de entrada y clasificarlos de modo que dicho reenvío a dicho módulo de detección se realice en un orden cronológico para aquellos paquetes pertenecientes a un flujo específico de dichos datos de tráfico procedentes de dichas al menos dos líneas de entrada.

Otras realizaciones del método del primer aspecto de la invención se describen según las reivindicaciones 2 a 13 adjuntas, y en una sección posterior relacionada con la descripción detallada de varias realizaciones.

Un segundo aspecto de la presente invención se refiere a un sistema adaptado para implementar el método del primer aspecto. El sistema comprende:

- un módulo de captura dispuesto para adquirir datos de tráfico de una línea de entrada y para reenviar dichos datos de tráfico a un módulo de detección; y
- un módulo de detección dispuesto para recibir dichos datos de tráfico y para realizar un análisis de inspección profunda de datos de dichos datos de tráfico recibidos,

en el que dicho módulo de captura está configurado y dispuesto para adquirir dichos datos de tráfico de al menos dos líneas de entrada y para clasificarlos de modo que dicho reenvío a dicho módulo de detección se realice en un orden cronológico para aquellos paquetes pertenecientes a un flujo específico de dichos datos de tráfico procedentes de dichas al menos dos líneas de entrada.

Otras realizaciones del sistema del segundo aspecto de la invención se describen según las reivindicaciones 15 a 19 adjuntas, y en una sección posterior relacionada con la descripción detallada de varias realizaciones.

Breve descripción de los dibujos

Las ventajas y características previas y otras se entenderán mejor a partir de la siguiente descripción detallada de realizaciones, con referencia a lo que se adjunta, que debe considerarse de manera ilustrativa y no limitativa, donde

La figura 1 ilustra la arquitectura de soluciones DPI actuales.

La figura 2 ilustra una arquitectura alternativa para soluciones DPI actuales.

La figura 3 ilustra un ejemplo de un encaminamiento habitual de paquetes de un flujo de usuario en un punto de presencia (PoP) con dos posibles trayectos por dirección.

La figura 4 ilustra un ejemplo de análisis de los trayectos individuales en DPI independiente.

La figura 5 ilustra un ejemplo en el que se usa un concentrador antes del análisis DPI.

La figura 6 muestra el diagrama de bloques usado en el proceso descrito en la invención del documento WO2011051750.

La figura 7 muestra un ejemplo de la arquitectura básica usada en la presente invención.
La figura 8 muestra una posible implementación de la presente invención, según una realización de la presente invención.
La figura 9 muestra un ejemplo del diagrama de bloques del sistema de monitorización modular escalable.
La figura 10 muestra un ejemplo de la facilidad para introducir nuevas fases de procesamiento debido al sistema de comunicación unificado.
La figura 11 muestra un ejemplo de implementación del sistema de monitorización modular escalable, según una realización de la presente invención.
La figura 12 muestra un ejemplo de uso del sistema de monitorización modular escalable usando divisores ópticos, según una realización de la presente invención.

Descripción detallada de varias realizaciones

La presente invención propone un sistema de monitorización de tráfico modular escalable.
El sistema de monitorización modular escalable (SMMS) es una arquitectura que soluciona las desventajas presentadas de los métodos de análisis DPI. Este sistema soluciona el problema de escalabilidad mientras permite el análisis de múltiples trayectos de tráfico. Esto se realiza clasificando el tráfico como parte del análisis de manera que se garantice que todos los paquetes de un mismo flujo llegan al mismo módulo de análisis y en orden cronológico. La figura 7 ilustra esta clasificación interna.
Esta invención se define como una construcción modular basada en diferentes fases. Cada nivel del sistema contiene un tipo diferente de módulo y el número de módulos en cada nivel es completamente independiente de otros niveles. Este número de módulos por fase de análisis sólo depende de la carga de procesamiento que deben asumir puesto que todos los módulos de un mismo nivel contribuyen por igual a los mismos procesos.
Este sistema está compuesto por dos fases de alto nivel:

- Fase de captura de tráfico ("RX1, RX2, RX3" en la figura 7). Estos módulos reciben tráfico de una o varias interfaces físicas y lo reenvían a la siguiente fase. Usando las cabeceras del paquete y el número de módulos en la siguiente fase, los módulos RX reenvían paquetes garantizando que cada módulo DPI recibe flujos completos. Cada módulo procesa un número limitado de líneas de entrada ("I1, I2, I3" en la figura 7) de modo que se garantiza que todos los módulos tienen una carga apropiada (sin problemas de escalabilidad) y se consigue que todo el tráfico se consolide y divida en flujos por flujos para la siguiente fase.
- Fase de módulos de detección de tráfico ("DX1, DX2, DX3, DX4, DX5" en la figura 7). Estos módulos realizan el análisis DPI. Este nivel puede dividirse en fases de modo que las tareas que tiene que asumir cada nivel están más especializadas y esto facilita el trabajo de los módulos en las siguientes fases.

Los logros principales de esta arquitectura son:

- Altamente modular y escalable. El diseño permite añadir más módulos del tipo necesario de modo que el sistema pueda analizar un mayor número de líneas de entrada, o analizar mayores volúmenes de tráfico.
- Complejidad reducida. No hay necesidad de coordinación o dependencia entre módulos de la misma fase, y la comunicación entre módulos de diferentes fases está limitada a un esquema de productor a consumidor.
- Capacidad para analizar tráfico de diferentes trayectos de tráfico sin necesidad de usar ninguna solución de *hardware* adicional para unificar el tráfico en un mismo flujo.
- Capacidad para analizar de manera apropiada tráfico dividido entre líneas independientes.

La manera en que se diseñó esta arquitectura permite llevar a cabo esta invención completamente con *hardware* disponible en el mercado, incluyendo el uso de tarjetas de interfaz de red (NIC) de propósito general en lugar de tarjetas/dispositivos de captura de alta velocidad específicos para la adquisición de tráfico. Esto reduce drásticamente el coste de implementación del sistema mientras que aumenta su flexibilidad puesto que puede implementarse para su ejecución en una única máquina multinúcleo, aprovechándose de la arquitectura multinúcleo, o dividirse entre varias máquinas.
El sistema de monitorización modular escalable (SMMS) es un sistema diseñado para permitir la inspección profunda de paquetes (DPI) de una manera escalable garantizando su compatibilidad con los enfoques DPI que han demostrado que introducen beneficios importantes en este campo de monitorización de tráfico, tal como se produce con el documento WO 2011051750.
Esta invención se divide en diferentes fases o niveles, teniendo cada una/uno múltiples módulos que replican una funcionalidad/conjunto de funcionalidades. Todos los módulos pertenecientes al mismo nivel trabajan de manera independiente, pero comparten la carga del trabajo para el que están diseñados. Esto se consigue gracias a la distribución de carga que realizan los módulos del nivel previo.
Existe una flexibilidad total en el número de módulos que van a incluirse en cada fase, siendo la única consideración a tener en cuenta asignar más módulos para la tarea más intensa, garantizando de esta manera que no se conviertan en un cuello de botella. Esta flexibilidad permitiría incluso especificar el número de módulos a crear en cada fase en el momento de inicio del sistema.
La comunicación entre módulos en diferentes fases se realiza usando un esquema de productor-consumidor. En esta comunicación los módulos que asumen el papel de productor garantizan que enviarán al consumidor

apropiado los paquetes que debe procesar, haciendo de este modo innecesario cualquier tipo de coordinación entre módulos en la fase posterior.

Existe un mínimo de dos fases básicas para la implementación de la invención, “adquisición de tráfico” y “análisis de tráfico”, aunque la última puede dividirse en más fases como se detallará mas adelante.

1. Fase de adquisición de tráfico (RX). Esta fase captura tráfico de una o varias interfaces físicas, y envía los paquetes a la siguiente fase garantizando que los paquetes pertenecientes al mismo flujo, independientemente de dónde proceda la interfaz de entrada, se reenvían al mismo módulo de análisis. Los módulos de adquisición de tráfico analizan cabeceras de paquete, capas 2, 3 y 4, y usan esta información para clasificar los paquetes en flujos, equilibrando entonces los flujos entre los módulos de la siguiente fase.

La función que determina a qué módulo debe enviarse el paquete, toma como parámetro de entrada la información de flujo, es decir, la encapsulación de capa 2, la versión IP (no IP, v4 o v6), la IP de origen y destino (en caso de paquete IP), el tipo de protocolo de capa 4 (TCP, UDP, …), los puertos de origen y destino (en caso de TCP o UDP). Esta información se usa para calcular el módulo de salida al que entregar el paquete.

Módulo de salida = función (encapsulación L2, IP_version, IP_addr_source, IP_addr_dest, layer_4_protocol, port_source, port_dest, output_modules_number)

Esta función es simétrica en el sentido de que si se conmuta IP_addr_source con IP_addr_dest y port_source con port_dest; se produce el mismo resultado, es decir el resultado de aplicar la función a un paquete que se desplaza de A→B es igual al resultado obtenido cuando se aplica a un paquete que se desplaza de B→A, garantizando de este modo que ambos sentidos de flujo se envían al mismo módulo de análisis. La misma función se aplica en todos los módulos RX con el fin de garantizar la entrega apropiada de todos los paquetes de un flujo al mismo módulo de análisis.

La función puede implementarse usando una función *hash*. Como el resultado obtenido de aplicar la función *hash* es normalmente mayor que el número de módulos de la siguiente fase, es necesario reducir el número de posibles salidas de la función *hash* al número de colas de salida. La implementación más simple para este fin es aplicar una función de módulo.

Como en esta fase es necesario hallar en las cabeceras diferentes campos interesantes (por ejemplo encapsulación, tipo de tráfico, protocolo, etc.), se usa la información que hace referencia a estos campos de modo que en fases de análisis posteriores estos valores puedan consultarse directamente. Esta información adicional se envía junto con el paquete a la siguiente fase. La información ampliada sobre cómo se intercambia el contenido entre fases se aplica más adelante.

2. Fase de análisis de tráfico (DPI). Esta fase aplica las firmas a los paquetes con el fin de clasificar el tráfico de servicio y proporciona estadísticas de flujo, es decir información de contabilidad.

Esta fase puede dividirse en varias simplificando las tareas y aumentando la modularidad y escalabilidad. En caso de que se divida la fase, la invención se convierte en una arquitectura multifase. La figura 8 ilustra posibles esquemas que dividen la fase DPI en los siguientes niveles:

- Generación de estadísticas de contabilidad (ACC). Estos módulos combinan los paquetes en flujos, generan información de contabilidad de flujo (número de paquetes, bytes, tamaño medio de paquetes, varianza, etc.) y realizan un seguimiento del inicio y el fin de los flujos. Calculan las estadísticas de cada flujo diferente, que se identifican por las direcciones de IP/puerto junto con la encapsulación L2.
- Fase de predetección (PRE). Estos módulos implementan filtros rápidos y realizan una primera clasificación de paquetes que son candidatos para generar metadatos, rechazando los paquetes que no son interesantes y reenviando los interesantes para su análisis posterior. Las reglas de filtro que van a aplicarse en estos módulos están diseñadas para consumir muy poco en cuanto a potencia de procesamiento, de modo que sean adecuados para procesar un gran volumen de paquetes por segundo. Esta metodología no garantiza que todos los paquetes que pasan por el filtro sean útiles para generar metadatos, aunque reduce significativamente la cantidad de información que procesan los módulos de detección (con más demanda en términos informáticos).
- Fase de detección (DET). Estos módulos realizan un análisis completo buscando firmas conocidas, en paquetes con el fin de clasificarlos por tipo o servicio usando el enfoque de DPI de tráfico real [1] [2] [3] o simplemente pueden buscar paquetes interesantes con el fin de extraer la información META para su análisis posterior usando el enfoque del documento WO 2011051750 y el documento PCT/EP2011/070875. En cualquiera de estos casos las firmas/filtros se dividen en las fases de predetección y detección, mejorando de esta manera la escalabilidad de la invención.
- Generación de información de contabilidad (TRA). Recopila y genera la información de contabilidad de todos los módulos ACC, consolidando los resultados en intervalos periódicos.
- Generación de informes de tráfico o información de metadatos (META). Recopila y genera los informes de tráfico obtenidos o la información de metadatos de los diferentes módulos, consolidando toda la información.

La división de la fase DPI en las fases ACC, PRE, DET, TRA, y META sigue las mismas reglas que la división entre las fases RX y DPI. Es decir, no hay necesidad de coordinación entre módulos pertenecientes a la misma fase y no hay dependencia en el número de módulos usados en cada fase. La única diferencia en este caso comparado con la comunicación RX-DPI es que los flujos ya se ordenan en la entrada de DPI. Esto implica que no siempre es necesario comunicar cada módulo con todos los módulos en la siguiente fase. Por ejemplo en la figura 8 a la izquierda, las fases ACC y PRE contienen el mismo número de módulos, de modo que puede establecerse un único consumidor-único productor.

La información intercambiada entre fases consiste en el propio paquete más una cabecera con campos que se rellenan en las diferentes fases de la invención con información relevante descubierta en cada una; de este modo la información generada por una fase puede usarse por las posteriores. Por ejemplo, cuando un paquete se predetecta como uno interesante para su análisis posterior, esta información se incluye en la cabecera adicional del paquete de modo que el módulo DET conozca el tipo de paquete y así el tipo de análisis que aplicar.

La cabecera puede contener entre otros campos los siguientes fragmentos adicionales de información:

• Información general de paquete, como tiempo de llegada, inicio de carga útil de paquete, tamaño de carga útil, dirección del paquete (A→B o B→A) que se introduce en la fase RX.

• Encapsulación de capa 2, direcciones IP y de puerto, versión IP y protocolo de capa 4. Resultado de la función de asignación, que introduce la fase RX. Los valores no relevantes se rellenan con ceros y se marcan como tales, por ejemplo la dirección IP en un paquete no IP.

• Tipo de paquete como resultado de la predetección del paquete que se genera por la fase PRE.

La comunicación entre fases se realiza haciendo pasar el paquete con la cabecera adicional usando colas. Dependiendo de como están distribuidos los módulos hay dos alternativas principales para esta comunicación, i) si las fases se implementan en procesadores de la misma máquina con una memoria compartida, la información anterior permanece en la memoria y se intercambian punteros entre las fases. ii) Si los módulos se distribuyen entre diferentes máquinas, toda la información (paquete+cabecera adicional) se envía normalmente usando protocolos IP/UDP o IP/TCP.

La figura 8 muestra algunas posibles implementaciones del SMMS. A la izquierda se implementan todos los niveles mencionados mientras que a la derecha de la figura 8, los niveles ACC y PRE se han fusionado en uno haciendo que estos módulos ejecuten ambas funcionalidades.

Debe observarse que estos niveles pueden fusionarse o pueden dividirse en módulos más simplificados, siendo posible incluso incluir niveles nuevos para tareas nuevas que no se consideren en estos ejemplos, como por ejemplo la detección de patrones basándose en el tiempo de llegada entre paquetes.

Algunas características técnicas de sistema de monitorización modular escalable son:

• Diseño modular y multinúcleo

La funcionalidad se distribuye entre muchas fases o conjuntos de funciones que pueden realizarse por diversos módulos, ejecutándose cada uno en una unidad o núcleo de procesador separado. Esto significa que diferentes conjuntos de núcleos ejecutarán funcionalidades específicas, por ejemplo en la figura 9 el primer grupo de núcleos podrían ser módulos de adquisición de tráfico y el segundo grupo de núcleos podrían ser módulos para la generación de contabilidad.

Los módulos pueden replicarse sin ninguna restricción en cuanto al número de módulos en fases previas o siguientes. Esto implica que a aquellas funcionalidades que requieren más potencia de procesamiento se les pueden asignar más recursos, replicando más módulos.

La figura 9 representa este diseño modular en el que los recursos se asignan basándose en las necesidades de los conjuntos de funciones que se definen. En esta figura también se representan puertos de 4x10 Gbps. Seguir esta arquitectura, que puede soportar el análisis de puertos de capacidad incluso más alta, sólo dependería de la cantidad de recursos de hardware disponibles.

• Arquitectura escalable

En la invención presentada los módulos descritos pueden implementarse en una única máquina o como sistema distribuido, dividiendo la potencia de procesamiento necesaria entre todos los núcleos de las diferentes máquinas. De este modo, cuantos más recursos de *hardware* estén disponibles más carga podrá soportar el sistema. Por ejemplo teniendo suficientes recursos libres, la adición de una nueva interfaz de captura de 10 Gbps es tan simple como insertar una nueva NIC genérica y reasignar los recursos para incluir la nueva interfaz en la captura.

• De ejecución adecuada (pero no limitada) en *hardware* disponible en el mercado

El sistema de monitorización modular escalable se ha diseñado para permitir su implementación usando *hardware* de propósito general. No hay necesidad de usar tarjetas/dispositivos de captura de alta velocidad específicos para la adquisición de tráfico ni concentradores de tráfico.

Permitir que la invención se implemente por completo usando *hardware* comercial disponible en el mercado implica, por ejemplo, el uso de tarjetas de interfaz de red de 10 Gbps genéricas, reduciendo drásticamente el coste de la implementación de la invención.

• Método de comunicación unificado

El formato de comunicación entre diferentes módulos se ha definido como independiente de la funcionalidad que implementan. Consiste, como se explicó anteriormente, en una cabecera y un contenido de paquete genéricos. Esto permite introducir nuevas funcionalidades de una manera normalizada y sin la necesidad de modificar conjuntos de funciones existentes. La figura 10 representa la inserción de una nueva fase.

La comunicación entre módulos se realiza usando colas de paquetes por consumidor-productor. Cada módulo recibe paquetes de los módulos de fase previa, los procesa, añade la información obtenida relevante como resultado del procesamiento y lo envía a la cola de salida correspondiente de la siguiente fase. Esta comunicación puede realizarse dentro de la máquina en un procesador multinúcleo o entre máquinas para un escalado.

Cada módulo usa un resultado de función de asignación (por ejemplo función *hash*) calculado en la primera fase (RX) con el fin de calcular la cola de salida a la que debe enviarse el paquete. Esto

garantiza un equilibrio del tráfico entre módulos y que todos los paquetes de un mismo flujo se procesan siempre por el mismo módulo.

Las colas de comunicación se establecen automáticamente entre módulos cuando se asignan los recursos. El hecho de que un mismo núcleo reciba información de diferentes colas no afecta al orden del paquete puesto que la lectura de los paquetes en orden es una funcionalidad requerida por todos los módulos, es decir, cada módulo comprueba el "tiempo de llegada" de los paquetes (que se escribió en la cabecera por la fase RX) en todas las colas y procesa primero el más antiguo.

Realizaciones de la invención

Con el fin de ilustrar el sistema de monitorización modular escalable, se presenta una implementación particular de la invención en la que se han implementado las fases básicas mencionadas. Esta implementación se basa en la agrupación de funcionalidades en 5 tipos de módulos: RX, PRE, DET, MET y AGG. En esta implementación específica las funcionalidades se han agrupado de la siguiente manera:

Rx: Recepción

- Adquirir tráfico de la NIC
- asignar sello de tiempo
- equilibrar la carga
- ordenación de flujos

Pre: Predetección+contabilidad

- Predetección: obtener paquetes relevantes usando firmas que comprueban las posiciones conocidas en los primeros *bytes* de carga útil. Sólo se reenvían los paquetes interesantes a la fase de detección, eliminando el resto de paquetes después de usarlos para la contabilidad.
- Gestión de contabilidad n.º 1: inserción de contadores de nuevos flujos y flujos de incremento en tablas para contabilidad.
- Actualización de estadísticas

Det: Detección

- Analizar paquetes relevantes y obtener información de metadatos. En esta fase se realiza un análisis DPI completo con el fin de hallar y extraer la información META.

AGG: Agregación de información de contabilidad

- Gestión de contabilidad n.º 2: Liberar recursos cuando finaliza un flujo
- Generar mensajes de contabilidad

MET: Generación de registros de metadatos

- Generar mensajes de metadatos

La invención se ha implementado de manera que el número de módulos de cada tipo y el núcleo en el que se ejecutan se especifican en un archivo de configuración que se carga en el momento de inicio. El número de puertos que van a analizarse y la asociación entre puertos y módulos RX también se especifica en este archivo de configuración. Esta manera flexible de especificar la configuración de la invención permite volver a configurar la invención en segundos adaptándola a demandas específicas del tráfico que va a analizarse, por ejemplo la necesidad empezar a analizar un puerto adicional. El siguiente texto es el segmento del archivo de configuración para los módulos de la invención ilustrados en la figura 11.

```
[RX]
tamaño de cola rx=512; tamaño de cada cola NIC, por defecto 256
índice de núcleo=1
índice de puerto=0
índice de núcleo=2
índice de puerto=1
índice de núcleo=3
índice de puerto=2
índice de núcleo=4
índice de puerto=3

[PRE]
    tamaño de cola rx=512; tamaño de cada anillo de entrada de este núcleo, por defecto 256
    índice de núcleo=5
    índice de núcleo=6
    índice de núcleo=7
    índice de núcleo=8

    [DET]
    tamaño de cola rx=256; tamaño de cada anillo de entrada de este núcleo, por defecto 256
índice de núcleo=9
índice de núcleo=10
índice de núcleo=11

[AGG]
```

tamaño de cola rx=256; tamaño de cada anillo de entrada de este núcleo, por defecto 256
tamaño de ráfaga de paquete rx=32; número de paquetes que intenta leer en cada ráfaga, por defecto 32
índice de núcleo=12

[MET]
tamaño de cola rx=256; tamaño de cada anillo de entrada de este núcleo, por defecto 256
tamaño de ráfaga de paquete rx=32; número de paquetes que intenta leer en cada ráfaga, por defecto 32
índice de núcleo=13

La figura 11 representa los diferentes módulos usados en este caso particular y la manera en que se combinan con el fin de analizar tráfico.

En este ejemplo se analiza el tráfico de cuatro enlaces de 10 Gbps. Dependiendo de en qué interfaz llega un paquete se analiza por un módulo RX diferente. Esta implementación se diseño para analizar tráfico en un punto de presencia (PoP) siguiendo el esquema ilustrado en la figura 12. Como puede observarse en esta figura, se usan divisores ópticos para enviar una copia del tráfico de 40 Gbps al sistema de monitorización modular escalable.

En esta invención, cuando un paquete llega a un módulo RX se añade una cabecera adicional. Esta cabecera se usa para almacenar información adicional generada por los diferentes módulos cuando se analiza el paquete. Uno de los fragmentos de información que se rellena en esta cabecera es el momento en que el paquete entró en el módulo RX, este momento se denominará a partir de ahora sello de tiempo de paquete. Una función *hash* simétrica se alimenta con campos específicos de cabeceras del paquete (capas 2, 3 y 4) de modo que el *hash* resultante es el mismo para todos los paquetes que se desplazan de A→B y para los paquetes que se desplazan de B→A. Este *hash* también se incluye en la cabecera adicional de paquete. Una vez que se ha aplicado la función *hash*, se usa este valor para realizar un equilibrio de carga de los paquetes para los diferentes módulos PRE; de este modo se garantiza que, independientemente de en qué interfaz se reciba un paquete, todos los paquetes de un mismo flujo llegarán al mismo módulo PRE. Este procedimiento envía los paquetes al módulo PRE apropiado realizando al mismo tiempo un equilibrio de carga entre todos ellos.

Es importante indicar que a pesar de que el paquete se desplaza virtualmente por toda la invención, en esta implementación específica, el paquete se guarda en memoria con la cabecera adicional descrita y que sólo los punteros para los paquetes se hacen pasar entre los módulos. Esta implementación permite tener exactamente el mismo tipo de colas para recibir paquetes en todos los módulos y optimiza el uso de memoria. En caso de que los módulos se distribuyan entre diferentes máquinas esta comunicación podría realizarse enviando los paquetes con la cabecera adicional usando un zócalo UDP o TCP.

Cada módulo PRE tiene tantas colas de entrada como módulos RX en el sistema y sigue comprobando las diferentes colas para recibir nuevos paquetes. Con el fin de garantizar que los paquetes se analizan en orden, el paquete más antiguo entre todas las colas de entrada es el primero en procesarse. Una vez que el paquete se ha quitado de la cola, se obtiene la longitud de paquete y se usa para la contabilidad de flujos.

Otra funcionalidad implementada en el módulo PRE es la predetección, es decir el uso de reglas de fácil comprobación que permitan determinar si un paquete es interesante. Éstos son algunos ejemplos de posibles criterios para determinar si un paquete es sospechoso de ser un paquete relevante para análisis:

- La carga útil del paquete TCP empieza con “GET”. Este paquete es sospechoso de ser un mensaje HTTP GET.
- La carga útil del paquete TCP empieza con “POST”. Este paquete es sospechoso de ser un mensaje HTTP POST.
- La carga útil del paquete TCP empieza con “HTTP”. Este paquete es sospechoso de ser un mensaje de respuesta HTTP.
- Paquete UDP con puerto de origen igual a 53. Este paquete es sospechoso de ser una resolución DNS.

Pueden definirse otras reglas para protocolos más complejos, por ejemplo si los bytes 3 y 5 de carga útil tienen valores específicos se considera que es un paquete de una determinada aplicación. Un ejemplo de un tipo de paquete que directamente no se considera para su análisis posterior son los mensajes ACK.

Una vez que se ha usado la longitud del paquete para la contabilidad y se ha determinado si su uso tiene interés para un análisis más profundo es el momento de hacer que avance al siguiente módulo o de rechazarlo. En caso de que los paquetes no se hayan clasificado como interesantes, se eliminan, liberando la memoria en la que estaban guardados, si no, usando el valor *hash* calculado en el módulo RX se selecciona un módulo DET y el puntero del paquete se inserta en su cola incluyendo en el mismo el resultado de la predetección.

Cada módulo DET tiene tantas colas de entrada como módulos PRE y los paquetes se sacan de la cola en orden cronológico. Una vez que se saca un paquete de la cola, se analiza según la clasificación previa que se realizó en el módulo PRE. Durante este análisis, mucho más completo que en PRE, puede ocurrir que se descubra que el paquete se haya clasificado previamente de forma incorrecta y, como no es interesante para su análisis posterior, se elimina. En caso de que el paquete se haya clasificado previamente de forma correcta, se extraen los campos relevantes de la carga útil para generar un paquete de metadatos. Por ejemplo, para un mensaje HTTP GET algunos campos interesantes que guardar son la petición HTTP, el *host* y el agente de usuario. Toda

esta información relevante para generar el mensaje de metadatos se guarda con el propio paquete y el puntero para el paquete se envía al módulo MET.

El módulo MET recibe paquetes de todos los módulos DET y, según su sello de tiempo, los saca de la cola en orden. Este módulo genera los paquetes META, que son mensajes que contienen sólo los campos más relevantes de los paquetes capturados más significativos. El resultado de este módulo combinado con los paquetes generados por el módulo AGG forman la interfaz de metadatos. A continuación se muestra un ejemplo, convertido a caracteres legibles, de algunos mensajes de metadatos. Como puede observarse determinados campos tales como dirección MAC, sello de tiempo, IP, protocolo y puertos están presentes para todos los paquetes mientras que otros campos más específicos son particulares de cada tipo de metadatos.

002290A1D4C0 1301447734.403577 1395792141:26303 > 1249764936:80 TCP 777 737 18 VLAN_Q 50 GET_HTTP 07 GET /complete/search?hl=es&q=cubiertasyserr HTTP/1.1 Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; GTB6.6) clients5.google.com
001193A38B40 1301447734.406502 3653517288:25841 > 1395069353:17581 UDP 357 329 VLAN_Q 50 BITORRENT_SIGNALING_1 n_peers: 1 1395069353:17581
001193A38B40 1301447734.414819 1394796038:63420 > 1334864642:11340 TCP 73 33 18 VLAN_Q 50 EMULE_SIGNALING_1 E0FF5C9A9CBFE2A2556B056B425A9578 9 000000440040000080
002290A1D4A8 1301447734.415134 1395069941:54649 > 3269476883:80 TCP 458 418 18 VLAN_Q 50 GET_HTTP 07 GET /rsrc.php/v1/za/r/3HjfDY8tjji.gif HTTP/1.1 Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; GTB6.3) static.ak.fbcdn.net
001193A38B40 1301447734.446418 1056264837:17103 > 1395069289:7456 UDP 383 355 VLAN_Q 50 BITORRENT_SIGNALING_1 n_peers: 5 3178423692:16880 1395069289:7456 3654909470:28732 1107856780:57159 1296258272:16880

El módulo AGG se encarga de limpiar periódicamente la tabla de información de agregación generando mensajes de contabilidad. Cada vez que se limpia la tabla se genera un mensaje indicativo del número de flujos almacenados en la tabla y a continuación un mensaje por cada uno de los flujos indicando encapsulación, protocolo, IP, puertos, número de bytes y paquetes en cada dirección y campos adicionales que pueden usarse en el propio sistema de monitorización o por un sistema adicional que implemente el método para minimizar el posprocesamiento de tráfico de red, como en el documento PCT/EP2011/070875. A continuación se muestra un ejemplo, convertido a caracteres legibles, de un mensaje de contabilidad que contiene la información de contabilidad de 5 flujos. La primera línea indica en el tercer campo que las siguientes 5 líneas contienen la información de contabilidad de 5 flujos. Esta información de agregación corresponde al miércoles, 30 de marzo de 2011, 01:16:00 GMT, que se expresa en formato de tiempo Epoch en el segundo campo de la primera línea.

000000000000 1301447760.000000 5:0 > 0:0 UNK 0 0 00 ACCOUNTING_INFO
1394646547:2883 1305587570:40939 TCP 0 1 0 75 VLAN_Q 50 00 00
3558207238:4321 1395069156:4026 TCP 2 0 2680 0 VLAN_Q 50 00 00
3678294024:13977 1395069353:17581 UDP 1 0 129 0 VLAN_Q 50 00 00
1371041298:51158 1394646813:45262 TCP 0 1 0 40 VLAN_Q 50 00 00
1474150759:3923 1395791977:12000 TCP 0 1 0 48 VLAN_Q 50 00 00

En todos los módulos se guardan estadísticas, que permiten comprobar en cualquier momento el número de paquetes capturados, el número total de paquetes previamente clasificados como interesantes, el número de paquetes de metadatos generados, el número de paquetes procesados por cada módulo, etc. Comprobando estas estadísticas es posible determinar si la configuración de módulos elegida es apropiada o si debe cambiar modificando el número de módulos de cada tipo cambiando el archivo de configuración.

Ventajas de la invención

Las ventajas principales del sistema de monitorización modular escalable son flexibilidad, escalabilidad, modularidad y la posibilidad de implementar la invención completamente con hardware disponible en el mercado. Esta solución permite capturar tráfico de diferentes líneas de entrada y vuelve a ordenar los paquetes con el fin de garantizar que todos los paquetes pertenecientes a un mismo flujo se proporcionan en orden cronológico a un módulo DPI. Este procedimiento evita la necesidad de concentradores de tráfico, y permite paralelizar el análisis DPI usando diferentes módulos de análisis.

Se garantiza la escalabilidad puesto que la replicación de módulos con el fin de distribuir la carga de tráfico no afecta a la lógica usada para el procesamiento de paquetes. Esto se consigue puesto que todos los paquetes de los mismos flujos se procesan siempre por el mismo módulo, algo que no era posible usando las soluciones previas [1] [2] [3] y el documento WO 2011051750.

La invención se ha definido de manera modular, permitiendo la distribución de funcionalidades entre diferentes fases según sus necesidades en cuanto a potencia de procesamiento. De este modo es posible reducir el cuello de botella del sistema únicamente implementando el número de módulos que ejecutan una determinada funcionalidad. Adicionalmente, el diseño modular permite dividir la funcionalidad entre diferentes máquinas de *hardware*. La interrelación entre módulos se reduce a un mínimo, con sólo colas de entrada y de salida, aumentando significativamente la cantidad de tráfico que puede procesarse comparado con otras alternativas.

Flexible, puesto que la invención permite añadir nuevos módulos con funcionalidades adicionales gracias al método común de comunicación entre módulos. Así, la invención es fácilmente ampliable con nuevos conjuntos de funcionalidades como por ejemplo la detección de patrones estadísticos, la detección de un comportamiento sospechoso de ataque, etc.

La invención puede implementarse usando *hardware* disponible en el mercado, disminuyendo de este modo significativamente el coste de implementar sistemas DPI de alta capacidad.

ACRÓNIMOS

ADSL	*Asymmetric Digital Subscriber Line*; línea de abonado digital asimétrica
AR	*Access Router*; encaminador de acceso
BRAS	*Broadband Remote Access Server*; servidor de acceso remoto de banda ancha
DPI	*Deep Packet Inspection*; inspección profunda de paquetes
HTTP	*HyperText Transfer Protocol*; protocolo de transferencia de hipertexto
IP	*Internet Protocol*; protocolo de Internet
PoP	*Point of Presence*; punto de presencia
SMMS	*Scalable Modular Monitoring System*; sistema de monitorización modular escalable
TCP	*Transmission Control Protocol*; protocolo de control de transmisión
UDP	*User Datagram Protocol*; protocolo de datagrama de usuario
VLAN	*Virtual Local Area Network*; red de área local virtual
QinQ	*802.1Q en 802.1Q (VLAN inside VLAN)*; 802.1Q en 802.1Q (VLAN en VLAN)

REFERENCIAS

[1] Sandvine. http://www.sandvine.com/
[2] iPoque. http://www.ipoque.com/
[3] Cisco SCE (Servicio Control Engine)
[4] Gigamon. http://www.gigamon.com/
[5] Enlace http://www.endace.com/

REIVINDICACIONES

1. Método para monitorización de tráfico de red, que comprende las etapas de:

a) adquirir con un módulo de captura, datos de tráfico de una línea de entrada y reenviar dichos datos de tráfico a un módulo de detección; y

b) recibir, dicho módulo de detección, dichos datos de tráfico para realizar un análisis de inspección profunda de datos de dichos datos de tráfico recibidos para realizar una detección de los mismos,

estando caracterizado dicho método porque dicha etapa a) comprende, adquirir dichos datos de tráfico de al menos dos de dichas líneas de entrada y clasificarlos de modo que dicho reenvío a dicho módulo de detección se realice en un orden cronológico para aquellos paquetes pertenecientes a un flujo específico de dichos datos de tráfico procedentes de dichas al menos dos líneas de entrada siendo enviados al mismo módulo de detección, y porque la detección de dicha etapa b) se realiza por al menos dos módulos de detección para realizar tareas intensas de manera distribuida.

2. Método según la reivindicación 1, en el que dichos datos de tráfico procedentes de dichas al menos dos líneas de entrada se adquieren en dicha etapa a) por al menos dos módulos de captura para realizar tareas intensas de manera distribuida.

3. Método según la reivindicación 1 ó 2, que comprende trabajar en al menos dos niveles diferentes para dicho módulo de captura y dicho módulo de detección.

4. Método según la reivindicación 1, que comprende además equilibrar por igual dicho flujo de dichos datos de tráfico entre dichos al menos dos módulos de detección.

5. Método según la reivindicación 2, en el que cada uno de dichos módulos de captura trabaja independientemente del otro.

6. Método según la reivindicación 1, 4 ó 5, en el que cada uno de dichos módulos de detección trabaja independientemente del otro.

7. Método según la reivindicación 1, que comprende, con el fin de realizar dicha clasificación, analizar la cabecera de dichos paquetes pertenecientes a un flujo específico de dichos datos de tráfico.

8. Método según la reivindicación 7 cuando depende de la reivindicación 6, que comprende determinar a cuál de dichos módulos de detección dicho módulo de captura reenvía dicho paquete perteneciente a un flujo específico de dichos datos de tráfico según la siguiente función:

Módulo de salida = función (encapsulación L2, IP_version, IP_addr_source, IP_addr_dest, layer_4_protocol, port_source, port_dest, output_modules_number)

en la que *encapsulación L2* es dicha cabecera; *IP_version* es la versión de protocolo IP (no IP, v4 o v6); *IP_addr_source*, *IP_addr_dest* son las IP de origen y destino (en caso de que dicho paquete sea IP); *layer_4_protocol* es el tipo de protocolo (TCP, UDP); *port_source, port_dest* son los puertos TCP o UDP de origen y destino, donde dicho paquete es TCP o UDP, y *output_modules_number* es el número de módulos en la siguiente fase.

9. Método según la reivindicación 8, que comprende además calcular, sobre el resultado de dicha función, una función *hash* simétrica con el fin de determinar a cuál de dichos módulos de detección debe enviarse dicho paquete perteneciente a un flujo específico de dichos datos de tráfico, dicha función *hash* simétrica produce el mismo resultado para paquetes que se desplazan en cualquier dirección.

10. Método según la reivindicación 9, que comprende procesar, cada uno de dichos módulos de detección, dicho paquete recibido y añadir, a dicho paquete, información relevante obtenida como resultado de dicho procesamiento.

11. Método según la reivindicación 1, que comprende dividir la funcionalidad entre diferentes máquinas de hardware para realizar las tareas de dichas etapas a) y b).

12. Sistema para monitorización de tráfico de red, que comprende:

- un módulo de captura (RX1, RX2, RX3) dispuesto para adquirir datos de tráfico de una línea de entrada (I1, I2, I3) y para reenviar dichos datos de tráfico a un módulo de detección (D1, D2, D3, D4, D5); y

- un módulo de detección (D1, D2, D3, D4, D5) dispuesto para recibir dichos datos de tráfico y para realizar un análisis de inspección profunda de datos de dichos datos de tráfico recibidos,

en el que dicho sistema está caracterizado porque dicho módulo de captura (RX1, RX2, RX3) está configurado y dispuesto para adquirir dichos datos de tráfico de al menos dos líneas de entrada (I1, I2, I3) y para clasificarlos de modo que dicho reenvío a dicho módulo de detección (D1, D2, D3, D4, D5) se realiza en un orden cronológico para aquellos paquetes pertenecientes a un flujo específico de dichos datos de tráfico procedentes de dichas al menos dos líneas de entrada (I1, I2, I3).

13. Sistema según la reivindicación 12, que comprende una estructura modular formada por al menos dos niveles diferentes, un nivel que contiene al menos dos módulos de captura (RX1, RX2, RX3) y otro nivel que contiene al menos dos módulos de detección (D1, D2, D3, D4, D5).

14. Sistema según la reivindicación 13, en el que el número de dichos módulos de captura (RX1, RX2, RX3) es independiente del número de dichos módulos de detección (D1, D2, D3, D4, D5) siendo dicho/s módulo/s de captura (RX1, RX2, RX3) y dicho/s módulo/s de detección (D1, D2, D3, D4, D5) configurados, respectivamente, para realizar las etapas a) y b) del método según cualquiera de las reivindicaciones 1 a 13.

15. Sistema según la reivindicación 12, estando implementado dicho sistema en una única máquina.

16. Sistema según la reivindicación 12, estando implementado dicho sistema en un sistema distribuido.

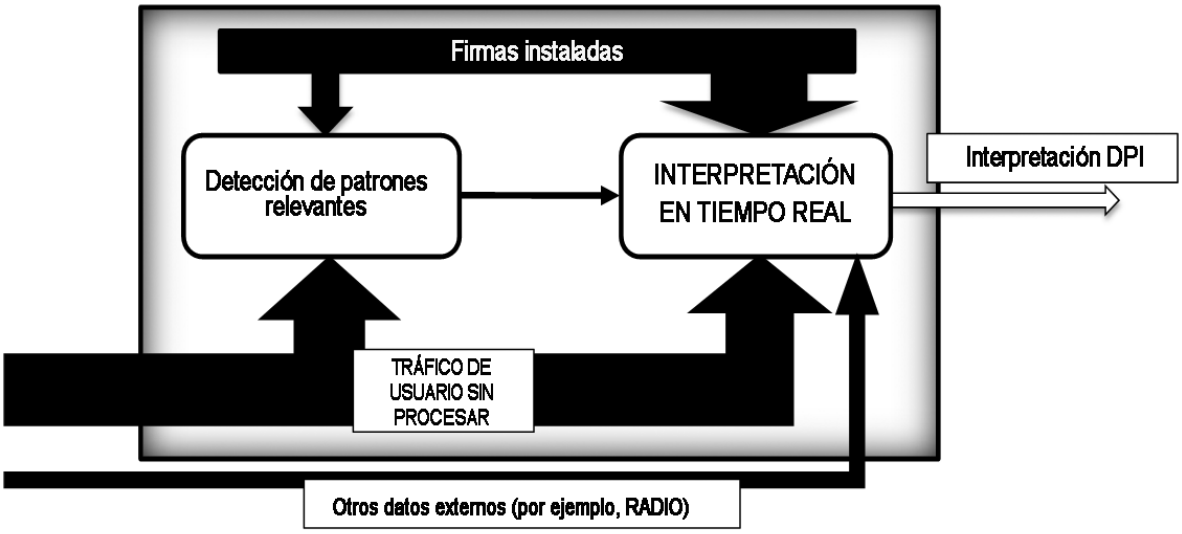

Figura 1

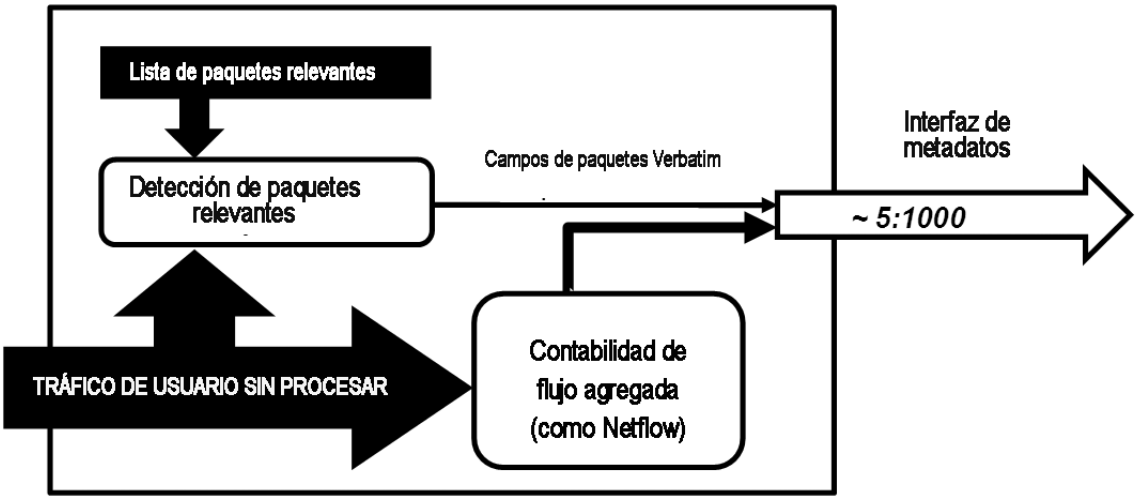

Figura 2

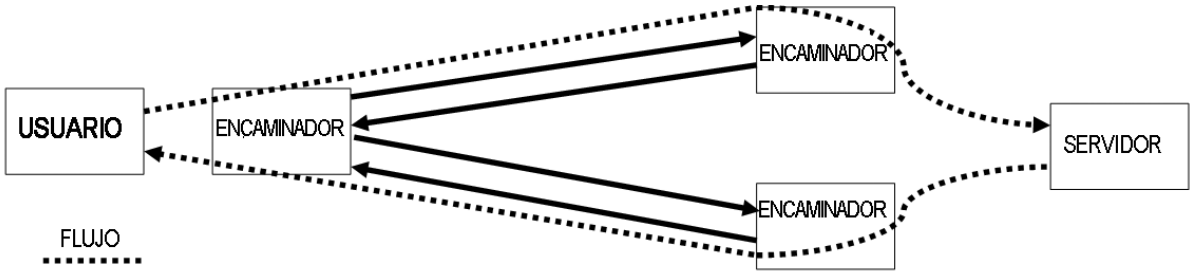

Figura 3

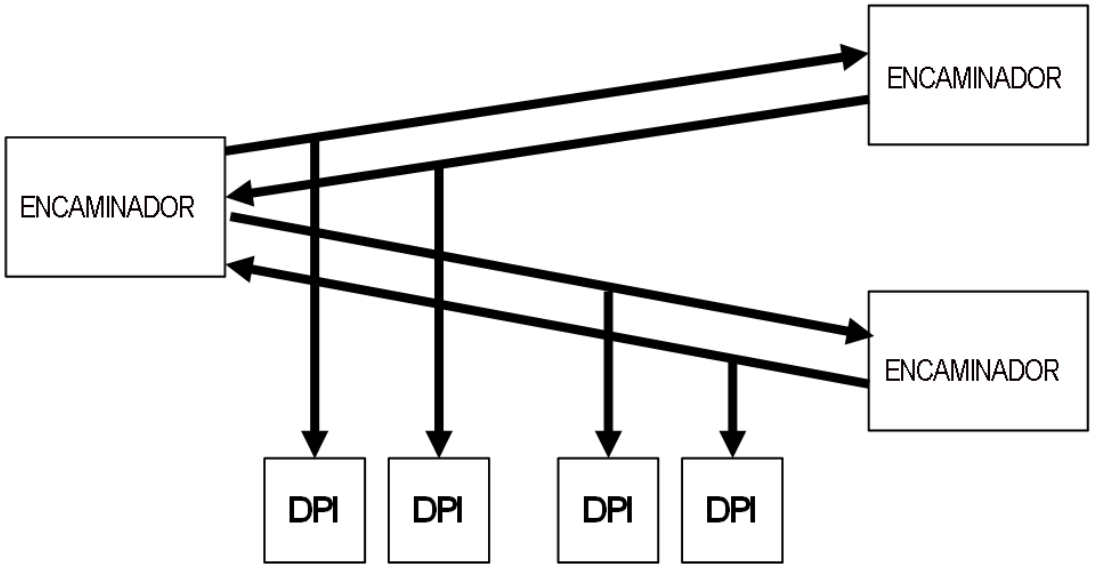

Figura 4

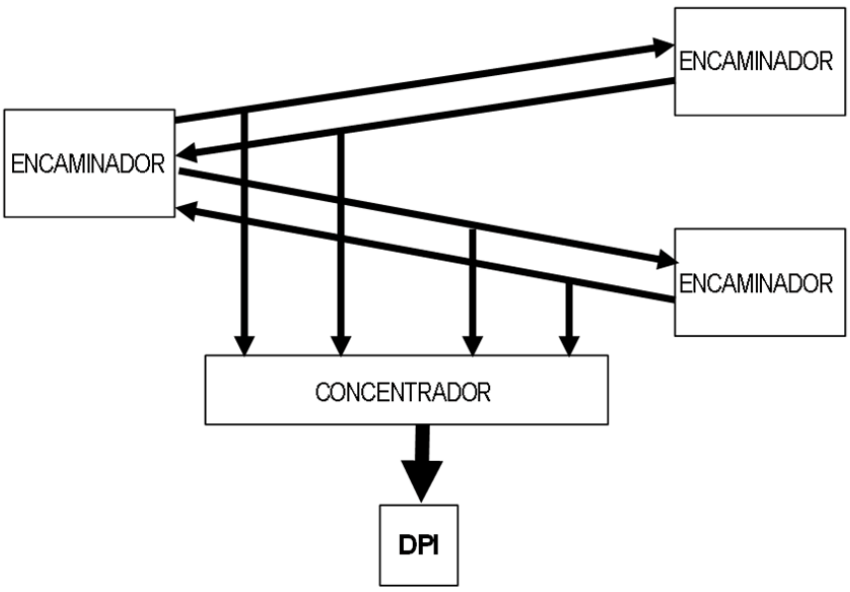

Figura 5

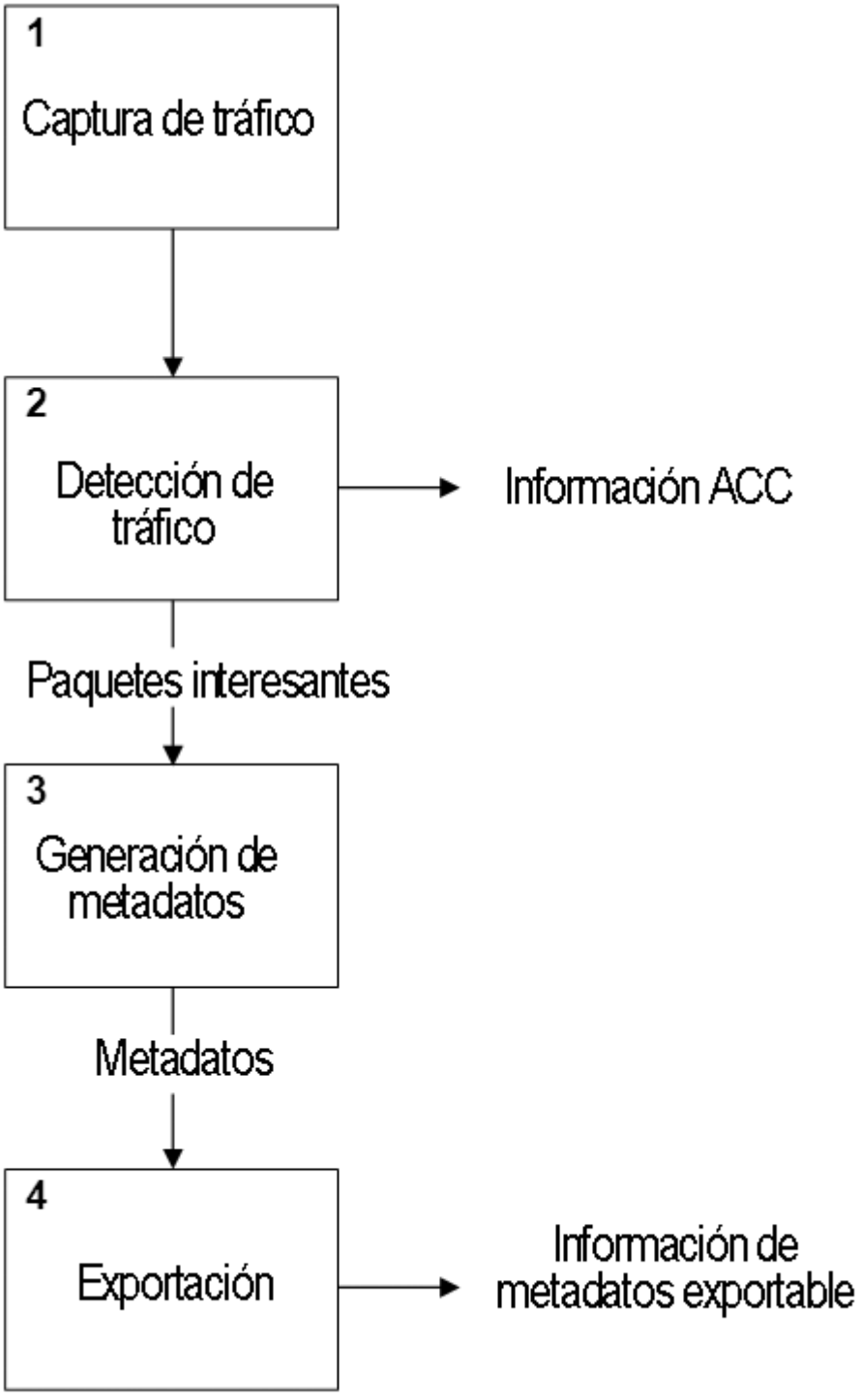
1
Captura de tráfico
2
Detección de tráfico
Información ACC
Paquetes interesantes
3
Generación de metadatos
Metadatos
4
Exportación
Información de metadatos exportable

Figura 6

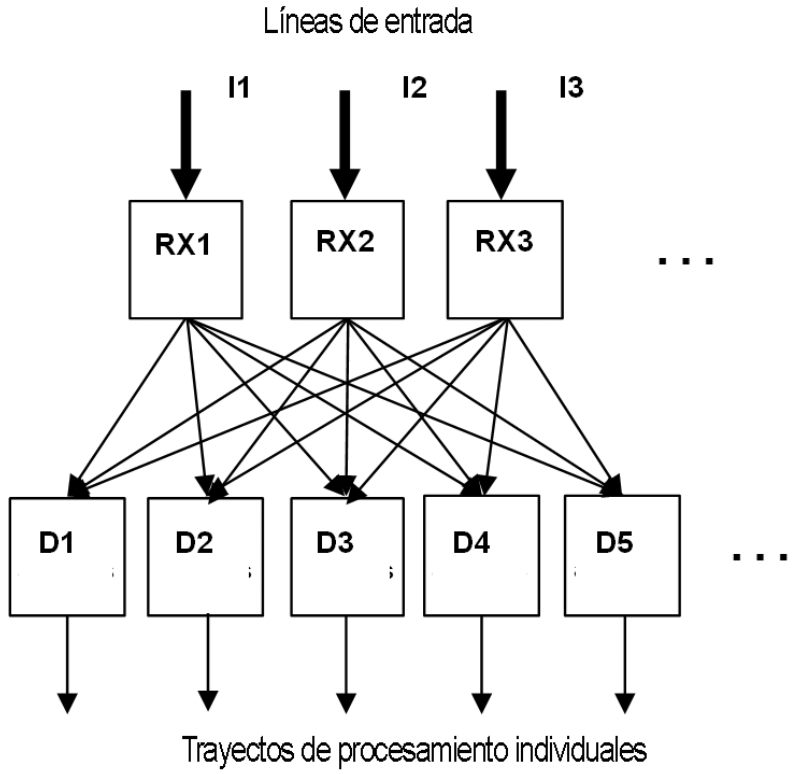

Figura 7

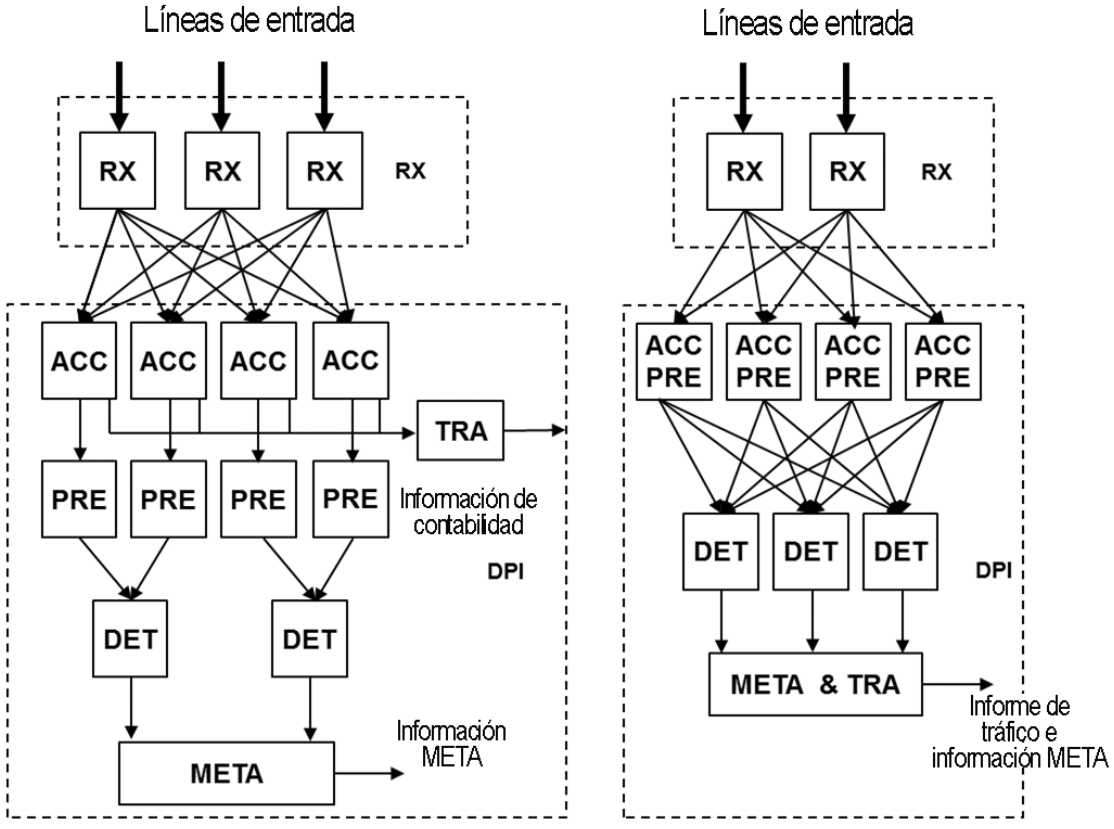

Figura 8

Figura 9

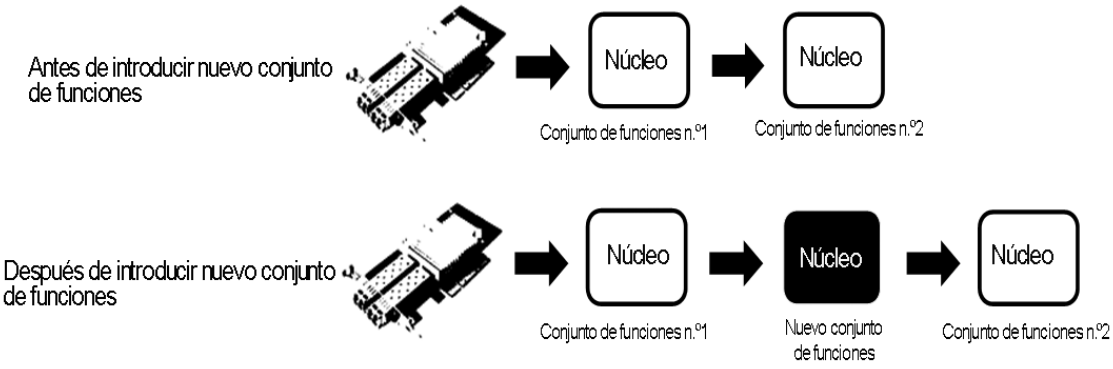

Figura 10

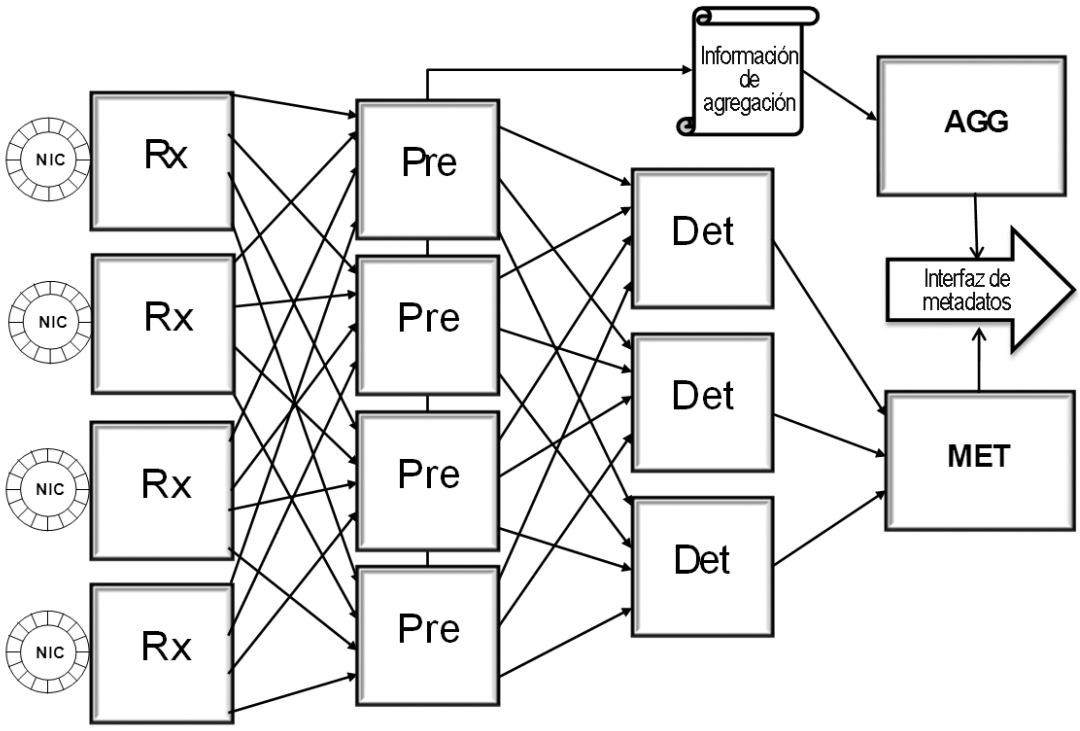

Figura 11

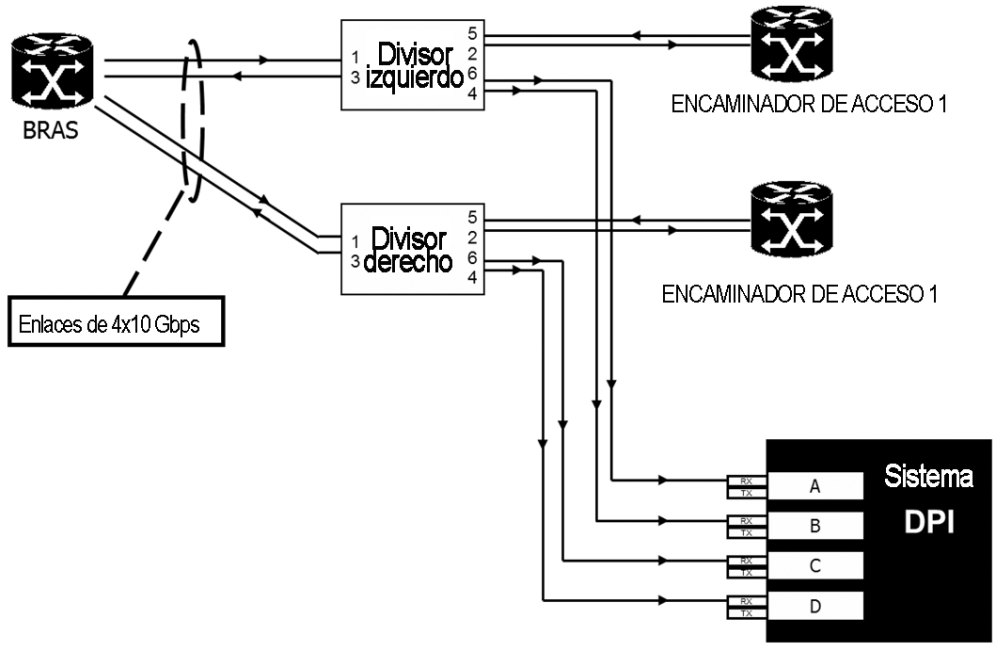

Figura 12

OFICINA ESPAÑOLA
DE PATENTES Y MARCAS

ESPAÑA

(21) N.º solicitud: 201230419

(22) Fecha de presentación de la solicitud: 20.03.2012

(32) Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TECNICA

(51) **Int. Cl. :** ***H04L12/26*** (2006.01)

DOCUMENTOS RELEVANTES

Categoría	(56) Documentos citados	Reivindicaciones afectadas
X	MATTHIAS VALLENTIN et al.: "The NIDS Cluster: Scalable, Stateful Network Intrusion Detection on Commodity Hardware II, RECENTADVANCES IN INTRUSION DETECTION; SPRINGERBERLI N HEIDELBERG, 05.09.2007, PÁGINAS 107-126, todo el documento.	1-16
A	EP 1511229 A1 (FUJITSU LTD) 02.03.2005, párrafos 15-61,66-70; figuras 1,7,8B,10.	1-16
A	PAXSON V: "Bro: a system for detecting network intruders in real-time", COMPUTER NETWORKS, ELSEVIER SCIENCE PUBLISHERS B.V., AMSTERDAM, NL, 14.12.1999, vol. 31, no. 23-24, páginas 2435-2463, ISSN: 1389-1286, DOI:10.1016/51389-1286(99)00112-7, todo el documento.	1-16
A	KRUEGEL C et al.: "Stateful intrusion detection for high-speed networks", PROCEEDINGS 2002 IEEE SYMPOSIUM ON SECURITY AND PRIVACY - 12-15 MAY 2002 - BERKELEY, CA, USA; [PROCEEDINGS OF THEIEEE SYMPOSIUM ON SECURITY AND PRIVACY J, IEEE COMPUT. SOC - LOS ALAMITOS, CA, USA, páginas 285-293, DOI: 10.1109/SECPRI.2002.1004378 ISBN: 978-8-7695-1543-4, todo el documento.	1-16
A	GIORGOS VASIALIDIS et al.: "MIDeA: a multi-parallel intrusion detection architecture", PROCEEDINGS OF THE 18TH ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 01.01.2011, páginas 297-308, New York, NY, USA ISBN: 978-1-45-030948-6, todo el documento.	1-16

Categoría de los documentos citados
X: de particular relevancia
Y: de particular relevancia combinado con otro/s de la misma categoría
A: refleja el estado de la técnica
O: referido a divulgación no escrita
P: publicado entre la fecha de prioridad y la de presentación de la solicitud
E: documento anterior, pero publicado después de la fecha de presentación de la solicitud

El presente informe ha sido realizado

☒ para todas las reivindicaciones ☐ para las reivindicaciones nº:

Fecha de realización del informe	Examinador	Página
04.02.2014	J. Santaella Vallejo	1/4

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

H04L

Bases de datos electrónicas consultadas durante la búsqueda (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

INVENES, EPODOC, WPI

Fecha de Realización de la Opinión Escrita: 04.02.2014

Declaración

Novedad (Art. 6.1 LP 11/1986)	Reivindicaciones	2-16	**SI**
	Reivindicaciones	1	**NO**
Actividad inventiva (Art. 8.1 LP11/1986)	Reivindicaciones		**SI**
	Reivindicaciones	1-16	**NO**

Se considera que la solicitud cumple con el requisito de aplicación industrial. Este requisito fue evaluado durante la fase de examen formal y técnico de la solicitud (Artículo 31.2 Ley 11/1986).

Base de la Opinión.-

La presente opinión se ha realizado sobre la base de la solicitud de patente tal y como se publica.

1. Documentos considerados.-

A continuación se relacionan los documentos pertenecientes al estado de la técnica tomados en consideración para la realización de esta opinión.

Documento	Número Publicación o Identificación	Fecha Publicación
D01	MATTHIAS VALLENTIN et al.: "The NIDS Cluster: Scalable, Stateful Network Intrusion Detection on Commodity Hardware II, RECENTADVANCES IN INTRUSION DETECTION; SPRINGERBERLI N HEIDELBERG, PÁGINAS 107-126, todo el documento.	05.09.2007
D02	EP 1511229 A1 (FUJITSU LTD)	02.03.2005
D03	PAXSON V: "Bro: a system for detecting network intruders in real-time", COMPUTER NETWORKS, ELSEVIER SCIENCE PUBLISHERS B.V., AMSTERDAM, NL, vol. 31, no. 23-24, páginas 2435-2463, ISSN: 1389-1286, DOI:10.1016/51389-1286(99)00112-7, todo el documento.	14.12.1999
D04	KRUEGEL C et al.: "Stateful intrusion detection for high-speed networks", PROCEEDINGS 2002 IEEE SYMPOSIUM ON SECURITY AND PRIVACY - 12-15 MAY 2002 - BERKELEY, CA, USA; [PROCEEDINGS OF THEIEEE SYMPOSIUM ON SECURITY AND PRIVACY J, IEEE COMPUT. SOC - LOS ALAMITOS, CA, USA, páginas 285-293, DOI: 10.1109/SECPRI.2002.1004378 ISBN: 978-8-7695-1543-4, todo el documento.	12.05.2002
D05	GIORGOS VASIALIDIS et al.: "MIDeA: a multi-parallel intrusion detectionarchitecture", PROCEEDINGS OF THE 18TH ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, páginas 297-308, New York, NY, USA ISBN: 978-1-45-030948-6, todo el documento.	01.01.2011

2. Declaración motivada según los artículos 29.6 y 29.7 del Reglamento de ejecución de la Ley 11/1986, de 20 de marzo, de Patentes sobre la novedad y la actividad inventiva; citas y explicaciones en apoyo de esta declaración

Método para monitorización de tráfico de red, que comprende las etapas de:

1. Adquirir con un módulo de captura, datos de tráfico de una línea de entrada y reenviar dichos datos de tráfico a un módulo de detección; comprende, adquirir dichos datos de tráfico de al menos dos de dichas líneas de entrada y clasificarlos de modo que dicho reenvío a dicho módulo de detección se realice en un orden cronológico para aquellos paquetes pertenecientes a un flujo específico de dichos datos de tráfico procedentes de dichas al menos dos líneas de entrada siendo enviados al mismo módulo de detección (figura 1; sección “Distribution Load”).

2. recibir, dicho módulo de detección, dichos datos de tráfico para realizar un análisis de inspección profunda de datos de dichos datos de tráfico recibidos para realizar una detección de los mismos, donde la detección se realiza por al menos dos módulos de detección para realizar tareas intensas de manera distribuida (figura 1; sección “Distribution Analysis”).

Por lo tanto a la luz de D01, la invención no es nueva tal como se establece en el artículo 6 de la Ley de Patentes 1986.

Reivindicaciones 2-11
La característica como el numero de módulos de captura o detección se describe en el documento D01 proporcionan las mismas ventajas que la presente solicitud. El experto en la materia podría por lo tanto considerar como opción normal de diseño incluir esta característica para resolver el problema planteado.

Por lo tanto a la luz de D01, las reivindicaciones 2-11 son nuevas pero carece de actividad inventiva tal como se establece en los artículos 6 y 8 de la Ley de Patentes 1986.

Reivindicaciones 12-16
Se considera que las características de diseño divulgadas en las reivindicaciones de sistema 12-16 son meras ejecuciones del método 1-11, siendo un modo particular de realización para un experto en la materia.

Por lo tanto a la luz de D01, las reivindicaciones 12-16 son nuevas pero carece de actividad inventiva tal como se establece en los artículos 6 y 8 de la Ley de Patentes 1986.