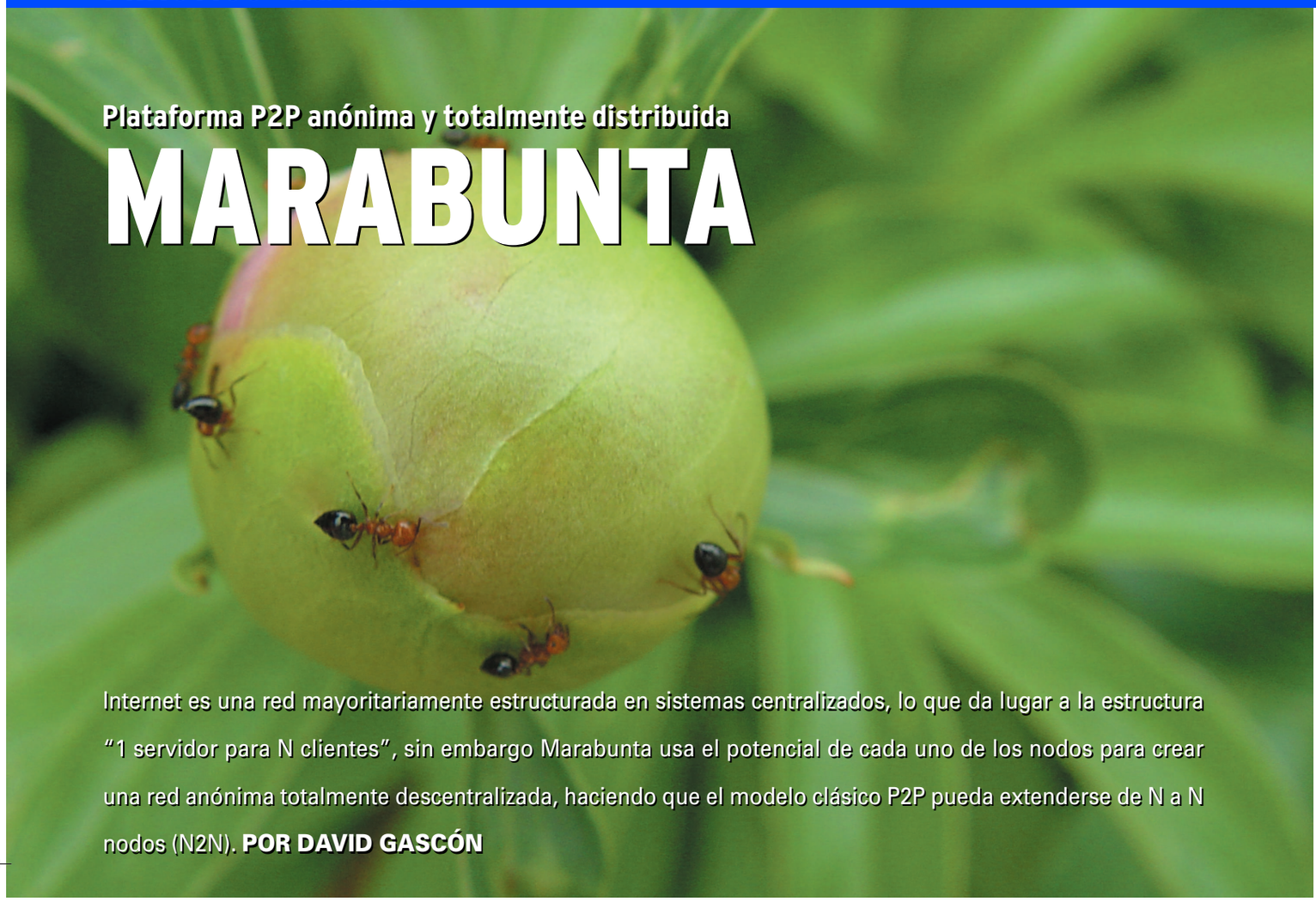

Plataforma P2P anónima y totalmente distribuida

MARABUNTA

Internet es una red mayoritariamente estructurada en sistemas centralizados, lo que da lugar a la estructura "1 servidor para N clientes", sin embargo Marabunta usa el potencial de cada uno de los nodos para crear una red anónima totalmente descentralizada, haciendo que el modelo clásico P2P pueda extenderse de N a N nodos (N2N). **POR DAVID GASCÓN**

Internet se está convirtiendo cada día más en un campo de batalla al hablar de privacidad y censura en las comunicaciones, debido a que los usuarios empiezan a ser conscientes de que Internet no es la "panacea" que se creía en términos de anonimato y privacidad.

En Internet todas las comunicaciones se registran (se "tracean"), por lo que cada vez que conectamos a un servidor para visitar una página web o para descargar nuestro correo queda grabada información esencial como nuestra IP, lo que al fin y al cabo es de forma unívoca nuestra identidad en la red. Esta información es almacenada en los ficheros del registro de las comunicaciones de la empresa que nos brinda el servicio de acceso a Internet (ISP, Internet Server Provider) y del servidor final accedido.

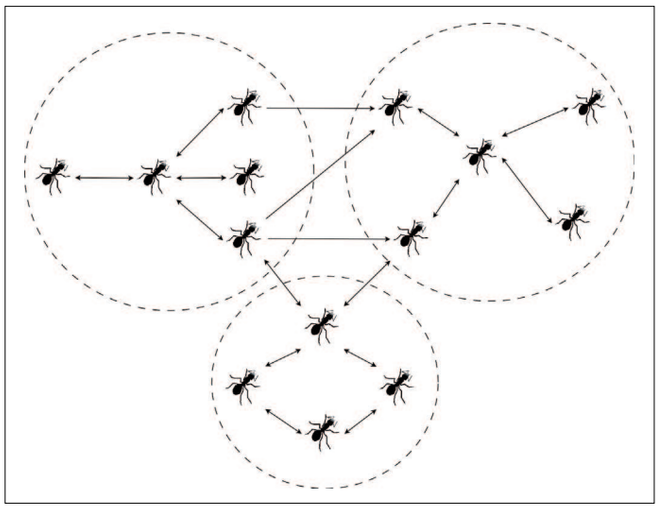

Figura 1: Podemos ver cómo los nodos se conectan entre ellos usando conexiones directas (P2P), de la misma manera que lo hacen los grupos de éstos mediante la conexion de algunos de sus elementos con algunos de otros grupos.

Lo que está claro es que la privacidad de los usuarios en la red es algo efímero, y lo que es peor, el acceso a la información puede ser fácilmente censurado debido a su almacenamiento en unos pocos servidores centralizados.

Marabunta intenta usar todos los nodos de la red como máquinas enrutadoras de paquetes ("routers") de forma que cada uno se comporta como un *cliente* y como un *servidor* al mismo tiempo.

Objetivo

El primer servicio implementado en la red es una *Lista de Mensajería Anónima*. Hay 4 categorías principales: Filosofía, Tecnología, Política, General. La idea es que cada nodo sea capaz de mandar mensajes dentro de alguna de estas 4 temáticas al resto de forma anónima. Esta tarea es desarrollada por todos los elementos de la red que se encargan de *distribuir* el mensaje a los nodos a los que están conectados de forma directa mediante conexiones P2P, eliminando la identidad del inicial que mandó el mensaje a distribuir.

Además de comunicación anónima, la plataforma creada por Marabunta usando

conexiones P2P entre los nodos proporciona una red la cual es **NO sensible a la censura** debido a que no existen servidores centrales para gestionar el intercambio de mensajes entre los nodos. De esta forma, para "tirar" la plataforma abajo tendrían que parar por completo el funcionamiento de los Servidores de Acceso a Internet (ISPs).

¿Cómo funciona?

Primero tenemos que diferenciar los 2 algoritmos principales que hacen que Marabunta se comporte como un cliente y como un servidor.

Ambos algoritmos son procedimientos que se ejecutan simultáneamente usando una estructura en hilo ("thread execution") y una zona de memoria compartida.

Como un Servidor:

- Escucha a todos los nodos que quieren acceder a la red: Petición de Conexión
- Escucha a todos los nodos conocidos (hermanos), los cuales necesitan que se les distribuyan mensajes por la red: Petición de Difusión de Mensaje
- Escucha a todos los nodos que necesitan más conectividad para asegurar su integridad en la red: Petición de nuevos Hermanos

Como un cliente:

- Busca nuevos nodos para incrementar la conectividad y la estabilidad en la red: Búsqueda de Hermanos
- Mandar mensajes propios a la red: Difusión de Mensaje
- Difusión de los mensajes en espera de difusión provenientes de otros nodos de la red: Retransmisión de Mensajes
- Envío de peticiones de búsqueda de nuevos hermanos enviadas por otros nodos: Retransmisión de Peticiones de Búsqueda de Hermanos

Principio Fundamental

Cada nodo es tanto cliente como servidor y realiza tareas de enrutado de la información.

La idea es establecer conexiones directas (estructura P2P) entre grupos de nodos para conseguir que todos los nodos en la red sean accesibles a través de diferentes rutas.

Una de las principales características de Marabunta es que la comunicación entre los nodos se realiza usando conexiones *UDP/IP*. Este tipo de comunicaciones son *no orientadas a conexión*, lo que es realmente interesante en términos de anonimato debido a que el origen de un datagrama UDP/IP no puede ser demostrado, pues no necesita confirmación al punto de conexión que envió el paquete, haciendo de Marabunta una de las plataformas de comunicación que mejor conservan el anonimato de sus usuarios.

El proceso de descubrimiento de nuevos nodos en la red es un elaborado algoritmo, el cual tiene en cuenta diferentes aspectos para asegurar el anonimato de los nodos ya presentes en la red. Por ejemplo, en las figuras 2 y 3 podemos ver cómo funciona el proceso de búsqueda y conexión de nuevos nodos. Cada nodo decide si dar a conocer su identidad a un nuevo nodo o no.

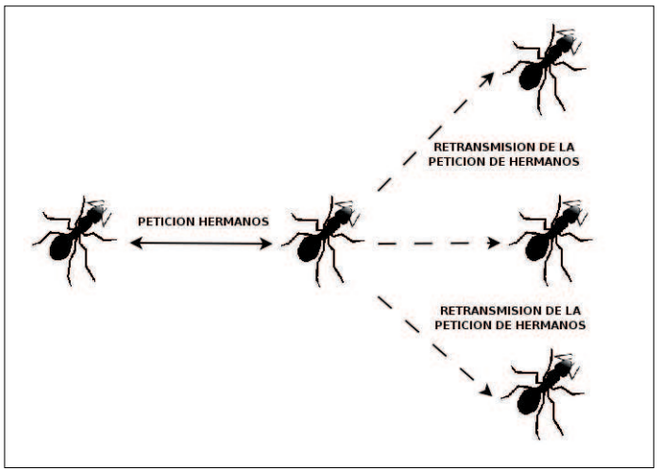

Figura 2: Cuando un nodo necesita una mayor conexión en la red manda una petición de "busca hermanos" a los nodos ya conocidos para que lo ayuden a conseguir una mayor conectividad.

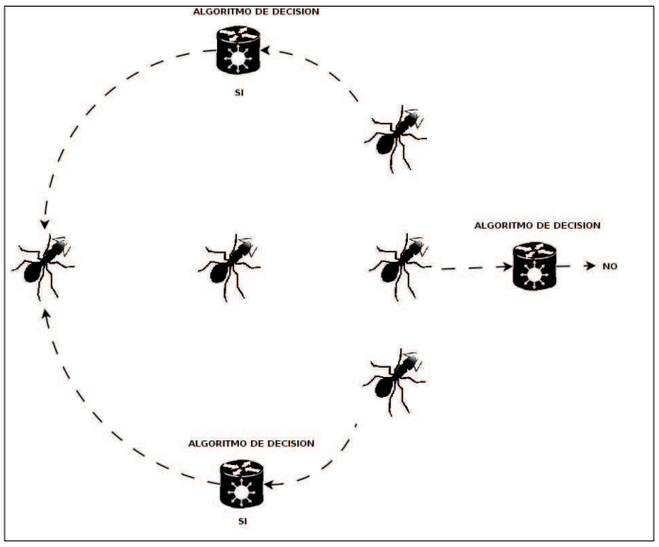

Figura 3: Cada nodo decide entonces si darse a conocer al nuevo nodo para ayudarle a tener una mayor conectividad o seguir siendo anónimo para él.

Ahora que ya sabemos qué es Marabunta y cuál es su propósito, vamos a ver cómo se maneja para poder conseguir conectarnos a la red o incluso crear nuestra propia "subred" anónima.

Primeros Pasos

Se puede descargar el código fuente y los binarios para la arquitectura i386 en [3] .

El paquete debian está en [4], aunque se puede intentar descargarlo desde alguno de los repositorios oficiales (actualmente está en proceso de promoción).

Compilación

Si decidimos compilarlo hemos de asegurarnos de tener instalado en la carpeta de "includes" del sistema los ficheros de desarrollo de las librerías QT4.

Normalmente, las distintas distribuciones GNU/Linux incluyen en el gestor de paquetes predeterminado la posibilidad de instalar el entorno necesario para la compilación : "qt4-development".

Para compilar, seguiremos los pasos habituales:

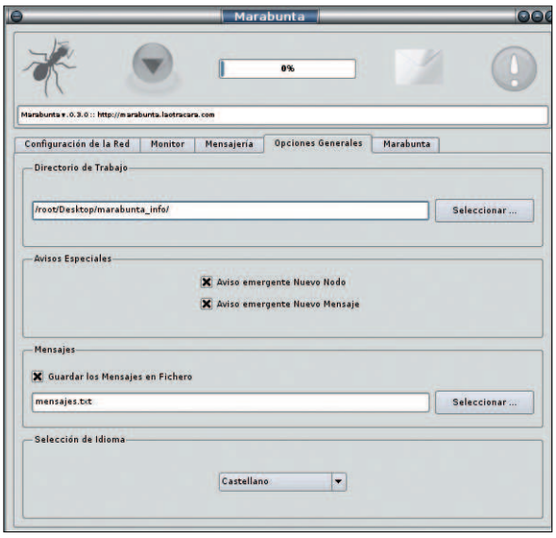

Figure 4: Panel General de Opciones: podemos configurar todas las opciones relacionadas con los directorios donde salvamos la información que genera la aplicación.

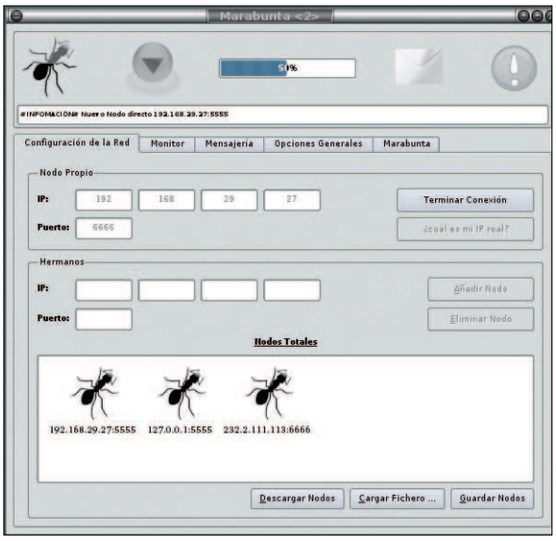

Figura 5,: Panel de Configuración de Red: todas las opciones para permitir la conexión de nuestro nodo a la red.

```
./configure
./make
```

Una vez que hayamos compilado el código o instalado los binarios o el paquete Debian, el primer paso es configurar las opciones de la aplicación.

Todas ellas pueden ser cambiadas desde el interfaz gráfico o directamente desde el fichero de configuración "marabunta.cfg", el cual se crea con las opciones por defecto la primera vez que arranca la aplicación. El directorio donde se almacena este fichero es el mismo en el que la aplicación está ejecutándose, o la ruta habitual para los ficheros de configuración si hemos instalado el paquete Debian: */etc/marabunta/marabunta.cfg*.

Veamos cómo configurar todos los parámetros desde el interfaz gráfico. En la Figura 4, vemos el panel de Opciones Generales donde podemos elegir el directorio de trabajo (*home*) para Marabunta, y donde se almacenará toda la información que genera la aplicación, como los mensajes recibidos en cada una de los canales temáticos, los fichero de log, etc.

Conexión a la Red

Lo primero de todo es que hemos de conocer nuestra IP pública y un puerto UDP disponible. Normalmente el elegir un puerto con un número alto como 6666 suele ser una buena elección para evitar problemas con los privilegios del sistema.

Si no sabemos nuestra IP podemos presionar sobre el botón de *¿Cuál es mi IP?* de forma que sea la propia aplicación la que se encargue de descubrirla por nosotros, la podremos ver en el panel de eventos situado en la parte superior de la aplicación.

Antes de continuar tenemos que ver si el sistema está usando también la IP pública o una privada. Mediante el comando *ifconfig* obtenemos la IP usada por el interfaz de red. Si esta IP es la pública, coincidirá con la obtenida por la aplicación y no es necesario que hagamos ningún paso extra. Sin embargo si no coinciden, significará que estamos usando una IP privada, por lo que nos estamos conectando a través de un NAT (Network Address Translator). En caso de que esto ocurra es bastante probable que la IP devuelta por *ifconfig* sea de la forma *192.168.xxx.xxx*, aunque no siempre es así.

Lo que tenemos que hacer para poder conectarnos a la red es crear una ruta de reenvío de paquetes UDP desde el router hasta nuestra máquina ("Port Forwarding"). Para ello hemos de acceder al panel de Administración de nuestro router, normalmente es posible hacerlo vía web. Por ejemplo, si nuestra IP privada es de la forma *192.168.0.xxx* es probable que el gateway tenga la dirección *192.168.0.254* o *192.168.0.1*, de todas formas podemos asegurarnos mediante la información mostrada en el campo *gateway* por el comando *route*.

Probamos entonces a acceder vía web a la IP del *gateway*, si no podemos tal vez sea porque haya que acceder a un puerto distinto al estándar de conexiones HTTP (80), o configurarlo mediante conexión vía serie, por lo que lo mejor es buscar información del modelo de router que tenemos en la red.

Una vez llegamos al panel de Administración tenemos que establecer una nueva ruta *PORT FORWARDING*, con la finalidad de que todos los paquetes que vayan dirigidos a un puerto UDP concreto de la IP pública que está asignada a nuestro router puedan sean redirigidos al mismo puerto de nuestra IP privada.

Por ejemplo, si tenemos la IP privada 192.168.0.10 y queremos usar el puerto 6666 para las conexiones entrantes a Marabunta tenemos que crear una regla que diga: todos los paquetes con destino el puerto 6666/UDP tienen que ser redirigidos al puerto 6666/UDP de la IP privada: 192.168.0.10

Una vez tengamos este problema resuelto podemos pasar a la configuración propia de la aplicación. Ponemos nuestra IP pública y puerto en las cajas de texto del panel de Conexión, como se muestra en la Figura 5.

Ahora ya podemos conectar a la red. En este momento estamos escuchando a todas las conexiones entrantes, y podríamos empezar a

Figura 6: Panel de Mensajes: Toda la información relacionada con la temática y filtrado de los mensajes.

comunicarnos en el momento que se conectara algún nodo a nosotros, sin embargo si queremos además conectar a la red ya existente tenemos varias alternativas:

- Si sabemos la IP y puerto de un nodo que ya forma parte de la red, simplemente lo añadimos en los campos de información sobre hermanos ya conocidos
- Si tenemos un fichero con información (pares: IP-puerto) sobre nodos que pudieran estar en la red, salvados por ejemplo de una anterior sesión de Marabunta, podemos cargarlo mediante el botón Cargar Fichero.
- Podemos presionar sobre el botón Descargar Nodos para conseguir información sobre algunos nodos con alta probabilidad de estar conectados a la red

Una vez la aplicación tiene una lista de "nodos posibles" comenzará a intentar conectar con ellos. Podemos ver en todo momento la calidad de conexión que tenemos en la red mirando la barra de porcentaje de conexión en la cabecera de la aplicación.

Mandando y Filtrando Mensajes

Como ya hemos visto anteriormente, el primer servicio de Marabunta es ser una plataforma que dé servicio a una Lista de Mensajes, la cual puede ser configurarada desde el panel de "Mensajería", ver Figura 6.

- La temática de los mensajes, nos permite seleccionar qué tipo de mensajes nos interesa recibir
- El filtrado de contenido nos permite elegir dentro de las categorías seleccionadas los mensajes que contengan un determinado patrón.

Supongamos por ejemplo que queremos visualizar los mensajes sobre la manifestación de hubo en nuestra ciudad ayer en contra de la política de algunos gobiernos por el uso de programas como *Echelon* o *Carnivore*, (los cuales, por cierto, atentan directamente contra la privacidad personal). Seleccionaríamos el canal de Política con la palabra clave *manifestación*.

Es importante tener claro que este "filtrado" de la información está relacionado únicamente con la información mostrada al usuario, no con la información que trata la aplicación, pues recordemos que continuamente está enrutando mensajes de otros nodos para que se incluyan entre los elegidos para ser visualizados o no.

Una buena forma de uso suele ser seleccionar la política de filtrado que queramos y dejar la aplicación funcionando durante horas, seleccionando además la opción de almacenar los mensajes en un fichero de forma que al final tengamos todos los mensajes en un fichero de texto y podamos leerlos tranquilamente. Para especificar el nombre del fichero donde volcar los mensajes podemos ir al panel de Configuración (Figura 4).

Creando nuestra propia Red Anónima

En la Figura 5 vimos que una de las posibilidades para conseguir conectar con la red es descargar una lista de nodos que han accedido a la red en un intervalo de tiempo no muy grande, por lo que hay altas probabilidades de que sigan dentro.

Lo que ejecuta esta acción es una consulta a un script escrito en PHP, el cual ejecuta un algoritmo de anonimato y privacidad sobre la información disponible y genera la lista más acertada de nodos disponibles.

Se puede descargar el script desde [5]. Buscamos el fichero con extensión *.php.txt*, lo cambiamos a *.php* y le damos permiso de ejecución una vez lo hayamos colocado en nuestro servidor. El otro fichero *marabunta.nodes* es donde se almacena la información relacionada con los nodos a los que va accediendo al script, por lo que ha de tener permiso de escritura en el servidor.

Ahora hemos de cambiar el fichero de acceso al script PHP (*nodos.txt*), el cual se encuentra en el mismo directorio que el fichero de configuración de la aplicación. Para conseguir que la aplicación trabaje sólo con vuestro servidor, será preciso borrar las URLs existentes e insertar la que habéis generado al subir el script PHP con el fichero para almacenar la información a vuestro servidor. ■

EL AUTOR

David Gascón es estudiante de Ingeniería Informática en el Centro Politécnico Superior de la Universidad de Zaragoza.

Actualmente se encuentra desarrollando el proyecto final, el cual versa sobre los sistemas de Anonimato en Redes Wireless Ad-hoc. Contacta con él en david@laotracara.com o visita su web en http://www.laotracara.com.

RECURSOS

[1] Marabunta *http://marabunta.laotracara.com*

[2] Apeiron (Redes Libres) *http://apeiron.laotracara.com*

[3] Descargas de Marabunta: *http://marabunta.laotracara.com/descargas/*

[4] Paquete Debian: *http://marabunta.laotracara.com/descargas/debian/*

[5] Script para la generación de listas de nodos: *http://marabunta.laotracara.com/descargas/phpNodos/*