Creación de una red superpuesta para el despliegue de servicios de colaboración

D. Prieto, E. Barra, S. Pavón, C. Barcenilla, J. Mejía

Resumen— Las aplicaciones de trabajo colaborativo hacen uso se múltiples flujos de datos, viéndose dificultado su funcionamiento por la presencia de cortafuegos y equipos que traducen las direcciones de red. En este artículo se describe una solución a estos problemas consistente en la creación de una red superpuesta a través de la cual se encapsula todo el tráfico del servicio de colaboración. Se detallan el funcionamiento multipuerto y otras características de la red superpuesta que la hacen especialmente interesante para cualquier aplicación que tenga problemas con dispositivos de red intermedios. Finalmente, se exponen los resultados de aplicar esta solución a un programa de multiconferencia (Isabel) y se explica cómo extender la solución a otras aplicaciones.

Palabras clave— Red superpuesta (*overlay network*), aplicaciones para trabajo colaborativo (*computer supported cooperative work*), red privada virtual (virtual private network), túnel (*tunnel*), OpenVPN, cortafuegos (*firewall*), traducción de direcciones (*network address translation*), Isabel.

I. Introducción

ESTE artículo describe los problemas de conectividad que típicamente afectan a las aplicaciones y servicios de trabajo colaborativo, y plantea una solución a ellos mediante el encapsulamiento del tráfico en una red de túneles.

Bajo el término de aplicaciones para el trabajo colaborativo (*computer supported cooperative work, CSCW* [1]) se incluye un amplio abanico de programas que facilitan la realización de trabajos en grupo. Los desarrolladores de herramientas de trabajo colaborativo estudian el trabajo en grupo, sus efectos sociales y psicológicos, y consideran cómo aplicar las nuevas tecnologías en estos campos. Ejemplos de este tipo de herramientas son las aplicaciones de videoconferencia y las de edición conjunta de documentos.

Desde el punto de vista de la red, los servicios de colaboración tienen una serie de características comunes. Suelen ser aplicaciones distribuidas, al permitir el trabajo en grupo de personas separadas físicamente, y suelen contar con servidores en Internet fácilmente accesibles. Unas veces son programas embebidos en el navegador, como es el caso de Google Docs [2] o Marte 3.0 [3], y otras veces son aplicaciones independientes que se ejecutan directamente sobre el sistema operativo, como la mayoría de las aplicaciones de videoconferencia. En cuanto a la estructura de red [4], suelen seguir el modelo cliente-servidor, que en ocasiones se vuelve más compleja con la presencia de servidores de flujos (*flowservers*), cuando manejan volúmenes importantes de tráfico. Estas estructuras de red más complejas son típicamente estructuras en forma de árbol, lo que permite evitar bucles, y distribuir la carga en distintos puntos de la red.

Este tipo de aplicaciones se suelen desplegar en entornos de red bastante restringidos, que se caracterizan por tener estrictas políticas de gestión de red. Aunque las características de distintos entornos pueden ser bastante heterogéneas, hay una serie de dificultades que se repiten frecuentemente en todos ellos, como son la presencia de NATs (*Network Address Translation*) y cortafuegos (*firewall*). Un *firewall* generalmente limita las conexiones entrantes a la red de la organización y, en muchos casos, también limita las salientes. Esto se debe a las políticas que los administradores de red deben aplicar para mantener la seguridad en las redes que controlan, evitando conexiones no permitidas desde el exterior y un uso no contemplado de la red desde el interior. Además, los responsables de red son reacios a permitir nuevos tipos de tráfico y los procedimientos para conseguirlo son muy lentos, especialmente cuando la organización es grande. Las características de estos entornos hacen recomendable que las aplicaciones de trabajo colaborativo tengan el origen de sus conexiones en el cliente y utilicen un número de puertos lo más reducido posible.

Además de los *firewalls*, y muchas veces combinados con estos, se encuentran los NATs. Este mecanismo, consistente en alterar las direcciones origen de los paquetes IP, se ha extendido mucho debido a la escasez de dichas direcciones IP y supone un grave problema para el desarrollo de aplicaciones distribuidas. En primer lugar, porque impide el acceso desde el exterior a una máquina que se encuentra detrás de un NAT, y, en segundo lugar, porque afecta a protocolos de aplicación que utilizan la dirección IP. La presencia de este mecanismo refuerza la necesidad de que las conexiones tengan su origen en el cliente.

Estos dispositivos han sido el origen de distintas soluciones,

D.Prieto, E. Barra, C. Barcenilla y J. Mejía realizan su tesis doctoral dentro del grupo de Internet de Nueva Generación del Departamento de Ingeniería de Sistemas Telemáticos de la Universidad Politécnica de Madrid (Av. Complutense, s/n, 28040 Madrid) (correos e.: dprieto@dit.upm.es; ebarra@dit.upm.es; barcenilla@dit.upm.es; mejia@dit.upm.es).

S. Pavón es Profesor Titular del Departamento de Ingeniería de Sistemas Telemáticos de la Universidad Politécnica de Madrid (correo e.: spavon@dit.upm.es).

generalmente muy complejas. Los desarrolladores de programas de voz sobre IP y de juegos online han desarrollado distintas soluciones propietarias para conseguir que sus aplicaciones funcionen detrás de NATs y *firewalls*. También se han intentado desarrollar protocolos abiertos que permitan a las aplicaciones convivir con estos mecanismos, como pueden ser STUN (*Session Traversal Utilities for NAT [5]*), su extensión TURN (*Traversal Using Relays around NAT* [6]) o sus usos ICE (*Interactive Connectivity Establishment* [7]) y SIP Outbound [8]. Sin embargo, la falta de estandarización de los NATs ([9] y [10]) ha causado que distintas implementaciones de estos dispositivos tengan comportamientos completamente distintos en cuanto a la relación entre direcciones internas y externas del NAT, o la forma de redireccionar el tráfico que llega del exterior del NAT hacia la parte interna de éste. Debido a estas razones, este tipo de soluciones son demasiado complejas, puesto que tienen que averiguar en un primer momento el comportamiento de los equipos intermedios y adaptarse posteriormente a dicho comportamiento.

Este artículo presenta una solución a estos problemas de configuración de redes basada en la creación de una red superpuesta. Esta propuesta es mucho más simple conceptualmente que las descritas previamente y parte de la característica común de todos estos dispositivos intermedios: es más sencillo iniciar las conexiones en sentido saliente hacia Internet y, una vez enviado tráfico desde el interior, mantener una relación entre la dirección interna y la dirección externa. Explotando esta característica, se propone encapsular todo el tráfico de la aplicación dentro de una red superpuesta que garantice la comunicación entre las distintas estaciones de la aplicación.

El concepto de red superpuesta (*overlay network* [11]) engloba a todas las redes construidas sobre otras redes. El principal aspecto de estas redes es que permiten acercar dos puntos que a nivel IP estén muy distanciados y abstraer a los niveles superiores de la aplicación de las distintas dificultades que aparecen a nivel de red. Existen diferentes tipos de redes superpuestas, entre los que se ha escogido una red superpuesta IP sobre IP soportada por un software existente ampliamente probado.

Aunque el estudio se ha realizado orientado a servicios de colaboración, como respuesta a las necesidades de conectividad que éstos demandaban, el resultado es aplicable a cualquier aplicación distribuida, que, al igual que las aplicaciones para el trabajo colaborativo, encuentren dificultades debido a la presencia de NATs y *firewalls*. La red superpuesta desarrollada es un potente recubrimiento, fácilmente adaptable y útil para una gran variedad de software.

II. Creación de una red superpuesta mediante OpenVPN

La solución propuesta consiste en montar una red superpuesta como una red de túneles entre todos los equipos de la aplicación. Por esta red superpuesta se envía todo el tráfico generado por la aplicación. Una vez establecida la red de túneles, los protocolos superiores de la aplicación (protocolos de aplicación propiamente dichos) se pueden despreocupar completamente de la existencia de equipos intermedios que puedan dificultar la conectividad dado que para ellos es como si estuviesen conectados directamente unos a otros. Una vez establecida la red superpuesta, el nivel de aplicación puede funcionar como si todas las máquinas estuvieran en una misma red local, sin ningún tipo de restricción para comunicarse entre sí.

Por otro lado, el establecimiento de la conexión se iniciará desde la parte más delicada de la estructura de red, consiguiendo que la probabilidad de establecimiento de la red superpuesta sea lo más elevada posible. En un caso típico y simple de aplicación distribuida, el servidor se encuentra en la Internet con una dirección IP pública, y los posibles *firewalls* que limiten el acceso a éste no afectan al funcionamiento de la aplicación. En cambio, los clientes pueden encontrarse en un entorno mucho más hostil para la aplicación, detrás del *firewalls* de una corporación con grandes restricciones para salvaguardar la seguridad de los activos internos, o en una red doméstica que sólo goza de una dirección pública y traduce las direcciones privadas del interior a ésta. En este caso, se aprecia sencillamente que los clientes son el punto débil a la hora de establecer la conexión y que será mucho más sencillo iniciar cualquier nueva conexión desde el cliente que desde el servidor. En la figura 1 se muestra un caso típico de situación de red que se encuentra una aplicación cliente-servidor. Es sencillo ampliar esta problemática a una estructura de red más complicada, en la que se incluyan *flowservers*, unidades de control multipunto (*Multipoint Control Units*, MCU) o cachés, y seguir manteniendo que los clientes extremos serán los que tendrán las mayores dificultades de red.

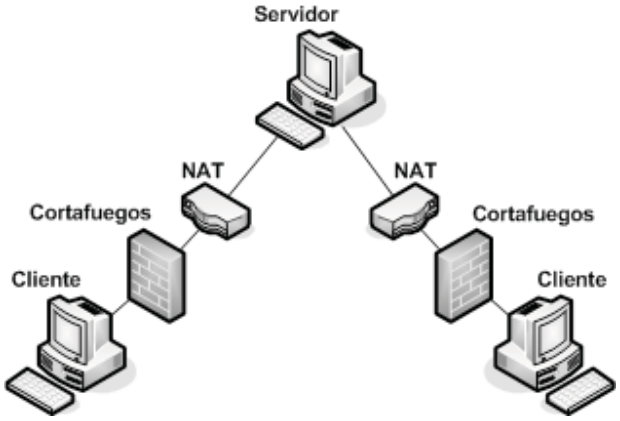

Fig. 1. Arquitectura de red típica en aplicaciones cliente-servidor.

La implementación se ha llevado a cabo utilizando OpenVPN [12], un software bastante extendido, que destaca por su estabilidad y configurabilidad. Además, está portado a la gran mayoría de sistemas operativos existentes (Linux, Windows 2000/XP/Vista, OpenBSD, FreeBSD, NetBSD, Mac OS X y Solaris), su presencia no reduce la velocidad de transferencia y consume muy pocos recursos. Estas características (una descripción más detallada de estas en [13])

convierten a OpenVPN en un software muy adecuado para la red superpuesta. Por otro lado, se ha desarrollado un pequeño recubrimiento para configurar el comportamiento de OpenVPN. Denominaremos al conjunto "módulo de gestión de red".

OpenVPN se ejecuta como servidor en todos los equipos. Cuando un equipo A se conecta a otro equipo B, se establece un túnel entre el cliente OpenVPN de A y el servidor OpenVPN de B. De esta manera, se recubre mediante túneles todo el árbol de distribución de la aplicación. Para que todo el tráfico de la aplicación sea encaminado a través de los túneles es necesario que el módulo de gestión de red sea el primero que establezca una conexión. En la figura 2 se muestra como un equipo (Equipo B) se conecta a través del cliente OpenVPN al nodo superior de la estructura (Equipo A), mientras que recibe cualquier número de conexiones de nodos clientes a través del servidor OpenVPN (Equipo C y Equipo D).

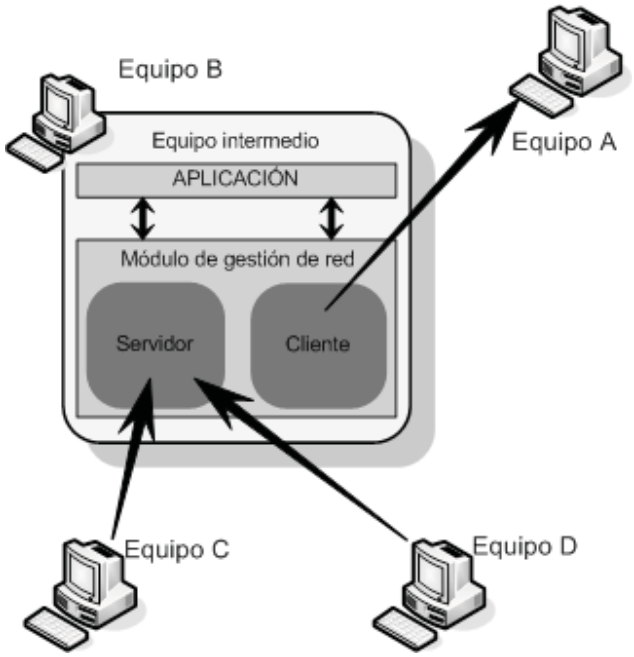

Fig. 2. Módulo de gestión de red recubriendo una estructura de árbol

Aunque la implementación inicial del módulo de gestión de red permitía el recubrimiento de una estructura de árbol, se ha ampliado su funcionalidad permitiendo configuraciones de red más complejas. Para ello se permite que en una misma máquina existan varios clientes simultáneamente. De esta forma, se puede recubrir cualquier estructura de red y un equipo de ésta puede recibir cualquier número de conexiones y, además, poder conectarse a cualquier número de equipos. En la figura 3 se puede apreciar como el módulo de gestión de red del equipo B, soporta conexiones de dos árboles de red distintos. Añadiendo al módulo de gestión de red un servicio de gestión de túneles como el descrito en su aplicación a Isabel (ver apartado IV), se pueden introducir a través de éste cualquier número de servicios diferentes, independientemente de las arquitecturas de red seguidas por éstos.

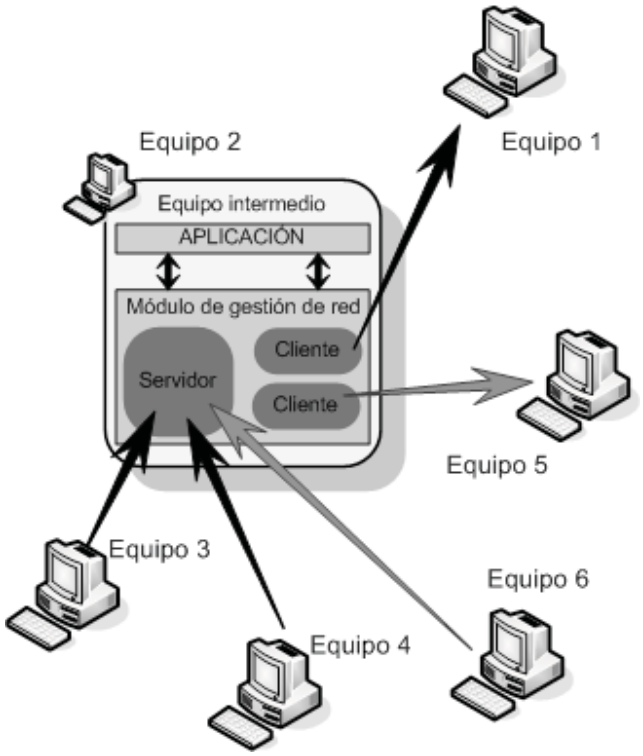

Fig. 3. Módulo de gestión de red recubriendo una estructura de red compleja

La red de túneles OpenVPN ofrece al nivel superior un conjunto de interfaces de red virtuales (una interfaz del servidor OpenVPN y una nueva interfaz por cada cliente OpenVPN), cada una con su respectiva dirección IP. Un problema que solventa el módulo de gestión de red es el direccionamiento, que debe utilizar rangos privados. La forma de configurar las direcciones IP en OpenVPN es suministrar un rango de direcciones al servidor OpenVPN. Éste escoge para sí mismo la primera dirección del rango y utiliza el resto para los clientes. El módulo de gestión de red garantiza que los rangos de direcciones utilizados no se solapen. Los rangos de direcciones IP tampoco deben solaparse con los ya existentes en la máquina en otras interfaces. El módulo de gestión también evita que las interfaces de red creadas por OpenVPN sean utilizadas para otros propósitos mediante reglas de *iptables* [14], restringiendo su uso a los puertos que utiliza el nivel de aplicación.

Una última funcionalidad del módulo de gestión de red es un funcionamiento multipuerto, que permite intentar establecer la red superpuesta por aquellos puertos que más probablemente estén abiertos en las redes de las organizaciones. Para ello, el servidor de OpenVPN escucha simultáneamente en varios puertos. De esta forma, cuando el cliente intenta establecer una conexión, prueba en todo un rango de puertos hasta lograrlo. En el lado del servidor se utiliza nuevamente *iptables* para redirigir, al puerto en el que realmente escucha el servidor OpenVPN, el resto de puertos que se desean añadir. El módulo de gestión de red comprueba que los puertos indicados no estén siendo utilizados por otra aplicación en el servidor antes de redirigirlos. Por su parte, el cliente intentará establecer la conexión en todos los puertos de

forma cuasi-simultánea, aumentando las posibilidades de éxito sin aumentar el tiempo de establecimiento de la conexión. En la figura 4 se puede ver de forma gráfica el procedimiento seguido para establecer la conexión utilizando un número reducido de puertos. Se aprecia que de los puertos empleados en el intento de conexión, el puerto 53018 y el puerto 1720 no logran atravesar los equipos intermedios, mientras que a través de los puertos 53 y 5060 se logra llegar al servidor. Los puertos utilizados en el ejemplo son puertos UDP y corresponden al puerto utilizado oficialmente por Isabel (53018), y a puertos utilizados por protocolos que tienen una utilización elevada: DNS, H.323 y SIP.

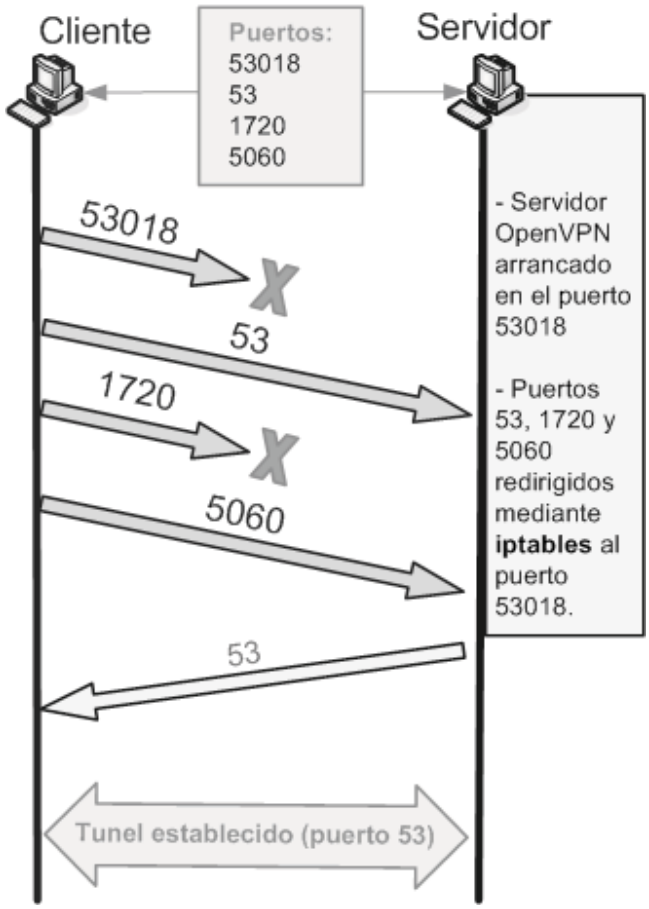

Fig. 4. Ejemplo de establecimiento de conexión mediante un funcionamiento multipuerto

OpenVPN funciona tanto en UDP como en TCP, por lo que se pueden utilizar ambos protocolos. Si se desean utilizar simultáneamente se deben ejecutar dos instancias del servidor OpenVPN, una para cada protocolo. El puerto 80 de TCP es el puerto utilizado por HTTP, en el que escuchan la mayoría de los servidores web del mundo, y por ello suele estar abierto en casi todas las redes en sentido saliente hacia Internet, para permitir a los usuarios navegar por páginas web. Otro detalle que se debe tener en cuenta a la hora de seleccionar un protocolo de transporte para la red superpuesta son las características del tráfico de la aplicación. Por ejemplo, TCP no es adecuado para el tráfico de tiempo real, como el audio o el vídeo de una videoconferencia, puesto que incluye retransmisiones en caso de pérdidas que aumentan el retardo de la comunicación. Esta es la razón por la que, en principio, para aplicaciones con requisitos importantes de retardo y variación de retardo se recomienda utilizar UDP para la red superpuesta, al igual que se utiliza UDP como protocolo de transporte en las mismas condiciones.

OpenVPN es utilizado habitualmente para establecer redes privadas virtuales [15]. Debido a ello, incluye múltiples configuraciones posibles en lo que respecta a autenticación y cifrado. En principio, el módulo de red abstrae esa complejidad, adoptando una configuración sencilla en ese sentido. Es posible modificar este comportamiento para dotar de seguridad a la aplicación aprovechando el potencial de OpenVPN, estableciendo por ejemplo un sistema de autenticación basado en LDAP o en usuario y contraseña, y un cifrado AES o BlowFish. OpenVPN también incluye posibilidades de compresión que a su vez pueden resultar interesantes en el encapsulamiento del tráfico de la aplicación. Documentación sobre cómo configurar OpenVPN está disponible en [16].

Otra característica interesante de OpenVPN es la estabilidad que ofrece [17]. Se puede ajustar el funcionamiento de los túneles para que no se cierren bajo ninguna circunstancia. Incluye mecanismos de envío de tráfico periódico, para evitar que los equipos intermedios den por terminada la conexión, y mecanismos que facilitan la rápida recuperación en caso de fallos en la red.

Al explicar el módulo de gestión de red se ha hablado de *iptables*, que es un componente ligado al núcleo de Linux. La implementación del módulo de gestión de red se ha realizado en este sistema operativo, aunque siempre pesando en una posible portabilidad a otros sistemas operativos. Tanto OpenVPN como Java (lenguaje en el que se ha programado el recubrimiento de OpenVPN que conforma junto con éste el módulo de gestión de red) son multiplataforma, pero la gestión de red está siempre muy ligada al sistema operativo.

III. Rendimiento del sistema

A continuación se exponen los resultados más interesantes de las pruebas realizadas al sistema y en las que se comprueba la perfecta validez de la solución propuesta. Se destacan dos aspectos: coste de CPU y sobrecarga de red.

Se han realizado pruebas de rendimiento para estimar el coste de CPU que supone el uso del módulo de gestión de red. Las máquinas utilizadas han sido dos Intel Celeron a 2,66 Ghz conectados a través de una red de área local. Para las prueba se ha utilizado el software *iperf* [18]. En una primera prueba se ha comprobado que el coste de CPU más significativo es introducido por OpenVPN y está asociado al volumen de tráfico cursado y al tipo de cifrado empleado. Se ha comprobado que con tasas de transferencia de hasta 30 Mbps y con cualquier tipo de cifrado el consumo de CPU es mínimo, nunca superior al 1%.

En cuanto a la sobrecarga de red, se ha comprobado que la red superpuesta introduce una sobrecarga en torno al 6%. Esta

sobrecarga se produce principalmente por la doble presencia de cabeceras a nivel de red (IP) y de transporte (UDP, TCP) que implica el concepto de túnel.

IV. Aplicación al caso de Isabel

Un aspecto interesante de la red de túneles es que se puede incluir de forma sencilla en una aplicación ya existente que precise alguna de las funcionalidades que la red de túneles provee. Esta aplicación debe lanzar el módulo de gestión de red en primer lugar y, posteriormente, utilizar las direcciones privadas que proporciona éste en vez de las direcciones públicas que utilizaría normalmente. El proceso de adaptación es rápido y sencillo, como ha sido el caso de su inclusión en Isabel [19].

Isabel es una herramienta de videoconferencia avanzada para PC, desarrollada por completo dentro del grupo Internet de Nueva Generación del Departamento de Ingeniería Telemática de la ETSIT[20]. Con esta herramienta se realizan multitud de clases, reuniones y congresos, con clientes muy variados desde universidades a empresas en distintos países. Tales son el Internet NG[21], el Telecom I+D [22] o una gran variedad de cursos[23]. Cada uno de los clientes que se conecta tiene una configuración de red distinta y la organización de un evento de tamaño medio a grande es complicada, dado que hay que ir avisando a las distintas sedes que participarán de que deben tener abierto un rango de puertos, en modo bidireccional y tanto en TCP como en UDP. Estos detalles a veces se olvidan y se abren los puertos tan sólo para salida de tráfico o tan sólo para TCP.

Isabel utiliza los puertos 53020 a 53032 de UDP para los flujos multimedia y 53009 a 53023 de TCP y 53009 a 53017 de UDP para flujos de control. Estos son los puertos que hay que pedir a las organizaciones que abran en sus *firewalls*. En cambio, aplicando esta solución que aquí explicamos con un único puerto UDP es suficiente, dado que sobre el túnel creado se encapsula todo el tráfico. Isabel, sin embargo, no cambia y sigue utilizando ese amplio rango de puertos sobre las interfaces virtuales, y es el módulo de gestión de red el que encapsula dentro de la red superpuesta de túneles OpenVPN todo el tráfico. El amplio rango de puertos utilizados ha sido una fuente continua de problemas a la hora de desplegar Isabel, enfrentando a los usuarios con sus administradores de red y, en muchos casos, empeorando la imagen de la aplicación al no funcionar algún componente porque no se podía utilizar el rango completo.

Por todo esto se integró el sistema de túneles con Isabel. Con la intención de no modificar el código de Isabel, o hacer modificaciones mínimas, y hacer este sistema de túneles lo más genérico, se diseñó un recubrimiento del sistema que llamamos “servidor de túneles”. Este sistema es autónomo. Es un simple servidor de XML-RPC que escucha en un puerto local de la máquina peticiones de cualquier aplicación para crear un túnel, que en nuestro caso lo usa tan sólo Isabel, pero en un caso genérico lo podrían usar varias aplicaciones. Cada aplicación puede solicitar al servidor los túneles que necesite y el servidor de túneles manejará las peticiones sin que interfieran unas con otras.

Si el túnel que se solicita al servidor ya está establecido, bien porque esté así predefinido o porque otra aplicación o servicio lo haya solicitado anteriormente y aún lo esté usando, el servidor no establece un nuevo túnel al sitio, sino que devuelve el ya existente. Si se le pide que cierre un túnel que está en uso por otra aplicación ignorará esta petición. De tal modo que lleva un control total de los túneles que se abren y cierran, haciendo un uso eficiente de éstos para no cargar de manera innecesaria la máquina.

El servidor de túneles está integrado y se distribuye con la aplicación Isabel, aunque como se ha comentado puede distribuirse de un modo autónomo. Las aplicaciones que deseen usar el servidor de túneles simplemente tienen que implementar la llamada XML-RPC a dicho servidor para crear el túnel y para destruirlo cuando ya no lo usen.

Isabel, con el servidor de túneles integrado, se ha utilizado en la realización de grandes eventos de hasta 40/50 sedes participantes, comprobando así que esta solución es viable: la red superpuesta se establece y se usa con normalidad, desaparece cuando se deja de usar y evita a los organizadores de los eventos tener que configurar o pedir que configuren los *firewall* de cada sede participante, haciendo que como máximo tengan que solicitar que se abra un único puerto UDP. Además, se comprueba que el establecimiento de esta red no interfiere con el uso de otras aplicaciones, que siguen funcionando con normalidad.

V. Conclusiones

El módulo de gestión de red y el servidor de túneles aquí presentados permiten la creación de una red superpuesta que facilita en gran medida el despliegue de cualquier servicio distribuido, en especial de servicios de colaboración. Presenta muchas ventajas, como el cifrado del tráfico dentro de los túneles para proporcionar seguridad o la búsqueda de los puertos conocidos (*Well Known Ports*) que son más probables que estén abiertos en el *firewall*.

La integración de este trabajo en cualquier aplicación es relativamente sencilla, ya que gracias al servidor de túneles tan sólo hay que implementar en la aplicación las llamadas XML-RPC que establecen y cierran el túnel al comenzar y terminar de usarlo. Como ejemplo de uso hemos expuesto el caso de Isabel, con la que se han realizado grandes congresos de hasta 40/50 participantes con esta red superpuesta, que ha funcionado a la perfección y ha permitido simplificar la gestión por parte de la organización.

En cuanto a futuros trabajos, será interesante la migración del módulo de gestión de red a otros sistemas operativos, especialmente a Windows por ser el sistema más extendido. En este caso se podría utilizar WIPFW [24] en lugar de *iptables*. Esta migración no será muy complicada y permitirá ofrecer el módulo de gestión de red a un gran número de nuevas aplicaciones. Como se ha explicado anteriormente, la mayor dificultad de esta migración es la dependencia del sistema operativo que tiene la gestión de red.