Principios y Aplicaciones de las Redes Activas

[†]María Calderón Pastor
E-mail: mcalderon@fi.upm.es

[★]Marifeli Sedano Ruiz
E-mail: marifeli@aut.alcala.es

[†]Santiago Eibe García
E-mail: seibe@fi.upm.es

[†]Facultad de Informática. Universidad Politécnica de Madrid
Campus de Montegancedo, 28660 Boadilla del Monte (Madrid)

[★]Escuela Politécnica. Universidad de Alcalá de Henares
28871 Alcalá de Henares (Madrid)

Abstract

This paper presents an overview of a new network technology: active networks. The active networks goal is to produce a new networking platform, flexible and extensible at runtime to accommodate the rapid evolution and deployment of networking technologies and also to provide the increasingly sophisticated services demanded by users. A snapshot of the architecture being developed in DARPA active networks program is presented. Finally, potential applications of active networks are highlighted, along with some of the challenges that must be overcome to make them a reality.

1. Introducción

En las redes de paquetes actuales, los nodos intermedios (encaminadores o conmutadores) son sistemas cerrados e integrados verticalmente. El desarrollo de nuevos protocolos de red depende en primer lugar de comités de estandarización. Es necesario un largo proceso hasta que se logra una solución satisfactoria que sea consensuada por todos los miembros. Asimismo, el despliegue de nuevos protocolos también está sujeto a un largo proceso. Los fabricantes, de encaminadores o conmutadores, integran los nuevos protocolos únicamente en sus nuevos equipos. Consecuentemente aparecen problemas de incompatibilidad con los equipos antiguos que no soportan el nuevo protocolo. De aquí, que en muchos casos, los usuarios realmente no perciben los beneficios reales del nuevo protocolo hasta que pasado el tiempo, no son sustituidos los equipos antiguos.

Las tecnologías de redes de datos han evolucionado considerablemente a lo largo de las tres últimas décadas. Sin embargo estos avances no han modificado sustancialmente la funcionalidad básica consistente en el transporte transparente de bloques de datos de usuario entre dos puntos de terminación de red. En los modelos de red tradicionales los datos de los usuarios son transmitidos de forma opaca, la red es insensible a los datos que transporta y éstos son encaminados de forma transparente y sin modificación. El proceso que se realiza dentro de la red es muy limitado: procesamiento de la cabecera en una red de conmutación de paquetes con un procesamiento adicional para la señalización, en el caso de que la red proporcione servicio orientado a conexión. Este procesamiento de las cabeceras del paquete y de la información de señalización es independiente de la aplicación o proceso de usuario que genera el paquete. Por otro lado, debido a que el servicio es único para todas las aplicaciones (con un estrecho margen de opciones y facilidades) es preciso realizar estudios detallados de caracterización de aplicaciones, y alcanzar compromisos de diseño para satisfacer la parte común de los requisitos del máximo número de aplicaciones.

Las Redes Activas [1] ofrecen un paradigma distinto permitiendo la programación de los nodos intermedios de la red, lo cual supone un gran salto conceptual en la evolución de las tecnologías de red. Las redes activas constituyen una arquitectura de red novedosa en la que los nodos de la misma pueden realizar procesamiento "a medida" sobre los paquetes que los atraviesan. Las redes activas producen un cambio en el paradigma de red: de llevar bits de forma pasiva, al de una máquina computacional más general. Otro gran salto conceptual se produce debido a que se propone que el comportamiento de los nodos pueda ser programado *en línea*, bien por el administrador de la red o por los propios usuarios en determinadas circunstancias, o lo que es lo mismo permite la modificación dinámica del comportamiento de la red.

La programación de la red puede resultar útil a distintos niveles, siendo sus principales potencialidades las siguientes:

- Acelerar la evolución de las redes, al permitir que nuevos protocolos y servicios sean introducidos en la red sin la necesidad de largos procesos, de estandarización y despliegue en la red. Desde el punto de vista de los proveedores de servicio de red, las redes activas permiten reducir el tiempo necesario para desarrollar y desplegar nuevos servicios de red.
- Permitir a los usuarios crear nuevos servicios o adaptar los ya existentes de acuerdo a las necesidades concretas de sus aplicaciones. Asimismo es posible imaginar usuarios que adapten los servicios, en base a opciones proporcionadas por "terceras partes" que vendan servicios de red mejorados.

- Facilitar la experimentación de nuevos protocolos o nuevos servicios de red. Desde el punto de vista de los investigadores, las redes activas ofrecen una plataforma en la cual experimentar con nuevos servicios de red de forma real, mientras que ésta continúa operando normalmente.

Un aspecto importante a reseñar es, que la utilidad de las redes activas está relacionada con la velocidad de la red [2]. En redes con poco ancho de banda, el procesamiento de todos los paquetes es factible. Pero cuando el ancho de banda aumenta, la capacidad de computación disponible por byte decrementa, hasta tal punto, que el único procesamiento que se puede realizar por paquete es el relacionado con la función en encaminamiento. De aquí que sólo una pequeña fracción de los paquetes pueda ser procesados, y es necesario que en cada paquete se indique si se necesita un procesamiento *rápido* (el asociado al encaminamiento) o *lento*. La mayoría de los paquetes sólo necesitarán un procesado rápido (denominado usualmente *Fast-path processing* en terminología inglesa).

Con esta idea en mente, los investigadores en redes activas se han planteado como requisito de diseño, la construcción de un nodo activo que además de realizar procesamiento a medida sobre algunos paquetes que lo atraviesan, sean capaces de encaminar datagramas IP tradicionales, a una velocidad comparable a la conseguida por los encaminadores IP "pasivos" existentes en la actualidad.

Este artículo se estructura de la siguiente forma. En la siguiente sección se exponen una serie de aspectos que se han identificado como claves para el desarrollo de una plataforma de red activa, y que marcan las diferencias esenciales entre las distintas propuestas planteadas hoy en día. En la sección tercera se presentan las ideas básicas de la arquitectura de redes activas que está siendo desarrollada por DARPA. En la sección cuarta se plantean algunas posibles áreas de aplicación de las redes activas, haciendo especial énfasis en el área de multipunto fiable. Por último se presentan las conclusiones obtenidas.

2. Aspectos claves de una plataforma de red activa

En esta sección plantearemos diversos aspectos, que se han identificado como claves, al diseñar una plataforma de red activa. El modelo y funcionalidad de una plataforma concreta de red activa están determinados por las soluciones tecnológicas elegidas para resolver cada uno de estos aspectos.

2.1. API de Red

La programación de la red se hace en base a la utilización de una *Application Programming Interface de red* (API de red). En nomenclatura de redes activas, una API de red define una *máquina virtual* que interpreta un lenguaje específico. El API de red para IP incluye el lenguaje que define la sintaxis y la semántica de la cabecera IP y su efecto en los encaminadores de la red. En las redes tradicionales la maquina virtual es fija, y el poder expresivo del lenguaje limitado. Si podemos ver la cabecera IP de una red tradicional como la *entrada de datos* a una máquina virtual, podemos ver que en una red activa la entrada de datos a la máquina virtual son, además de los datos, los programas que porta el propio paquete.

Poder expresivo del lenguaje

De entre todos los aspectos que determinan cómo es el API de red, el más crucial es el lenguaje utilizado para programar la red.

El grado de programabilidad del API de red puede ser muy diverso. Las posibilidades van desde la selección de una simple lista de parámetros seleccionados de entre un conjunto predeterminado de posibilidades, a la utilización de un lenguaje completo equivalente Turing capaz de describir cualquier tipo de computación. Entremedias existe una infinita gama de posibilidades basadas: en restringir de distintas formas los lenguajes de propósito general o en la utilización de lenguajes de propósito específico.

Un aspecto esencial del lenguaje de programación es la facilidad de acceso a los recursos del nodo (por ejemplo, colas de salida o tablas de encaminamiento), para esto es determinante cuál es el conjunto de abstracciones asociados a éstos que proporciona el lenguaje.

La elección de un determinado lenguaje de programación está muy relacionada con la arquitectura de seguridad de la red activa. Los nodos se pueden programar, pero cualquier plataforma de redes activas debe ofrecer ciertos niveles de seguridad que garanticen que un programa inyectado en el nodo, ya sea de forma intencionada o inintencionada, no haga cosas "prohibidas" que pongan en peligro el funcionamiento del nodo. Con cosas prohibidas nos estamos refiriendo a programas que se meten en un bucle infinito, que acceden a zonas de memoria protegidas, o que hacen un uso abusivo de los recursos.

La ventaja de utilizar un lenguaje restringido es que limita los posibles comportamientos del nodo, simplificando el análisis de que el comportamiento programado es correcto, y por otro lado, acota de algún modo el efecto que un determinado paquete puede hacer en un nodo. En esta línea algunos de los lenguajes propuestos no admiten la existencia de estructuras de control del tipo bucles o bifurcaciones [3], o no admiten la utilización de punteros [4].

Alexander [2] ha identificado una serie de propiedades que debería poseer un lenguaje de programación válido para redes activas, siendo las más relevantes las siguientes: ser fuertemente tipado, disponer de un recolector de basura, posibilitar la carga dinámica, tener un mecanismo que permita definir diferentes interfaces para acceder a un mismo módulo y ofrecer buen rendimiento.

Almacenamiento temporal en el nodo

Otra característica importante del API de red es si permite que un programa incluido en un determinado paquete activo, deje residente información (por ejemplo información de estado), que pueda ser utilizada posteriormente por otros paquetes (por ejemplo que pertenecen al mismo flujo) que atraviesen el nodo posteriormente. Normalmente esta memoria es denominada *soft-state* pues tiene asociado un tiempo de vida pasado el cual si no se ha accedido a dicha información de alguna forma (para leerla o modificarla), dicha información se borrará del nodo sin que exista ningún tipo de notificación. Obviamente cuando se ofrece esta posibilidad, deben incluirse mecanismos que protejan de accesos no autorizados a dicha información de estado, y mecanismos que acoten la cantidad de memoria que puede utilizar un determinado usuario o flujo. La existencia o no, de memoria del tipo *soft-state* hace que sea muy diferente qué es posible programar en una red activa.

Composición dinámica de servicios

La meta última de las redes activas es el hacer más fácil el desarrollo de nuevos servicios de red. En concreto, debería proporcionar cierto soporte al proceso de creación de servicios. De aquí que una importante característica que debe incorporar el API de red sea la posibilidad de poder *componer* servicios a partir de bloques básicos. Estos bloques básicos son usualmente denominados *componentes*. Un API de red debería contener un mecanismo de composición para poder construir servicios compuestos en base a determinadas componentes. Existen trabajos que apuntan soluciones en esta línea siguiendo distintas aproximaciones, como por ejemplo el sistema LIANE [5] que plantea un sistema de composición basado en eventos, o el Netscript [6] donde se utiliza un modelo de computación de flujo de datos.

2.2. Sistema de Distribución de Código

Existen dos enfoques principales para realizar la descarga de código en el nodo: uno denominado discreto o fuera de banda y otro llamado integrado o en banda [1].

Enfoque discreto. Se mantiene el formato de paquete existente y se proporciona un mecanismo que soporta de forma separada la descarga de programas (*extensiones activas*), independizando esta descarga de programas, del proceso realizado sobre los paquetes. Cuando llega un paquete a un nodo se examina su cabecera y se lanza el programa adecuado para procesar dicho paquete. Esta separación puede resultar interesante en los casos en los que la selección de los programas que se deben inyectar en la red es realizada por administradores de red, en lugar de por usuarios finales. En este enfoque, el nodo programable no es tan "activo" como puede ser el enfoque integrado, pero es más fácil de implementar y desarrollar en las redes actuales.

Enfoque integrado. En este enfoque, los paquetes pasivos de las redes tradicionales son reemplazados por *paquetes activos* que incluyen tanto los datos como el programa activo, relativamente pequeño, que será ejecutado en los nodos cuando los vaya atravesando el paquete activo. En realidad, en el paquete activo puede ir realmente el código o sólo una referencia a dicho código, de ahí las dos posibilidades siguientes:

- *Código embebido*, en el que el código asociado al paquete activo va incluido en el propio paquete activo [3]. Este enfoque es muy adecuado cuando el código asociado al paquete activo tiene un tamaño muy pequeño o cuando se trata de paquetes activos que se mandan muy pocas veces.
- *Carga bajo demanda*, en el que los paquetes activos identifican el proceso que se debe ejecutar en el nodo, y este se carga la primera vez que se recibe un paquete activo de ese tipo [7]. De esta forma el proceso que trata un tipo de paquetes activos sólo se cargará una vez por sesión o mientras se esté mandando paquetes activos de ese tipo.

En ambos casos, enfoque discreto o integrado, deberán existir mecanismos que permitan la carga dinámica de código. Este mecanismo de carga dinámica permite que el código se cargue en el nodo mientras este se encuentra operativo, sin que sea necesario interrumpir temporalmente su funcionamiento.

2.3. Seguridad

En una red activa existen dos tipos de funcionalidades encargadas de proporcionar seguridad al sistema.

Por un lado existe una funcionalidad, usualmente denominada seguridad (*security* en terminología inglesa) que se encarga de evitar que usuarios no autorizados puedan causar daños intencionados al sistema, incluso hasta el punto de dejar un nodo o la red no operativa. Estos daños pueden consistir en dejar a un nodo sin recursos (por ejemplo memoria), cambiar las tablas de encaminamiento, modificar una extensión activa o el código embebido en un paquete generado por otro usuario. Generalmente, los mecanismos utilizados para proveer esta funcionalidad se basan en utilizar mecanismos de autorización y autenticación, firmas digitales y listas de control de acceso.

Por otro lado, existe una funcionalidad (denominada unas veces *safety* y otras *robustness* en terminología inglesa) que se encarga de evitar que usuarios autorizados, ya sea de forma intencionada o no, puedan causar daños al sistema o a otros usuarios. Estos daños pueden consistir en enviar a un nodo un programa que se mete en un bucle infinito, acceder al espacio de memoria que está siendo utilizado por otro usuario o por el propio nodo, etc. Los mecanismos utilizados para lograr este tipo de seguridad están en muchos casos relacionados con las propiedades del lenguaje utilizado para programar la red, como es la utilización de lenguajes fuertemente

tipados y con recolector de basura. También se ha propuesto la utilización de técnicas formales que permitan probar la *corrección* de un programa antes de que éste sea introducido en la red [8].

Uno de los principales retos para definir una arquitectura de seguridad satisfactoria es el coste de proceso de los mecanismos de seguridad asociados, que en muchos casos la hace inviable en una red activa en producción.

Dado que este aspecto constituye un factor crítico en las redes activas, en la arquitectura de red activa propuesta por DARPA y presentada en la sección 3 se plantea como objetivo de diseño que, deben siempre estar limitadas las consecuencias que puedan tener las acciones realizadas por un usuario, aunque éste disponga del nivel más alto de autorización.

2.4. Granularidad del Control

Este atributo se refiere a cuanto tiempo se mantendrá modificado el comportamiento del nodo (por ejemplo un cambio en el algoritmo de encaminamiento del nodo). Una primera posibilidad es que este cambio del comportamiento del nodo afecte a *todos* los paquetes que atraviesen el nodo hasta que no llegue otro paquete que modifique de nuevo dicho comportamiento. Otra posibilidad es que este cambio afecte únicamente al paquete que lleva el programa, granularidad por paquete. Y entre estas dos posibilidades existe otra intermedia que hace que el cambio en el comportamiento se restrinja a todos los paquetes pertenecientes a un determinado flujo. Siendo un flujo un conjunto de paquetes que comparten una característica común como por ejemplo tener una determinada dirección origen.

2.5. Propuestas actuales

A continuación haremos una breve descripción de las principales propuestas de modelos de red activa que se han realizado hasta la actualidad.

El proyecto Switchware [9] de la Universidad de Pensilvania. En Switchware se permite la opción de código embebido (programas PLAN contenidos en los programas activos) o la aproximación discreta de instalación de extensiones activas (*switchlets*). El PLAN (Programming Language for Active Networks) es un lenguaje funcional de propósito específico basado en OCAML. PLAN es muy simple y está diseñado teniendo como objetivo que los programas PLAN no puedan violar la política de seguridad[3]. Esta política se pretende que sea lo suficientemente restrictiva como para que los administradores de red permitan que los programas PLAN se ejecuten sin ser necesaria una autenticación previa. En esencia PLAN permite unas cuantas primitivas básicas, composición secuencial e invocación a *servicios externos* que pueden ser: extensiones activas (*switchlets*) o facilidades embebidas en el propio sistema. Estos servicios externos pueden requerir de una autenticación y/o autorización antes de permitir el acceso a los recursos.

En Switchware, la granularidad de control es por paquete para el caso del código PLAN, mientras que los *switchlets* están disponibles para todos los paquetes que atraviesen el nodo. En este proyecto han planteado una arquitectura de seguridad denominada SANE [10].

La herramienta ANTS [7] del MIT. El API de red ANTS consiste en una máquina virtual Java (JVM) aumentada con determinadas clases ANTS, que implementan métodos para permitir que los paquetes activos (*cápsulas* en terminología ANTS) sean decodificados e interpretados. ANTS permite disponer de *soft-state* en los nodos activos. La granularidad de control puede ser por paquete o por flujo. Como método de descarga de código utiliza el enfoque de carga bajo demanda.

El ANTS utiliza los mecanismos de seguridad propios de Java y además incorpora mecanismos de huella digital.

Dado que el estándar JVM no soporta acceso a recursos de transmisión a un nivel suficientemente bajo, la implementación de ANTS en plataformas estándares no puede soportar capacidades de QoS y está limitada a las capacidades de red básicas proporcionadas por Java.

El proyecto Smart Packets de BBN [4]. El objetivo central de este proyecto consiste en investigar cómo las redes activas pueden hacer más eficientes las tareas de administración y gestión de red. Para la distribución de código utiliza la aproximación de código embebido dentro de los paquetes activos. Uno de sus requisitos de diseño es que un paquete activo debe caber en un único paquete de nivel de enlace. Dada la hegemonía de Ethernet ellos han limitado que un programa activo debe tener como máximo 1024 bytes a los que se suman las cabeceras de control e información de autenticación. Han desarrollado dos lenguajes: el Sprocket y el Spanner. El Sprocket es un lenguaje similar a C modificado, para mejorar la seguridad (no dispone de punteros) y para añadir primitivas de red. Spanner es un lenguaje ensamblador, diseñado para permitir una codificación densa, en el que se compila Sprocket de modo que los paquetes activos portan código Spanner.

El API de red se implementa mediante una máquina virtual Spanner que se ejecuta como un demonio en un nodo activo.

Smart packets incorpora sistemas de seguridad basados en mecanismos de autenticación mediante el intercambio de certificados.

3. Arquitectura de Red Activa

En esta sección presentamos las principales ideas de la arquitectura de red activa que está siendo desarrollada por DARPA (Defense Advanced Research

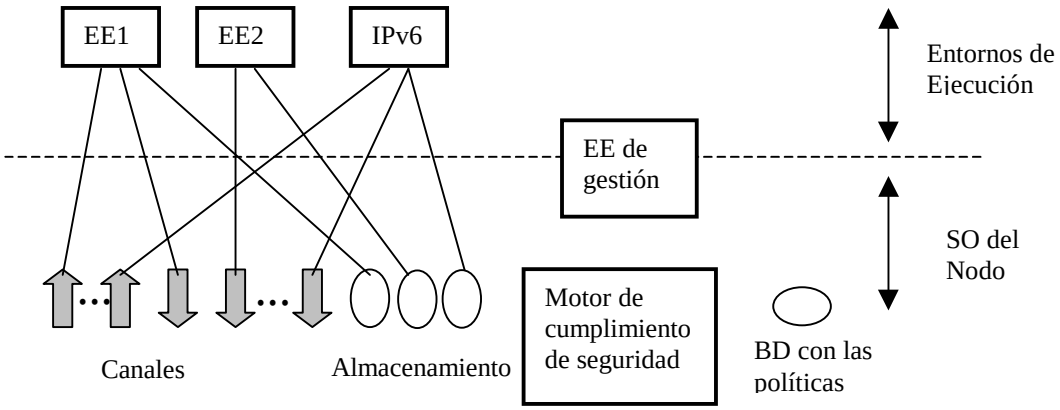

Fig. 1: Componentes

Projects Agency) dentro del programa de redes activas[11]. Esta arquitectura tiene como propósito definir cuales son los principales componentes de una plataforma de red activa y los interfaces entre éstos. Uno de sus requisitos es permitir que exista más de una API de red. Esto tiene varios objetivos. Por un lado, actualmente han sido propuestas varias APIs de red y todavía no existe la experimentación suficiente como para decidir cual es la mejor o incluso si debe existir una única API de red. Por otro lado, permite un marco en el que incorporar el "procesamiento rápido" de paquetes que hemos comentado previamente. Y por último, resuelve problemas de incompatibilidad, dado que la funcionalidad de IPv4 o IPv6 puede ser vista simplemente como otra API de red.

La funcionalidad de un nodo de la red activa se divide entre los entornos de ejecución (EE) y el sistema operativo del nodo (NodeOS). En la figura 1 se muestra la organización general de estos componentes que a continuación se analizan con más detalle.

3.1. Entorno de Ejecución

Cada EE exporta un API de red o máquina virtual que los usuarios pueden programar o controlar enviándole paquetes. En general, la ejecución de estas instrucciones puede causar que haya una actualización en la información de estado del EE o incluso del NodeOS, y puede causar que el EE transmita uno o más paquetes, ya sea inmediatamente o pasado un cierto tiempo.

Posibles EEs residentes en un nodo podrían ser: ANTs que sería un EE de propósito general o el Smart Packets que sería un EE específico de gestión de red. El EE oculta al NodeOS los detalles de la interacción con los usuarios.

Cada nodo activo dispone de *un entorno de ejecución de gestión* a través del cual se controlan ciertos aspectos de la configuración del nodo. Por ejemplo, el EE de gestión puede permitir la modificación de la base de datos con la política de seguridad del nodo (ver figura 1) o puede permitir el arranque de otros EEs.

3.2. Sistema Operativo del Nodo

El NodeOS es el nivel intermedio que opera entre los EEs y los recursos físicos (transmisión, computación y almacenamiento). El NodeOS proporciona la funcionalidad básica sobre la cual los entornos de ejecución construyen las abstracciones presentadas al usuario. El NodeOS gestiona los recursos del nodo activo y media entre los distintos EEs cuando se solicitan dichos recursos. El NodeOS aísla a un entorno de ejecución de los efectos que puede tener el comportamiento de otros entornos de ejecución.

Los usuarios y otras entidades de la red se representan por una abstracción denominada *principal*. Esta abstracción se utiliza en temas de seguridad y de contabilización de uso de recursos. Cuando un EE solicita un servicio del NodeOS, está petición se acompaña del identificador (y posiblemente una credencial) del principal, en cuyo nombre se está realizando la petición. Dicho principal puede ser el propio EE u otra entidad (por ejemplo, un usuario) en cuyo nombre actúa el EE. El NodeOS presenta esta información al motor de cumplimiento de la seguridad que verifica la identidad (ver figura 1) y autoriza o no al principal a obtener el servicio solicitado.

Procesamiento de los paquetes

El NodeOS implementa la abstracción de *canal de comunicaciones* sobre la cual se envían y se reciben paquetes. Existen dos tipos de canales: los *anchored* y los *cutthrough*. Los canales *anchored* están anclados a un determinado EE y son la abstracción sobre la cual los EEs envían y reciben paquetes. El flujo lógico de los paquetes a través de un nodo activo es el siguiente: cuando un nodo activo recibe un paquete desde un enlace físico, lo clasifica en base al contenido del mismo (normalmente con la cabecera), y de acuerdo con esto: el paquete será descartado o bien asignado a un determinado canal.

El NodeOS utiliza patrones para asociar cada paquete recibido a un determinado canal. Dichos patrones son proporcionados por el EE cuando crea dicho

canal (por ejemplo paquetes con una determinada combinación de protocolo IP y puerto TCP).

Además de los canales *anchored* también pueden existir los canales *cuthrough* que por el contrario no están asociados a ningún EE. Cuando un NodeOS recibe un paquete sobre un canal de este tipo, dicho paquete no será interceptado ni procesado por ningún EE, y la única operación que se realizará con dicho paquete será la de almacenamiento-reenvío, típica de una red *pasiva*. Cómo se puede observar esta abstracción se utiliza para implementar el procesado rápido de paquetes ya antes comentado.

3.3. Active Network Encapsulation Protocol

El *Active Network Encapsulation Protocol* (ANEP) [12] proporciona los mecanismos para que un usuario pueda especificar a que EE particular debe dirigirse el paquete que está generando. Un campo en la cabecera ANEP es un identificador del EE que debe ejecutar dicho paquete. En la actualidad estos identificadores de EE son asignados por la Autoridad de Números Asignados para Redes Activas.

Es interesante señalar que un paquete que no contiene una cabecera ANEP puede ser procesado por un EE siempre que dicho EE cree el canal *anchored* adecuado. Este podría ser el caso de un paquete que ha sido generado en un sistema final no activo y se desea que se procese en un EE. Un ejemplo de aplicación podría ser un servicio de mejora del control de congestión de TCP implementado en un EE.

4. Algunas Areas de Aplicación

Ya hemos mencionado que el control dinámico de la red activa permite que los servicios “se ajusten a medida”, de las aplicaciones y de las condiciones cambiantes de la red. Este ajuste a medida tiene como objetivo mejorar el rendimiento percibido por las aplicaciones, en comparación con las soluciones extremo-a-extremo. En esta sección proporcionamos varios ejemplos de cómo las redes activas pueden mejorar el rendimiento de las aplicaciones.

Los esfuerzos de mejorar el rendimiento de las aplicaciones usando las redes activas cubren una amplia variedad de áreas [13]. Entre estas áreas las que parecen más prometedoras son las siguientes.

4.1. Adaptación dinámica

Un área interesante de investigación está relacionada con la adaptación dinámica a las condiciones cambiantes de la red.

Principalmente, las redes activas pueden aportar mejoras de rendimiento significativas en situaciones donde sea crucial la respuesta rápida ante cambios en la información local (en un nodo o conjunto de nodos de la red).

Por ejemplo, la calidad de servicio obtenida por una aplicación se puede degradar de forma significativa, en situaciones de congestión en la red o de enlaces con una tasa alta de pérdida de paquetes. La aproximación tradicional para resolver este tipo de situaciones ha sido que la fuente se vaya adaptando a las condiciones de la red, pero este planteamiento tiene una serie de limitaciones como es el tiempo que necesita la fuente para detectar el cambio y reaccionar en consecuencia. Al introducir en los nodos inteligencia sobre como adaptarse a los cambios en la red, se hace que la adaptación se produzca antes con el consiguiente aumento de rendimiento.

En esta área ya se han obtenido resultados interesantes en el Protocolo Boosters [14]. En este trabajo se muestra como un protocolo se puede adaptar a condiciones tales como un aumento en la tasa de pérdidas en la red añadiendo dinámicamente funcionalidad de corrección de errores alrededor del área de la red donde se están produciendo las pérdidas.

Otros trabajos en esta área, relacionados con control de congestión son: el de Zegura y otros [15] que han propuesto estrategias inteligentes de descarte de paquetes en situaciones de congestión para preservar la calidad de vídeo MPEG en esos periodos, y el de Faber [16] que ha planteado como introduciendo en la red inteligencia para actuar en situaciones de congestión, TCP puede mejorar su rendimiento en un 18%.

4.2. Gestión de Red

La manera tradicional de realizar la gestión de una red, consiste en recolectar información de los nodos gestionados, en base a solicitar el valor de unas variables determinadas y comprobar si se detectan anomalías. Este enfoque concentra la inteligencia en las estaciones de gestión, lo que puede provocar cuellos de botella. Además esta aproximación limita seriamente la posibilidad de seguir la pista a problemas detectados, en escalas de tiempo razonables.

El principal grupo que ha realizado aportaciones en esta área ha sido BBN con su proyecto Smart Packets [4]. Este grupo ha planteado que la programación de la red hace posible que los nodos gestionados sean así mismo nodos programables. De esta forma, los centros de gestión pueden enviar programas a los nodos gestionados. Este enfoque aporta tres ventajas. Primero, la información de gestión que vuelve al centro de gestión puede ajustarse en tiempo real a los intereses que en un momento dado tenga dicho centro de gestión, reduciendo por tanto, el tráfico y la cantidad de información que debe analizar dicho centro de gestión. Segundo, muchas de las reglas de gestión empleadas por el centro de gestión pueden ser incluidas en programas que se envían al nodo gestionado, permitiéndole identificar y corregir problemas automáticamente sin la intervención del centro de gestión. Tercero, permite hacer más cortos los ciclos de monitorización y control.

4.3. Comunicaciones Multipunto

Por último finalizaremos esta sección mostrando como las redes activas pueden mejorar el rendi-

miento de un servicio de comunicaciones de multipunto fiable.

Después de una década investigando en protocolos de multipunto fiable ésta continúa siendo una línea de investigación abierta, debido a que las aproximaciones actuales presentan alguna de las siguientes deficiencias: (1) plantean la resolución de la problemática de aplicaciones específicas, no ofreciendo servicios genéricos que cubran un amplio rango de requisitos; (2) utilizan enfoques que les permiten escalar correctamente, cuando el número de receptores es muy elevado o cuando los participantes de la comunicación están muy alejados geográficamente, a costa de limitar el grado de fiabilidad proporcionado; (3) no optimizan el retardo ni el ancho de banda de la red. Con objeto de resolver estas deficiencias en [17] se presenta el RMANP *(Reliable Multicast Active Network Protocol*) un protocolo de multipunto fiable que opera sobre la tecnología de redes activas.

Las redes activas pueden facilitar la provisión de un servicio de multipunto fiable que resuelva las deficiencias planteadas anteriormente. El hecho de que el número de receptores sea potencialmente muy elevado conlleva asociadas dos problemáticas: la implosión de ACKs y la implosión de NACKs. Respecto a la implosión de ACKs, el utilizar la tecnología de redes activas permite que cuando dichos paquetes atraviesen la red vayan siendo fusionados en determinados nodos activos que se encuentran en el camino entre los receptores y la fuente. La *fusión de ACKs* consiste en que el nodo activo por cada "n" ACKs que recibe, reenvía hacia el emisor un único ACK pero que lleva fusionada la información de los "n" ACKs recibidos. Con relación a la implosión de NACKs, el soporte que pueden ofrecer las redes activas consiste en realizar un filtrado de NACKs en determinados nodos que están en el camino entre los receptores y la fuente. El *filtrado de NACKs* consiste en que los nodos activos recuerdan los NACKs que han reenviado hacia la fuente, o lo que es lo mismo, los datos para los que se ha solicitado retransmisión, y cuando reciben un NACK lo reenvían si solicita otros datos distintos, o lo filtran (no lo retransmiten) si solicita datos ya pendientes de retransmisión. Con la fusión de ACKs y el filtrado de NACKs se elimina la problemática de la implosión de ACKs y NACKs, respectivamente.

Muchas aplicaciones multipunto muestran sensibilidad al retardo. Con relación a este aspecto, el soporte de las redes activas tiene dos vertientes: 1) realizar un almacenamiento, bajo la política del mejor esfuerzo posible, de los datos multipunto en determinados nodos activos en el camino entre la fuente y los receptores, 2) realizar un control intermedio de secuencia. El *almacenamiento intermedio* de los datos permite implementar un esquema de retransmisiones locales que evita que todas las retransmisiones tengan que realizarse desde la fuente. Cuando un nodo activo recibe un NACK viajando hacia la fuente, y tiene almacenado el dato solicitado, dicho nodo filtra el NACK y retransmite el dato solicitado. El realizar *retransmisiones locales* permite: 1) disminuir el retardo extremo a extremo, 2) optimizar el consumo de ancho de banda, 3) repartir la carga asociada a las retransmisiones entre la fuente y los nodos activos que llevan a cabo el almacenamiento intermedio de los datos, 4) realizar la recuperación ante diferentes errores de forma concurrente e independiente, en distintas partes de la red. El *control intermedio de secuencia* consiste en que un nodo activo lleva control de los datos multipunto recibidos y cuando detecta un salto en la secuencia genera un NACK que envía hacia la fuente. El control intermedio de secuencia permite una detección prematura de las pérdidas lo cual conlleva una disminución del retardo extremo a extremo.

Otra necesidad de algunas de las aplicaciones multipunto es el conocimiento de la identidad de los receptores por parte de la fuente. El soporte de las redes activas con relación a este aspecto está basado en la función de *agregación de información*. Los receptores pueden informar sobre su identidad enviando ACKs hacia la fuente. Dada la posibilidad de las redes activas de realizar una agregación de estos ACKs, la fuente conocerá la identidad de sus receptores sin que se produzca implosión. La fuente en lugar de procesar un ACK por cada receptor procesará un número reducido de ACKs, cada uno con información relativa a muchos receptores.

Adicionalmente a los requisitos específicos de las aplicaciones multipunto, todo servicio de comunicaciones debe tener como objetivo el optimizar el uso de recursos de la red. Las redes activas ofrecen en este sentido la posibilidad de realizar retransmisiones de ámbito restringido y filtrado de retransmisiones. La *retransmisión de ámbito restringido* evita la retransmisión de los datos solicitados a través de todas las interfaces de un nodo activo. En este caso, el soporte de los nodos activos consiste en recordar a través de que interfaces se solicitó un paquete de datos, restringiendo la retransmisión de ese paquete a dichas interfaces. Los nodos activos utilizan el *filtrado de retransmisiones* para evitar el envío de múltiples retransmisiones del mismo paquete de datos, cuando éstas han sido pedidas en paralelo por un conjunto de receptores o nodos activos que cuelgan del mismo interfaz.

El RMANP ha sido implementado sobre ANTS [7] habiéndose realizado un análisis de los requisitos de memoria en los nodos activos, y de los tiempos de proceso necesarios para procesar los paquetes activos RMANP en los nodos activos [18].

5. Conclusiones

Aquí hemos presentado una panorámica de la tecnología de redes activas y hemos discutido algunas de sus potenciales aplicaciones.

Las redes activas son un área investigación prometedora que en estos momentos se encuentra en una fase

relativamente inmadura debido a que todavía no se ha realizado la suficiente experimentación y son muchas las líneas de investigación abiertas. Sin embargo, se han identificado dos retos que si no son superados de forma satisfactoria, comprometerán el éxito de la tecnología:

- La *arquitectura de seguridad*. El principal problema de los mecanismos de seguridad tradicionalmente utilizados es su coste, en tiempo y computación, que los hacen inviables para una red activa en producción, de aquí que la pregunta, aún sin resolver, sea: ¿Qué clase de mecanismos de seguridad son necesarios para poder seguir la pista de los permisos asignados a millones de usuarios en miles de encaminadores activos desplegados por todo el globo?
- El *rendimiento*. En los modelos de redes activas propuestos hasta el momento se pueden identificar dos tendencias: una primera, liderada por el MIT con su ANTS, consistente en ofrecer un API de red muy flexible que permita programar casi cualquier cosa aún a costa de obtener rendimientos bastante pobres, y una segunda, liderada por la Universidad de Pennsylvania con su Switchware, consistente en utilizar un API de red muy restringido y hacer que la red se programe esencialmente en base a las extensiones activas introducidas en el nodo por el proveedor de red, obteniendo buenos rendimientos a costa de limitar la flexibilidad. Quizá la solución pase por buscar un compromiso entre la flexibilidad y el rendimiento una vez resuelta la cuestión clave: ¿qué tipos de flexibilidad son realmente útiles?

Referencias

[1] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall and G.J. Minden. *"A Survey of Active Network Research"*. IEEE Comunications Magazine, pp. 80-86, January 1997.

[2] S. Alexander. *"A Generalized Computing Model of Active Networks"*. Ph.D. U. de Pennsylvania. December 1998.

[3] M. Hicks, J. T. Moore, D. S. Alexander, C. A. Gunter, S. M. Nettles. *"PLAN: A Programming Language for Active Networks"*. Proceedings of International Conference on Functional Programming (ICFP) '98, September 1998.

[4] B. Schwartz, W. Zhou, A. W. Jackson, W. T. Strayer, D. Rockwell and C. Partridge. *"Smart Packets for Active Networks"*. January 1998. http://www.net-tech.bbn.com/smtpkts/smart.ps.gz

[5] Kenneth L. Calvert, Samrat Bhattacharjee, Ellen W. Zegura and James Sterbenz. *"Directions in Active Networks"*. IEEE Communications Magazine, pp. 72-78 ,October 1998.

[6] Y. Yemini and S. Da Silva. *"Towards Programmable Networks"*. IFIP/IEEE International Workshop on Distributed Systems: Operations and Management. L'Aguila. Italy, October 1996. http://www.cs.columbia.edu/~dsilva/netscript. html

[7] D. Wetherall, J. Guttag and D.L. Tennenhouse. *"ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols"*. Proceedings of IEEE OPENARCH'98, San Francisco, CA, April 1998.

[8] Samrat Bhattacharjee, Kenneth L. Calvert and Ellen W. Zegura. *"Reasoning About Active Network Protocols"*. Proceedings of ICNP '98, Austin, TX, October 1998.

[9] S. Alexander, W. A. Arbaugh, M. W. Hicks, P. Kakkar, A. D. Keromytis, J. T. Moore, C. A. Gunter, S. M. Nettles and J. M. Smith. *"The SwitchWare Active Network Architecture"*. IEEE Network, Special Issue: Active and Programmable Networks, 12(3):29-36, May/June 1998.

[10] S. Alexander, W. A. Arbaugh, , A. D. Keromytis and J. M. Smith. *"A Secure Active Network Environment Architecture: Realization in SwitchWare"*. IEEE Network, Special Issue: Active and Programmable Networks, 12(3):37-45, May/June 1998.

[11] Programa DARPA de redes activas. http://www.darpa.mil/ito/research/anets

[12] S. Alexander, B.Braden, C. A. Gunter, A. W. Jackson, A. D. Keromytis, G. J. Minder and D. Wetherall. *"Active Network Encapsulation Protocol (ANEP)"*. RFC DRAFT, 1997.

[13] U. Legedza, D. Wetherall and J. Guttag. *"Improving the Performance of Distributed Applications Using Active Networks"*. Proceedings of IEEE INFOCOM'98, San Francisco, April 1998.

[14] D. Feldmeier, A. McAuley, J. Smith, D. Bakin, W. Marcus and T. Raleigh. *"Protocol booster"*. IEEE Journal on Selected Areas in Communications, 16(3):437-444, April 1998.

[15] Samrat Bhattacharjee, Kenneth L. Calvert and Ellen W. Zegura. *"An Architecture for Active Networking"*. Proceedings of High Performance Networking (HPN'97), White Plains, NY, April 1997

[16] T. Faber. *"ACC: Using Active Networking to Enhance Feedback Congestion Control Mechanisms"*. IEEE Network, Special Issue: Active and Programmable Networks, 12(3):61-65, May/June 1998.

[17] M. Calderón, M. Sedano, A. Azcorra and C. Alonso. *"Active Network Support for Multicast Applications"*. IEEE Network, Special Issue: Active and Programmable Networks, 12(3):46-52, May/June 1998.

[18] M. Sedano Ruiz. *"Soporte de Redes Activas para un servicio integrado de multipunto fiable con control de Congestión*". Ph.D. Facultad de Informática, Universidad Politécnica de Madrid. Febrero 1999.