



APLICACIÓN DE REDES ACTIVAS A LAS WLANS

Alvaro Suárez Sarmiento y Javier Jesús Sánchez Medina

Grupo de Arquitectura y Concurrencia
Departamento de Ingeniería Telemática - ULPGC
Campus Universitario de Tafira, Aula Magna
35017- Las Palmas de G. C.
alvaro@dit.ulpgc.es, jsanchez@ayacata.teleco.ulpgc.es

RESUMEN

En la actualidad las arquitecturas de redes son estáticas en el sentido de que el software de control de los componentes hardware de la red está empotrado y generalmente cerrado (propiedad del fabricante de los componentes hardware). Esto hace que instalar nuevos servicios lleve una cantidad de tiempo elevadísimo. Recientemente se han implantado algunos proyectos de redes activas que evitan estos problemas. En este artículo presentamos algunas ideas básicas sobre las redes activas, sus aplicaciones y algunos proyectos. Finalmente presentamos el trabajo inicial que hemos realizado sobre este tipo de redes y algunas conclusiones y trabajo futuro.

Palabras Clave: Arquitectura de red, Internet, Redes programables, Redes Activas, programación distribuida.

1. INTRODUCCIÓN

En los últimos años la telecomunicación ha experimentado un desarrollo que ha conducido a que la práctica totalidad de las personas se puedan comunicar, desde cualquier lugar geográfico, con otras personas o dispositivos electromecánicos. Para que esto sea posible ha sido necesario elaborar estándares de comunicación determinados y aceptados por la «Comunidad Internacional». Para establecer estos estándares, las Arquitecturas de Redes (especificación de un conjunto de capas y protocolos de comunicación) [1] deben ser homologadas por todas las partes implicadas.

Sin embargo este proceso de estandarización mundial es extremadamente lento debido a que se requiere poner de acuerdo a multitud de entidades académicas, industriales, sociales y políticas. La estandarización de una nueva arquitectura de red puede durar muchos años. Esto hace que nunca lleguen a tiempo las mejoras que proporcionarían. Otro problema importante es el de la compatibilidad de la nueva arquitectura con las anteriores: Es necesario mantener una compatibilidad con las infraestructuras existentes para que sea posible implantar progresivamente la nueva arquitectura. El verdadero problema de las estrategias de diseño de las arquitecturas de redes actuales es que no se suele separar la concepción del hardware y el software de control de red. Debido a esto cuando los usuarios piden nuevos servicios a las compañías de comunicación, éstas no pueden proporcionarlos rápidamente debido a que se deben ajustar: 1. Al funcionamiento de los equipos de red (conmutadores, enrutadores, etc.) y 2. A los algoritmos que implementan los protocolos de comunicación, que han sido diseñados especialmente para una arquitectura de red determinada. En la figura 1

se muestra un ejemplo gráfico de esta idea. Se muestra una arquitectura red muy simplificada que consta básicamente de dos niveles muy directamente relacionados entre ellos haciendo muy poco flexible la introducción de nuevos protocolos y servicios.

Recientemente han surgido con fuerza las redes de comunicación programables [32] que pueden solventar estos y otros problemas. El objetivo básico de este tipo de redes es facilitar la implantación rápida, explotación y evolución de nuevos protocolos de red y en definitiva de nuevos servicios demandados por los usuarios. En la

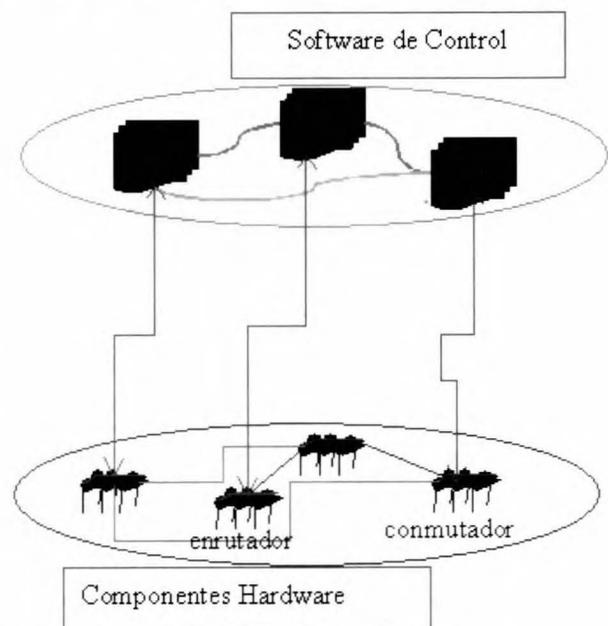


Figura 1. Esquema de una arquitectura de red simplificada rígida.

práctica existen dos Escuelas de pensamiento para la implantación de la Redes Programables:

* Señalización abierta (Opensig). La idea de los seguidores de esta propuesta consiste en considerar a los componentes hardware (conmutadores, enrutadores, etc.) como un conjunto de objetos distribuidos y crear un conjunto de interfaces de red programable mediante las cuales los operadores de red podrían programar libremente a la red. Un proyecto que merece atención es el IEEE 1520. En la figura 2.a se muestra un ejemplo gráfico de esto.

* Redes Activas (AN). La idea [1] [2] [25] de las redes activas es desacoplar el hardware (y las capas software inferiores) y la programación de los servicios ofrecidos. En contraposición a: las redes de software empotrado de la actualidad y las que siguen el modelo de programación cuasi-estático anterior, éstas dotan a la red de facilidades para permitir la inyección de programas en tiempo real sobre los componentes hardware de la misma. Para ello se basan en el concepto de «paquete activo», que es un datagrama que además de datos porta código o referencias a código. De esta manera se puede generalizar y acelerar la instalación de un nuevo servicio sólo con programarlo en estos paquetes y migrarlo de un componente hardware a otro de la red. En la figura 2.b se muestra un esquema de esta nueva idea.

Las redes activas son muy interesantes por varios motivos: imponen un nuevo modelo de arquitectura de red, utiliza un modelo de programación distribuido novedoso y permite experimentar con nuevos protocolos en muy poco tiempo, a la vez que se pueden implantar utilizando el protocolo estándar IP. Además de las redes activas, hoy en día existen un gran número de proyectos de diseño de nuevas arquitecturas de red entre las que destacan [33] y [34].

La estructura del artículo es la siguiente: en el apartado 2 se presentan las ideas generales acerca de las redes activas. En el apartado 3 comentamos el trabajo que estamos realizando en esta tecnología y finalmente en el apartado 4 se presentan algunas conclusiones sobre nuestro trabajo y algunas ideas de futuro.

2. LAS REDES ACTIVAS

Las redes actuales están programadas de una manera muy rígida. Normalmente consisten en un conjunto de componentes hardware programados con un software embebido en los mismos, lo cual limita críticamente la computación distribuida en la red. Además los paquetes que transportan no pueden ser modificados para garantizar la integridad de la información. Esto hace inviable la implantación de nuevos protocolos. Para solucionar estos problemas se inventaron las redes activas [1] [2]. En estas redes se puede: 1. Inyectar programas para su ejecución «inmediata o en tiempo real» sobre los componentes hardware y 2. Los datos portados en los paquetes pueden ser modificados en tiempo real. Tanto los datos como los programas que los procesarán, o referencias a ellos se encapsulan en paquetes activos. Los componentes hardware/software que ejecutan el código para transformar los datos portados en los paquetes activos se denominan nodos activos. Existen dos tipos de red activa. Si los paquetes activos están arquitectónicamente separados del código que los procesa [1] las redes activas se denominan discretas o fuera de banda (el administrador de cada nodo activo carga estáticamente el código que procesa los paquetes activos). Ejemplo de este tipo de redes son: [13] y [15]. En cambio si la distribución de los paquetes y del código utilizan la misma arquitectura, éstas se denominan integradas o en banda (el código se carga dinámicamente según se necesite). Ejemplo de este tipo de redes son: ANTS (Active Node Transfer System) [3], [4] y [7], el M0

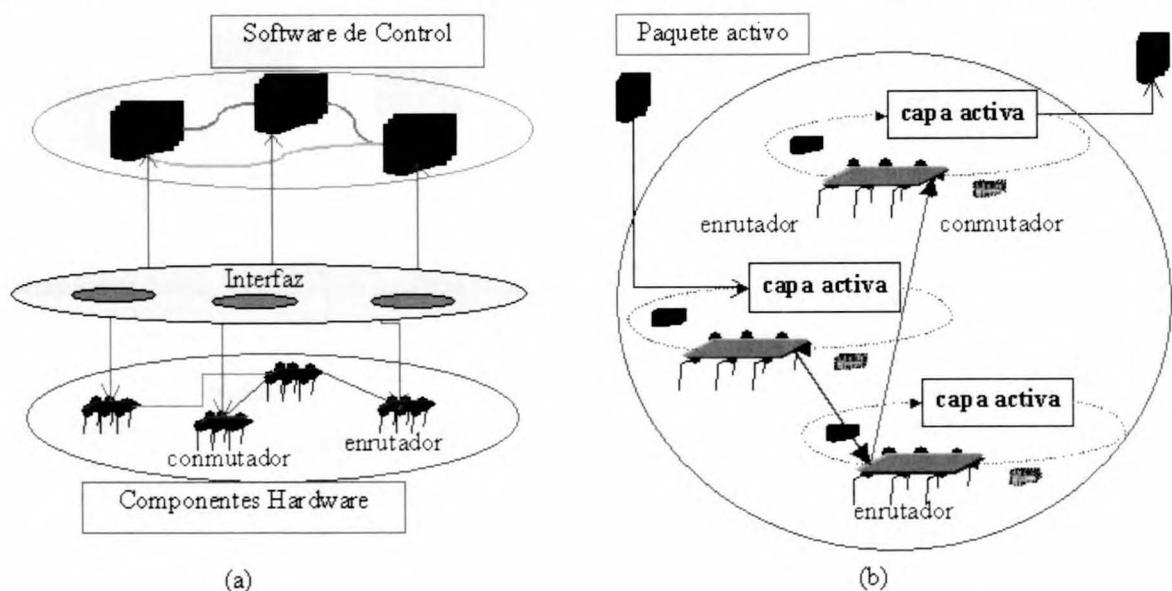


Figura 2. Esquemas de redes programable: a) Señalización abierta, b) redes activas

[26], los SmartPackets [23]. Existen aproximaciones mixtas como el proyecto Switchware [10] y [11].

Explicamos los nodos activos basándonos en la figura 3. Un entorno de ejecución (EE, Execution Environment) es el conjunto de recursos software y hardware que se asignan a un determinado flujo de paquetes en un nodo activo. Tanto los recursos que se asignan como el nivel de libertad para operar con ellos son, en general, determinados de antemano, antes de la creación del entorno de ejecución por motivos de seguridad. En un nodo activo se llevan a cabo cuatro actividades: 1. Distribución de paquetes: Los paquetes entrantes según su tipo, que se indicará por medio de campos en los mismos, son distribuidos al entorno de ejecución que corresponda. 2. Procesamiento de mensajes: Cada entorno de ejecución lleva a cabo la computación que proceda sobre los paquetes que le son asignados y haciendo uso de los códigos que corresponda. Estos códigos en general se encontrarán en el almacén de componentes o en los mismos paquetes. 3. Almacenamiento de componentes: Se reservarán recursos para el almacén de bibliotecas de código que podrán ser

cargadas o descargadas de manera estática o dinámica. En general, para cada entorno de ejecución se especificará a qué componentes de este almacén puede acceder. 4. Planificación/transmisión: Los flujos de paquetes de salida de cada entorno de ejecución son planificados y transmitidos.

2.1. APLICACIONES

Seguidamente indicamos una lista de posibles aplicaciones de las redes activas clasificadas según sean integradas o discretas.

2.1.1 Integradas

Posibles aplicaciones para la aproximación integrada son: Aplicaciones de Ping, Herramientas de diagnóstico y gestión de red, Multicasting activo, Control de calidad del servicio y Señalización.

En general la opción integrada se utiliza para aplicaciones que no requieren un elevado ancho de banda pero

MIT	ANTS [3], [4] y [7].	Plataforma escrita en Java. Facilidad para la implementación de protocolos nuevos. Bajo rendimiento.
MIT	PAN [9]	Similar a ANTS pero utiliza código máquina. Mejor rendimiento que el de ANTS. Problemas de seguridad
BBN	SmartPackets [23]	Diagnóstico de red extendido. Gran mejora frente a SNMP
Georgia Tech	CANE [20]	Procesamiento de datos específico de la aplicación (ejemplo: control de la congestión en streams de vídeo MPEG)
Georgia Tech	Caches auto-organizativas [19]	Modelos analíticos y simulaciones para estudiar las mejoras obtenidas mediante el cacheo de código en la red
U. Pennsylvania	Switchware/SANE [10] y [11]	Combinación: discreta e integrada. Código portado y referencias a código cargado estáticamente en los nodos, sobre una plataforma de red segura (SANE) que garantiza la integridad del entorno
U. Pennsylvania	P4 [13]	FPGAs para implementar el proceso de los paquetes en hardware (ejemplo: corrección de errores en los paquetes)
U. Arizona	Joust/Scout [27], [28] y [29]	Versión del ANTS sobre el Scout (sistema operativo modular orientado a comunicaciones) y el Joust (JVM para Scout). Rendimiento: entre 2 y 3 veces superiores al ANTS original
U. Columbia	Netscript [24]	Middleware para programar los nodos intermedios. Simplifica el desarrollo de sistemas interconectados y la programación remota
ETH Zurich y U. Washington	ANN [15]	Arquitectura hardware y software de un nodo activo para alto volumen de tráfico y de ancho de banda. Muy buen rendimiento
U. P. M.	RMANP [14]	Protocolo de multicast activo con tres niveles de fiabilidad distintos sobre ANTS. Encuentran problemas de rendimiento

Tabla 1. Algunos proyectos de investigación de redes



sí una gran flexibilidad. En [25] se muestra un protocolo de multicast robusto de audio y otro de multicast de vídeo en capas. Ambos protocolos se implementan en dos plataformas distintas: el ANTS (los paquetes activos portan referencias a código que se carga dinámicamente) y el M0 (cada paquete activo porta todo el código necesario para su procesamiento). En [14] se hace un estudio del rendimiento de un protocolo de multicast con tres tipos de fiabilidad implementado en el ANTS y se llega a la conclusión de que ANTS es muy sencillo de utilizar para implementar nuevos protocolos pero tiene un rendimiento pobre. En [16] se explica la plataforma PANTA [17] (escrita en Python) para analizar un esquema de reserva del ancho de banda de cada enlace en una red activa.

2.1.2. Discretas

Algunas aplicaciones implementadas son las siguientes: Caches de Web con auto-aprendizaje, Algoritmos de control de congestión, Acciones On-line, Mezcla de datos de sensores y Conmutadores de alta velocidad.

Como puede verse, este tipo de aplicaciones tiene unos requerimientos de ancho de banda muy elevados. Por otra parte, el procesamiento de los paquetes activos no necesita ser muy pesado. Por lo tanto estamos ante una relación de compromiso entre programabilidad de la red y ancho de banda. En este caso se pierde programabilidad en favor de una mayor velocidad de enrutamiento. Es necesario que se implemente en hardware la mayor parte posible de la funcionalidad de los nodos, utilizando por ejemplo FPGAs. Es decir, que se hace apropiado realizar la red activa a muy bajo nivel. En [15] se presenta un conmutador de alta velocidad activo (la arquitectura hardware/software y los rendimientos obtenidos). En [13] se utiliza una FPGA para dotar a un conmutador de reprogramabilidad.

2.2. ALGUNOS PROYECTOS DE REDES ACTIVAS

En la tabla 1 presentamos algunos proyectos de investigación de redes activas. Primero presentamos el centro de investigación en el cual se llevan a cabo, segundo el nombre del proyecto y tercero un comentario breve.

3. NUESTRO TRABAJO EN REDES ACTIVAS

A continuación presentamos el trabajo realizado en nuestro grupo de investigación en redes activas. Estamos diseñando protocolos activos que creemos tendrían buenos rendimientos en redes inalámbricas y redes híbridas alámbricas-inalámbricas de área local. Para ello estudiamos qué problemas presentan este tipo de redes y su solución aplicando redes activas. Hemos elegido la plataforma ANTS para implantar nuestros protocolos con el objeto de realizar una batería de pruebas con ellos.

Según [30] las redes inalámbricas de área local adolecen básicamente de tres problemas: 1. El ancho de banda disponible es más pequeño que en una red alámbrica. 2. El ancho de banda disponible es variable en el tiempo. 3. El enlace inalámbrico es susceptible de perderse en un momento dado.

Actualmente trabajamos en protocolos para el cálculo distribuido del producto de dos matrices, de ping y de multicast orientado a tiempo real. En el siguiente apartado presentamos su funcionamiento y resultados obtenidos con el ANTS.

En el ANTS los protocolos se programan extendiendo tres clases: Protocol, Capsule y Application. La topología de la red se configura mediante el archivo «.config» asignándole a cada nodo un nombre, la aplicación que correrá sobre él (si existe), la dirección IP/UDP del canal a utilizar y un manager.

3.1 Cálculo distribuido de matriz por matriz

La topología utilizada para el cálculo distribuido se muestra en la figura 4. Una estación maestro distribuye el trabajo y recolecta los resultados, y un conjunto de estaciones esclavos se encargan de realizar el cálculo que se les ha solicitado.

En el maestro se centralizan el control y la planificación del cálculo distribuido (asigna tareas a los esclavos, recoge resultados de éstos y gestiona la utilización de la arquitectura) para realizar el producto $C=A \times B$. Realiza un único broadcast distribuyendo la matriz B a todos los esclavos. Luego distribuye ordenadamente filas de A a esclavos. Los esclavos devuelven el resultado (filas de C) al maestro.

Para realizar esto utilizamos dos tipos de cápsulas que extendemos de la clase abstracta Datacapsule del ANTS, BCapsule (para el broadcast inicial de B) y UniCapsule (para el unicast de A y C). Ambos tipos de cápsulas solamente portan los datos del cálculo. El enrutamiento es el estándar IP en el caso de la UniCapsule y en el caso de BCapsule se envían clones de esta cápsula a todos los vecinos. Además se crean dos tipos de aplicación (MApplication y SApplication) extendiendo la clase Application del ANTS, que implementan las operaciones del nivel de aplicación del nodo maestro y el esclavo respectivamente.

Una vez implantado este protocolo en el ANTS realizamos una batería de pruebas cambiando la configuración de la topología y comprobando que el resultado del cálculo distribuido coincidía siempre con el mismo cálculo secuencial. Hemos realizado las pruebas con todos los nodos activos locales ejecutándose en una sola máquina. Llegamos a las dos siguientes conclusiones: 1. El número de comunicaciones que realiza es el mínimo posible. Esto es muy importante en una red inalámbrica debido a la

latencia de los enlaces. 2. Se trata de un protocolo ligero eficiente implementable sobre UDP. Para implantarlo a bajo nivel habría que tener en cuenta mecanismos de detección de pérdida y retransmisión de paquetes. Esto se haría en un estudio posterior.

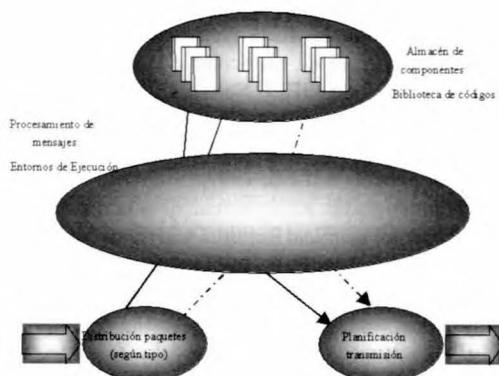


Figura 3. Visión conceptual de un nodo activo.

3.2 Ping Broadcast

En la aplicación distribuida anterior cada estación podría querer averiguar qué otras están disponibles en la red en un momento determinado. Cada estación podría saberlo haciendo «ping» sobre las otras. Debido a la naturaleza de los enlaces inalámbricos la estación que desea realizar este ping no tiene confianza en ninguna de las otras estaciones. En redes híbridas alámbricas-inalámbricas es necesario que el ping se propague por todos los segmentos de la red.

El protocolo que hemos diseñado se resume en los siguientes pasos: 1. Una estación inicia la petición de ping haciendo un broadcast de la misma en el medio de comunicación. 2. Las estaciones presentes en el medio compartido escuchan la petición de ping. 3. En el caso de no haber recibido una petición de ping con el mismo origen en un tiempo muy cercano (que se trataría de un «eco» de otro ping), retransmite la petición de ping haciendo un broadcast nuevamente a todos los vecinos. Además envía una cápsula unicast de ACK al nodo origen del ping que quedará enterado de que este nodo está disponible.

Para implantarlo en ANTS hemos extendido la clase DataCapsule en dos tipos de cápsula, uno para la inundación inicial de la red (FloodCapsule) y otro para la respuesta de los nodos (ACKCapsule). El enrutamiento en estos dos tipos de cápsula es equivalente a los del ejemplo anterior. Las cápsulas de inundación se difunden y las de respuesta se enrutan de manera estándar IP. Extendemos la clase abstracta Application del ANTS en la clase PingApplication que realiza todas las operaciones de nivel de aplicación de este protocolo.

Las conclusiones que obtuvimos del prototipado y prueba de este protocolo son: 1. Las pruebas realizadas demuestran que una implementación distribuida de este protocolo podría ser de gran utilidad en redes en las cuales es necesario comprobar el estado de los enlaces periódicamente, por ejemplo las redes inalámbricas. 2. Una implementación de este protocolo a nivel de red, mejoraría en gran medida su rendimiento.

3.3 Multicast en tiempo real

Este protocolo tiene tres características que lo hacen eficiente: 1. Las peticiones de suscripción no se cursan hasta el nodo suministrador del servicio. Al contrario se gestionan solamente entre los nodos suscriptores y el nodo intermedio anterior. 2. Se realiza un cacheo de los paquetes recientes en los nodos intermedios para retransmitirlos en el caso de que haya pérdidas. 3. En el caso de que no se disponga de los paquetes solicitados, se ahorra ancho de banda concentrando las peticiones de retransmisión de todos los nodos que cuelgan de uno dado en un solo paquete que se cursa al nodo intermedio anterior. 4. También se ahorra ancho de banda mediante un descarte de los paquetes obsoletos (que no cumplen las restricciones de tiempo real) en todos los nodos de la red.

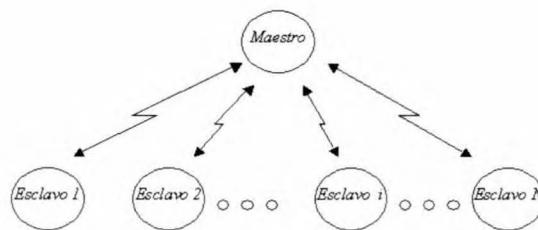


Figura 4: Arquitectura del sistema

Para implantarlo en ANTS hemos extendido la clase DataCapsule en cuatro tipos de cápsula: 1. MDataCapsule para los datos multicast. 2. NACKCapsule para las peticiones de retransmisión. 3. SCapsule para las peticiones de suscripción. 4. RdataCapsule para la retransmisión de cápsulas. El enrutamiento en el tipo MDataCapsule es el único que no es unicast, sino que sigue una lista de distribución. También hemos extendido la clase abstracta Application en tres tipos de aplicación: 1. SenderApplication para el nodo suministrador del servicio. 2. NodeApplication para los nodos intermedios en la red. 3. ReceiverApplication para los nodos terminales de la red, que pueden suscribirse al servicio multicast.

4. CONCLUSIONES Y TRABAJO FUTURO

En este artículo hemos presentado algunas ideas principales sobre un nuevo tipo de arquitectura de redes denominada redes activas. La idea básica de este tipo de redes es considerar la transformación de los paquetes de datos inyectados en la red dentro de los nodos activos. De

esta manera, el código puede migrar desde un nodo a otro y los datos pueden ser alterados a lo largo del camino de comunicación entre un emisor y el receptor final.

Estas redes tienen muchas aplicaciones y la ventaja principal es que se pueden instalar nuevos protocolos en un periodo de tiempo muy reducido adaptando la red de esta manera a las demandas de nuevos servicios por parte de los usuarios. Quizás en el futuro los nuevos proveedores de servicios de Internet tengan la posibilidad de ofrecer nuevos servicios de rápida y eficientemente a sus usuarios y quizás sea este hecho el que les pueda colocar en una posición fuerte frente a la competitividad tan fuerte que existe hoy día en este mercado.

En nuestro grupo de trabajo hemos desarrollado un conjunto de protocolos activos como primera aproximación a este tipo de redes y hemos utilizado la herramienta ANTS para prototiparlos y probarlos rápidamente.

En los próximos meses experimentaremos diseñando nuevos protocolos para aplicaciones interesantes que aprovechen las características positivas y atenúen los problemas de los entornos inalámbricos. Además, utilizando el ANTS realizaremos pruebas más exigentes y reales con los mismos. A medio plazo pretendemos utilizar los protocolos que en este periodo demuestren su eficiencia para implementarlos en un nivel inferior en la torre de protocolos, buscando mejorar su rendimiento.

BIBLIOGRAFÍA

- [1] D. L. Tennenhouse and D. Wetherall. Towards an active network architecture. In *Multimedia Computing and Networking 96*, San Jose, CA, Jan 1996.
- [2] D. Tennenhouse et al. A Survey of Active Network Research. *IEEE Communications Magazine*, 1997.
- [3] D. Wetherall. Developing Network Protocols with the ANTS Toolkit. En la página web de David Wetherall en la universidad de Washington. <http://www.cs.washington.edu>.
- [4] D. Wetherall et al. ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols. En *1st Conf. On Open Architectures and Network Programming (OPENARCH'98)*, páginas 117-129, San Francisco, CA, Abril 1998. IEEE.
- [5] U. Legedza, D. Wetherall and J. Guttag, «Improving the Performance of Distributed Applications Using Active Networks», *Proc. INFOCOM'98*, 1998.
- [6] D. Wetherall, U. Legedza, and J. Guttag, «Introducing New Internet Services: Why and How», *IEEE Network*, May/June 1998.
- [7] D. Wetherall, J. Guttag and D. L. Tennenhouse, «ANTS: Network Services Without the Red Tape», *IEEE Computer*, April 1999.
- [8] D. Wetherall, «Active network vision and reality: lessons from a capsule-based system», in *SOSP'99*, Dec. 1999.
- [9] E. L. Nygren, S. J. Garland, and F. Kaashoek, «PAN: A High-Performance Active Network Node Supporting Multiple Mobile Code Systems», *IEEE*, 1999.
- [10] D. S. Alexander et al., «The SwitchWare Active Network Architecture», *IEEE Network*, May/June 1998.
- [11] D. S. Alexander et al., «A Secure Active Network Environment Architecture: Realization in SwitchWare», *IEEE Network*, May/June 1998.
- [12] M. Hicks, et al., «PLANet: An Active Internetwork», *IEEE*, 1999.
- [13] I. Hadzic and J. Smith, «On-the-fly Programmable Hardware for Networks», *Proc. GLOBECOM 1998*.
- [14] M. Calderón, et al., «Active Network Support for Multicast Applications», *IEEE Network*, May/June 1998.
- [15] D. S. Decasper, et al., «A Scalable High-Performance Active Network Node», *IEEE Network*, January/February 1999.
- [16] D. Williams, et al., «An Active Network Bandwidth Reservation Scheme», *IEEE*, 1999.
- [17] A. Fernando and B. Kummerfeld. «Pants: Active Node Transfer System Technical Report», University of Sydney, Australia, 1998.
- [18] S. Bhattacharjee et al. «An Architecture for Active Networking». *Proc. INFOCOM'97*, April 1997.
- [19] S. Bhattacharjee, K. Calvert and E. Zegura, «Self-Organizing Wide-Area Network Caches», *Proc. INFOCOM'98*, April 1998.
- [20] E. Zegura, «CANes: Composable Active Network Elements», Georgia Tech; <http://www.cc.gatech.edu/projects/canes/>
- [21] S. Bhattacharjee, K. Calvert and E. Zegura, «Active Networking and the End-to-End Argument», Georgia Tech;
- [22] S. Bhattacharjee, K. Calvert and E. Zegura, «Directions in Active Networks», *IEEE Communications Magazine*, October 1998.
- [23] B. Schwartz et al., «Smart Packets for Active Networks», Jan. 1998. Disponible en <http://www.net-tech.bbn.com/smpkts/smart.ps.gz>
- [24] Y. Yemini and S. DaSilva, «Towards programmable networks», *IFIP/IEEE Int'l. Wksp. Dist. Syst.: Operation and Mgmt.*, October 1996.
- [25] A. Branches et al., «Multicasting Multimedia Streams with Active Networks», *IEEE*, 1998.
- [26] C. Tshudin. The messenger environment M0 - a condensed description. En J. Vitek and C. Tshudin, editors, *Mobile Object Systems - Towards the Programmable Internet*, LNCS 1222, pages 149-156. Springer, Apr. 1997.
- [27] D. Masberger and D. Peterson. «Making paths explicit in the Scout operating system», *2nd Symp. Op. Syst. Design and Implementation*, Oct 1996.
- [28] A. B. Montz et al., «Scout: A Communications-Oriented Operating System», *IEEE*, 1995.
- [29] J. Hartman et al., «Joust: A Platform for Liquid Software», *IEEE Network*, July 1998.
- [30] Amit B. Kulkarni and Gary J. Minden, «Active Networking Services for Wired/Wireless Networks», *IEEE* 1999
- [31] Tanenbaum Andrew, «Redes de Computadoras», 3ª edición, Prentice Hall, ISBN 968-880-958-6, 1997.
- [32] Campbell A.T., De Meer H.G., Kounavis M.E., Miki K., Vicente J., and Vilella D., «A Survey of Programmable Networks», *Computer Communications Review*, April 1999.
- [33] Campbell, A.T., Kounavis M.E., Vilella D.A., Vicente J. (Intel), Miki K. (Hitachi), De Meer H.G., and Kalaichelvan K.S. (Nortel), «Spawning Networks», *IEEE Network Magazine* July/August 1999.
- [34] <http://www.ist-fain.org/index.html>