Arquitectura de Red Programable y sus Aplicaciones en Sistemas Móviles Inalámbricos[1]

D. Larrabeiti, M.F. Sedano, M. Calderón, B. Alarcos, M. Urueña, R. Romeral, M. Bagnulo, E. de la Hoz

Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid.
Av. Universidad 30 - 28911 Leganés (Madrid)
E-mail: {dlarra, maria, muruenya, rromeral, marcelo}@it.uc3m.es

Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid.
Ciudad Universitaria S/N - 28040 Madrid
E-mail: marifeli@gsi.dit.upm.es

Área de Ingeniería Telemática. Dpto. de Automática. Universidad de Alcalá.
Crta Madrid-Barcelona, Km 33,600 - 28871. Alcalá de Henares (Madrid)
E-mail: {bernardo, enrique}@aut.uah.es

Abstract*. This paper identifies a set of scenarios where intelligent programmable devices can find a field of application in the development of wireless networks, and presents a novel architecture for programmable nodes especially designed to ease the deployment of network applications in wireless networks, including a security architecture that enables scalable secure on-demand deployment of code. Finally, we describe preliminary results of this work on a prototype of this system before starting the deployment phase in a UMTS testbed.*

1. Redes activas, redes programables y redes móviles inalámbricas

Cada vez es más frecuente en la periferia de las redes IP, el uso de dispositivos con capacidad para realizar procesamiento sobre los paquetes más allá del simple encaminamiento de paquetes. Ejemplos de este tipo de dispositivos son cachés web transparentes, cachés de contenidos multimedia y de optimización de la distribución de video, cortafuegos y sistemas de detección de intrusión, multiplexores de direcciones y puertos NAPT, adaptadores de formato, traductores IPv4-IPv6, acondicionadores y marcadores de tráfico, etc.

El paradigma de las *redes activas* [2], hoy considerada una línea de investigación madura, llevó el estudio de este tipo de dispositivos hasta sus últimas consecuencias, analizando la posibilidad de que los propios usuarios finales distribuyeran código en los routers activos para cambiar el comportamiento del router sobre el tráfico. Dicho código podía transportarse en cápsulas [3] (paquetes conteniendo el propio código que debe procesarlos), debía ejecutarse en entornos de ejecución suficientemente seguros [4], y debería compatibilizar el uso compartido de los recursos del router entre las distintas aplicaciones activas [5]; todo esto constituye un reto de difícil implementación y aplicación industrial.

Más pragmáticamente, las *redes programables* [6], admiten un aceptable grado de programabilidad, ofreciendo un mayor control sobre las entidades capaces de desplegar dinámicamente los "programas de red". Estos programas, suficientemente probados y garantizando la compatibilidad entre ellos, son descargados desde almacenes de código seguros (nota: en ocasiones en la literatura los términos *redes activas* y *redes programables* aparecen usados indistintamente).

Entre las iniciativas de organizaciones de normalización que fomentan la programabilidad de la red se encuentran OPES [7], revelando la preocupación de IETF en el control de la alteración de contenidos por parte de dispositivos intermediarios, IEEE P1520 [8] y FORCES [9], donde se promueve el desarrollo de una arquitectura de router abierta con la definición de un interfaz entre elementos de reenvío y elementos de control. Asimismo, la industria ha respondido a esta necesidad con una amplia oferta de procesadores de red [10] como alternativa flexible pero de alta velocidad a la implementación de procesadores de interfaz mediante ASICs, tecnología esta última con una capacidad de reprogramación más limitada.

[1] Este trabajo está financiado por el Ministerio de Ciencia y Tecnología a través del proyecto TIC2001-1650-C02-01/02 AURAS (**A**rquitectura integrada **U**MTS-**R**edes **A**ctivas para la implantación rápida de **S**ervicios).

En todo este rango de posibilidades de programación del comportamiento de nodos de red se ha realizado una intensa investigación, tanto en la definición de arquitecturas [6] como en las posibles aplicaciones de las mismas [4]. La aplicación de estas tecnologías al campo de las redes móviles inalámbricas no ha sido una excepción. Así existen trabajos que intentan optimizar las prestaciones de sistemas inalámbricos, clásicamente rompiendo el principio de transporte extremo a extremo. Tal es el caso de TCP spoofing y relaying en PEP (Performance Enhancing Proxies) [11, 12] (especialmente útiles en escenarios con segmentos inalámbricos con alta tasa de errores y con protocolos de enlace sin capacidad de retransmisión), el de dispositivos que pretenden optimizar el transporte de flujos multimedia [13] (optimizaciones basadas en el conocimiento y caracterización de las áreas de cobertura y de la posición y la trayectoria del móvil), e incluso en la aplicación de técnicas de redes activas al cómputo de rutas en redes ad-hoc [14]. Por otra parte, algunos fabricantes de procesadores de red, como IBM [15], prevén una importante área de aplicación de estos dispositivos en el diseño de equipos de sistemas móviles 3G, aduciendo que su flexibilidad inherente para cambiar sus pilas de protocolos permitirá amortizar la inversión en estos equipos durante la transición a la futura arquitectura todo-IP [16].

En este contexto tecnológico, se sitúa el trabajo presentado en este artículo, que está siendo realizado en el marco del proyecto MCYT AURAS [1], y cuyo principal objetivo es el desarrollo de arquitecturas de nodos programables adecuadas para redes móviles y el estudio de sus aplicaciones en UMTS. En este artículo se propone una arquitectura de nodo programable que intenta dar soporte a aplicaciones de red para terminales móviles. En primer lugar se justifican los principios de diseño de esta arquitectura; a continuación, se describe brevemente la arquitectura de seguridad ideada para el despliegue seguro y escalable de código en la red; después se definen escenarios de aplicación de este sistema y, finalmente, se muestran resultados preliminares obtenidos sobre un prototipo que implementa la arquitectura propuesta.

2. Arquitectura de Red Programable SARA

2.1 Arquitectura de nodo programable

SARA (Simple Active Router Assistant) [17] es una arquitectura de nodo programable que tiene por objetivo principal la implementación y explotación de dispositivos de red reprogramables dinámicamente. Su programabilidad se basa en algunos conceptos desarrollados en el campo de las redes activas; es decir, propone la existencia de un entorno de ejecución dinámico en el nodo donde se ejecutan las aplicaciones de red; pero a diferencia de un nodo activo, las posibles aplicaciones a lanzar en este entorno se hallan controladas por el administrador de la red, y el usuario final simplemente queda habilitado para activarlas y cambiar su estado mediante señalización específica. Este procedimiento de distribución de código en la red y de activación del mismo mediante paquetes de señalización, se realiza de manera segura mediante la arquitectura de seguridad presentada en la sección siguiente. Asimismo se ha procurado dar soporte a aplicaciones para redes móviles, como se analiza en este trabajo.

Los principios de diseño de SARA son los siguientes:

I. Los routers convencionales deben delegar el procesamiento específico en asistentes. Partiendo de la existencia de routers convencionales de gama media con *fast path* hardware y *slow path* software, es evidente que la introducción de servicios de procesamiento específico en su CPU, mermaría rápidamente las prestaciones de los routers convirtiendo su capacidad de procesamiento en un cuello de botella. De hecho la carga de CPU es un factor que hoy en día es tenido muy en cuenta a la hora de introducir cortafuegos, túneles cifrados y conmutadores de nivel 4 en una red. Por consiguiente todo procesamiento complejo sobre los paquetes debería realizarse sobre elementos especializados, ajenos al propio router, y de la manera más desacoplada posible de su función principal de reenvío de paquetes. En definitiva, la arquitectura SARA propone que todo router que requiera ampliar su programabilidad con nuevas aplicaciones de red debe reprogramarse para delegar su procesamiento en procesadores co-ubicados (denominados asistentes) que pueden consistir en ordenadores conectados al router con una LAN de alta velocidad y conteniendo un entorno de ejecución apropiado para el tratamiento eficiente de paquetes. La penalización causada al router por esta nueva función queda determinada por el coste de identificar los paquetes que deben ser desviados, entre los paquetes que entraron en el router por sus interfaces convencionales, y desviarlos al asistente que debe procesarlos. El asistente puede realizar procesamiento transparente sobre los paquetes en capa 3 y superiores antes de reenviarlos al router para su encaminamiento normal, y debe disponer del mayor control posible sobre los recursos del router.

II. Cooperación router-asistente. La delegación de funciones al asistente/s precisa cooperación entre el router y el asistente. Los mecanismos establecidos para ello son:

A) Desvío de paquetes de señalización de aplicaciones de red. Esto permite lanzar, cambiar el estado o retirar aplicaciones del entorno de ejecución en los asistentes.

B) Vista del estado del router. Ciertas aplicaciones requieren consultar la tabla de rutas, capacidades y nivel de ocupación de los enlaces del router, longitud media de las colas, etc. El entorno de ejecución pone a disposición de las

aplicaciones esta información, mediante una vista seleccionada del estado del router – típicamente obtenida por SNMP – cacheada por motivos de eficiencia.

C) Protocolo Router-Asistente (RAP). Si se requiere disponer de capacidad de proceso de cualquier flujo convencional que atraviese el router, es necesario habilitar mecanismos de control más complejos y versátiles que los anteriores. RAP permite tareas tales como la programación de desvíos de flujos a asistentes, reparto de flujos entre varios asistentes o la salida de un paquete por un interfaz determinado del router (útil en retransmisiones sobre subárboles en aplicaciones de multicast fiable).

La figura 1 muestra la arquitectura software empleada en el asistente (derecha) y la interacción con el router preexistente. Lógicamente, existe una relación de compromiso entre funcionalidad disponible a las aplicaciones de red y sobrecarga en el router convencional. Por este motivo, la arquitectura ofrece dos niveles de cooperación: cooperación *ligera*, que requiere los mecanismos A) y B); y cooperación *completa*, basada en C). Nótese que la cooperación ligera es suficiente para muchas aplicaciones y sólo precisa pequeños cambios en los routers preexistentes.

III. Despliegue transparente de aplicaciones de red. Un objetivo de diseño de la arquitectura es facilitar la movilidad, ocultando la topología de la red a los terminales y la ubicación de los nodos programables. Las aplicaciones se lanzan en los nodos del trayecto automáticamente empleando paquetes de señalización direccionados al sistema final, no siendo necesario direccionar los nodos programables en el trayecto (o árbol) origen-destino(s). Para implementar esta funcionalidad de manera eficiente se emplea la opción *router alert* de IP (RFC2113 para IPv4 y RFC2711 para IPv6). La alternativa, la opción no transparente, muy frecuente en la mayoría de esquemas de redes activas donde se construye un overlay de nodos activos estableciendo vecindad explícita entre ellos, es también posible. En este caso pueden emplearse búsquedas multicast incrementales en anillo para localizar al nodo programable más cercano.

Los paquetes de señalización de aplicaciones de red llevan encapsulación ANEP, contienen referencias al entorno y a la aplicación que debe procesarlos. En caso de no hallarse residente la aplicación en el entorno, se procede a su descarga remota, a partir de su URI asociada, desde un almacén de código seguro (servidores de código). La aplicación tiene un tiempo de vida especificado en la propia señalización, que se refresca mediante mecanismos soft-state, con el fin de caducar aplicaciones que procesaban los paquetes de un móvil cuyas comunicaciones han dejado de atravesar el nodo en cuestión.

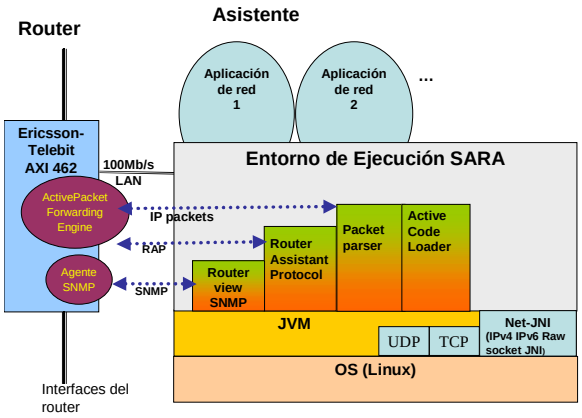

Figura 1. Arquitectura software SARA

2.2 Arquitectura de Seguridad

La introducción de programabilidad en la red introduce nuevos riesgos de seguridad que hay que abordar. Entre esos riegos cabe destacar la protección de la propia red programable frente a paquetes de señalización no auténticos que pueden provocar efectos negativos en la arquitectura del nodo programable. Uno de los principales efectos a evitar es el consumo de recursos por ejecución de aplicaciones no autorizadas. Para proteger a la red de este riesgo se ofrecen mecanismos de autenticación y autorización de los usuarios que solicitan un servicio y protección de los paquetes de señalización con autenticación e integridad. Con el objeto de cubrir estos riesgos se ha desarrollado una solución de seguridad para SARA denominada ROSA (Realistic Open Security Architecture for Active Networks) [18, 19].

El escenario real sobre el que se puede aplicar una red programable consta de usuarios que solicitan servicios, los cuales requieren un procesamiento por parte de los nodos programables. Para poder obtener este servicio, los usuarios establecerán sesiones que se identificarán de forma única mediante los siguientes parámetros, a los que denominaremos *parámetros de sesión*:

- Identificación del usuario, U, que solicita el servicio (es decir solicita la activación de una aplicación de red).
- Periodo de tiempo de validez de la sesión, marcado por el tiempo de inicio y fin de la sesión (SST, SET).
- Direcciones IP del origen y destino del flujo de paquetes de señalización correspondientes a la sesión, L.
- Identificador de código activo que deben ejecutar los nodos programables, Ci.

El planteamiento de seguridad se ha enfocado desde la perspectiva de protección de la red programable frente a usuarios maliciosos. Debido a esto, el principal riesgo es el uso no autorizado de una aplicación, es decir, el envío de paquetes de señalización no autorizados que provocan la

ejecución de dicho código. Para proteger la red de este riesgo hay que ofrecer mecanismos de autenticación y autorización de los usuarios que solicitan el servicio, y protección de los paquetes de señalización con autenticación e integridad. Por otro lado se ha buscado una solución que sea escalable en cuanto a sobrecarga de las cabeceras y sobrecarga de procesamiento de seguridad de los paquetes de señalización. Debido a esto se ha desestimado el uso de mecanismos basados en clave asimétrica. Se ha optado por proteger los paquetes de señalización con un código de autenticación de mensajes (generado con hmac) basado en clave simétrica, que tiene unos tiempos de ejecución rápidos y no sobrecarga en exceso las cabeceras con respecto a otros mecanismos como firma digital. El uso de clave simétrica sin embargo, plantea un problema de distribución de clave entre los distintos componentes de la red programable y el usuario.

ROSA implementa un mecanismo distribuido de generación de clave, en el que parte de la información que llevan los paquetes de señalización es usada como credencial de autorización. La red programable con seguridad estará formada (Fig. 2) por un *Servidor de Autorización* (AS), *Servidores de Código* (CS) y *Nodos Programables* (PN compuesto por un router y su asistente) que compartirán un valor secreto (Kci) asociado a cada aplicación (Ci). Los Kci son generados por el AS periódicamente y enviados a los CS por un canal confidencial. Cuando los PN descargan un código (aplicación) desde los CS también descargan el Kci asociado a dicho código. Podemos distinguir tres fases en el proceso de seguridad:

Solicitud de servicio: un usuario solicita el servicio a un AS, cuando éste comprueba que el usuario está autorizado, genera una clave de sesión (K) y se la da al usuario. Esta clave depende de los *parámetros de sesión* (U, SET, SST, L, Ci) y del *valor secreto* (Kci) asociado al código. Para generar K se utiliza una función de derivación de claves. La comunicación entre el usuario y el AS se hace con mecanismos de autenticación mutua y confidencialidad.

Generación de señalización protegida: el usuario envía paquetes de señalización desde el origen al destino. Los paquetes van protegidos con un código generado con hmac y la clave de sesión (K). Los paquetes de señalización llevan además los *parámetros de sesión* que identifican la sesión y con los cuales se ha calculado K.

Procesamiento en los nodos programables: cuando un paquete de señalización de una sesión llega a un PN por primera vez, este descarga el valor secreto (Kci) y la aplicación desde un CS. Con los parámetros de sesión que van en el paquete y Kci, el nodo genera la clave de sesión K. Con K y hmac verifica que el paquete es auténtico e integro. Con los parámetros de sesión, comprueba que el paquete está

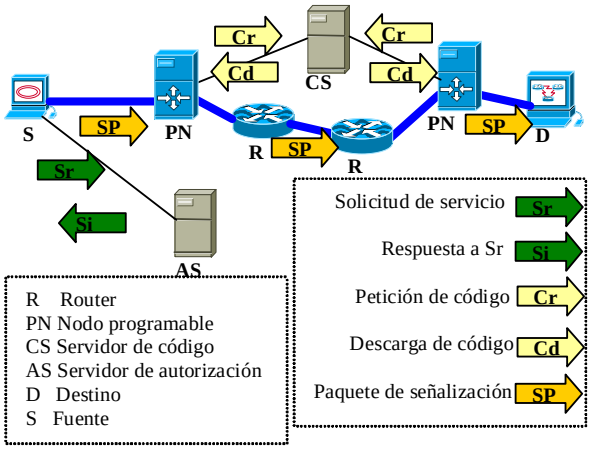

Figura. 2. Arquitectura ROSA

dentro de la ventana de tiempo de la sesión. El nodo comprueba de esta forma, que el paquete ha sido generado por un usuario que conoce K. Esto implica que el usuario ha sido autorizado previamente por el AS a enviar paquetes en el periodo de tiempo especificado, desde un origen a un destino y para ejecutar una aplicación de red concreta. Como vemos, los parámetros de sesión que van en el paquete actúan como credencial dado que permiten comprobar que el paquete está autorizado para ser procesado. Si la aplicación modifica el contenido del paquete de señalización, lo vuelve a proteger utilizando hmac y la clave K. Para el resto de paquetes de señalización de la sesión no es necesario descargar el código y el *valor secreto*, puesto que ya están en el nodo.

La solución de seguridad planteada soporta movilidad de los terminales dado que si un usuario se mueve por la red y este movimiento implica un cambio del nodo programable, el nuevo nodo será capaz de general la clave de sesión y comprobar la integridad y autenticad de los paquetes sin ningún tipo de procesamiento adicional.

La solución propuesta se ha implementado y se ha evaluado la influencia de la arquitectura de seguridad en el retardo extremo a extremo. Los resultados obtenidos muestran que ROSA introduce un pequeño incremento (7,6%) sobre el retardo extremo a extremo sin seguridad, cuando la descarga del *valor secreto* y de la aplicación desde el CS no es necesaria, es decir, para la mayoría de los paquetes procesados por los nodos programables. Sólo el primer paquete de señalización de la sesión experimenta un mayor retardo debido a la descarga segura del código y del *valor secreto*.

3. Aplicaciones en redes móviles inalámbricas

Como se ha descrito en la introducción, existen diversos escenarios en los que programas ejecutados dentro de la red pueden apoyar la comunicación en

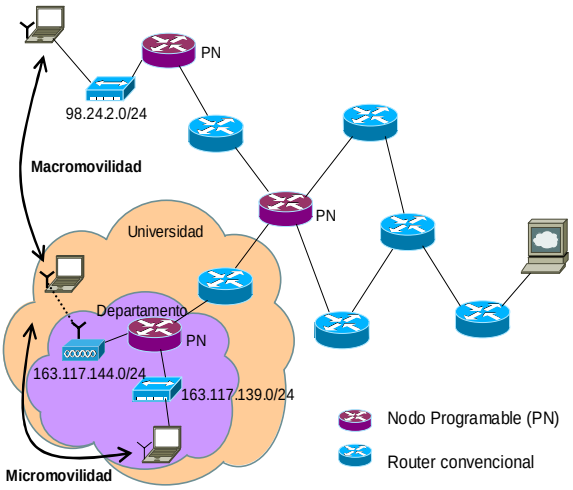

Figura 3. Micro y macromovilidad

entornos inalámbricos, entre ellas han sido muy estudiadas las aplicaciones encargadas de optimizar las prestaciones, como los PEP (Performance Enhancing Proxies) [12] y relays de aplicación con conocimiento exacto de cobertura y contexto del terminal móvil. Un sistema como SARA añade a este tipo de aplicaciones la posibilidad de ubicar automáticamente dichos proxies en los nodos óptimos del trayecto (en enlaces con tasas altas de errores o alto retardo). La información necesaria para caracterizar con precisión los enlaces de dicho trayecto puede obtenerse mediante paquetes de señalización que inspeccionan las vistas de estado de los routers. De este modo se da un cierto soporte a macromovilidad (Fig.. 3), en el sentido de que las aplicaciones de red necesarias vuelven a restaurarse al cambiar el trayecto de los paquetes debido al movimiento del terminal o cambios de enrutamiento.

Nótese que no se pretende dar una alternativa a la implementación de movilidad de terminal, ya que para ello existen protocolos específicos. Tampoco a la movilidad de aplicaciones, de la que se han ocupado ampliamente los estudios sobre movilidad de agentes. sino un soporte bastante más simple independiente de si el móvil ha precisado cambiar de dirección IP en su cambio de subred.

En un escenario de micromovilidad (Fig. 3) la programabilidad de un solo router en la red de una organización, sí nos permitiría una implementación rápida de mecanismos simples de movilidad. Por ejemplo, activando de manera controlada funciones de proxy-ARP. De esta manera un usuario puede desplazarse a otro departamento de la organización, y cambiar de subred física conservando su dirección IP y, con ella, sus permisos de acceso a sus recursos de intranet, etc.

UMTS

En UMTS se proponen distintos escenarios de aplicación de dispositivos programables de procesamiento de paquetes. En primer lugar, en la implementación completamente actualizable de los

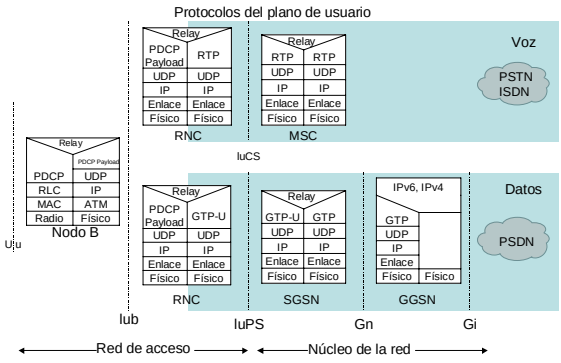

Figura 5. Plano de usuario UMTS all-IP

elementos del núcleo de la red. Así, algunos fabricantes de Network Processors [15] han añadido a sus diseños iniciales -sólo IP/IEEE802-, soporte de conmutación ATM con AAL2 y AAL5, segmentación y reensamblado (SAR). De esta manera pretenden que estos procesadores sean directamente aplicables a la implementación del plano de datos de BSC/RNC, SGSN, GGSN y MSC[2] desde la actual release 99 basada en ATM/IP (Fig. 4) a la futura arquitectura all IP (Fig. 5), prometiendo velocidades hasta OC-48 y haciendo innecesaria la actualización de los componentes físicos de los equipos.

Otras aplicaciones de red programables distintas de la mera implementación flexible de los protocolos UMTS que proponemos incluyen la optimización de prestaciones en aplicaciones TCP/IP en los terminales, de manera análoga a las propuestas existentes en la literatura para WiFi o satélite [12].

Las limitaciones en ancho de banda, de procesamiento y memoria de los terminales inalámbricos pueden mejorarse substancialmente con dispositivos activos dentro de la red (mediante retransmisiones, caching, control de flujo en función de la capacidad disponible en el segmento radio, etc) especialmente si éstos se ubican cerca de los terminales. Sin embargo, esto no es sencillo en UMTS debido a la movilidad y a los mecanismos de gestión de la misma. Por consiguiente, tanto en los nodos B como en la RNC sólo sería posible lanzar aplicaciones que no precisen traspasar su estado entre nodos B y RNCs. Si el traspaso del estado de la aplicación de red entre estos elementos fuera necesario, sería asimismo imprescindible implicar el traspaso de dicho estado en la señalización UMTS de traspaso gestionada por el SGSN. Esto no es imposible de realizar, pero no es general deseable, ya que impide mantener aislados los procesos de optimización de las aplicaciones de usuario de los

[2] Por limitaciones de espacio no se incluye en este artículo la descripción de los elementos, protocolos y acrónimos empleados en UMTS, remitiendo al lector a la norma para una documentación completa de los mismos.

procesos de señalización estándar. En definitiva, un funcionamiento compatible con la señalización actual de procesos que precisen traspaso de estado requiere que los procesos de usuario deban alojarse en equipos más allá del SGSN, donde no existe información específica de las condiciones del segmento radio móvil-nodo B. En este escenario la característica de SARA para el despliegue transparente de código a lo largo del trayecto de los paquetes daría soporte a un traspaso de GGSN, que no puede darse en la práctica.

Por último, la disponibilidad de interfaces abiertos de acceso a los recursos red como OSA/Parlay [21], abre en UMTS nuevas posibilidades para la explotación de las aplicaciones de red. La razón es que gracias al uso controlado de IP que se realiza en 3G se cubre una carencia específica de las aplicaciones de red en el contexto de IP: la imposibilidad práctica de aplicar un modelo de negocio en el uso de estas aplicaciones. Efectivamente, en 3G los usuarios están plenamente identificados y se dispone de mecanismos para facturar por el consumo que realizan de los recursos. Este procedimiento puede aplicarse al uso de las aplicaciones de red, agregándolas al conjunto de recursos contabilizados. La viabilidad de esta nueva perspectiva de uso del interfaz OSA/Parlay la estamos validando en la plataforma de pruebas del proyecto IST Opium [20]. En este caso, la aplicación de red es una caché transparente con pre-carga inteligente de objetos web ubicada tras el GGSN, y asociada al router de acceso a la internet pública. La figura 6 muestra la ubicación del nodo programable (etiquetado IWB en la figura) en la plataforma de pruebas.

4. Un prototipo de SARA

SARA [17] es un prototipo de nodo programable desarrollado en Java (y parcialmente C) para estudiar el paradigma *router*-asistente. Tal como se ha explicado anteriormente, el sistema es capaz de procesar transparentemente paquetes IP de señalización de las aplicaciones de red y cualquier flujo de paquetes de usuario que pasan por el router. Los primeros disparan la carga o refrescan el estado

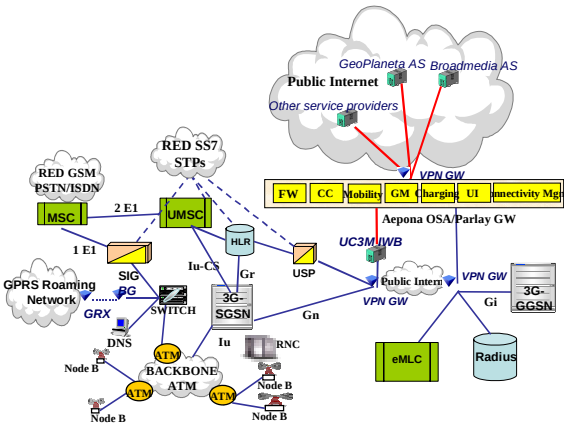

Figura 6. Escenario de uso de una pasarela OSA/Parlay para tarificación de un servicio de precarga web

de las aplicaciones, independientemente de su dirección destino. El router desvía los paquetes de señalización al asistente donde son analizados y procesados por las aplicaciones. Cada aplicación se ejecuta como un *thread* (nativo del S.O.) y puede hacer uso de las bibliotecas de análisis y modificación de sus cabeceras, extensión de JAVA (JNI/C para Linux) para dar servicio de *raw sockets*, vista del estado del router obtenida por SNMP, etc disponibles en el entorno.

Hoy en día hay dos posibles configuraciones. Ambas soportan IPv4 e IPv6, permitiendo a las aplicaciones activas un control total sobre los paquetes desviados. La primera está totalmente basada en Linux (cubriendo los dos papeles, como router y asistente) y la segunda es una plataforma híbrida donde se emplea un router Ericsson-Telebit AXI462 con un kernel modificado para interoperar con un PC asistente.

La experiencia preliminar con este prototipo indica que el techo de prestaciones de una aplicación es severo y viene impuesto por los cambios de contexto entre el kernel y el entorno de ejecución JAVA. Así una aplicación ejecutándose en la plataforma descrita en la figura 7, es capaz de analizar y procesar 2000 paquetes por segundo (Fig. 8), con tasas hasta 30 Mb/s, y con una penalización en retardo máximo de 2 ms por el desvío al asistente.

5. Conclusiones

La introducción de distintos grados de programabilidad en las redes de conmutación de paquetes está siendo objeto de investigación por parte de numerosos grupos de investigación, ya sea en forma de red activa, red programable o de procesadores de red. Como hemos presentado, uno de los más importantes campos de aplicación de esta tecnología puede ser la optimización de las comunicaciones en sistemas móviles. En este trabajo hemos presentado una arquitectura de red programable que incluye elementos que identificamos como esenciales para una aplicación práctica de esta tecnología en redes móviles: es una propuesta que permite una evolución progresiva a

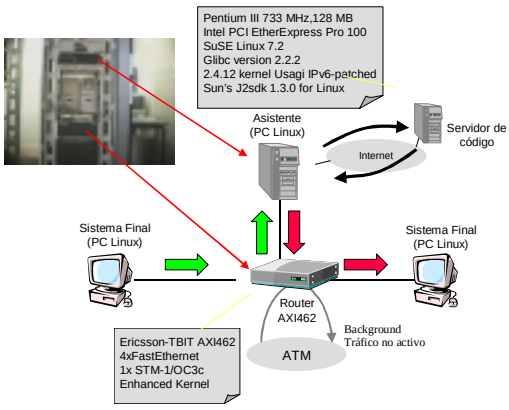

Figura 7. Banco de pruebas

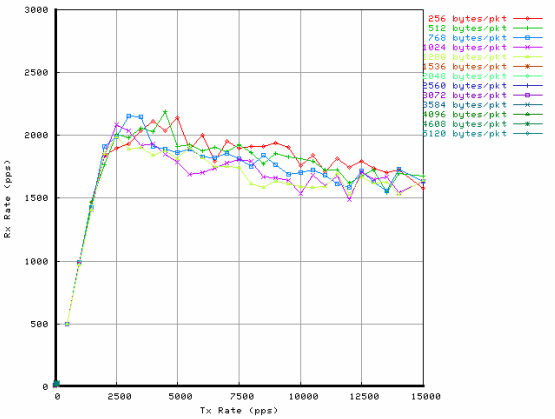

Figura 8. Tasa cursada vs tasa ofrecida (paquetes/s)

partir de routers convencionales, facilita la movilidad con su mecanismo de carga de código en nodos intermedios sin precisar su direccionamiento explícito, y presenta un compromiso razonable entre capacidad de control de los recursos del nodo y aislamiento de los procesos de tratamiento personalizado de paquetes. Además, la viabilidad de este enfoque se ha fundamentado con una arquitectura de seguridad que garantiza un uso controlado de las aplicaciones de red. Finalmente se han identificado escenarios de aplicación a implementar y se han descrito experiencias iniciales sobre una primera versión del prototipo. Queda por demostrar próximamente la viabilidad de algunas de las aplicaciones descritas sobre un escenario con tecnología UMTS real.

Referencias

[1] Proyecto MCYT AURAS. http://matrix.it.uc3m.es/~auras/.

[2] D. Wetherall, U. Legedza and J. Guttag, *Introducing new Internet services: Why and How*, IEEE Network Magazine, 1998.

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

[4] Konstantinos Psounis. *Active networks: Applications, security, safety, and architectures.* IEEE Communications Surveys, 2(1), Q1 1999.

[5] L. Peterson, Y. Gottlieb, M. Hibler, P. Tullmann, J. Lepreau, S. Schwab, H. Dandekar, A. Purtell and J. Hartman. *An OS Interface for Active Routers*. In IEEE Journal on Selected Areas in Communications, 2001.

[6] Andrew T. Campbell, Herman G. De Meer, Michael E. Kounavis, Kazuho Miki, John B. Vincente, and Daniel Villela. *A survey of programmable networks.* Computer Communication Review, 29(2):7-23, April 1999.

[7] S. Floyd and L. Daigle, "IAB Architectural and Policy Considerations for Open Pluggable Edge Services," *RFC 3238*, January 2002.

[8] IEEE P1520: Proposed IEEE Standard for Application Programming Interfaces for Networks. http://www.ieee-pin.org/

[9] IETF Forwarding and Control Element Separation http://www.ietf.org/html.charters/forces-charter.html.

[10] P. Cowley, M. Fiuczynski, J-L. Baer, and Bershad. *Characterizing processor architectures for programmable network interfaces.* In Proc. International Conference on Supercomputing, Santa Fe, 2000.

[11] A. Calveras, X. de Porrata. Utilización de Proxies para mejorar el rendimiento de comunicaciones móviles en Internet. X Jornadas Telecom I+D. Barcelona-Madrid. Noviembre 2000. .

[12] J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby, Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, *RFC 3135*. June 2001.

[13] V. Sunderam, J. Pascoe and G. Tonev. *Reconciling the Characteristics of Wired and Wireless Networks; The Janus Approach*. IEEE 4th Int. Workshop on Active Middleware Services. IEEE Computer Society, pp. 91-98, Edinburgh, Scotland. July 2002.

[14] Tschudin, C.; Lundgren, H.; Gulbrandsen, H. *Active routing for ad hoc networks* IEEE Communications Magazine, Volume: 38 Issue: 4 , pp: 122 -127Apr 2000.

[15] A. Millard. Tech.ical Report. *2.5G/3G wireless networks and the application of network processors.* http://www-3.ibm.com /chips/techlib/techlib.nsf/pages/main. August, 2002.

[16] Y-B Lin, A-C Pang, and Y-R. Haung and I. Chlamtac. *An All-IP approach for UMTS Third-Generation Mobile Networks*. IEEE Network, September/October 2002.

[17] D. Larrabeiti, M. Calderón, A. Azcorra and M. Urueña. *A practical approach to Network-based processing*. IEEE 4th International Workshop on Active Middleware Services. IEEE Computer Society, pp. 3-10. Edinburgh. July 2002.

[18] Marcelo Bagnulo, Bernardo Alarcos, María Calderón, Marifeli Sedano. "ROSA: Realistic Open Security Architecture for Active Networks". IWAN 2002, LNCS 2546, pp. 204-215. Zurich, Switzerland, December, 4-6 2002.

[19] Marcelo Bagnulo, Bernardo Alarcos, María Calderón, Marifeli Sedano. "Providing Authentication & Authorization Mechanisms for Active Service Charging". QofIS/ICQT 2002, LNCS 2511, pp. 337-456. Zurich, Switzerland, October 16-18, 2002.

[20] http://www.ist-opium.org/

[21] http://www.parlay.org/specs/