

# SISTEMA MULTIAGENTE PARA RADIOACCESO A REDES CORPORATIVAS

Javier Grande Gundín  
cibi7@retemail.es

David Sáenz Tagarro  
davidst@usuarios.retecal.es

José Ramón Villar Flecha  
diejvf@unileon.es

Ángel Alonso Álvarez  
dieaaa@unileon.es

Francisco J. Rodríguez Sedano  
diejrs@unileon.es

Área de Ingeniería de Sistemas y Automática,  
Dpto. de Ingeniería Eléctrica y Electrónica.  
Universidad de León,  
Campus de Vegazana s/n - 24071 León

## Resumen

*Ante el problema de la gestión del posicionamiento y comunicación entre agentes móviles en una red corporativa de amplia extensión, se plantea una solución basada en el modelo de funcionamiento del sistema APRS, que hace posible el envío de todo tipo de información y su gestión en una red, a través de radiofrecuencia y redes TCP/IP.*

**Palabras Clave:** Radiofrecuencia, redes, encapsulamiento, tunneling, agentes, alta disponibilidad, roaming, GPS, sistemas distribuidos, radiocontrol.

## 1 INTRODUCCIÓN

El proyecto se centra en el desarrollo de un sólido backbone de la red, a través de un sistema modular que permite la extensión y adaptación del sistema a gran variedad de situaciones y necesidades.

El proyecto se basa en la arquitectura APRS, contando así con un sistema jerárquico de servidores, cuyo núcleo central cuenta con hasta tres servidores primarios, a los que se pueden conectar tanto servidores secundarios, como clientes. (Vease figura 1)

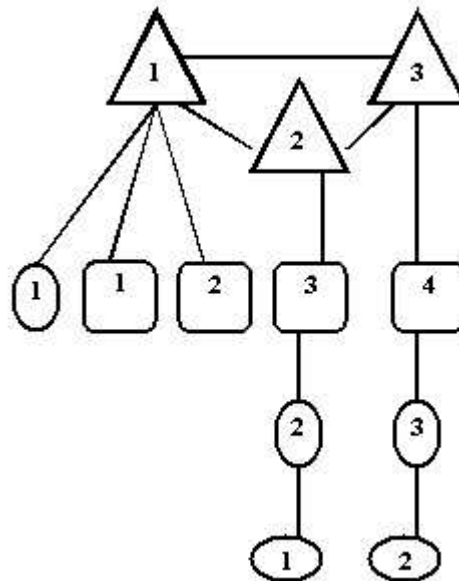


Figura 1

Los servidores están conectados entre sí a través de una red basada en TCP/IP, mientras que los clientes (agentes) son sistemas con las siguientes características:

- Sistemas con conexión directa TCP/IP
- Sistemas con conexión directa TCP/IP y un equipo de radiocomunicaciones y GPS
- Sistemas con un equipo de radiocomunicaciones y GPS

Así, cada ente con capacidad de radio emisión, transmite su posición adquirida a través de un sistema GPS al ente conectado a la red TCP/IP más cercano, y éste la propaga a través de su servidor.

Mediante esta técnica se proporciona la capacidad de envío de todo tipo de objetos en el campo “mensaje” de los paquetes, haciendo posible la transmisión de ficheros de cualquier tipo, paquetes de otros protocolos encapsulados, envío remoto de comandos (radiocontrol), e incluso la implementación de sistemas distribuidos, simplemente con utilizar la interfaz adecuada.

Los clientes pueden tener movilidad (roaming), ya que su situación es transmitida periódicamente y un deterioro severo en la calidad de la comunicación provocaría la reconexión con el ente con enlace de radio más cercano, siento esto totalmente transparente para el agente de cara al usuario

## 2 NOMENCLATURA

La especificación del protocolo establece una normativa para nombrar los entes del sistema que sigue el siguiente esquema:

<NombreServidorPrimario>.<NombreServidorSecundario>.<NombreClienteTCP/IP>\_<NombreClienteRadioFrecuencia>

Los nombres de los entes pueden contener cualquier carácter excepto “.” y “\_” que están reservados como separadores.

A través de esta nomenclatura se facilita que el módulo dedicado al encaminamiento de datos apoye sus algoritmos en un encaminamiento jerárquico. Ejemplo: figura 2.

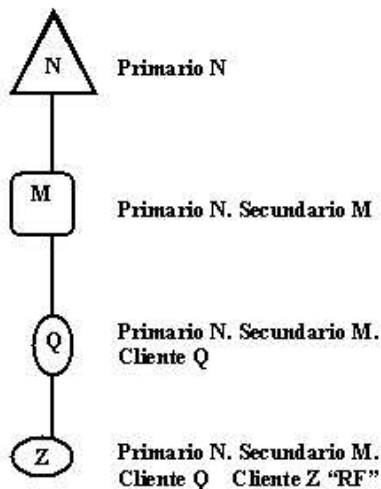


Figura 2

## 3 ENCAMINAMIENTO

El núcleo, tanto de los servidores como de los clientes, tiene un interfaz con un módulo de encaminamiento, que le debe proporcionar la información necesaria para hacer llegar las transmisiones a su destino.

Se ha desarrollado un módulo que realiza esta tarea mediante un algoritmo basado en estado del enlace y multidifusión, aprovechando los conocimientos con que se cuenta en cada momento sobre la estructura jerárquica de la red a partir de la nomenclatura de entes.

Al trabajarse en una red corporativa, el envío multicast a gran escala no debería suponer una barrera tecnológica, debido a la viabilidad de configuración de los dispositivos de red para retransmitir información multicast, con lo que se reduce tanto el número de envíos como la carga en la red frente a un módulo que enviase la información de encaminamiento directamente a cada nodo.

## 4 IDENTIFICACIÓN Y CARACTERIZACIÓN

Cada ente se identifica en el sistema de forma única mediante su nombre compuesto y su dirección IP y/o frecuencia de radio. El nombre se establece en cada reconexión con un servidor, manteniéndose la parte específica del mismo (<NombreCliente>) a la que se añaden las “cabeceras” necesarias, correspondientes con los servidores o cliente que tengan en el nivel superior.

Un dato a mayores para la identificación de un cliente determinado es su posición en coordenadas geográficas.

## 5 TRANSMISION Y COMUNICACIONES

El envío de datos se realiza mediante el protocolo de transporte TCP mientras que el módulo de encaminamiento utiliza UDP, provisto de segmentación independiente de datos en datagramas a nivel de aplicación.

Se está trabajando en la posibilidad de enviar datos a varios entes en el mismo envío (multidifusión) simplemente con incluir el nombre de los mismos en el campo destino, característica funcional en la próxima versión.

El envío en el nivel de radiofrecuencia puede realizarse mediante cualquier tecnología disponible, en función de las necesidades de la aplicación,

como packet radio (AX. 25), AMSAT, o telefonía móvil (GPRS...).

Para las radiocomunicaciones se cuenta con un interfaz que provee al entorno de independencia de la tecnología.

## 5.1 CONEXIÓN

Los servidores primarios se interconectan entre si formando un triángulo virtual, los servidores secundarios se conectan a un primario especificado en el inicio de la conexión y los clientes se conectan al servidor más cercano (primario o secundario) y con menor índice de ocupación automáticamente. Ahí es cuando se realiza la asignación del nombre compuesto al cliente.

De carecer de conexión a la red TCP/IP, los clientes se conectan por radiofrecuencia con el cliente más cercano con acceso a la red TCP/IP, que hace el papel de pasarela, de modo similar a los IGATE's de APRS.

## 5.2 TOLERANCIA A FALLOS

El entorno mantiene su funcionalidad con la existencia mínima de un solo servidor primario, por lo que al disponer de al menos tres servidores en el caso ideal que se recomienda en la especificación del protocolo, se puede caracterizar al sistema con alta disponibilidad.

En caso de haber un deterioro en las comunicaciones, tanto entre cliente – servidor como entre cliente – cliente de radiofrecuencia, se realiza una reconexión automática con reasignación de nombre, al servidor o cliente más cercano, según proceda. Esto plantea el problema de saber en cada momento (por ej.) el servidor al que está conectado un cliente, lo que se ha solucionado permitiendo enviar mensajes a un cliente indicando nada más su nombre específico.

En caso de conectarse a la red un cliente, si existiese un cliente con nombre igual, se le asignaría al cliente entrante un nuevo nombre consistente en el que proponía con un código alfanumérico aleatorio concatenado al final del mismo.

## 6 ESTADO DEL DESARROLLO

El desarrollo está realizado en el lenguaje de Sun Microsystems Java, y está en fase experimental, con plena funcionalidad en el nivel de la red TCP/IP debido a las limitaciones del proyecto para disponer de material para soportar tecnologías basadas en radiofrecuencia.

Se ha desarrollado un interfaz experimental para realizar los test y pruebas necesarios para garantizar el adecuado funcionamiento del sistema.

## 7 PRUEBAS EXPERIMENTALES

Utilizando el interfaz experimental anteriormente citado, y el módulo de encaminamiento basado en multicast, se han realizado mediciones en los tiempos de envío y recepción de paquetes, en una red Ethernet con conexión a un Switch y ancho de banda de 100Mbps y Full Duplex, en PCs x86 con sistema operativo Linux. No se ha podido probar la conexión con interfaces de radiofrecuencia debido a las limitaciones del proyecto.

Los test se han realizado con un tiempo medio de ping de 0,304 ms entre hosts sin tener en cuenta el tiempo empleado en la resolución ARP.

Se han contabilizado tiempos para desde el envío de un paquete con un mensaje de tipo `java.lang.String` con cuatro caracteres como contenido, hasta la recepción del acuse de recibo a nivel de aplicación, entre 2 ordenadores.

Se han realizado mediciones en las siguientes tres configuraciones:

- 1) Configuración mínima que consta de un servidor primario y dos clientes conectados al mismo. (Figura 3)

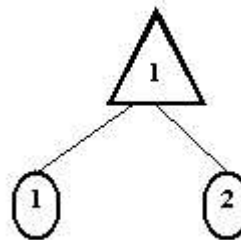


Figura 3

- 2) Configuración semióptima con dos clientes conectados a un servidor primario y un cliente conectado a otro primario. (Figura 4)

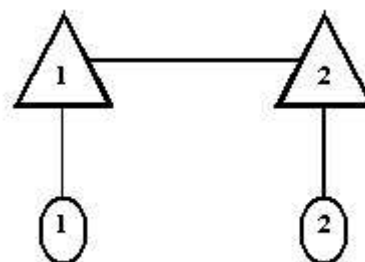


Figura 4

- 3) Configuración óptima basada en los tres servidores primarios conectados entre sí, un cliente directo a un primario y un secundario que cuenta con un cliente conectado a otro primario. (Figura 5)

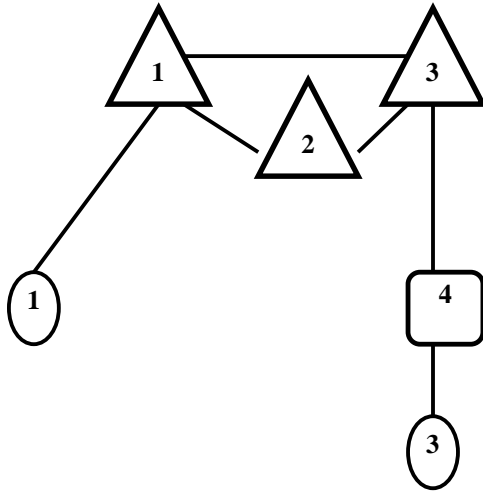


Figura 5

Los tiempos medios obtenidos son los correspondientes a la tabla 1:

Configuración	Tiempo medio hasta acuse de recibo (ms)
1	478
2	682
3	732

Así, mediante la siguiente ecuación deducimos una aproximación del tiempo de proceso en cada host:

$$T = Tr + Ne * Th$$

Donde:

T: Tiempo medio hasta el acuse de recibo [ms]  
 Tr: Tiempo del retardo en la red medido con la herramienta ping [ms]  
 Ne: Número de enlaces que atraviesa el mensaje en el sistema.  
 Th: Tiempo de proceso en 1 Host.

Así obtenemos que el tiempo medio de proceso de la información en 1 host es:

$$Th = 107ms$$

## 8 CONCLUSIONES

El proyecto ha resultado ser un complejo análisis de las soluciones de multitud de problemas que surgen

al enfrentarse al desarrollo de un sistema de esta envergadura.

La mayoría de los problemas han sido solucionados muy eficientemente, consiguiendo usos de CPU en Linux inferiores al 4% para un servidor primario.

El uso de memoria de un cliente no alcanza en ningún caso los 20MB teniendo en cuenta que se está contabilizando la máquina virtual de Java 1.4.2.

La carga en la red es despreciable en una red Ethernet de 100Mbps.

En el futuro están los objetivos de mejorar el algoritmo de enrutamiento para gestionar los cambios de estado (activo / inactivo) de los entes en tiempo real, y realizar un gestor de módulos (pluggins) uniformado para admitir fácilmente pluggins de terceras partes y facilitar el desarrollo de drivers para los dispositivos que sean necesarios en un sistema con funcionamiento real.

Ahora se está trabajando en la internacionalización de los interfaces públicos del núcleo, y la optimización de los clientes.

También se está trabajando en el área de las comunicaciones, optimizando los envíos para reducir el número de paquetes que se transmiten y su tamaño, lo cual fue una barrera tecnológica debido a un bug en el Java SDK utilizado en el desarrollo.

El gran tiempo necesario para el desarrollo y diseño del sistema, y la falta de medios, han dificultado el progreso, pero aun así, el resultado ha sido más que satisfactorio.

## 8.1 APLICACIONES

El sistema desarrollado en este proyecto cuenta con un amplio rango de aplicaciones, que se centran fundamentalmente alrededor de empresas de transporte y flotas de vehículos o embarcaciones de cualquier tipo, así como de dispositivos robotizados o personas que trabajen en lugares remotos y en general cualquier ente que puede ser radiocontrolado o del que se requiere información sobre él mismo, su entorno o su posición geográfica.

Un ejemplo sencillo de aplicación sería una empresa de transportes a nivel nacional que introduzca el hardware necesario en sus vehículos o dote de él a sus empleados, para saber en todo momento el estado de cada envío de cara a ofrecer mayor información a sus clientes y realizar una gestión más óptima de la eficiencia de la empresa.

El desarrollo en Java, facilita la migración e integración del cliente en un sistema empotrado con sistema operativo compatible. Sin ir más lejos, LynxOS, sistema operativo propiedad de la empresa Lynuxworks líder en sistemas operativos de tiempo real y empotrados de altas prestaciones,

cuenta con soporte Java compatible con Sun Microsystems.

### **Referencias**

- [1] D. L. Stevens, D. E. Comer, (1991), Internetworking with TCP/IP, vol1. Prentice-Hall.
- [2] D. L. Stevens, D. E. Comer, (1991), Internetworking with TCP/IP, vol2. Prentice-Hall.
- [3] D. L. Stevens, D. E. Comer, (1991), Internetworking with TCP/IP, vol3. Prentice-Hall.
- [4] George Coulouris, Jean Dollimore y Tim Kindberg, (2001), Sistemas Distribuidos. Conceptos y Diseño. Addison-Wesley
- [5] <http://www.ar1.org>
- [6] <http://www.aprs.org>
- [7] <http://www.xastir.org>
- [8] <http://www.linuxworks.com>

### **Agradecimientos**

Queremos expresar nuestro agradecimiento al personal implicado del Área de Ingeniería de Sistemas y Automática del Departamento de Ingeniería Eléctrica y Electrónica de la Universidad de León, por su tiempo, dedicación y apoyo por que saliera adelante este proyecto, y a nuestras familias por “sufrir” el trabajo llevado a casa, en especial a Isa que “sufrió” doble.