

Servicio de Distribución de Datos Meteorológicos

I. INTRODUCCIÓN

Durante la campaña del 2002-2003 en la base antártica Juan Carlos I, se ha desarrollado un sistema de visualización de los datos meteorológicos que adquiere la estación Campbell instalada en 2000. En el funcionamiento anterior a nuestra actuación, la estación almacenaba los datos cada diez minutos para posteriormente volcarlos a un PC conectado a la estación mediante un cable serie empleando el software de volcado y control del *datalogger* PC208W proporcionado por el fabricante. Dicho software, aunque muy potente, tiene las posibilidades de visualización condicionadas a la consola de un PC y por tanto la visualización de los datos se reduce a la estación. Con el nuevo diseño los datos son accesibles desde cualquier punto de la base empleando la red local.

La implantación de la red inalámbrica Wifi ha permitido la distribución de estos datos a cualquier punto de la base hasta un radio de 3.3 km. del punto de adquisición. Para la visualización se han empleado dos paneles LED rojos de 60 cm. de longitud total, con 14 caracteres de capacidad y altura de 6 cm. por carácter suministrados por ASEL S.L. ([1]) y un applet gráfico incrustado en página Web. Toda la programación, tanto de los paneles LED como de las applets gráficas, ha sido implementada en Java, siendo uno de los objetivos de este trabajo la evaluación de este lenguaje como herramienta para producir software de control de instrumentación siguiendo la filosofía de los nuevos entornos de medida distribuidos que estamos desarrollando dentro del programa LabVir ([3]).

II. DESCRIPCIÓN

A diferencia del sistema que actualmente se encuentra en funcionamiento en los buques oceanográficos españoles (SADO (Sistema de Adquisición de Datos Oceanográficos), [2]), que emplea para la distribución de datos el protocolo UDP, nuestro proyecto contempla mecanismos RMI (Remote Methode Invocation) implementados en Java para el transporte y acceso a objetos (datos, procesos, aplicaciones, etc.) por la red (Fig. 1).

Este desarrollo ofrece *a priori* mayores posibilidades de control remoto de las aplicaciones y acceso a datos

complejos, y de hecho constituye la primera prueba de la infraestructura de programación creada para la adquisición y el procesado de los datos en el proyecto que nos ocupa ([6], [5] y referencias en allí citadas).

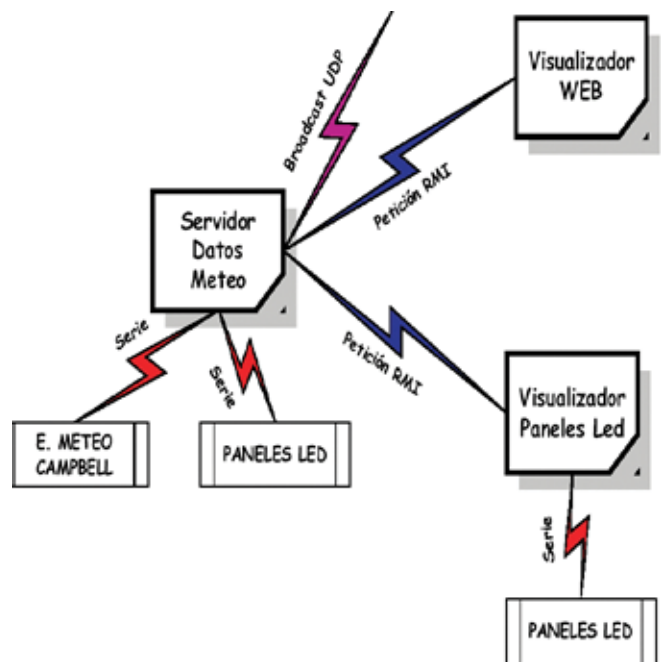


Fig. 1: Protocolos de transmisión de datos

La máquina virtual empleada en el desarrollo de la aplicación de visualización distribuida ha sido la 1.4.0.1 de SUN Microsystems. Entre las herramientas utilizadas cabe citar *Poseidón 1.5*, *Codeguide 5.0* y *Forte for Java 4.0 Community Edition*. Se han realizado tres aplicaciones.

- Un servidor de datos que controla y toma la información que el *datalogger* de la estación meteorológica guarda cada 10 minutos. El servidor procesa la información y la ofrece a la red como telegrama UDP *broadcast* y como servicio RMI.

- Un visualizador en pantalla de LEDs que interroga al servidor de datos pidiéndole los datos de la última lectura del *datalogger* y tras montar un texto de visualización lo muestra en un panel de LEDs.

- Un visualizador en página Web que interroga al servidor de datos pidiéndole los datos de la última lectura del *datalogger* y tras montar un texto de visualización y una imagen gráfica de la dirección y velocidad del viento, lo muestra en una página Web.

Los ficheros finales de los desarrollos en Java de este sistema se han almacenado en la máquina servidor. En ella se encuentran los archivos JAR y los scripts y páginas html que permiten su correcta ejecución.

A. El servidor de datos

La adquisición de datos meteorológicos y la puesta en la red de los mismos la realiza una aplicación Java empaquetada en un archivo JAR, cuya clase Java principal denominada «servidorMeteo.class» consulta al *datalogger* conectado al COM1 (/dev/ttyS0) cada 30 segundos obteniendo los datos almacenados en el último intervalo de diez minutos. La aplicación construye un objeto «meteo.class» con la información meteorológica, envía la expresión de este objeto en forma de cadena de caracteres al puerto UDP 12000 y al panel de LEDs conectado al puerto COM2 (/dev/ttyS1). La aplicación ofrece el objeto «meteo.class» bajo RMI como «servidor.bae» y como servicio «ServidorMeteoRMI». Los datos meteorológicos extraídos del *datalogger* son,

- Fecha
- Temperatura media
- Temperatura mínima
- Temperatura máxima
- Velocidad media del viento
- Velocidad máxima del viento
- Dirección media del viento
- Dispersión angular de la dirección del viento
- Humedad relativa
- Presión atmosférica
- Radiación solar

A ellos hemos añadido el parámetro de sensación de frío (*Windchill*), que nos da una idea aproximada de la sensación térmica, en este caso de frío, y que es de utilidad para los miembros de la base antártica que realizan su actividad al exterior. Para su cálculo se ha empleado una nueva formulación desarrollada por investigadores de la Universidad de Indiana Purdue y el Instituto de Medicina Ambiental y Protección Civil de Toronto en el año 2000 ([4]).

$$T_{sf} = 13.112 + 0.6215 T - 13.37 v^{0.16} + 0.3965 T v^{0.16}$$

donde T es la temperatura (en °C), v es la velocidad del viento (en km/h) y T_{sf} la temperatura de sensación (en °C).

La aplicación servidor de datos se ejecuta bajo un entorno *Linux Mandrake* al que se le ha instalado una máquina virtual Java (/usr/java/java2) complementada con los paquetes de clases «jdom.jar», que ofrece soporte para codificación XML, y RXTX.jar que ofrece control de puertos serie. Puesto

que la aplicación actúa como servidor RMI es necesario arrancar la aplicación /usr/java/java2/bin/rmiregistry que se encarga de hacer de puente y registrar los objetos que el servidor ofrece a las aplicaciones cliente.

Las clases de la aplicación se han empaquetado en un archivo JAR que facilita su carga en memoria y su distribución. La ejecución de la aplicación se realiza invocando al programa /usr/java/java2/bin/java con la opción «-jar» y parámetro el nombre y path completo del archivo JAR.

La aplicación manualmente se lanza desde una shell con un script de arranque localizado en /etc/init.d. El programa se comunica con el *datalogger* Campbell vía puerto serie /dev/ttyS0 a 9600, 8, n, 1, el cual responde a las teclas, Return, A, B, C, D y E devolviendo el *prompt* (*), referencia del *datalogger*, número del área de almacenamiento, año, día del año, hora, y datos de meteorológicos respectivamente.

El telegrama leído es interpretado para obtener la información meteorológica que simultáneamente se ofrece a la red y se muestra directamente en un panel de LEDs se controla conectado al puerto serie /dev/ttyS1 a enviando una cadena de bytes, con la información que queremos mostrar en el panel, el efecto de pantalla (inmediato, flash, invertido, etc.) y el identificador del panel destino. Hay que tener en cuenta que Java codifica internamente el tipo *short* usado para representar números enteros con sólo 2 bytes empleando el orden *BigEndian* mientras que los paneles tienen una representación *LittleEndian*. Hay pues que invertir el orden de los bits con la información del número de bytes del telegrama y *checksum* antes de enviarlo al panel por el puerto serie.

B. El visualizador de paneles LED

La visualización de los datos meteorológicos por paneles se realiza por una aplicación ejecutable en cualquier máquina a la que esté conectada un panel de LEDs en un puerto serie, con conexión a la LAN y a la que se le halla instalado una máquina virtual Java2 con las librerías «RXTX.jar» y «Jdom.jar».

La aplicación se ha desarrollado pensando en su ejecución bajo Linux. Dado que la nomenclatura de los puertos serie es distinta en cada uno de los sistemas operativos existentes en la base antártica (estaciones SUN, PC con Windows, PC con Linux y Mac), para aprovechar portabilidad del código que permite Java sería necesaria una ligera modificación en el código, pasando como parámetro a la aplicación el tipo de puerto serie, /dev/ttyS, COM1, etc..

En la campaña 2002-2003 se ha instalado una pantalla de LEDs para presentar los datos meteorológicos en el módulo de habitabilidad. Dicha pantalla está emplazada sobre la puerta del local Radio junto a la entrada de la cocina, es un enclave visible y próximo a un ordenador (Fig. 2 y Fig. 3).

La aplicación emplea la tecnología RMI para acceder

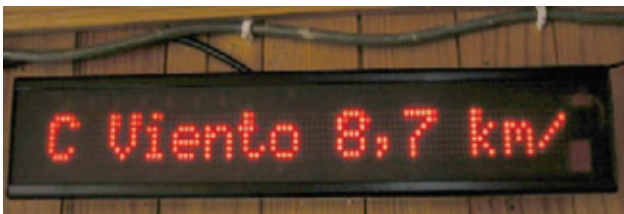


Fig. 2: Pantalla de LEDs



Fig. 3: Situación de uno de los paneles en la puerta del local Radio de la base antártica Juan Carlos I

directamente a los datos meteorológicos que se han generado en el ordenador, emplea las propiedades de representación alfanumérica de esos datos remotos y envía una cadena de caracteres con esa representación a un panel de LEDs conectado al puerto serie /dev/ttyS0. A diferencia del empleo de UDP utilizado en las aplicaciones SADO, la comunicación entre aplicaciones no se realiza con la transmisión por la red de una expresión en forma de cadenas de caracteres de unos datos, sino por el acceso directo a la representación lógica y formal de esos datos en el lugar en el que se han generado. Este planteamiento ofrece claras ventajas en cuanto a la escalabilidad de la aplicación no siendo necesario que la aplicación que pide los datos conozca la naturaleza exacta de la información que está pidiendo.

Obviamente estas consideraciones de diseño sobrepasan de alguna manera el sentido práctico de la aplicación y podríamos llegar a la conclusión que la tecnología RMI es excesiva para esta aplicación concreta que pudiera haberse resuelto también mediante tecnología UDP, pero lo cierto es que el esfuerzo necesario para su desarrollo ha sido realmente pequeño y constituye uno de los primeros bancos de pruebas de esta tecnología en entornos de adquisición de datos.

Al tratarse de una aplicación cliente RMI no es necesario ejecutar previamente al lanzamiento de la aplicación el «rmiregistry». Solamente es necesario que la aplicación conozca en que ordenador está el servicio que se demanda y bajo que nombre está registrado. El conocimiento del nombre del servidor puede ser un handicap si por algún motivo se cambia la máquina. Es por ello que se ha optado por pasar el nombre de la máquina como parámetro al ejecutar la aplicación.

La comunicación se establece desde una shell del ordenador servidor ejecutando un script en /etc/init.d/, el cual sólo produce mensajes de consola al inicio ya que todos datos y los mensajes de error se envían directamente al panel de LEDs.

Los errores que reporta la aplicación hacen referencia a la incapacidad de esta en establecer el enlace de red hacia la aplicación servidor de datos emplazado en el ordenador. Ante una situación como esta será necesario verificar las conexiones de red de ambos ordenadores y que la aplicación servidor de datos está ejecutando en el servidor.

C. El visualizador por página Web

La visualización de los datos meteorológicos por página Web se realiza por applets Java que se descargan directamente desde un servidor Web montado en la misma máquina en la que corre el servidor de datos.

El applet, diseñado como una aplicación gráfica de tres ventanas dispuestas en pestaña, muestra los valores numéricos de los datos de estación meteorológica y realiza un diagrama polar en el que se muestra la dirección del viento medio y su rango de variación (Fig. 5 y Fig. 4). Aunque los datos se refrescan en el servidor cada diez minutos (intervalo de almacenamiento del *datalogger*) el applet hace peticiones al servidor de datos cada diez segundos a fin de detectar con la mayor brevedad cualquier fallo del sistema.

El applet «AppletMeteo.class» es de hecho una aplicación cliente RMI con la misma estructura que el visualizador de datos de paneles descrito en el apartado anterior. En este caso la tecnología RMI resuelve las limitaciones que imponen los applets Java en tema de seguridad cuando se realizan operaciones de red. Si la aplicación se hubiera planteado como un applet que escuchara un *broadcast* de UDP, al ser ejecutada desde una página Web lanzaría una excepción de seguridad puesto que los applets sin certificado de seguridad no pueden acceder puertos de red que procedan de direcciones de red distintas a

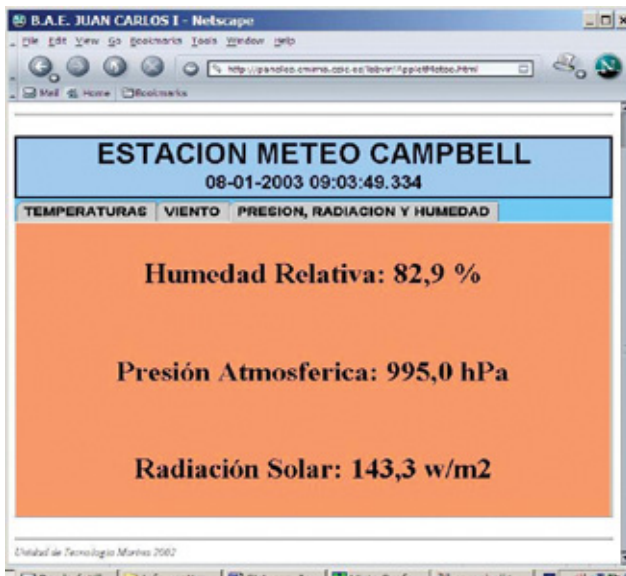


Fig. 4: *AppletMeteo.class* mostrando la humedad relativa, la presión atmosférica y la radiación solar medias de los últimos 10 minutos

la del servidor del que se han descargado. En el caso de los *broadcast* de UDP, esta condición siempre se da, pero empleando RMI podemos descargar un applet que conecte a un puerto IP del mismo ordenador del que ha sido descargado sin violar ninguna restricción de seguridad. Es por esta razón que en el ordenador «paneles.bae» se ha montado un servidor Web desde el que se descarga el applet «AppletMeteo.class». Una vía alternativa para no montar un Web server en el ordenador sería la construcción de un servlet instalado en el servidor principal de Intranet (servidor) que hiciera de puente entre las aplicaciones cliente y servidor. En un principio se desestimó esta solución por ser demasiado compleja para las dimensiones reales del proyecto.

III. CONCLUSIONES

Durante la campaña antártica del 2002-2003, se ha desarrollado e implementado un sistema de visualización de los datos meteorológicos en la base Juan Carlos I. Los datos de la estación meteorológica se distribuyen a cualquier punto de la base empleando la red de área local y la nueva red inalámbrica Wifi. Para la visualización se han empleado dos paneles LEDs rojos y un applet gráfico incrustado en página Web.

Este desarrollo ha sido un ensayo del empleo de Java, y las tecnologías de red asociadas, en el desarrollo de un entorno de distribuido de adquisición de datos, así como el ensayo de las estructuras de datos y jerarquía de clases ideadas para este proyecto. La aplicación de esta tecnología a cubierto las expectativas en cuanto a requerimientos específicos de tiempo y facilidad de programación.

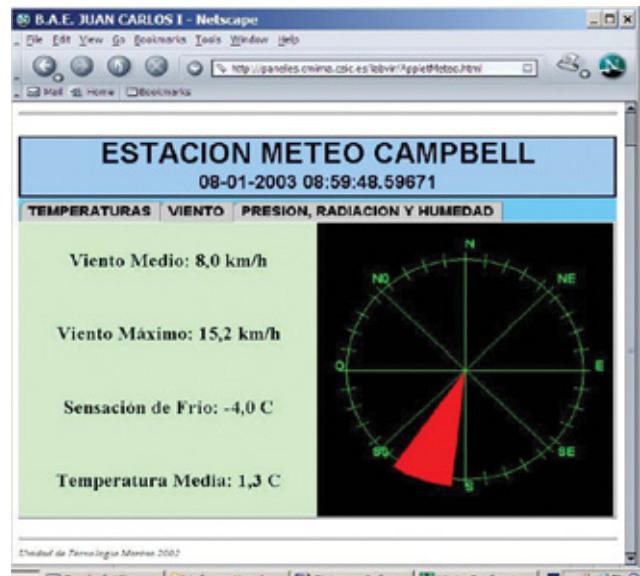


Fig. 5: *AppletMeteo.class* mostrando la velocidad del viento máxima y media, su dirección y variación en los últimos 10 minutos

Ante la expectativa de la disminución de los costes asociados a las comunicaciones vía satélite y a la mejora de los sistemas de energías alternativas, nuestro desarrollo permitirá el acceso a la información de la estación meteorológica y al control remoto de la instrumentación durante el largo invierno antártico en que la estación está deshabilitada, sin ninguna modificación.

Adicionalmente se ha realizado un control de las pantallas LEDs mediante Java abriendo las posibilidades de empleo de estos dispositivos en entornos de red.

El empleo de ordenadores de pequeño tamaño, reducido consumo y bajo coste, como son los PC embebidos tipo Tini, facilitarán la integración final del sistema de paneles a la red sin necesidad de dedicar un PC al control de estos.

REFERENCES

- [1] ASEL S.L. <http://www.asel.es>
- [2] R. Boza. *Sistema de Adquisición de Datos Oceanográficos (SADO)*, UGBO-CSIC-Inf., Marzo 2000
- [3] LabVir *Laboratorio Virtual Distribuido para la Adquisición y el Procesado de Datos Oceanográficos*. <http://arda.eupvg.upc.es>
- [4] NOAA *National Weather Service Plans to Implement a New Windchill Temperature Index*. <http://www.crh.noaa.gov>
- [5] J. Sorribas, E. Trullols, J. del Rio, R. Morillas, C. Samitier, A. Manuel. *Análisis y especificación de un sistema de adquisición de datos oceanográficos*, SAAE102., Septiembre 2002
- [6] E. Trullols, C. Samitier, J. Sorribas, A. Manuel, R. Morillas, J. del Rio. *CORBA for Distributed Measurements*, SCI02, Julio 2002