

Comunicación P-4

CAMBIO DEL SISTEMA DE COMUNICACIONES ENTRE EL CENTRO NACIONAL RADAR Y EL FUJITSU

Miguel Ángel Martínez Rubio
Fernando Aguado

Servicio de Teledetección (INM)

RESUMEN

Se ha migrado la aplicación informática del Centro Nacional de Radar desde el viejo VAX-785 a un nuevo ordenador VAX-4500 mucho más potente y compacto. Un aspecto interesante es el relativo a los programas de gestión de las comunicaciones SIRAM-SAIDAS que se han adaptado a TCP/IP. Para realizar una interfaz de protocolos que permita una comunicación entre los procesos de ambas aplicaciones se optó por utilizar «sockets». El sistema de comunicaciones resultante es compacto y estándar pudiendo ser migrado fácilmente a otras máquinas y construir sobre él un Sistema Distribuido.

1. Introducción

El sistema de intercambio de productos entre el Sistema de la Red de Radares (SIRAM) y el McIDAS se basa en la utilización de un *hardware* y *software* específico denominado KDX. Debido a esto, este intercambio tiene lugar únicamente entre los dos ordenadores centrales situados en Madrid, siendo necesario realizar la distribución y visualización de los productos en cada uno de los sistemas de forma redundante. El sistema puede verse en la Fig. 1.

La necesidad del sistema KDX se debe a que los ordenadores VAX-785 de DIGITAL utilizan para comunicaciones únicamente protocolo DECNET

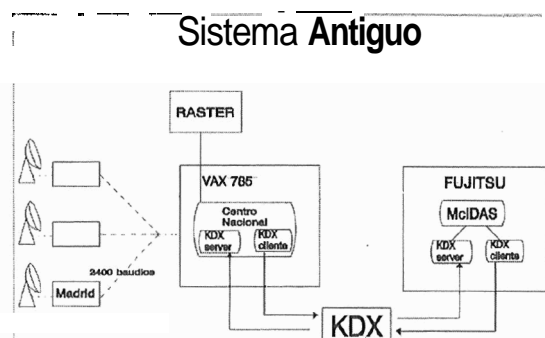


Fig. 1. Esquema de comunicaciones SIRAM-McIDAS utilizando sistema KDX

(propietario de la casa DIGITAL), mientras que el FUJITSU utiliza TCP/IP. La llegada de los nuevos ordenadores DIGITAL VAX-4500 para el Centro Nacional que incorporan además del protocolo DECNET el protocolo TCP/IP, y la desaparición del FUJITSU en el plazo de un año obliga a cambiar este sistema anticuado de comunicaciones. Para realizar este cambio se optó por mantener la estructura de los programas antiguos y sustituir las rutinas de comunicaciones utilizando *sockets* ya que los programas antiguos tenían una estructura cliente-servidor.

2. Arquitectura cliente-servidor

El uso de interfaces de programas de aplicaciones (APIs) que permitan la comunicación de máquinas heterogéneas es lo que permite que un sistema sea calificado como abierto. La arquitectura cliente-servidor surgió de la necesidad de definir un *interfaz* de protocolos, mediante la cual se definen servidores pasivos y clientes activos que crean conexiones hacia los servidores. El diálogo entre cliente y servidor siempre se inicia a petición del primero, teniendo en cuenta que a un mismo servidor pueden acceder varios clientes simultáneamente. Aunque generalmente la arquitectura cliente-servidor se emplea en *entornos* distribuidos no tienen por qué ser sinónimos, ya que tanto los clientes como los servidores pueden convivir en un único ordenador.

2.1. Sockets

Los *sockets* son uno de los muchos *interfaces* de programación de aplicaciones (APIs) a los protocolos de comunicaciones. La abstracción de *sockets* se encuentra presente en multitud de sistemas operativos. Un *socket* se puede abrir y cerrar, y sobre él se puede leer o escribir como si se tratara de un fichero. De esta forma un programa tratará un *socket* de igual forma que un fichero, con la diferencia de que lo que en él escriba será enviado a otro proceso (en su propia máquina o en cualquier otra de la red) y lo que de él lea estará generado por ese proceso interlocutor. La diferencia entre un descriptor de fichero y un descriptor de *socket* consiste en que el sistema operativo enlaza el primero con un fichero mientras que en el segundo crea una referencia a *socket*. De esta forma, se permite al programa que utilice el mismo socket para comunicarse con diferentes destinos. La integración entre los descriptors de *socket* y fichero es tan grande que el S.O. los localiza en el mismo conjunto. Cuando un programa desea crear un *socket* lo solicita al S.O. a través de una llamada al sistema y éste devuelve un número entero de referencia al *socket* de forma análoga a lo que ocurre durante la apertura de un fichero.

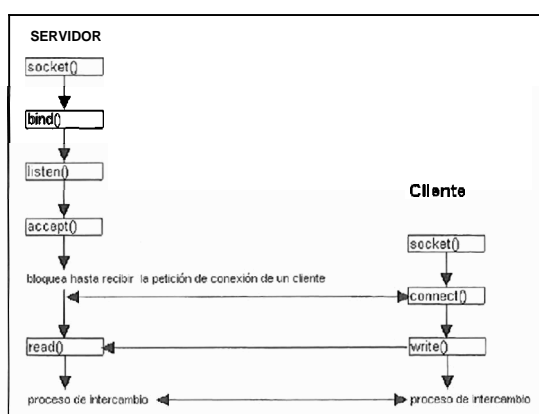


Fig. 2. Esquema cliente-servidor utilizando «socket»

La creación de un *socket* no supone la asociación a ninguna dirección ni de origen ni de destino. Desde el punto de vista del servidor será necesario disponer de una dirección local en la que aceptar la petición. En nuestro caso al usar TCP/IP, una dirección IP más un número de puerto (por ejemplo 25 para SMTP (correo electrónico), 80 para HTTP (usado en el WWW en los navegadores de INTERNET)). La asignación de la dirección local se realiza a través de la llamada al sistema *bind* (Fig. 2).

La llamada al sistema *accept* suspende la ejecución del programa hasta que recibe una petición de algún cliente. La recepción de datos a través del *socket* se realiza mediante la llamada al sistema *read* (análogamente para el envío de los datos se dispone de *write*).

Desde el punto de vista del cliente, una vez creado el *socket*, será necesario realizar las peticiones a una dirección destino determinada (dirección IP y puerto) en la que se encontrará esperando el servidor. La asignación de la dirección destino se realiza a través de la llamada *connect* que inicia la conexión. El envío de datos a través del *socket* se realiza mediante la llamada al sistema *write*. Es necesario sincronizar los envíos y recepciones entre el cliente y el servidor.

3. Sistema nuevo

Las rutinas de comunicaciones que se encargaban del intercambio entre el SIRAM y el McIDAS a través del KDX se sustituyeron por otras rutinas FORTRAN que hacen las llamadas al sistema *socket* en el FUJITSU y por las llamadas al sistema *socket* en el VAX-4500. Este cambio presenta las siguientes ventajas:

- Aumenta la velocidad de transmisión.
- No es necesario *hardware* especial.
- Soporta estructura cliente-servidor mediante *sockets* (TCP/IP).
 - Es portable (fácilmente actualizable y generalizable).
 - Facilita el mantenimiento, y ocupa poco volumen.

Aunque tiene los siguientes inconvenientes:

- No permite usar las terminales *Raster*.
- Aún no es un sistema distribuido.

4. Conclusiones

Cada vez surge con más fuerza la necesidad de integrar información procedente de distintos sistemas de observación, para obtener productos finales más adecuados a las necesidades reales de los usuarios, prueba de ello es el producto desarrollado en nuestro servicio «RAINSAT» que combina información procedente de los radares y del satélite METEOSAT.

Con este espíritu de integración surgió la necesidad de crear un canal de comunicaciones entre los dos sistemas de explotación del servicio, el SIRAM y el McIDAS, apoyándonos en las ayudas disponibles *on-line* en el VAX y en la SUN.

Posteriormente, tomamos conciencia de que este sencillo *socket* se había convertido en un canal de intercambio de datos que abría numerosas posibilidades a todo tipo de intercambio entre ambos sistemas. Y además, sencillas implementaciones permitirán la ingestión tanto en el SIRAM como en el McIDAS de cualquier otro sistema de observación.

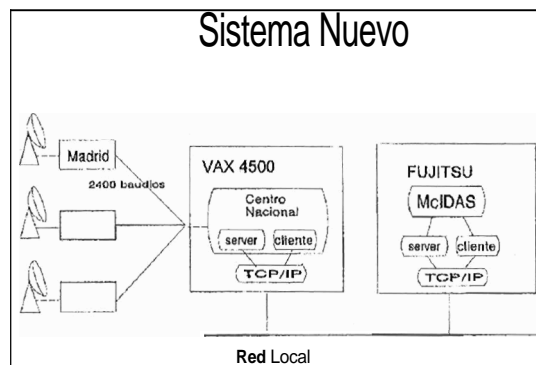


Fig. 3. Esquema de comunicaciones SIRAM-McIDAS utilizando TCP/IP mediante «socket»