Universidade Federal de Santa Catarina | www.periodicos.ufsc.br ISSNe 2175-8018
IJIE
IBEROAMERICAN JOURNAL OF INDUSTRIAL ENGINEERING
REVISTA IBEROAMERICANA DE ENGENHARIA INDUSTRIAL
REVISTA IBEROAMERICANA DE INGENIARÍA INDUSTRIAL

SISTEMA DE COMUNICACION Y TRANSMISIÓN DE DATOS DESDE ESTACIONES METEOROLÓGICAS

Ing. David Schvartzman[*]
Ing. Jaime Julio Saavedra[**]
Prof. Jorge Andrés Molina[***]
Prof. Delia Cohenca[****]
Prof. Fernando Pio Barios[*****]

RESUMEN: En el presente trabajo se desarrolla un programa (en lenguaje C) simple, flexible e inteligente para obtener variables meteorológicas de estaciones remotas y transmitirlas con eficiencia y seguridad a un servidor central encargado de almacenarlas en una base de datos, a través de cualquier tipo de redes. Los resultados obtenidos con respecto al conjunto de programas cliente/servidor desarrollados muestran que estos funcionan adecuadamente permitiendo efectuar la transmisión de datos meteorológicos entre un cliente remoto y el servidor de una manera segura y confiable. También se verificó que los programas facilitan la extracción de los datos meteorológicos de forma ordenada y segura a través de las técnicas de encriptación y detección de errores CRC16.

Palabras clave: Transmisión de datos. Variables climáticas. Bases de datos. Redes. Lenguaje C.

1 INTRODUCCIÓN

El querer tener al alcance de la mano la información ha sido siempre un problema y hoy en día se ha convertido en casi una necesidad. Muchas veces la información se encuentra a grandes distancias y no siempre se dispone del tiempo para esperar a que sea transportada hasta donde se requiere.

En la actualidad, la mayor parte del conocimiento que se tiene del comportamiento

[*] Ingeniero Electrónico , Universidad Nacional de Asunción, Laboratorio de Mecánica y Energía, Facultad de Ingeniería, david.schvartzman@gmail.com
[**] Ingeniero Electrónico, Universidad Nacional de Asunción, Laboratorio de Mecánica y Energía, Facultad de Ingeniería, jsaavedra@intercomingenieria.com.py
[***] Doctor en Física por el Centro Brasileiro de Pesquisas Físicas, Brasil, Universidad Nacional de Asunción Laboratorio de Mecánica y Energía, Facultad de Ingeniería, jmolina@ing.una.py
[****] Magíster en Energía para el Desarrollo Sostenible, Universidad Nacional de Asunción, Laboratorio de, Mecánica y Energía, Facultad de Ingeniería, deliac@ing.una.py
[*****] Licenciado en Física , Universidad Nacional de Asunción, Laboratorio de Mecánica y Energía, Facultad de Ingeniería, fbarrios@ing.una.py

climático procede de las mediciones realizadas con distintas estaciones meteorológicas, las que sirven para suministrar los datos necesarios para las simulaciones numéricas, que son las que, en definitiva, van a dar la predicción final sobre el sistema climático.

Debido a la creciente generación de tecnologías que aprovechan las energías renovables, nace la necesidad de obtener la información de las variables climatológicas (radiación solar, dirección y velocidad del viento, temperatura, humedad, precipitación, etc.) en tiempo real y, además, obtener el histórico, ya sea para optimizar el funcionamiento de aparatos o determinar qué regiones poseen potenciales para sustituir la energía convencional por energías renovables y limpias (TESTER et al., 2005).

Es sabido que el problema de la energía en el mundo está atravesando una fase crítica debido al aumento del consumo mundial, en particular en los países "emergentes", y también debido al carácter limitado de los recursos de combustibles fósiles y el agotamiento progresivo de ellos. Hace más de una década que tanto los países desarrollados como los que están en vías de desarrollo, se interesan por fuentes alternativas de energía para mantener el nivel de desarrollo y satisfacer la demanda cada vez más exigente de los sectores industriales, el transporte y los sectores terciarios.

Ante estos problemas, se plantea entonces la siguiente pregunta: de entre todas las fuentes de energía disponibles en una región particular (solar, eólica, hidráulica, etc.), ¿cuál sería la solución alternativa, con mayor eficiencia y menor impacto ambiental, a implementar en dicha región?

Esta pregunta se responde diseñando un sistema capaz de entregar información precisa, actualizada y libre de errores sobre la magnitud de las variables meteorológicas en distintas zonas de nuestro país a lo largo del tiempo. Para esto, ponemos en comunicación todas las estaciones meteorológicas a través de los programas desarrollados en el presente trabajo y conformamos en un servidor central una base de datos en formato SQL, la cual almacena el histórico de variables medidas en las distintas estaciones remotas.

2 DESARROLLO

En esta sección el problema se plantea de manera cualitativa y se describe la propuesta para solucionarlo, mediante el diseño de un sistema de comunicación que involucra la instalación de equipos de comunicación y la implementación de un conjunto de programas para hacer a la comunicación de estas estaciones meteorológicas con el servidor central, transmitiendo eficientemente la información a través de dichos equipos.

2.1 Formulación del problema

Centrados en la necesidad, de obtener las variables climatológicas en tiempo real o histórico, se ha desarrollado el presente proyecto, en el cual se ha planteado el problema de comunicar y transmitir datos entre dos estaciones meteorológicas.

Los datos correspondientes a las variables censadas en la estación remota deben ser enviados a un servidor de manera a que los mismos puedan ser publicados en forma grafica o tablas, para cualquier usuario que requiera utilizar los datos en tiempo real o histórico.

Mediante la comparación de los sistemas de comunicación a diseñar, se podría instalar una central meteorológica de varias maneras, eligiendo para cada caso en particular la mejor opción desde el punto de vista de la ingeniería. Y usarla en cualquier sitio del país permitiendo obtener mediciones precisas y actualizadas de las variables climáticas, siendo todos los datos transportados por la red de telefonía celular existente GSM (*Global System Mobile*), o bien mediante enlaces dedicados de ondas radioeléctricas, a un punto central (Facultad de Ingeniería – UNA).

Y así conformar una base de datos sobre todas las mediciones para crear estadísticas climáticas, prever futuros desastres naturales, y poder informar a cualquier interesado sobre las variables climáticas en tiempo real, con una información fiable y libre de errores.

2.2 Propuesta de ingeniería de diseño

En la Figura 1, se presenta el diseño de la red actualmente implementada y funcional, a la cual se hace referencia en la presente investigación.

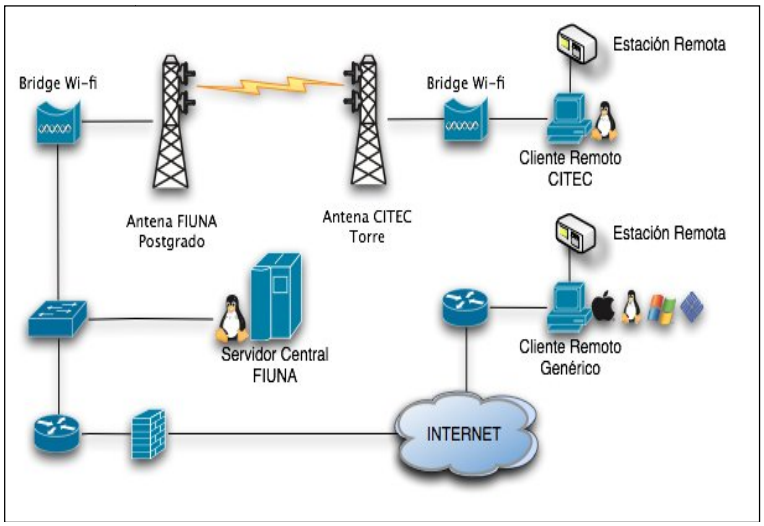

Figura 1 – Diagrama de Red
Fuente: Elaborado por el autor

2.3 Sistema de transmisión propuesto e implementado

Dentro del marco del presente trabajo, se realizó un enlace que une las dos sedes de la Facultad de Ingeniería. La sede ubicada en el campus de la Universidad Nacional de Asunción (FIUNA) y la sede ubicada en Isla Bogado, Luque (CITEC). En la Figura 2 se observa una vista superior de los puntos enlazados.

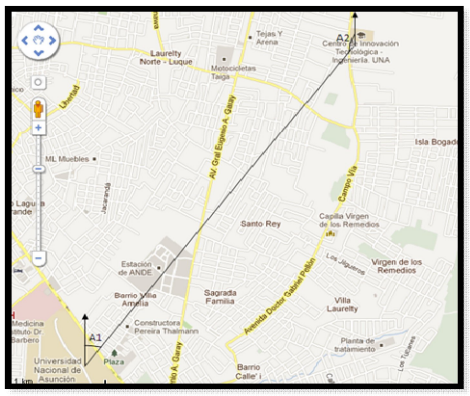

Figura 2 - Enlace FIUNA – CITEC
Fuente: http://maps.google.es/

Para ello, primeramente se realiza el siguiente cálculo:

Distancia geográfica entre dos puntos sobre la superficie terrestre (Ecuación 1):

$$d_{PP'} = \frac{40000}{360}\arccos\left[\begin{array}{l}\cos\varphi\cos\varphi'+\\ sen\varphi sen\varphi'\cos(\lambda-\lambda')\end{array}\right] \quad (Km) \qquad (1)$$

Donde: $d_{PP'} = 5{,}06Km$

Este es el punto de partida para decidir el tipo de equipos que finalmente se utilizarían para este enlace (CARDAMA et al., 2002).

El equipo que mejor se adapta a los requerimientos para realizar un puente de capa 2 (*bridge*) y que abarca estas distancias es el *Nanobridge* M5, de la marca *Ubiquiti* (USA). La robustez y el protocolo que utilizan (*AirMax*, MIMO 2x2 TDMA) son algunos motivos que hacen a estos equipos los más adecuados. Con el AirMax habilitado se consigue un CCQ del 100%, lo que quiere decir que no hay pérdidas en la transmisión (sin el *AirMax* el CCQ fue de como máximo 80%).

Estas antenas poseen buena ganancia (22 dBi), son fáciles de instalar, de tamaño pequeño, amigable interfaz de configuración y poseen gran variedad de opciones para los modos de funcionamiento como dispositivo de red. En el enlace realizado en esta investigación una de las antenas trabaja como *Access Point* WDS (*Wireless Distribution*

*Syste*m) y la otra en el modo *Station* WDS. Así, la primera actúa como estación maestra y la otra como esclavo (TOMASI, 2003).

La antena instalada en la FIUNA se encuentra ubicada sobre el edificio de postgrado, a una altura relativa al terreno de 27 metros. Mediante la siguiente formula se realiza el cálculo de altura de la antena ubicada en el CITEC (Ecuación 2):

$$azimut = arcsen\left[\frac{sen\varphi.sen(\lambda - \lambda')}{sen\left[\arccos\left[\cos\varphi.\cos\varphi' + sen\varphi.sen\varphi'\cos(\lambda - \lambda')\right]\right]}\right] \quad (2)$$

Donde: $h_{a2} = 28.22[m]$

El cálculo del azimut se realiza mediante la siguiente ecuación:
Valorando, para la FIUNA tenemos: azimut = 33º 8’ 6.06’’
Valorando, para el CITEC tenemos: azimut = 213º 8’ 6.05’’

De acuerdo a los resultados obtenidos y de manera que no haya ninguna posibilidad de que el enlace tenga obstrucción (TOMASI, 2003), se optó por colocar la antena del CITEC a los 50 metros; una altura equivalente a casi el doble de la altura calculada teóricamente. Esto puede observarse en la Figura 3 (der).

Finalmente, se obtuvieron resultados satisfactorios. Se aclara que estos cálculos teóricos fueron respaldados Radio Mobile, una herramienta ampliamente utilizada para realizar cálculos de enlace.

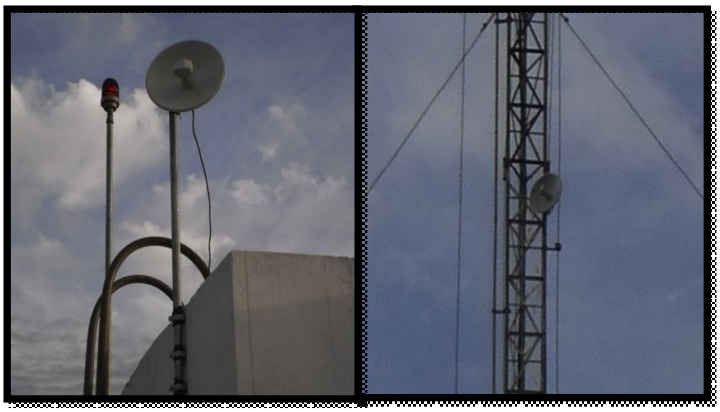

Figura 1 – Antenas instaladas: FIUNA (izq.), CITEC (der.)
Fuente: Elaborado por el autor

En esta sección se ha desarrollado un modelo del sistema propuesto. Teniendo en cuenta las ecuaciones (1) y (2) y el diseño general de la red mostrada en la Figura 1 se contempla una la primera parte, de lo que el presente trabajo pretende demostrar. Interprentando el problema planteado anteriormente y la solución mostrada aquí, se resuelve la transmisión física de datos desde distintos puntos a una ubicación central donde se almacenan ordenadamente los mismos.

2.3 Sistema de programas propuestos e implementados

En la presente sección se describirán los programas desarrollados para conseguir una transmisión eficiente, robusta, segura, y que permiten a traves de los medios físicos de comunicación obtener los resultados deseados. En la sección A se decribirán de forma cualitativa las carateristicas del programa general desarrollado, estudiando inclusive su diagrama de flujo. En B, se explicará la implementación del programa servidor desarrollado, encargado de recibir información meteorológica de multiples fuentes y almacenarla ordenadamente. En la sección C se hace referencia al diseño del programa cliente, cuyo objetivo fundamental es obtener información meteorolófica de la estación y transmitirla con ciertos criterios de seguridad y robustez al programa servidor. Por último en la sección D se describirá como los programas de las secciónes B y C interactúan para conformar un sistema seguro, a la vez flexible y confiable para producir el efecto deseado.

2.3.1 Obtención de datos de estaciones *Davis Vantage Pro/Pro2/Vue*

La sub-función más importante incluida en el código fuente desarrollado, y que hace a la comunicación con la estación meteorológica (Figura 4), es la sub-función estacion().

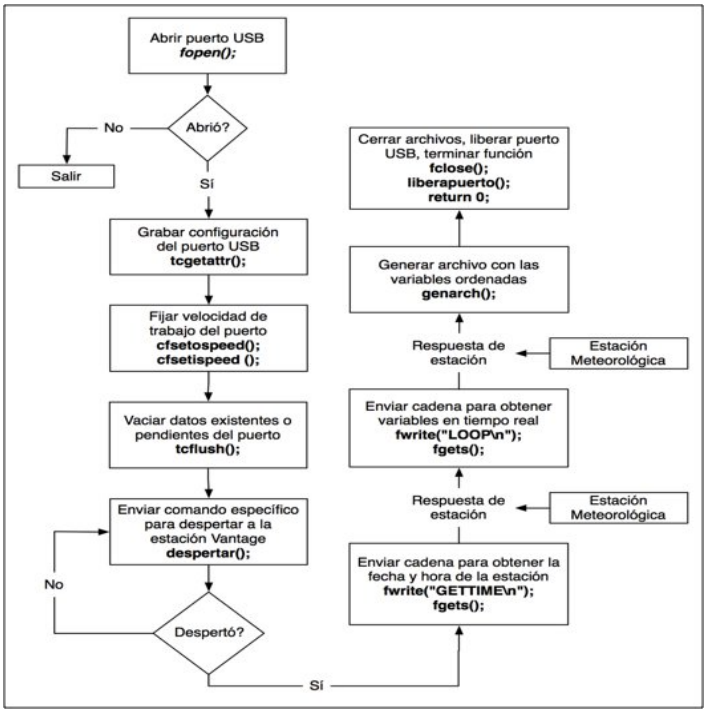

Figura 4 – Diagrama de flujo de la función *estación()*

Fuente: Elaborado por el autor

Mediante un llamado a esta, el programa envía una cadena de caracteres, denominada *Davis LOOP string*, y la estación meteorológica responde con todas las variables medidas en ese instante de tiempo. El puerto utilizado para esta comunicación es el USB (*Universal Serial Bus*). La función estación () genera un archivo en el cual se almacenan estos datos (un archivo utilizado como buffer local) con un formato específico. Una vez que se genera el archivo, el programa cliente lo abre y lo envía ordenada y secuencialmente al servidor.

Cabe resaltar que la función solo mantiene la comunicación con la estación por uno o dos segundos, durante la solicitud de los datos en tiempo real; una vez que la estación meteorológica los envía, se corta la comunicación y se libera el puerto USB, de manera que otros programas podrían estar trabajando paralelamente con la estación meteorológica sin problemas.

2.3.2 Programa servidor

El programa servidor desarrollado se instala en la computadora que hará de servidor de base de datos. Esta computadora recibirá constantemente solicitudes de los clientes, que van conectados a las estaciones meteorológicas remotas y transmiten las variables obtenidas de la

estación en tiempo real; el programa servidor almacena estas variables con un formato específico determinado por el lenguaje MySQL, para luego ser cargadas en una base de datos SQL. Es importante recordar que el programa servidor requiere que la interfaz de red que estará utilizando posea asignada una dirección IP estática, de manera que los clientes remotos puedan conectarse automáticamente a esta dirección IP con seguridad de que el programa servidor estará esperando para establecer la conexión. Otro parámetro sumamente importante a tener en cuenta respecto al programa servidor es el número de puerto, en el cual estará recibiendo las solicitudes: este número debe ser fijo, no pertenecer al conjunto de puertos bien conocidos y no debe haber otra aplicación escuchando al mismo puerto (HALL, 2009).

El programa servidor desarrollado posee varias cualidades importantes que se describen a continuación:

- Autenticación de usuarios

La seguridad del programa servidor es fundamental debido a que el servidor posee una dirección IP estática y un puerto fijo en el que escucha conexiones de clientes remotos. Se debe autenticar la identidad de cada cliente que trate de conectarse al servidor, ya que en caso contrario un usuario malintencionado podría conectarse al mismo y realizar acciones que puedan perjudicar al sistema (STEVENS; FENNER; RUDOFF, 2003), incluso con la eliminación total de las bases de datos globales acumuladas. Por esta razón el servidor escucha inicialmente solo segmentos del protocolo UDP (*User Datagram Protocol*) y cuando recibe de algún cliente dos mensajes consecutivos, correspondientes exactamente a un usuario y contraseña válidos que este posee registrados, entonces solicita a dicho cliente establecer una conexión segura TCP (*Transmission Control Protocol*) para iniciar la transmisión de datos. Esto se puede apreciar en la Figura 5. El registro de usuarios y contraseñas correspondientes almacenado en el servidor puede ser modificado directamente en el código fuente del programa.

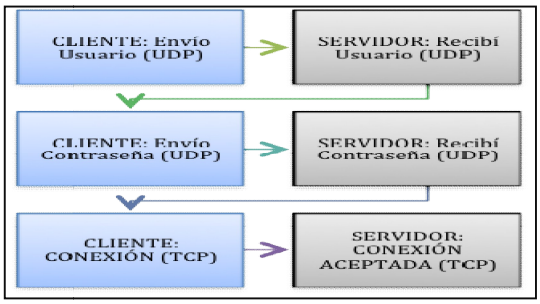

Figura 5 – Autenticacion de usuarios
Fuente: Elaborado por el autor

- Encriptación de usuarios y contraseñas

De manera a incrementar la seguridad, los mensajes intercambiados entre los programas servidor y cliente, y que llevan nombres de usuario y contraseñas, son encriptados por un simple, pero ingenioso algoritmo desarrollado. El código fuente de la función de encriptación, desarrollado en lenguaje C, se incluye a continuación:

```
void encripta(char mensaje[], char llave[])
    {
          int z=0, tam=0;

          tam = strlen(mensaje);

          char llave2[tam];

          srand ( time(NULL) );

          for (z=0; z<tam; z++)
          {llave[z]=(int)(rand()%22+65);}

          llave[z]=0;

          for(z=0; z<tam; z++)
          {llave2[z] = llave[tam-1-z]*2-60;}
          llave2[z]=0;


          for(z=0; z<tam;z++)
          {mensaje[z] = (mensaje[z]+llave2[z]);}
    }
```

Las variables de entrada son:

- ✓ *char mensaje[]* – el propio mensaje a encriptar.
- ✓ *char llave[]* – una cadena de caracteres de igual longitud que el mensaje; inicialmente se encuentra vacía.

La sub-función no presenta una salida explícita, pero dado que *llave[]* es una cadena, la misma es pasada a la sub-función *por referencia* y es efectivamente modificada al finalizar el llamado a la función *encripta*. Por tanto, la salida es la variable *llave[]*.

Note, en el código fuente, que es la cadena *llave2[]* la que se suma al mensaje para producir la encriptación del mismo.

Entonces, se envía al receptor el *mensaje[]* encriptado y la *llave[]*; este debe, primeramente, calcular la *llave2[]* y, luego, aplicar la operación inversa al mensaje para

desencriptarlo. Precisamente esto hace la función: *void desencripta(char mensaje[], char llave).*

Lo que hace interesante a esta simple técnica de encriptación es que la llave que se utiliza para encriptar y desencriptar los mensajes es variable, ya que es generada de manera aleatoria por la función *rand()* (cuya semilla también es aleatoria, generada por *srand(time(NULL));*) y, por tanto, la llave de encriptación/desencriptación varía dinámicamente.

Esto es importante ya que los programas utilizados para romper mensajes encriptados generalmente buscan claves constantes. Para romper la encriptación de los mensajes hay que conocer en detalle las técnicas de encriptación, lo cual es prácticamente imposible al ser la llave dinámica.

- Conexión y transmisión de datos ordenada

Una vez que un cliente autentica correctamente su identidad, se establece la conexión TCP con el servidor y empieza la transmisión ordenada del archivo que posee los datos meteorológicos.

La transmisión se realiza abriendo a nivel binario el archivo y enviando ordenadamente mensajes de tamaño fijo igual a 5 bytes al receptor. Debido a que la transmisión se puede realizar a través de cualquier tipo de redes, se contempla que se produzcan errores en los mensajes recibidos. A medida que se aumenta el tamaño de los mensajes enviados, aumenta la probabilidad de errores en mensajes recibidos por el receptor (STEVENS; FENNER; RUDOFF, 2003).

Entonces, para aumentar la velocidad de transmisión efectiva sin aumentar considerablemente los errores en los mensajes recibidos, se envían simultáneamente 5 mensajes de 5 bytes cada uno (25 bytes en total por envío). Una vez que el receptor confirma que ha recibido los 5 mensajes de manera correcta, se prosigue con los siguientes mensajes. Este proceso se repite secuencialmente hasta que se envía completamente el archivo.

- Detección de errores

Como se mencionó en el punto anterior, se contempla que los mensajes puedan llegar con errores al receptor. Esto se considera inaceptable ya que se precisa tener una base de datos cuyas mediciones sean reales y correctas para que luego esta sea de utilidad.

Para responder a este problema, se implementaron varias técnicas de corrección y detección de errores en el código fuente del programa. Las técnicas de corrección

implementadas fueron el método de Hamming y la codificación Reed-Solomon (HALL, 2009). Realizando pruebas con estas técnicas, a través de una conexión inalámbrica (*Wi-fi*), se observó que solo el 75% de los mensajes podían corregirse; los demás eran incorregibles debido a la cantidad de errores que poseían.

Se desarrolló finalmente un programa que implementa la técnica de detección conocida como CRC (*Cyclic Redundancy Check*). Se utiliza en la implementación realizada el polinomio generador estándar de la CCITT, conocido como CRC16.

Con esta técnica se consigue detectar todo tipo de fallas en mensajes recibidos y los mensajes que llegan con errores son solicitados nuevamente por el receptor, de manera que el transmisor vuelve a enviarlos (STEVENS; FENNER; RUDOFF, 2003).

- Atención a múltiples clientes en simultáneo

Un servidor debe poder atender solicitudes de varios clientes de manera simultánea. Esto se consigue en el programa desarrollado mediante la función *fork*. Mediante una llamada a la función *fork*, se crea un proceso idéntico al que corre en memoria (STEVENS; FENNER; RUDOFF, 2003; KERRISK, 2010). A este proceso creado se lo conoce como proceso hijo y al proceso que llama a la función *fork* se lo conoce como proceso padre. El proceso principal del servidor escucha permanentemente el puerto y cuando un cliente desea conectarse crea un proceso hijo mediante una llamada a *fork*; es el hijo quien se encarga de atender al cliente determinado que se está conectando, como se muestra en la Figura 6 (KERRISK, 2010).

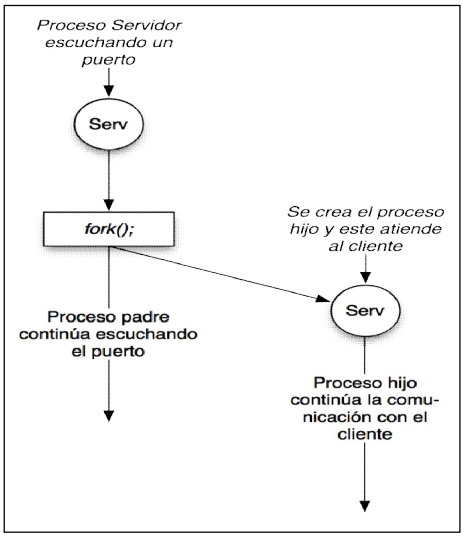

Figura 6 – Atención a múltiples clientes
Fuente: Elaborado por el autor

El proceso hijo creado mantiene la comunicación con el cliente hasta que el cliente se desconecta. De esta forma el proceso principal (padre) se dedica exclusivamente a atender un puerto y gestionar las conexiones entrantes. Mediante esta técnica, el proceso servidor puede atender a una cantidad indefinida de clientes.

2.3.3 Programa cliente

El programa cliente va instalado en las computadoras remotas, que se conectan a las estaciones meteorológicas. Este programa solicita a la estación las variables meteorológicas en tiempo real mediante una llamada a la sub-función estación, comentada previamente. Posteriormente, se crea el archivo a enviar y se ordenan los datos en el mismo. Una vez obtenidos los datos y conformado el archivo a transmitir, el programa cliente intenta conectarse al servidor remoto para iniciar la transmisión del archivo. Tal como el programa servidor, el cliente incluye las características de encriptación y desencriptación, la técnica de detección de errores CRC y la transmisión ordenada de datos.

Además de estas, el programa cliente posee otras cualidades que se describen a continuación.

1) Protección contra entornos hostiles y desatendidos

Dado que el programa cliente estará instalado en una computadora en un sitio remoto, posiblemente desatendido, se prevén ciertos tipos de fallas.

- Una falla común es la pérdida de la conexión (o enlace) al servidor. En caso de que la conexión se pierda, los datos son obtenidos de la estación periódicamente y son almacenados localmente en un archivo. Cuando regresa la conexión, el programa cliente envía automáticamente todos los datos obtenidos en el periodo de desconexión. Además, en el programa se observa una advertencia de que no se estableció conexión.
- Podría ocurrir otro tipo de falla si la consola de la estación meteorológica fuese desconectada (o no estuviera bien conectada) de la computadora, o bien si las baterías de la misma se agotaran. Este tipo de falla es grave, ya que no se obtienen los datos. El programa cliente muestra una advertencia explicando esta situación pero sigue corriendo normalmente. Una vez que el problema se resuelve continúa obteniendo los datos y transmitiéndolos normalmente.
- Si ocurre cualquiera de los tipos de fallas anteriores, el programa servidor los detecta e imprime en pantalla un mensaje de error, detallando el problema y sugiriendo la solución.

- El programa funciona de forma desatendida, transmitiendo los datos automáticamente cada cierta cantidad de tiempo (variable definida por el usuario).

2) Copia de seguridad de la base de datos en disco local

Para mantener duplicada la base de datos por cuestiones de redundancia y seguridad, el programa cliente copia la base de datos al disco de la computadora local. De esta forma, además, los datos pueden ser aprovechados y utilizados para investigaciones locales, sin necesidad de ser solicitados al servidor central. Es importante recalcar que para la utilización del programa cliente no se requiere conocimiento informático alguno. Simplemente se hace doble-click en el ejecutable y se inicia el programa, que realizará sus tareas automáticamente de forma pre-configurada.

2.3.4. Comunicación cliente-servidor

Habiendo definido los programas cliente y servidor, se muestra a continuación cómo se implementa la comunicación entre estos programas desarrollados. La comunicación se realiza llamando a ciertas funciones especiales de manera ordenada y secuencial. En la Figura 7 se muestra un diagrama básico de las funciones que hacen a la comunicación efectiva (WILLIAMS, 1993; STEVENS; FENNER; RUDOFF, 2003;).

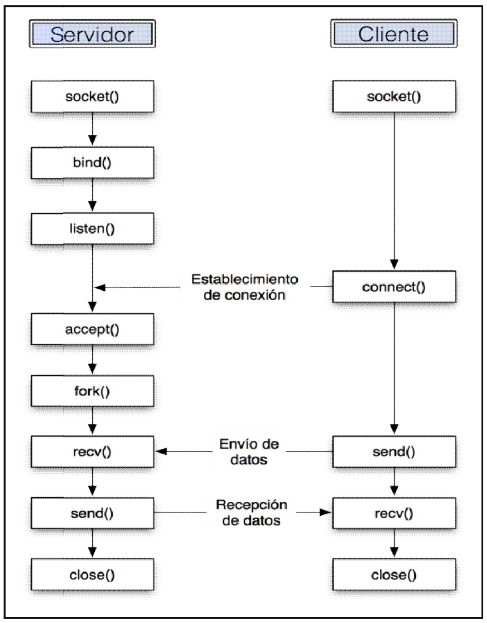

Figura 7 – Comunicación entre el cliente y el servidor
Fuente: Elaboración propia

En síntesis, luego de describir las etapas A, B y C en esta etapa, D, se juntan todas éstas para contemplar el sistema final de programas. La forma en que las etapas interactúan se resume en la Figura 7. Es de suma importancia recordar este esquema y las relaciones existentes entre el programa cliente y el programa servidor en lo que resta del trabajo, ya que son éstos los que permiten al conjunto de equipos de comunicación realizar efectivamente la transmisión.

3 ANÁLISIS DE LOS RESULTADOS

Habiendo implementada la red mostrada en las etapas anteriores, e instalado el conjunto de programas en las debidas computadoras, se observan resultados prometedores del sistema. Se observa que los programas clientes se han sincronizado con sus respectivas estaciones meteorológicas y empezaron a transmitir la información al servidor. Por otro lado, el programa servidor recibe conexiones de múltiples clientes simultáneamente y almacena ordenada y periódicamente la información recibida de los clientes en su base de datos. Esta demás repetir que en las transmisiones entre cliente y servidor, los clientes son autenticados y las comunicaciones son encriptadas. A continuación evaluamos cuantitativamente los resultados obtenidos.

3.1 Enlace de microondas FIUNA-CITEC

Los resultados obtenidos con respecto al enlace de microondas que conecta los Laboratorios de Mecánica y Energía de la Facultad de Ingeniería de la UNA (LAAMEN-FIUNA) con los laboratorios de electrónica de potencia y control del centro de innovaciones tecnológicas (CITEC) son muy positivos y se detallan a continuación:

1. En cuanto al ancho de banda teórico (BWteórico) obtenido en modo full dúplex, se presentan valores de 60 Mbps/ 60 Mbps (TX/RX). Se observa esto en la Figura 8.

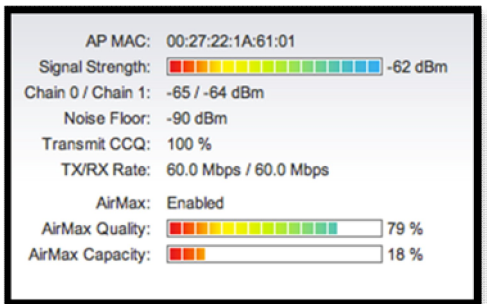

Figura 8 – Parámetros del enlace FIUNA – CITEC
Fuente: AirMax

2. En cuanto a la relación Señal a Ruido (SNR), se observa en la Figura 8 que el nivel de señal recibido es de -62 dbm y que el nivel del ruido es de -90 dbm, por tanto (Ecuación 3):

$$SNR_{dB} = P_{señal\ dB} - P_{ruido\ dB} =$$
$$= -62 - (-90) = 28$$

Se obtiene una SNR muy buena e igual a 28.

3. Se realizaron pruebas de *ping* o *Round-trip Delay Time* (retardo de ida y vuelta de un paquete) de extremo a extremo en la red y el resultado obtenido fue un retardo de 8.965 ms en promedio, como se puede apreciar en la Figura 9. Nótese, además, que al haber transmitido 2280 paquetes y haber recibido exactamente la misma cantidad, la eficiencia en la transmisión es del 100% (en coherencia con el CCQ = 100% que se observa en la Figura 8).

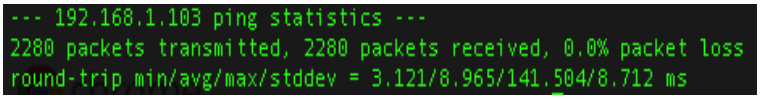

Figura 9 – Resultados del *round-trip delay time* (*ping*)
Fuente: Elaborado por el autor

4. Para medir el ancho de banda real del enlace se realizaron pruebas de transmisión semi-dúplex en tiempo real y los resultados indican un promedio de 45 Mbps, como se observa en la Figura 10.

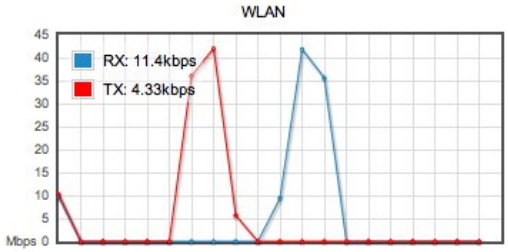

Figura 10 – Resultados de la prueba de ancho de banda real
Fuente: AirMax

3.2 Programas Cliente/servidor desarrollados

Los resultados obtenidos con respecto al conjunto de programas cliente/servidor desarrollados muestran que estos funcionan adecuadamente y permiten efectuar una segura y confiable transmisión de datos meteorológicos entre un cliente remoto y el servidor. Además, dado que estas estaciones probablemente serán instaladas en zonas desatendidas y en

ambientes hostiles (donde la pérdida de conexión es un problema común), se contemplan ciertos criterios de protección contra fallos para minimizar la pérdida de datos y, de esta manera, aumentar la versatilidad del programa.

En la Figura 11 se puede apreciar el programa servidor, que estuvo corriendo por un periodo de prueba de dos semanas, conectado a un cliente remoto.

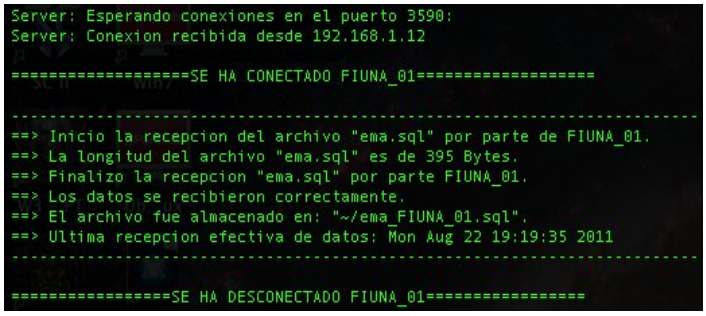

Figura 11 – Programa Servidor
Fuente: Elaborado por el autor

De acuerdo a esta Figura 11, el usuario conectado es FIUNA_01, el cual transmite un archivo llamado ema.sql, que posee un conjunto ordenado de variables meteorológicas, cuya longitud es de 395 Bytes.

A continuación se muestra, en la Figura 12, al programa cliente que estaba transmitiendo al servidor de la Figura 11.

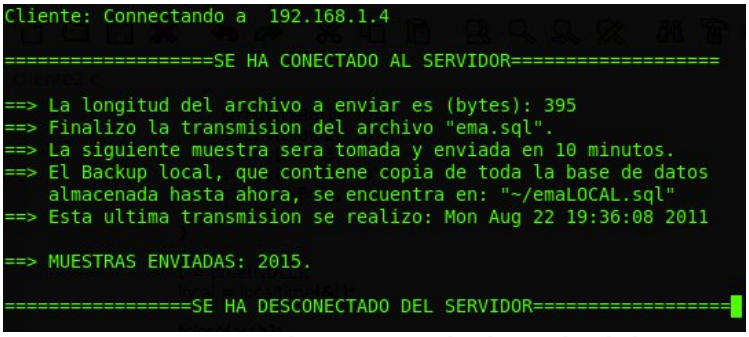

Figura 12 – Programa cliente conectado al servidor de la Figura 11
Fuente: Elaborado por el autor

El programa anuncia que se ha conectado al servidor y recuerda que enviará la siguiente muestra en 10 minutos. Además, se observa que se imprime en pantalla la fecha y hora de la última muestra enviada, la cantidad de muestras ya enviadas y la ubicación de la base local *"~/emaLOCAL.sql"*.

4 CONCLUSIONES

Los resultados obtenidos en el presente trabajo final de grado han permitido verificar que la transmisión de datos en función a los programas desarrollados no requieren de conocimiento informático especializado para su instalación y empleo, debido a que están pre-configurados y programados de manera precisa; la característica descripta resulta muy conveniente pues posibilita que todo usuario acceda a los mismos y efectúe la transmisión de datos. Los criterios contemplados para la protección contra fallos (como ser la conectividad o pérdida del enlace, o la desconexión de la consola a la PC) permiten a los programas funcionar de forma autónoma y desatendida. Por tanto los programas facilitan la obtención de datos de forma ordenada y segura a través de las técnicas de encriptación y detección de errores (CRC16) que fueron aplicadas en el presente trabajo.

Los programas hacen uso del medio de comunicaciones de manera eficiente (ya sea el enlace físico FIUNA-CITEC u otra red), manteniendo la conexión solo durante la transmisión, y una vez que culmina la transmisión se desconectan liberando totalmente el medio.

Todos los programas fueron desarrollados en lenguaje ANSI C, lo que da portabilidad al código fuente, permitiendo su compilación en varios sistemas operativos (Linux, Mac OS, Windows, BSD, otros), que permite su uso en múltiples plataformas. Se constata además que los programas desarrollados no pueden operar con recursos de cursos de hardware muy sencillos. Siendo así, se libera a los usuarios de las estaciones de tener que incurrir en costos de adquisición -anual- correspondientes a las licencias del software del fabricante de las estaciones meteorológicas.

El hecho de poseer un enlace dedicado entre la FIUNA y el CITEC garantiza que los datos de la estación meteorológica instalada en el CITEC podrán transmitirse regularmente en distintas condiciones. A su vez, el elevado ancho de banda de la red -60 Mbps Full Dúplex- desplegada puede ser aprovechado para compartir archivos, servidores, bases de datos y otros tipos de aplicaciones.

COMUNICATION AND TRANSMISSION OF DATA FROM WEATHER STATIONS

ABSTRACT: The present research develops simple, flexible and intelligent software written in C language, to retrieve the variables from weather stations and transmit them efficiently and securely to a central database server through any type of network. The results obtained with them shows that the client/server programs works in a secure and dependable way for the

transmission of meteorological variables. We also verify that those programs facilitate the extraction of the data in an ordered and secure way using encryption and CRC16 error check technique**.**

Keywords: Data transmission. Climatic variables. Data basis. Networks. C language.

BIBLIOGRAFIA

CARDAMA, A. et al. **Antenas.** Edicions de la Universitat Politècnica de Catalunya, Segunda Edición, 2002.

HALL, B. **Beej's guide to network programming:** using Internet Sockets. Version 3.0.14, 2009, pp. 22-59.

KERRISK, M. **The Linux programming interface**. A Linux and UNIX System programming Handbook, No Starch Press, Inc., p. 513-525, 2010.

STEVENS R.; FENNER B.; RUDOFF A. **UNIX Network Programming Volume 1**, Third Edition: The Sockets Networking API, Addison Wesley, 2003, p. 121-153, 2003.

TESTER, Jefferson W. et al. **Sustainable Energy:** choosing among options. The MIT Press, 2005, p. 41-45, 2005.

UBIQUITI WIKI, **AirOS AirMAX:** user's guide for installation and administration. v 5.0. Ubiquiti Networks, 2011.

TOMASI, W. **Sistemas de comunicaciones electrónicas**. Pearson Education, Cuarta Edición, 2003.

WILLIAMS, R.**A Painless Guide to CRC Error Detection Algorithms**: everything you wanted to know about CRC algorithms, but were afraid to ask for fear that errors in your understanding might be detected. Rocksoft Pty, 1993.

Originais recebidos em: 29/12/2011

Aceito para publicação em: 04/06/2012