ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN MONITOREADOR PARA REDES USANDO TCP/IP Y PARADIGMA CLIENTE-SERVIDOR.

Nestor Arreaga Alvarado [1], Carlos Calero Pèrez [2], Christian Romo Andrade [3], Willie Siavichay Pino [4], Guido Caicedo Rossi [5].

[1] Ingeniero Eléctrico en Computación 1998.
[2] Ingeniero Eléctrico en Computación 1998.
[3] Ingeniero Eléctrico en Computación 1998.
[4] Ingeniero Eléctrico en Computación 1998.
[5] Director de Tópico, Ingeniero Eléctrico especialización Computación, Escuela Superior Politécnica del Litoral, 1990, Master en Ciencias en Computación (State University of New York at Buffalo, USA.), 1993, Profesor de la ESPOL desde 1993.

RESUMEN

Se ha diseñado e implementado un sistema, el cual usa principalmente la tecnología de comunicación de datos, de redes de computadoras y el paradigma Cliente-Servidor. El sistema permite conocer la cantidad de información que circula en una red local, es decir, permite monitorear su tráfico. Para ello consta de dos partes:

- La primera que es donde se ejecuta el servidor del monitoreo (o simplemente SERVIDOR), que se encarga principalmente de la captura de los paquetes que circulan en la red, aparte de atender los requerimientos de los usuarios,
- y la otra (CLIENTE), que permite dos tipos de usuarios: Administrador, el cual puede configurar los parámetros necesarios para arrancar el monitoreo en el SERVIDOR, y el Monitoreador, el cual puede obtener información estadística acerca del tráfico de la red local.

INTRODUCCIÓN

En el curso correspondiente al Tópico I, se diseñó e implementó bajo DOS, usando Borland C++ 3.0, el software ***monitor.exe***, el cual monitorea los paquetes de tipo TCP/IP[1] que circulan en una red local y muestra los resultados clasificándolos por protocolos, puertos y direcciones (Direcciones IP de máquinas específicas que generan tráfico en la red).

El presente proyecto usa el software ***monitor.exe,*** como base de la aplicación *SERVIDORA*, el cual fue modificado para que esta vez no muestre ninguno de sus resultados en pantalla, solo los grabe en archivos tipo texto. Para el diseño de las aplicaciones *CLIENTE* y *SERVIDORA* se realizó un Análisis Orientado a Objetos y se eligió *JAVA* como el lenguaje más óptimo para realizar la implementación, ya que se ajusta al mismo paradigma y además existía el requerimiento de que el *CLIENTE* esté embebido en un browser de Internet, el cual es capaz de desplegar las curvas que representan el tráfico de manera gráfica.

OBJETIVOS

El objetivo de este proyecto es monitorear el tráfico TCP/IP existente en una red Ethernet basándose en la información contenida en los paquetes que circulan por ella y mostrar los resultados del monitoreo de manera gráfica a través de un browser de WWW, aplicando el paradigma Cliente-Servidor.

JUSTIFICACIÓN DEL PROYECTO

Podemos citar las siguientes razones por las cuales este proyecto se realiza:

- La necesidad de llevar estadísticas acerca del tráfico existente en una red.
- Tener una apreciación comparativa de ese tráfico para conocer en qué intervalos de tiempo existe más carga en la red.
- Dar la posibilidad a los usuarios de conocer la carga de tráfico en la red.

---

[1]Arquitectura de comunicaciones de Internet.

- Permitir al administrador manejar el sistema con una interface fácil de operar y de manera remota.

CONTENIDO

1.1. DESCRIPCIÓN GENERAL DEL PROYECTO

El *SERVIDOR* se ejecuta en una máquina que se encuentra en una red LAN que usa TCP/IP, y atiende los siguientes tipos de requerimientos:

- Arranque del proceso de monitoreo, basado en una configuración nueva o anterior.
- Detención del proceso de monitoreo.
- Envío de datos estadísticos del monitoreo que se está ejecutando o del anterior a los *CLIENTES monitoreadores*.

Los clientes se pueden ejecutar en cualquier máquina que se encuentre conectada a Internet. El cliente *Administrador* es el que puede ejecutar las tareas siguientes:

- Configurar el monitoreo,
- Arrancar el monitoreo, siempre y cuando se haya configurado uno.
- Detener el monitoreo.

El cliente *Monitoreador* es el que puede ejecutar las tareas siguientes:

- Ver los resultados del monitoreo actual, en linea.
- Ver los resultados del monitoreo actual, histórico.
- Ver los resultados del monitoreo anterior, histórico.

Para visualizar mejor la comunicación entre los *CLIENTES* y el *SERVIDOR* observar la Figura#1.

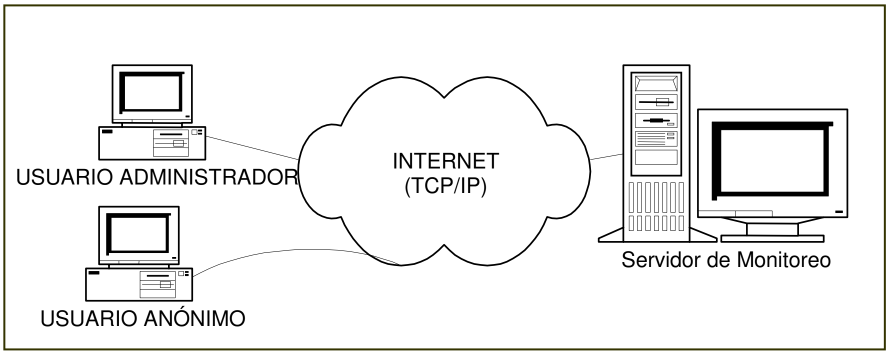

Figura # 1. Interacción cliente- servidor en una red.

FUNCIONALIDAD DEL SERVIDOR:

La función principal de esta aplicación es capturar los paquetes que pertenecen al protocolo TCP/IP para su análisis y clasificación. Se pueden configurar los siguientes parámetros de muestreo:

- **Protocolos udp y tcp a monitorear:** Estos son los protocolos TCP/IP que van a ser monitoreados. Si no se especifican formarán parte de un grupo denominado "otros".
- **Puertos udp y tcp a monitorear:** Estos son los puertos TCP/IP que van a ser monitoreados. Los que no se especifiquen formarán parte de un grupo denominado "otros".
- **Direcciones a monitorear:** Estas son las direcciones IP de las máquinas que se desean monitorear y examinar el nivel de tráfico desde o hacia ellas.
- **Tiempo máximo de monitoreo:** Es el tiempo máximo de monitoreo. Puede interrumpir el proceso de monitoreo.
- **Unidad de tiempo:** Es el intervalo de tiempo en que se registrarán los muestreos, puede ser en segundos, minutos u horas, el programa permite cualquiera de las tres opciones. Esta unidad de tiempo es como mínimo un cincuentavo (1/50) del tiempo máximo de monitoreo.[2]

---

[2] Esto no influye en la captura de los paquetes ya que estos son capturados conforme se transmiten. La cantidad de paquetes obtenidos es simplemente la suma de todos los paquetes obtenidos durante el intervalo elegido.

- **Fecha de inicio del monitoreo:** Indica la fecha y hora de inicio del proceso de monitoreo, es decir se puede realizar un monitoreo programado para un tiempo futuro.

En la siguiente gráfica se puede apreciar como el *CLIENTE Administrador*, el cual se menciona en el tema **Funcionalidad del cliente**, configura los parámetros para poder arrancar el monitoreo.

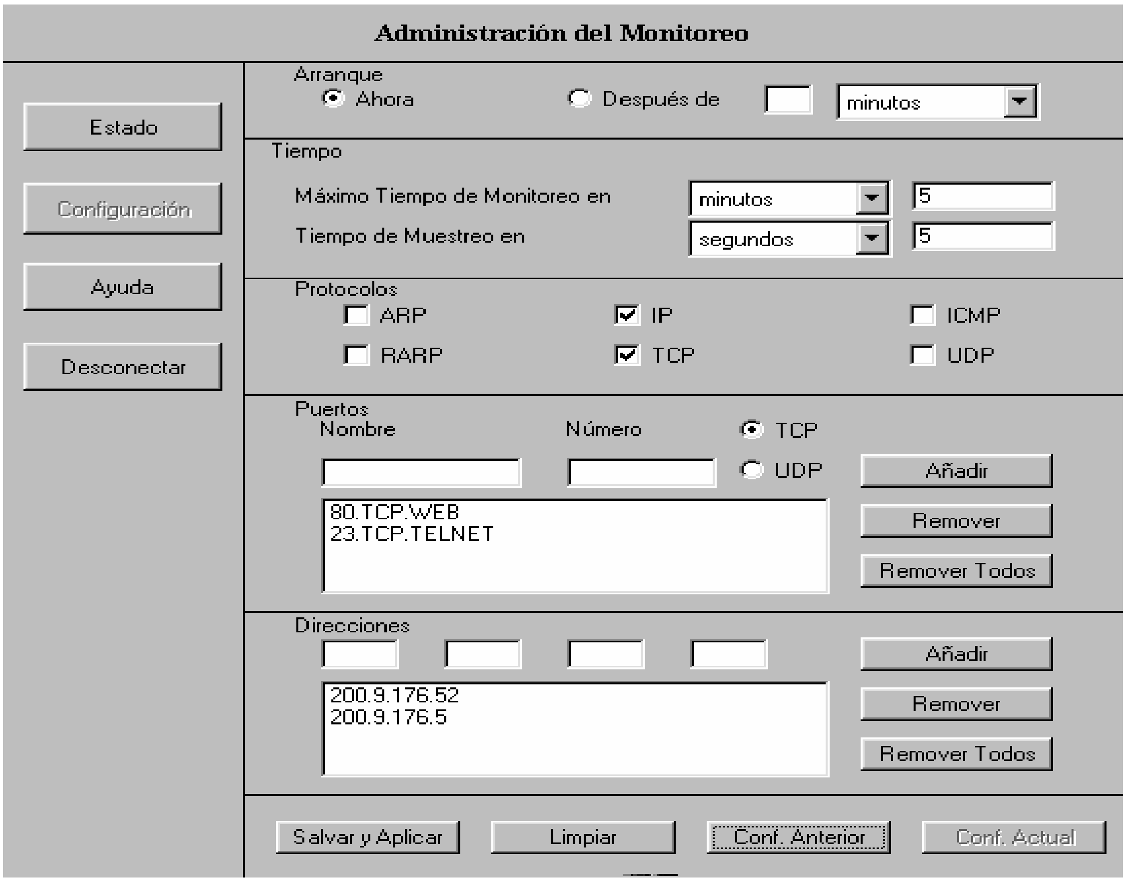

Figura # 2 Pantalla de Configuración.

FUNCIONALIDAD DEL CLIENTE:

El *CLIENTE* es un applet implementado en lenguaje JAVA. Se accede al cliente usando un browser (Internet Explorer, Netscape, etc.) que soporte JAVA. Hay que distinguir dos tipos de clientes (applets):

- usuario *Administrador* y,
- usuario *Monitoreador*.

El applet del usuario *Administrador* permite configurar, arrancar o detener el proceso de monitoreo en el servidor. El applet del usuario *Monitoreador* permite seleccionar las estadísticas que se desea observar y en qué formato (gráfico o texto). Se puede elegir entre los datos del monitoreo anterior o del monitoreo actual. En los gráficos, el *eje de las x* indica el número de muestreos (unidad de tiempo), mientras el *eje de las y* indica:

- Bytes por unidad de tiempo,
- Paquetes por unidad de tiempo, y
- Tamaño promedio de los paquetes en la unidad de tiempo.

Se pueden observar los siguientes gráficos:

- **Tráfico por protocolos.-** Se muestra un gráfico de hasta seis líneas de diferentes colores, uno para cada protocolo monitoreado.
- **Tráfico por puertos.-** Se muestra un gráfico de hasta seis líneas de diferentes colores, los cinco primeros colores indican los puertos que mayor tráfico han causado en la red y el último para indicar el "resto" de los puertos no especificados.
- **Tráfico por direcciones IP.-** Se muestra un gráfico de hasta seis líneas de diferentes colores, los cinco primeros colores indican las direcciones IP seleccionadas y el último para indicar el "resto" de las direcciones IP no especificadas. En este tipo de gráfico también se podrán elegir entre bytes, paquetes y tamaño promedio de recibidos o transmitidos.

A continuación se muestra la gráfica de tres protocolos que son monitoreadas en la red.

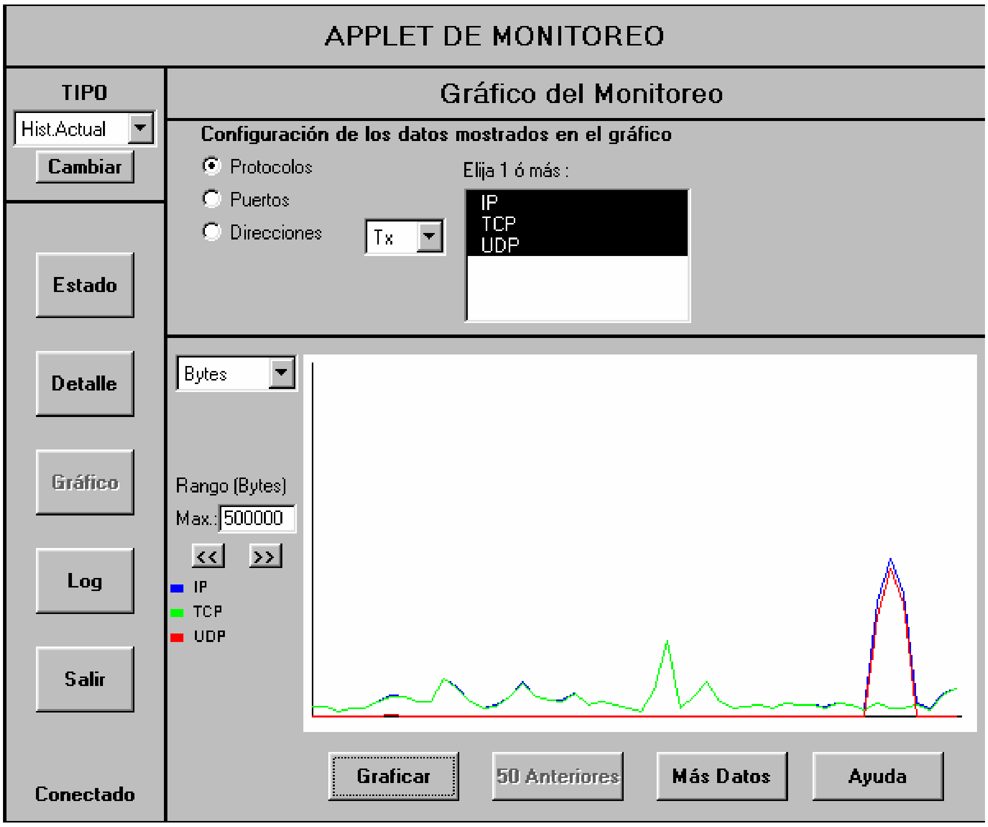

Figura # 3 Pantalla de Gráfico del Monitoreo

ALCANCE

El presente proyecto va a ser desarrollado en su mayor parte en JAVA, ya que la aplicación que realiza el monitoreo se mantendrá en Borland C ++ 3.0. El monitoreo sólo permitirá registrar datos sobre un máximo de 6 protocolos, 10 puertos y direcciones.

Para versiones futuras de este software se puede migrar la aplicación que realiza el monitoreo a un lenguaje de más bajo nivel (Ensamblador). Además se puede incrementar la cantidad de protocolos, puertos y direcciones a monitorearse. Otra mejora es que podría desarrollarse la aplicación servidora en un sistema operativo de red (UNIX, Windows NT, etc.). Se puede implementar una opción para monitorear otro tipo de tráfico en la red (IPX/SPX, etc.). Si tenemos mejoras en la tasa de transmisión que la red soporta, se puede evitar limitar la cantidad de datos estadísticos que serán transmitidos al cliente.

CONCLUSIONES Y RECOMENDACIONES

Al realizar este proyecto aprendimos a manejar hilos concurrentes, procesos, métodos, instancias, sockets, puertos, entre algunos de los conceptos que envuelve el paradigma orientado a objetos y conceptos básicos de programación Cliente-Servidor.

Cuando la aplicación desarrollada es usada por un administrador de una red, esta le puede indicar comportamientos y preferencias de los usuarios de la LAN, ya que se puede llegar a establecer si la red es usada principalmente para usar el WWW, o para realizar una sesión FTP que les permita obtener información acerca de cualquier tema de interés particular.

Se recomienda tratar de migrar la parte del servidor que está codificada en Borland C a Java para que el servidor esté codificado en un solo lenguaje. Esta migración no se la realizó debido a que ello implicaba invertir una gran cantidad de tiempo, con el cual no se contaba.

Cuando se realicen cambios o mejoras en el código escrito en JAVA, si se obtienen resultados inesperados durante el proceso de compilación o ejecución, se recomienda probar con otros compiladores y/o programas, ya que ciertas consideraciones respecto a la forma de compilación, principalmente, varían dependiendo del fabricante. Las herramientas que se usaron durante la fase de implementación fueron Visual Café de Symantec, Visual J++ de Microcosoft, Jdk1.1.5 de Sun.

REFERENCIAS

Douglas E. Comer, Redes Globales de información con Internet y TCP/IP, principios básicos, protocolos y Arquitectura. (3era. Edición; Mexico: Prentice-Hall Hispanoamericana, S.A., 1996).

Douglas E. Comer and David L. Stevens, Cliente-Server Programming and Applications, Volumen III (Prentice-Hall , Inc. Upper Saddle River, New Jersey 07458).

Howard W. Sams & Campany, Turbo C Biblie, Volumen 2.0 (Indianapolis, Indiana: The Waite Group, Inc., 1989).

Mary Campione & Kathy Walrath, The JAVA Tutorial (2da. Edición; San Antonio Road, Palo Alto: Sun Microsystems, Inc.,1995).