Herramienta de Gestión en Tiempo Real de Redes de Área Extensa

[1]E. MAGAÑA, [2]J. ARACIL, [3]J. VILLADANGOS
DEPARTAMENTO DE AUTOMÁTICA Y COMPUTACIÓN
UNIVERSIDAD PÚBLICA DE NAVARRA
CAMPUS ARROSADÍA S/N, 31006 PAMPLONA
TFNO: [1,3] +34 948 169645 [2] +34 948 169733 FAX: [1,2,3] +34 948 169281
E-MAIL: {[1]eduardo.magana, [2]javier.aracil, [3]jesusv}.upna.es

Abstract:

This paper presents PROMIS[1], a new network management tool based on a distributed architecture of traffic probes. It allows network managers to monitor traffic from any wide area network segment in real time using a WWW console or a Tcl/Tk graphic interface. Keywords: distributed sniffer, network management, real time traffic monitoring.

1. Introducción

A la hora de gestionar una red de comunicaciones de medio y gran tamaño, se encuentra la necesidad de herramientas de gestión automatizadas, como la que se presenta en este trabajo, que permitan llevar un control continuo de todas las redes. Este proceso consiste en la definición de alarmas e identificación automática de problemas que notifiquen al gestor de manera inmediata las desviaciones del comportamiento habitual de la red. De esta forma, se puede aislar el problema y actuar en consecuencia.

Medidas sobre redes locales (LAN) y de área extensa (WAN) muestran características autosimilares (*self-similarity*) en el tráfico [2] que provoca alta variabilidad en la carga de las redes. Por tanto, es necesario una monitorización precisa del tráfico en los periodos de congestión, que es cuando las herramientas sobre SNMP (Simple NEtwork Management Protocol) y RMON (Remote network MONitor) muestran su vulnerabilidad por la posibilidad de pérdida de datos. PROMIS ofrece las funcionalidades típicas de los sistemas SNMP/RMON pero con una mayor fiabilidad debido al uso de un protocolo de transporte fiable.

2. Estado del arte

Las herramientas de gestión que vienen siendo utilizadas son principalmente los analizadores de protocolos y las plataformas de gestión SNMP.

La aplicación de los analizadores de protocolos o *sniffers* a la gestión de red se limita al ámbito del segmento local que monitorizan, no ofreciendo más información.

Las plataformas de gestión SNMP/RMON (HP Openview, SunNet Manager, IBM NetView y Cabletron Spectrum) sí proporcionan un control global de todas las subredes de una red de área extensa pero la escala de tiempo en la que se lleva a cabo la monitorización debe ser necesariamente elevada. Esto se debe al esquema de *polling* gestor/agente en el que se basan y en el uso de UDP (User Datagram Protocol) como protocolo de transporte.

En entornos con grandes pérdidas de paquetes el uso de un protocolo de transporte no fiable favorece la pérdida de información de monitorización precisamente en instantes de congestión. Estos instantes de congestión en el caso de la Internet pueden tener lugar en escalas de tiempo grandes [3,4]. Para tener una información detallada de congestión es preciso por tanto enviar información *en tiempo real* y con transporte fiable de agente a gestor. Las plataformas basadas en SNMP/RMON, sin embargo, no pueden satisfacer este requisito.

Para ilustrar este hecho, en la *Fig. 1* se muestran una serie de gráficas de porcentaje de paquetes perdidos en función del tamaño, obtenidas a partir de 70 series de 100 pings (ICMP) para diferentes escenarios con la red local cargada al 80%. Se observa que las pérdidas aumentan con el tamaño del paquete y que son elevadas cuando se sale a la red Internet, lo que provoca otra vez un gran impacto si pretendemos monitorizar en tiempo real con una plataforma SNMP/RMON.

Presentamos un sistema, el Sniffer Distribuido PROMIS, que evita los problemas de los sistemas anteriores, gracias al uso de un protocolo de transporte fiable y de un gestor multiplataforma de fácil utilización en Internet.

3. Arquitectura y características del Sniffer Distribuido PROMIS

El Sniffer Distribuido consta de una aplicación central de gestión, que es lo que denominaremos consola, que obtiene y presenta los datos de interés referentes a una determinada red en tiempo real. Para ello, se colocan una serie de elementos hardware/software, que denominaremos sondas, en cada segmento de red a monitorizar. Las sondas ven el tráfico de la subred en que están conectadas, procesan todos los paquetes que circulan por ella y obtienen la información, desde el factor de utilización medio hasta los bytes/sg generados por determinado servicio.

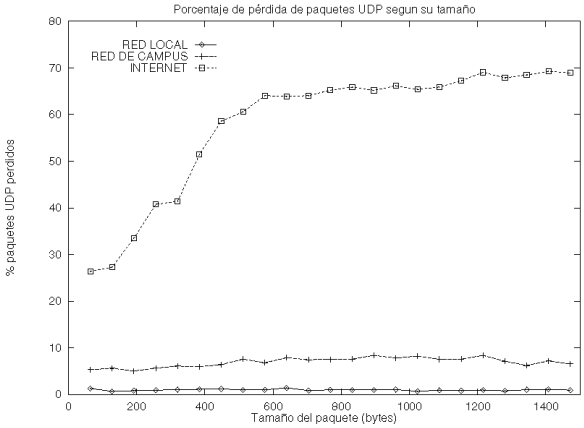

Figura 1: Pérdida de paquetes UDP

Como el procesado se realiza en la sonda, se añade un tráfico mínimo a la red. Además es posible tener una o varias consolas simultáneas. De esta forma, la consola se pone en contacto con las sondas para requerirles la información que interese en cada momento. Esta comunicación se realiza sobre un protocolo de transporte fiable como TCP (Transmission Control Protocol), lo que proporciona las ventajas de fiabilidad debido a la ausencia de pérdida de datos y alarmas, y la verdadera monitorización en tiempo real.

A continuación se indican los datos a los que es posible acceder desde la consola:

- Modo Monitor: permite observar parámetros de red en tiempo real como bytes/sg, paquetes/sg y utilización media tanto globales como filtrados por máquina, protocolo y servicio. En esta parte se incluyen los informes de problemas generados por el sistema experto.
- Modo Captura: permite la captura programada del parámetro del tráfico requerido a disco duro.
- Alarmas: definidas ante parámetros del tráfico que sobrepasen ciertos umbrales.

4. Componentes del Sniffer Distribuido: consola y sonda

Se ha optado por hacer la consola multiplataforma, mientras que la sonda se ha diseñado para PCs con Linux exclusivamente.

4.1. Sonda

Se ha implementado sobre un PC Pentium 133MHz, 32MB RAM, 1.5GB de disco duro y tarjeta Ethernet ISA 3c509, con sistema operativo Linux. Básicamente, el programa, escrito en lenguaje C, consta de 3 procesos que funcionan de forma concurrente intercomunicados por un sistema de IPC (InterProcess Communication) como son las *pipes*, tal y como se aprecia en la *Fig. 2*.

El proceso Lector es el encargado de capturar todos los paquetes de la red Ethernet gracias a poner la tarjeta de red en modo promiscuo.

El proceso Filtra se encarga de obtener la información de interés de cada paquete. También procesa las alarmas y capturas.

Cualquier petición por parte de la consola lanza un subproceso del Gestor de Peticiones que según su cometido requerirá o no datos del proceso Filtra a través de *pipes* activadas dinámicamente.

4.2. Consola

Se han implementado dos variantes de consola: una sobre navegador WWW y la otra como aplicación Tcl/Tk.

La versión sobre navegador WWW conjuga el uso de programación HTML, CGIs y applets Java. En todo caso la comunicación entre la sonda y la consola se realiza vía TCP/IP, ya sea por medio del protocolo HTTP del sistema de páginas web o por conexiones directas del Java con la sonda.

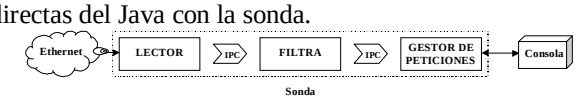

Figura 2: Estructura de bloques de la sonda

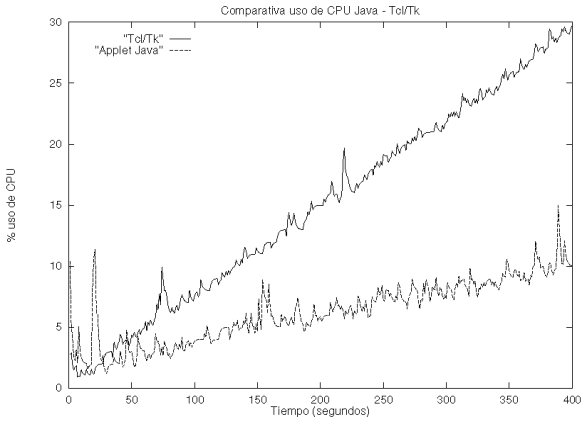

Figura 3: Comparativa uso de CPU Java - Tcl/Tk

Las ventajas de esta implementación son la accesibilidad a la información de las sondas desde cualquier ordenador del mundo provisto de un navegador web.

La consola como aplicación Tcl/Tk es portable a muchas plataformas y presenta las ventajas de que puede funcionar de manera autónoma con respecto a las sondas, el intercambio de información se reduce (se evita el tener que cargar en la consola las páginas web de las sondas, sólo se transmiten los datos necesarios) y cuando una sonda detecta alguna alarma en su red se la comunica de forma inmediata a la consola.

La *Fig. 3* presenta una comparativa de ocupación de CPU (medido en un Axil Ultima Sparc Station 167MHz) en Java y Tcl/Tk empleado en representar una gráfica a la que se va añadiendo un punto por segundo. El algoritmo consiste en reescalar el gráfico y dibujar una nueva barra cada segundo. Se aprecia la mayor velocidad del Java debido a su compilación en *bytecode* frente a Tcl/Tk que es un lenguaje *script* interpretado.

6. Conclusiones

Actualmente las redes de área extensa están evolucionando a escenarios de alta velocidad en los que se hace necesario una monitorización con alta granularidad y fiabilidad. Las herramientas basadas en SNMP/RMON no están diseñadas para tal entorno debido al uso de un protocolo de transporte no fiable y el esquema de *polling*. La herramienta PROMIS presentada supera esas limitaciones usando un protocolo de transporte fiable y una arquitectura software multiplataforma. Además, PROMIS ofrece una consola sobre WWW y Tcl/Tk permitiendo una gestión de red distribuida en tiempo real.

Referencias

[1] E. Magaña, J. Aracil, J. Villadangos. "PROMIS: A Reliable Real-time Network Management Tool for Wide Area Networks". Aceptado en *Proceedings of IEEE Euromicro 98*, Vaasteras, Suecia, Agosto 1998.

[2] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. "On the self-similar nature of Ethernet traffic". *IEEE/ACM Transactions on Networking*, **2**(1):1-15, January 1994.

[3] E. Mier, "Network world, bell labs evaluate SNMP on bridges". *Network Wold*, April 1991.

[4] E. Mier, "Network world, bell labs test router's SNMP agents". *Network World*, July 1991.