MIRA: Software para el análisis de tráfico IP sobre ATM

Contacto: Carlos Veciana-Nogués

Advanced Broadband Communications Center (CCABA)
Universitat Politècnica de Catalnya (UPC)
Jordi Girona 1-3, Campus Nord D6008, 08034 Barcelona, Spain
Phone. +34 93 401 7182, Fax. +34 93 401 7055
carlosv@ac.upc.es

La lista completa de autores del articulo se especificará en el articulo definitivo1
Palabras clave: IP, ATM, análisis de tráfico, gestión de redes, tarificación

Conocer el uso que se da a las redes es sumamente importante para el diseño y gestión de las mismas. El entorno de red actual dominado por la tecnología TCP/IP (Internet) es muy dinámico y en expansión. Nuevas aplicaciones aparecen día a día, las aplicaciones ya no se restringen a los servicios básicos de web, correo electrónico o transferencia de ficheros, sino que nuevos servicios irrumpen con fuerza (comercio electrónico, multimedia,...). Los servidores también se multiplican, así como los intermediarios (proxys). Los usuarios, a su vez, modifican sus hábitos (numero y tipo de peticiones, duración de las sesiones,...) aprovechando la mayor calidad y nuevos tipos de contratos a la hora de conectarse a la red (tarifa plana,...). Finalmente, la comercialización de Internet ha puesto de manifiesto el replanteamiento de los métodos de tarificación, ya que la tarificación aplicada en los servicios de telecomunicaciones existentes hasta el momento no parecen satisfacer los requisitos del nuevo medio de comunicación que es Internet.
En el Centro de Comunicaciones Avanzadas de Banda Ancha, se viene trabajando en los últimos años en mecanismos de captura y análisis de trafico en entornos de alta velocidad. Estos entornos son los que corresponden con los servicios ofrecidos por las operadoras y los servidores de acceso a Internet (ISP). Debido al elevado volumen de tráfico que se maneja y a las tecnologías de red utilizadas en estos entornos, existen diferencias notables con las herramientas de análisis de trafico utilizadas en entornos LAN. Podemos decir que las herramientas de análisis de tráfico en entornos LAN nos ofrecen mucha información sobre las características de la red y de cada uno de los usuarios, mientras que es difícil tener una visión global del comportamiento de la red y de los usuarios. Las herramientas de análisis de tráfico para entornos WAN, nos ofrecen datos muy interesantes respecto a las características del trafico (volúmenes, retardos, perdidas, conexiones,...), pero difícilmente nos permiten concretar sobre el comportamiento de clientes y servidores. Al mismo tiempo en entornos WAN, donde la visión de la red es mucho más amplia es mucho más fácil detectar tendencias en el uso de los servicios. Las herramientas que se han desarrollado dentro de los proyectos CASTBA [1] – MEHARI [2]- MIRA[1], han conseguido realizar un análisis de trafico en un entorno WAN (Red Académica Española, RedIris) con un nivel de detalle equiparable al que es posible conseguir en un entorno LAN. Yendo un paso más allá del análisis convencional, se ha caracterizado el tráfico no solo con la información contenida en las cabeceras de los paquetes (tipo de aplicación, dirección IP, volúmenes en bytes,...), sino que se ha realizado un análisis del contenido de los mismos (protocolo de aplicación, datos de usuario,...), y se han cruzado estos datos con Bases de Datos externas (Internet Routing Registry). Nuestro trabajo se enmarca dentro de las líneas definidas en OC3MON [3] y el RTFM-WG [4][5], añadiendo nuevos parámetros que nos permitan tipificar la información por su naturaleza. Los objetivos finales son varios: ofrecer tantos parámetros como sea posible para la tarificación del uso de la red, caracterizar el tráfico en clases para aplicar diferentes tarifas a cada clase, y, finalmente, detectar nuevas aplicaciones y los servidores que las ofrecen de cara a una correcta planificación de las infraestructuras (anchos de banda, proxies, políticas de QoS,...).

El objetivo de este artículo es presentar el software de análisis de datos. El software se ha estructurado como una jerarquía de módulos a tres niveles. Dentro de cada jerarquía se realizan funciones del mismo tipo, pero cada módulo esta especializado en una tarea determinada. La interfaz entre los distintos módulos sigue una especificación que permite encadenar módulos de diferentes jerarquías y combinar

[1] CASTBA, MEHARI y MIRA han sido proyectos financiados por la CICYT, en los que han colaborado grupos de investigadores de las Universidades Politécnica de Madrid, Universidad Carlos III de Madrid y Politécnica de Catalunya. Actualmente los trabajos se realizan dentro del proyecto financiado MIRA (2FD97-2234-C03-02).

módulos de la misma jerarquía para obtener resultados más complejos. También se ha definido un lenguaje de filtrado de datos orientado al análisis de tráfico. Este lenguaje permite un rápido prototipaje de procesos de análisis al mismo tiempo que permite definir secuencias de distintos procesos de tratamiento de datos. Finalmente, también se han desarrollado herramientas de tratamiento gráfico, tanto para los resultados (gráficas) como para el control de los procesos (GUI). El resultado final es una arquitectura de procesos de análisis de tráfico abierta, que integra, ademas de los procesos de análisis, todas las funcionalidades relacionadas con el tratamiento masivo de datos y la generación de resultados.

La arquitectura del sistema:

Los módulos que forman la plataforma MIRA se clasifican jerárquicamente en tres clases: Captura, Preprocesado, y Análisis. Los procesos de captura dependen de la tecnología del enlace que es objeto de la monitorización (en nuestro caso ATM) y se encargan de capturar el tráfico y proporcionar paquetes IP completos a los módulos de preprocesado. Los módulos de preprocesado analizan el contenido de los paquetes y recopilan toda la información significativa, descartando el resto de información. Los módulos de Análisis son los que se encargan de ofrecer resultados finales y combinaciones de los mismos para conseguir una mayor información (resúmenes, históricos y cruzado de datos). Además, resultados obtenidos con módulos de analisis, sirven como configuración de otros módulos de análisis o de la jerarquía superior (preprocesado). En [2] se presentaron las funcionalidades de las tres primeras clases, y en este artículo nos centraremos en los procesos de análisis.

Los procesos de captura se ejecutan en máquinas especializadas, basadas en arquitectura PC, tarjetas de red ATM y sistema operativo Freebsd. Toda la información capturada es enviada por una red privada de alta velocidad (Ethernet a 100Mbps) a las máquinas donde residen los módulos de preprocesado, y análisis de datos. Estas máquinas son PC’s de altas prestaciones, con 256 Mbytes de memoria, con dispositivos de almacenamiento masivo y de alta velocidad de acceso.

Módulos funcionales

Los módulos de preprocesado son cuatro: módulo de usos, módulo de orígenes y destinos, módulo de análisis de cabeceras de aplicación, y módulo de clasificación de flujos. Cada uno de éstos módulos funciona de forma independiente, y caracteriza la información capturada identificando flujos y caracterizándolos con atributos directamente derivados de la información de los paquetes, o deducidos mediante heurísticos y consultas a bases de datos. Este procesado de la información debe ser ligero y simple, para poder mantener un ritmo alto de captura y procesado en tiempo real.

Una vez preprocesados los datos, los módulos de análisis cruzan los resultados obtenidos, los enriquecen con nuevos atributos y finalmente los cruzan y acumulan para obtener informes útiles para administradores y gestores del sistema.

Tanto los datos obtenidos a partir de los módulos de preprocesado, como los resultantes de los módulos de análisis, siguen todos un formato definido en la arquitectura, que permite su tratamiento desde una herramienta de análisis de propósito general. Esta herramienta, que llamamos filter, es un intérprete de comandos interactivo, programable con un lenguaje de filtrado de datos, que permite definir nuevos módulos de análisis de forma intuitiva.
Se ha definido un lenguaje de script, que permite crear cadenas de procesado de datos, invocando módulos de análisis externos y módulos de análisis en lenguaje de script, de los que se obtienen los resultados finales en forma de informes y gráficas. Además dispone de funciones de acumulados de datos que permiten la generación de resultados parciales y de históricos.

Interfaz de control

El número de módulos que componen el sistema es muy elevado, y la arquitectura abierta definida permite introducir nuevos módulos. Para simplificar la tarea de integración de los módulos bajo un mismo aspecto, y simplificar la gestión de los mismos, se ha definido la interfaz de interacción con los módulos (ejecución, configuración). Se ha definido un lenguaje en el que se especifican los tipos de parámetros que puede recibir un proceso y unas marcas que estructuran estos parámetros en grupos y les dan

significado semántico. La interpretación de los ficheros de configuración por parte de una aplicación gráfica, hace posible la generación dinámica de la interfaz de usuario. Por ejemplo, se generan ventanas para cada grupo de parámetros, se posicionan los parámetros en pestañas, se insertan líneas de ayuda y botones de navegación, se permite la edición de ficheros y la visualización de resultados,... De esta manera, cualquier modulo de MIRA, puede ser gestionado desde una GUI sin necesidad de programar en un lenguaje de este tipo. Cabe remarcar que la mayor parte de los procesos se programan con lenguajes de bajo nivel y que por cuestiones de licencia se pretende obtener un producto de bajo precio.

Las gráficas

Es evidente que la mejor forma de analizar la información es de forma gráfica. Mas aún cuando los volúmenes son muy elevados o la información muy diversa. Los proyectos anteriores confiaron en aplicaciones comerciales para la generación de los gráficos finales. Las gráficas más ricas se consiguieron con aplicaciones en entorno Windows, pero esto suponía un problema de integración con las aplicaciones de captura y análisis (UNIX) además de problemas de licencia. Se evaluaron diversas aplicaciones gráficas en entorno UNIX, y finalmente se opto por el uso de una librería de libre distribución que nos permitió definir todos los tipos de gráficas necesarios en el proyecto. Se trata de la librería GD, sobre la que se ha programado un proceso de nivel superior capaz de interpretar ficheros en formato MIRA, y configurable también con parámetros en formato MIRA (lo que permite su uso desde la interfaz gráfica de usuario).

El resultado

El resultado es un conjunto de procesos para el análisis de tráfico con un conjunto de utilidades que permiten añadir nuevas funcionalidades. La definición de las interfaces entre módulos (ficheros de entrada y salida, parametrización y control) permiten adaptar otras aplicaciones (Netramet[6], NetFlow[7]) de forma fácil. El lenguaje de script permite crear nuevos procesos de análisis que una vez evaluados pueden ser programados de forma más eficiente e integrados al sistema como módulos especializados. El lenguaje de script permite experimentar al administrador del sistema, y decidir el tipo de análisis que desea obtener. Los informes finales pueden ser muy heterogéneos, desde clasificaciones del trafico por tipo de aplicación, por servidor (dirección IP), por dirección de Red, SA, enlace, grupo de entidades, tipo de entidad, tipo de tráfico,...
En el artículo se presentara la arquitectura del sistema mira a nivel de procesos (módulos) y se describirá el módulo de análisis de propósito general filter, junto con el módulo de interfaz gráfica dinámica y de generación de gráficos. Los resultados se apoyaran en medidas de campo realizada con tráfico real.

References

[1] M.Alvarez et al. “CASTBA: Internet Traffic Measurements over the Spanish R&D ATM Network”, HP-OVUA Workshop, Rennes (France), April 1998
[2] P. Lizcano et al “MEHARI: System for Analysing the Use of the Internet Services”,Computer Networks 31 (1999), pag.2293-2307

[3] J.Aspirdof et al. “OC3MON: Flexible, Affordable, High Performance Statistics Collection”, INET’97, Lamasya, june 1997.
[4] . Nevil Brownlee et al. “Traffic flow measurement: Architecture” RFC 2063, january 1997.
[5] . Nevil Brownlee “SLR: A Language for Describing traffic Flows and Specifying Actions for Flow Groups” IETF draft August 1999
[6] . Nevil Brownlee “Traffic Flow Measurements. Experiences with Netramet” IETF RFC 2123, March 1997
[7]Cisco Netflow White paper: http://www.cisco.com/warp/public/732/netflow/nflow_wp.htm