ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad en Ingeniería en Electricidad y Computación

TESIS DE GRADO

“DISEÑO Y AUTOMATIZACIÓN DE LAS POLÍTICAS DE ADMINISTRACIÓN DE REDES DE LA ESPOL”

Previa a la obtención del Título de:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

Presentada por:

Neil Alfredo Núñez Montiel

Guayaquil – Ecuador

Año

2005

AGRADECIMIENTO

Al Ing. Guido Caicedo Director de Tesis por su valiosa ayuda y paciencia, a mis compañeros del CSI pos su colaboración y a todas las personas que de una u otra forma me ayudaron con la elaboración de este trabajo.

DEDICATORIA

A Dios
A mis padres
A mi novia
A mis hermanos

TRIBUNAL DE GRADUACIÓN

Ing. Miguel Yapur
Sub-Decano de la FIEC

Ing. Guido Caicedo
Director de Tesis

MBA Ruth Alvarez
Vocal Principal

Ing. Albert Espinal
Vocal Principal

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral"

Neil Núñez Montiel

RESUMEN

Anteriormente la ESPOL contaba con un computador tipo mainframe que centralizaba los servicios administrativos y académicos y que era accedido a través de terminales. La administración de estos terminales y su acceso al computador principal, era algo sencillo que requería pocas personas.

Con las nuevas redes, y nuevas tecnologías que se instalaron en la ESPOL(ATM, backbone, cliente-servidor, etc.), la administración y mantenimiento del hardware, software y servicios se hace compleja, ya que implica controlar la operación y configuración de algunos equipos de comunicaciones y redes de computadoras, que se encuentran distribuidos geográficamente en los Campus Gustavo Galindo y Campus Peñas. La administración de estos equipos y redes se vuelve una tarea cada vez más laboriosa que demanda el incremento del personal que los administra y la organización y distribución de sus tareas. De allí la necesidad de plantear políticas, y ciertos mecanismos automatizados de administración de redes para un funcionamiento adecuado de las mismas.

Este proyecto plantea el desarrollo de esas políticas sobre la base del estudio de la administración de redes en lo que tiene que ver con la configuración, rendimiento, seguridad, manejo de fallas y manejo de usuarios, además de implementar el uso apropiado de los protocolos y recursos de administración de redes disponibles

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

Capítulo I

Áreas de la Administración de Redes

1.1 Introducción

Años atrás, la responsabilidad de la administración de los sistemas de cómputo estaba centralizada y asignada a un operador que controlaba dicho equipo desde un solo lugar. En la actualidad, el desarrollo de las redes de computadoras ha ocasionado que los sistemas sean distribuidos entre los componentes computacionales de la red cambiando por lo tanto el modelo de administración. La tarea que antes hacía una persona, desde un solo sitio, ahora se distribuye entre expertos ubicados en distintos puntos de la red que controlan diferentes partes de los sistemas lo que ha ocasionado la necesidad de asegurar el funcionamiento correcto y eficiente de la red. Por ejemplo, en una red actual, un problema que surge en un lugar en particular puede deberse a fallas de hardware o software de equipos ubicados en otro sitio, por lo que los administradores tienen que comunicarse, ubicar el problema y luego solucionarlo.

Existen algunos modelos para resolver los problemas de la administración de redes, cada uno de ellos con un esquema propio diseñado para manejar un problema en particular. Sin embargo en todos

ellos el objetivo sigue siendo el mismo: Maximizar la eficiencia y productividad en el uso de la red para garantizar la operatividad de los sistemas computacionales.

Con estos antecedentes, la ISO(International Standars Organization) ha desarrollado un modelo general que establece que el manejo de una red tiene cinco áreas funcionales, que son:

- manejo de fallas,
- manejo de la seguridad,
- manejo de la configuración,
- manejo del rendimiento,
- registro de eventos y administración de usuarios.

A pesar que cada una de las cinco áreas tiene su importancia, la implementación de la administración de redes para un caso particular, no necesariamente cubre todas las áreas, y mucho depende del tipo del ambiente de la red a manejar. Lo que puede ser importante para un ambiente, puede no serlo para otro.

Para entender la relación de las áreas funcionales con la administración de las redes, es necesario definir primero los componentes involucrados en la arquitectura básica de administración de redes que son:

- Dispositivos Administrados.- Son dispositivos de la red como computadoras, u otros componentes que requieren alguna forma de monitoreo.
- Dispositivos Administradores.- Son sistemas computacionales que ejecutan un programa que permite recibir alertas, consultar y verificar los dispositivos a los cuales administra. La consulta de los dispositivos que hace la estación de administración, puede ser automática o iniciada por un usuario. Los agentes en los dispositivos administrados responden a las consultas hechas por la estación de administración, como lo muestra la figura 1.
- Administradores de la red.- Son los encargados de ejecutar las tareas de administración basados en las políticas y procedimientos de administración que posea la organización. En algunos casos, también se encargan de establecer las políticas y procedimientos de administración que la institución necesita.
- Políticas y procedimientos de administración.- Son las reglas que gobiernan la administración de los recursos. Para muchos ambientes, las políticas son un conjunto simple de reglas que pueden o no ser cumplidos.

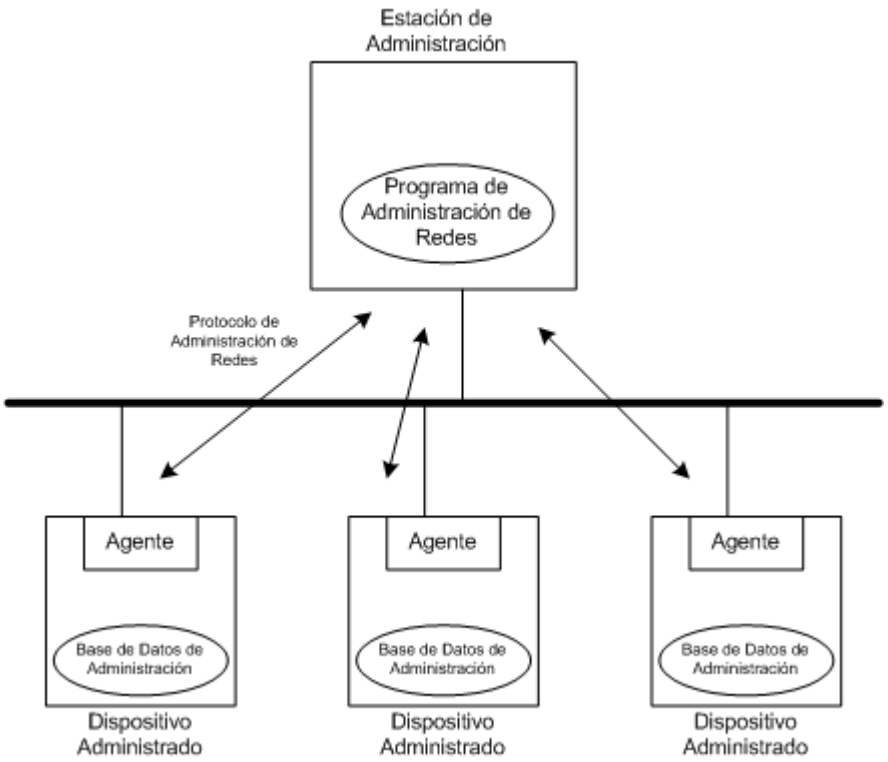


Figura 1: Relaciones entre la Estación de Administración y los Dispositivos Administrados (Ref 1)

1.2 Administración de las Fallas de la Red

El manejo de fallas es el proceso de localizar, aislar y corregir los problemas de la red. De las muchas tareas envueltas en la administración de la red, el manejo de las fallas es probablemente la más importante.

1.2.1 Localización de las Fallas de la Red

La localización de fallas consiste en recolectar información del estado de la red para lo cual se usan dos mecanismos: El primero es la recolección de eventos de la red, y el segundo es la verificación periódica del dispositivo.

En el primer mecanismo, los eventos son transmitidos por los dispositivos de red al sistema administrador cuando éstos experimentan una condición de falla. Esta condición puede ser: La caída de un enlace, la falla de uno de los componentes del dispositivo, el reinicio del componente, etc. En muchos casos, la información enviada no provee toda la información necesaria del evento para un manejo efectivo de las fallas, o en el peor de los casos si el equipo falla completamente, éste no puede notificar el evento a la estación de administración. Esto nos lleva a concluir que los mecanismos basados solo en la recolección de eventos no siempre son efectivos en mantener la información del estado actual de los dispositivos de la red.

El segundo mecanismo para la localización de fallas consiste en examinar periódicamente el estado de los dispositivos para poder encontrar fallas en intervalos de tiempo determinados. Una forma es realizar inspecciones a los dispositivos enviando mensajes a través de

la propia red y esperando como respuesta en el mensaje de regreso una copia del original, para lo cual se usan protocolos como el ICMP Echo, Appletalk Echo, y el Bayan Vines Echo, entre otros.

El examinar automáticamente los dispositivos de manera periódica requiere la utilización de la red. Esta utilización de la red se incrementa a medida que disminuye el tiempo entre cada consulta. Un objetivo de la administración de fallas está en determinar el tiempo entre cada consulta que permita una detección adecuada de fallas y un uso mínimo de la red. Otros factores a considerar son el número de dispositivos a examinar y la velocidad de los enlaces.

Para entender como este mecanismo afecta el tráfico de la red, revisemos el siguiente ejemplo: Si para examinar un dispositivo, cada consulta y respuesta tuviera una longitud de 100bytes, incluyendo los datos y la información de cabecera y para una red con 30 dispositivos, se enviaran 100 bytes por la consulta, el equipo enviaría 100 bytes con la respuesta

(100 bytes + 100 bytes) * 30 dispositivos = 6000 bytes

(6000 bytes * 8 bits/byte) = 48000 bits

O sea que por cada consulta se generarían 48000 bits de tráfico. Si la consulta se realizara cada 60 segundos, en promedio serían

(48000bits / 60segundos) = 800bits/segundo

con lo cual tendríamos la información de la red actualizada cada minuto. En una hora, esto significaría

800 bits/segundo * 3600 segundos/hora * 60 consultas * hora =

172800000bits

O sea aproximadamente 173 megabits de tráfico se generarían para consultar los 30 dispositivos. Dependiendo de la capacidad de llevar tráfico de la red, esto puede o no puede significar un problema. Si quisiéramos reducir el tráfico generado por las consultas podríamos examinar la red cada 10 minutos, lo cual representaría alrededor de la décima parte del valor anterior. Sin embargo, la desventaja sería de que si un equipo falla, el administrador de la red no tendría conocimiento de la falla durante 10 minutos.

Debido a los problemas que tiene cada uno de los métodos explicados, es bueno en muchos casos, utilizar ambos para lograr una localización efectiva de las fallas. Para demostrar el beneficio de complementar los métodos, revisemos los siguientes casos:

Caso 1: Sólo se usa el primer mecanismo para la localización de las fallas

Revisemos la figura 2 y supongamos que en el Sitio 1 el dispositivo B tiene problemas y no puede enviar un evento al sistema de administración que se encuentra en el Sitio 2. Como el dispositivo A no envía un evento de falla del enlace porque el enlace está operativo, *el administrador de la red no tiene conocimiento del problema y por lo tanto no puede tomar las medidas encaminadas a la resolución de la falla*. El administrador se enterará cuando un usuario que se encuentre en el Sitio 1, le notifique verbalmente que no puede comunicarse con los sistemas que se encuentran en el Sitio 2, o un usuario que se encuentra en el Sitio 2 le notifica verbalmente que no puede comunicarse con los sistemas que se encuentran en el Sitio 1.

Caso 2: Sólo se usa el segundo mecanismo para la localización de las fallas

Revisemos la figura 2 y supongamos que el sistema de administración está configurado para que realice consultas cada 10 minutos a todos los dispositivos de los dos sitios. Si de repente se pierde el enlace entre los dispositivos A y B*, el administrador será notificado de la falla la próxima vez que el sistema de administración realice la siguiente*

consulta, lo cual, en el peor de los casos, será 10 minutos después de ocurrida la falla.

Si disminuimos el tiempo entre consultas, el tráfico generado por las consultas puede llegar a saturar el enlace lo cual lo haría inútil para intercambiar información entre los sistemas de los sitios 1 y 2.

Caso 3: Utilización de los dos mecanismos para la localización de fallas

Revisemos la figura 2 y supongamos que los dispositivos A y B pueden enviar eventos al sistema de administración, y también el sistema de administración realiza consultas periódicas de los dispositivos cada 10 minutos.

Si ocurre la falla planteada en el caso 1, *entonces el administrador será notificado la próxima vez que el sistema de administración realice la consulta,* que en el peor de los casos será después de 10 minutos de ocurrida la falla.

Si ocurre la falla planteada en el caso 2, *entonces el dispositivo A envía una notificación al sistema de administración indicando que el enlace entre los dispositivos A y B se ha caído*.

Para los dos casos anteriores se muestra que los dos mecanismos se usan en conjunto para la localización de las fallas.

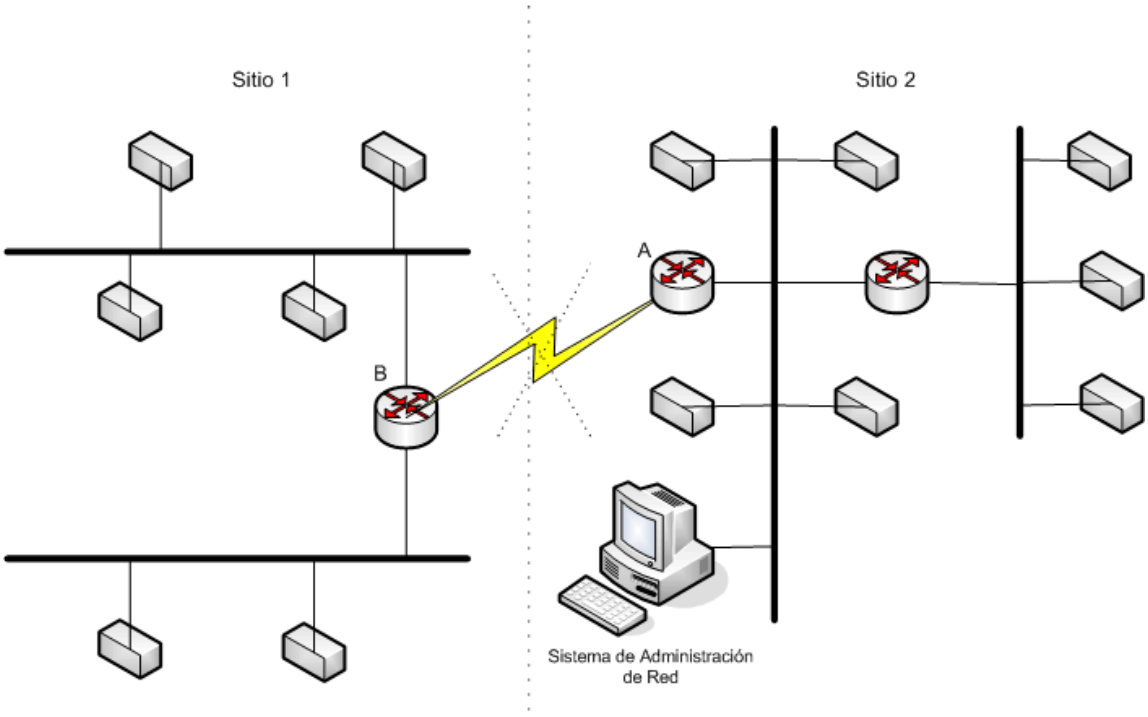


Figura 2: Ejemplo de Ocurrencia de una falla (Ref 1)

1.2.1.1 Elección de las Fallas a Manejar

No todas las fallas tienen la misma prioridad: Algunas requieren que el administrador de la red conozca de ellas para tomar las medidas adecuadas para corregirlas, mientras que otras pueden ser manejadas por el dispositivo administrador sin que el operador se entere. Las fallas que se deben administrar son las críticas para

el ambiente de red en particular, las justificaciones para esta afirmación son las siguientes: 1) si una gran cantidad de fallas ocurre en un mismo tiempo no siempre es posible manejarlas todas; 2) si se limita la cantidad de eventos que los equipos transmiten al dispositivo administrador, el tráfico debido a los eventos se reduce significativamente lo que reduce la utilización de la red para las tareas de administración.

La determinación de las fallas que el administrador debe manejar se establece de acuerdo a los dos factores siguientes:

- El alcance del control que el administrador tiene sobre la red, la cual afecta la cantidad de información que se puede obtener de los dispositivos de la red
- El tamaño de la red

En muchas organizaciones un departamento central se encarga de manejar el backbone(red principal) que puede consistir de routers, bridges, hubs, switches, etc., dejando a los administradores de las redes locales el manejo de sus propios dispositivos los cuales solo tienen influencia sobre un sector del sistema completo(figura 3).

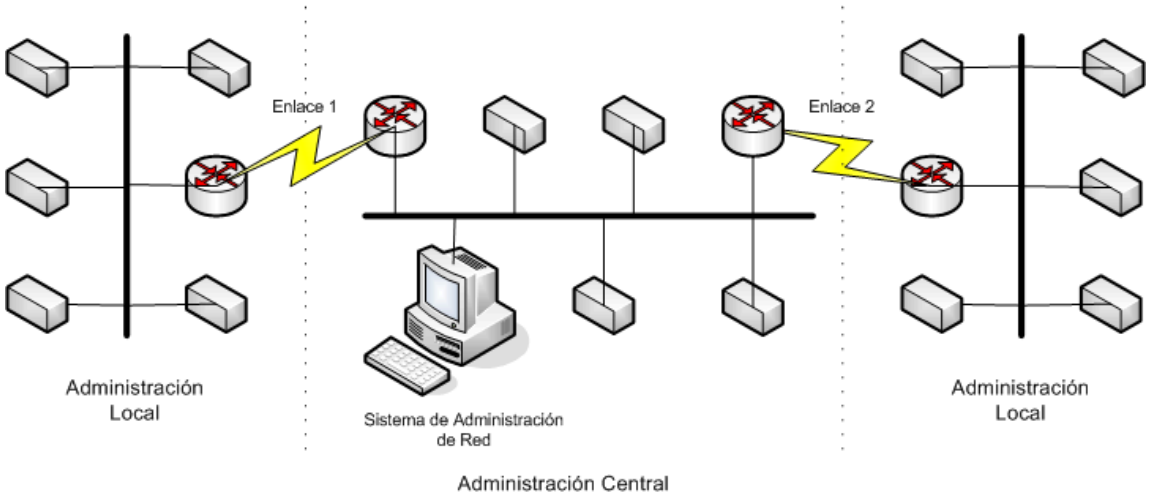


Figura 3: Alcance de la Administración Central (Ref 1)

El tamaño de la red puede influenciar en la decisión de que dispositivos manejar. En una red pequeña que incluye, unos cuantos dispositivos, el administrador de la red puede ser capaz de manejar personalmente todas las fallas. En una red mediana, el administrador puede ser capaz de manejar aquellas fallas críticas de los dispositivos. Finalmente en una red grande solo hay tiempo para examinar los eventos críticos de los dispositivos más importantes.

Otro factor a considerar son los procedimientos que se pueden aplicar cuando una falla específica ocurre. Si por ejemplo un enlace de un router se cae, puede establecerse un procedimiento para que el router automáticamente pueda levantar un enlace de respaldo. Estos

procedimientos preestablecidos pueden automatizar la administración de fallas, pero de todos modos, el administrador debe saber que la falla ha ocurrido para que pueda corregir el problema principal.

1.2.2 Aislamiento de las Fallas de la Red

La información recolectada por los métodos para la localización de fallas puede no ser suficiente. Supongamos que en la red de la figura 2, uno de los dos enlaces tiene problemas. El dispositivo de administración presentará una gran cantidad de eventos, indicando que todos los equipos del otro lado del enlace están fallando, y la tarea de localizar la causa principal del problema se volverá compleja. De situaciones como ésta nace la idea de relacionar los eventos de la red para que la localización de la causa principal de una falla pueda ser hallada y aislada para luego realizar los procedimientos requeridos para la corrección del problema.

Muchos dispositivos administradores usan un motor de inteligencia artificial para relacionar los distintos eventos de la red, pero con el costo de consumir demasiados recursos computacionales. Otros dispositivos administradores relacionan directamente los diferentes equipos para procesar los eventos de la red. Estas relaciones no

requieren gran cantidad de recursos de los sistemas, y además son muchos más prácticos de implementar. En la sección 1.1.4 se revisan las características principales de estas herramientas.

1.2.3 Corrección de las Fallas de la Red

Conociendo la falla, y sabiendo su origen, entonces podemos realizar los procedimientos de corrección que permitan a la red operar con normalidad.

La corrección de las fallas requiere procedimientos y planes para ser utilizados en el momento de la falla los cuales en algunos casos pueden ser automatizados. Por ejemplo, en el ejemplo de la figura 2, en la cual se produce una falla en el enlace, se puede implementar un mecanismo que automáticamente haga que entre en funcionamiento un enlace de respaldo que permita la interconexión entre las dos localidades. Esta automatización la puede realizar la herramienta para el manejo de las fallas en conjunto con las herramientas de configuración de los equipos la cual se revisará en la sección 1.3.2.

1.2.4 Herramientas para el Manejo de las Fallas de la Red en los Dispositivos Administradores

Luego de decidir como la información del estado de la red será recolectada y que problemas requieren manejo, el siguiente paso es la utilización de las herramientas necesarias para el manejo de las fallas. La efectividad de una herramienta depende en gran medida del tipo de información que los dispositivos administrados de la red provean.

Una herramienta simple puede indicar que un problema existe pero no indicar su causa. Por ejemplo una herramienta puede enviar los mensajes ICMP a cada máquina y dispositivo en la red para comprobar la conectividad de la capa de red IP. La información recolectada puede almacenarse en un archivo de registro u ocasionar cambios de colores en un mapa jerárquico. Después de que el dispositivo haya perdido la conectividad, la herramienta deberá alertar al administrador periódicamente de la existencia de la falla. El esquema de funcionamiento de esta herramienta se muestra en la figura 4.

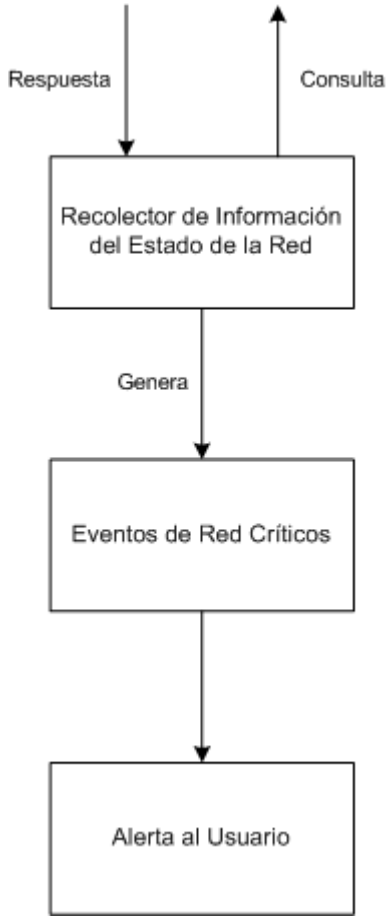


Figura 4: Diagrama de una herramienta de manejo de fallas simple (Ref 1)

Una herramienta más completa para la localización de fallas debe ser capaz de usar los dos mecanismos para la recolección de eventos. Para esto es importante que los dispositivos de la red sean lo suficientemente sofisticados para enviar eventos reportando su estado.

La figura 5 muestra el diagrama de una herramienta compleja para la localización de fallas, en el se muestra que ésta herramienta usa los dos mecanismos para la localización de fallas. En el bloque “Intérprete de los eventos de la red” se encuentra la funcionalidad que permite relacionar los eventos para facilitar la localización de fallas. Por ejemplo, podemos relacionar la utilización de memoria de un dispositivo (un router por ejemplo), y la posible falla del equipo para mantener la transmisión de datos de un sitio a otro.

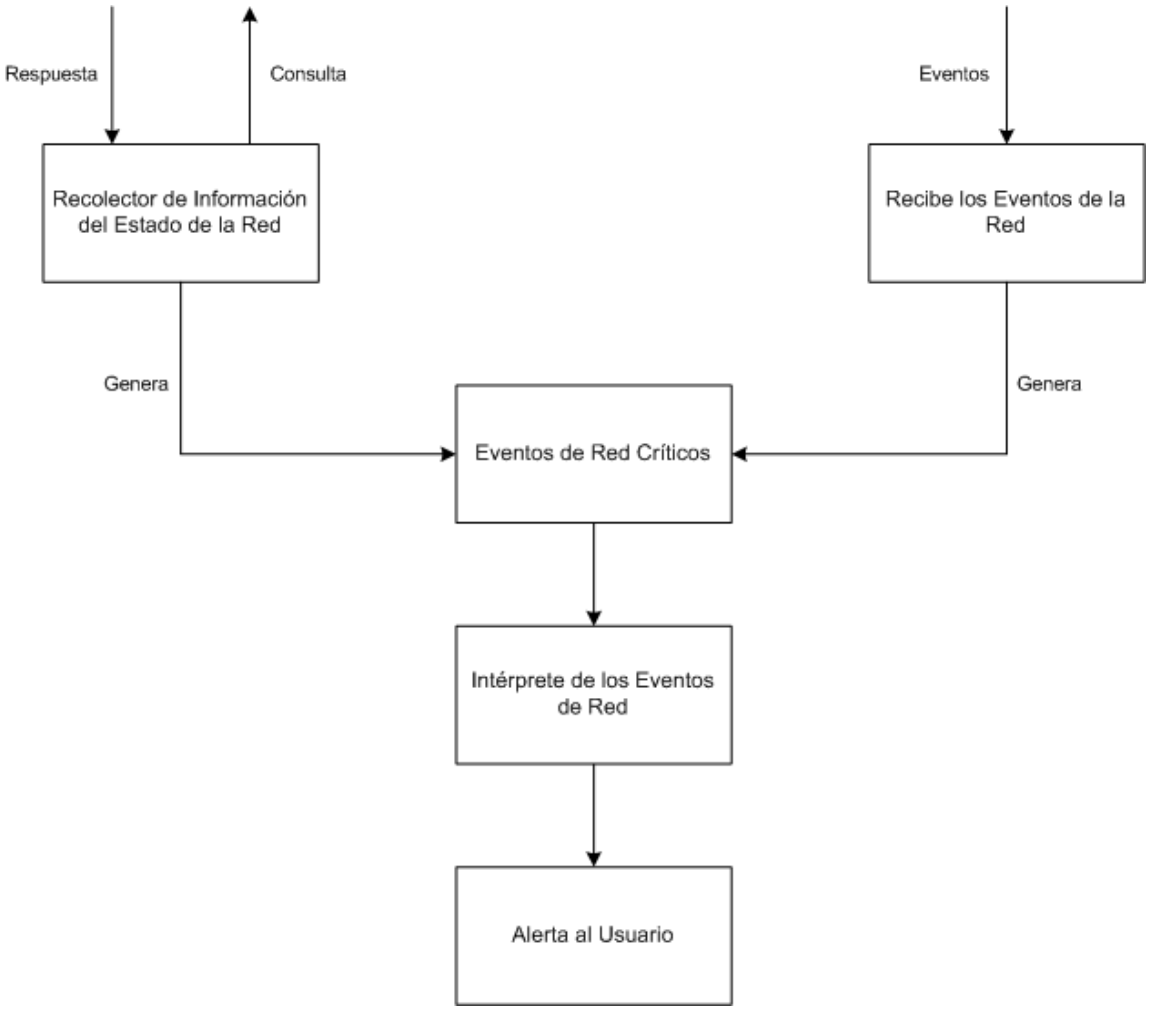


Figura 5: Diagrama de Una herramienta de administración de Fallas Compleja (Ref 1)

1.2.5 Impacto de las Fallas en la Red

Un sistema administrador de las fallas debe ser capaz de relacionar automáticamente como una falla puede afectar otras áreas de la red

ya que solamente así se puede lograr un análisis completo de las fallas.

Consideremos una situación en la cual un enlace de radio conecta dos localidades de una organización. Si el enlace falla, la herramienta informa del problema y reporta: **falla en el enlace del router Sitio A**; esta información es útil, pero, si se reportara: **falla en el enlace del router Sitio A, no hay comunicación entre A y B**; con esta información adicional se sabe que el problema requiere una atención inmediata.

Pero que tal si además del enlace de radio se tuviera una conexión de respaldo entre los dos sitios, de tal manera que si el enlace principal falla, entra en funcionamiento el enlace de respaldo. En este caso el reporte debiera ser**: falla en el enlace del router Sitio A, el enlace principal abajo, comunicación con enlace de respaldo.** De esta forma se sabe que la comunicación está seriamente afectada pero que aún existe.

1.2.6 Maneras de Reportar Fallas

Las maneras de reportar fallas son casi tan importantes como el proceso mismo de manejar las fallas. Las maneras más comunes pueden ser a través de mensajes de textos, mensajes gráficos y señales auditivas.

Los mensajes de textos son los más comunes porque son fáciles de programar, funcionan en cualquier terminal, sea gráfico o no, y no requiere de muchos recursos de hardware. Sin embargo tiene un gran inconveniente: el administrador debe estar siempre atento a los mensajes presentados dificultando la administración. Si la herramienta para el manejo de fallas solo usa mensajes de texto para notificar los eventos de la red al administrador, y ocurre una falla en uno de los equipos de la red, es posible que el administrador no lea el evento indicando la falla, porque no estaba atento o no se encontraba, y por lo tanto no realice los procedimientos de corrección de la falla.

Los mensajes gráficos son los más indicados y efectivos porque son más fáciles, para el administrador, de interpretar y visualizar, pero tiene el mismo inconveniente anteriormente explicado.

Para los mensajes gráficos, el sistema de administración usa una interfaz gráfica de usuario(GUI), que muestra la red en forma de un

mapa jerárquico en lo que por lo general la red es representada por una nube, y cada nodo dentro de la nube puede representar un edificio, ciudad, o un equipo; y además cada nodo puede ser abierto en otro mapa mostrando los nodos que contiene. Cada nodo tiene un color que indica el estado actual del mismo, este esquema de colores puede ser diferente de una aplicación a otra, pero los colores más comunes son:

- Verde: dispositivo normal, sin problemas
- Amarillo: dispositivo puede tener un problema
- Rojo: dispositivo en estado de error
- Morado: dispositivo revisado por el administrador

El verde se usa para indicar que el dispositivo está funcionando correctamente. El amarillo se usa cuando un dispositivo tiene una facilidad de respaldo automática para algún componente que falla; por ejemplo muchos dispositivos como routers, multiplexadores y hubs, tienen fuentes adicionales de respaldo, y cuando una fuente falla, la otra automáticamente entra en operación. El amarillo también significa que el dispositivo ha tenido un error pero lo ha corregido sin intervención manual. Esto se aplica por ejemplo para el caso de equipos que posean la característica de conexión de respaldo en los

que si el enlace principal falla, otro enlace entra en operación. Un dispositivo en color rojo indica que tiene problemas graves como temperatura de operación crítica, etc.; o que el dispositivo ha fallado completamente. También, el color rojo en un dispositivo indica que el sistema de administración no tiene acceso al dispositivo por problemas en la red o por cambios en la configuración del equipo.

El morado se usa para indicar que el dispositivo está siendo revisado por el administrador porque presentó una falla o puede ser usado para indicar que el equipo está en mantenimiento.

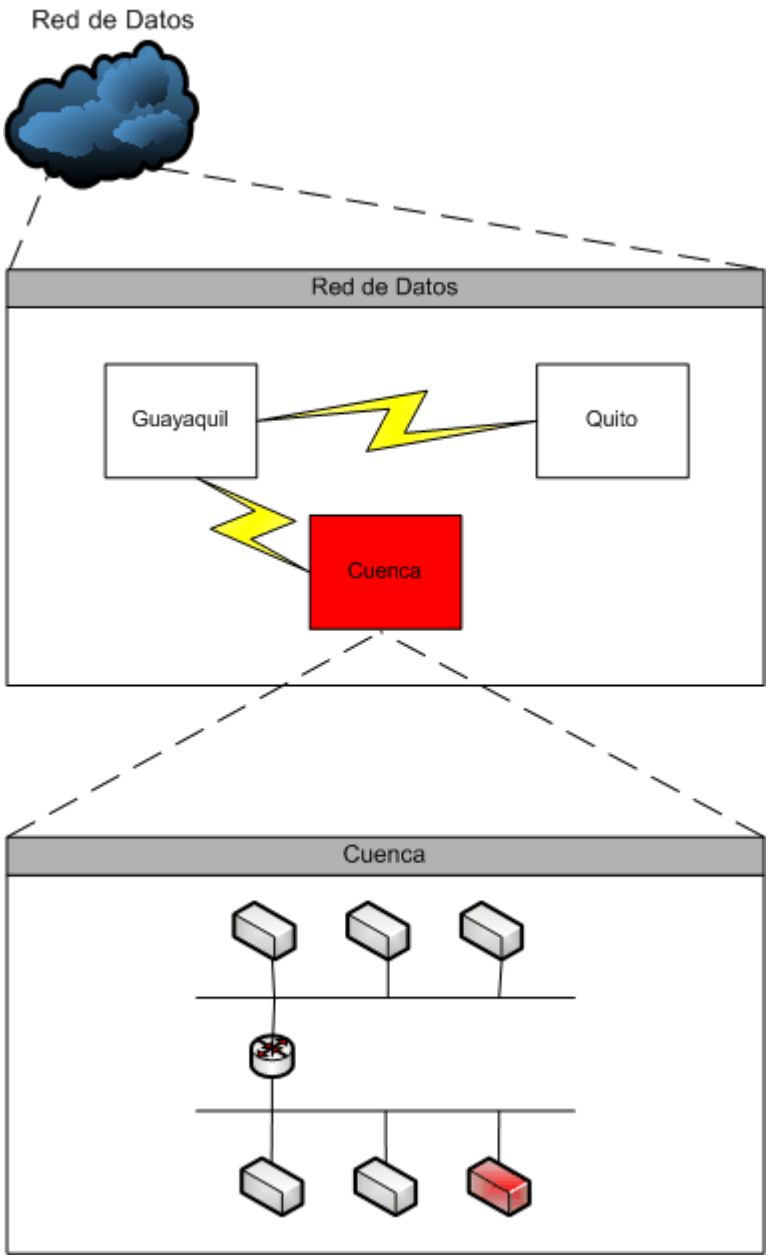


Figura 6: Mapa jerárquico usado para aislar fallas (Ref 1)

Las señales auditivas se usan para llamar la atención del administrador y son una ayuda a los mensajes de texto o gráficos. A más de mensajes de texto y gráficos, también se pueden usar buscapersonas (beepers) para reportar fallas a los administradores cuando están fuera del lugar donde se encuentra la estación de administración.

En la actualidad la combinación de mensajes de texto y mensajes gráficos es la más usada para reportar fallas. Por ejemplo, en la figura 6, el gráfico nos muestra que existe una falla en el dispositivo de color rojo, pero no nos indica el tipo de falla, por lo que un mensaje de texto es útil para indicar el motivo de la falla.

1.3 Manejo de la Seguridad de la Red

La administración de la seguridad tiene como fin principal proteger la información sensitiva en los dispositivos conectados a una red de datos controlando los puntos de acceso a la información[LEIN96]. La información sensible es cualquier dato del que una organización quiere mantener control sobre su acceso y manipulación, como roles de pago, información de clientes, investigaciones y desarrollos de la empresa.

La administración de la seguridad permite a los administradores proteger la información sensible controlando el acceso a las computadoras y dispositivos de red por parte de los usuarios (dentro y fuera de la organización) y registrando los intentos de violar las reglas de seguridad establecidas

El manejo de la seguridad de la red no debe ser confundido con la seguridad de la aplicación, del sistema operativo, o la seguridad física. La seguridad del sistema operativo envuelve configurar la protección de archivos, directorios y programas; la seguridad física tiene que ver con asegurar puertas de áreas de los sistemas, instalar sistemas de tarjetas de acceso, y proveer seguridad en los teclados de las computadoras. Estos tipos de sistemas de seguridad no son parte del manejo de la seguridad de la red, pero sin ellos, la administración de la seguridad será insuficiente.

El manejo de la seguridad es realizado a través de la configuración específica de dispositivos y de computadoras de la red para controlar los puntos de acceso en la red. Los puntos de acceso incluyen: servicios de software, componentes de hardware, y medio de transporte de la red. Los servicios de software son todos aquellos programas que ofrecen acceso a sus recursos a través de la red. Los componente de hardware pueden ser routers, hubs, etc.; o cualquier

otro dispositivo con lo cual un usuario pueda lograr acceso a la red. El medio de transporte de la red, es el medio físico por el cual viajan los datos. Este medio en particular, es un área vulnerable, si una persona tiene acceso al medio que lleva información importante, las medidas tomadas para asegurar la información en las computadoras o dispositivos de red serían insuficientes.

1.3.1 Beneficios del proceso de administración de la seguridad

Uno de los beneficios de tener una red es el de compartir la información contenida en las computadoras. Sin embargo al compartir la información también se corre el riesgo de que ésta pueda ser borrada o alterada, lo que representa un problema de seguridad. Para resolver este problema, una computadora que contenga información importante podría desconectarse de la red para luego transferir la información a través de medios removibles como cintas o discos ópticos. De esta forma solo las personas que tengan acceso físico al sistema pueden acceder a la información. Sin embargo este método aunque seguro, no es eficiente ya que elimina los beneficios de usar una red.

El manejo adecuado de la seguridad de la red puede ofrecer una alternativa más práctica que libere al usuario del problema de la seguridad e incremente la confidencialidad de la red.

Por ejemplo, si se usa un esquema de acceso mediante usuarios y claves se puede tener control de las personas que acceden a la información y los derechos que tienen sobre ella.

1.3.2 Implementación de la Administración de la Seguridad

Supongamos que en una compañía una computadora almacena la información de los roles de pago, la cual la organización la define como información sensible y que además se tiene un punto de acceso a través de la red, usando un programa de terminal remoto. Primero hay que controlar el programa de terminal remoto, asegurándonos que solo las personas debidamente autorizadas tengan cuentas de usuarios para examinar la información. Estas cuentas a su vez deben tener claves asignadas las cuales pueden generarse de manera aleatoria y requiriendo renovación periódica. Realizadas estas tareas uno puede confiar que la información sensible está más segura.

Sin embargo, este perímetro simple de defensa no es suficiente. Adicional al usuario y clave para cada cuenta, las conexiones solo se deberían hacer desde máquinas permitidas, es decir, se debería limitar los puntos de acceso, para obligar a que se use la combinación de usuario, clave y dirección de red.

Aún así no existe garantía de que alguien pueda escudriñar la información en tránsito a través de la red por lo que se hace necesario de alguna forma encriptar la información o restringir el acceso a la misma.

El manejo efectivo de la seguridad requiere que el administrador haga un balance entre la necesidad de asegurar la información sensible y las necesidades de los usuarios de acceder a la información pertinente para realizar su trabajo. La administración de la seguridad envuelve los siguientes pasos:

- Identificar la información sensible
- Encontrar los puntos de acceso
- Asegurar los puntos de acceso
- Mantener la seguridad de los puntos de acceso

1.3.2.1 Identificando la Información Sensible

Como se mencionó anteriormente, el primer paso para la implementación de la administración de la seguridad es determinar que máquinas tienen información sensible. La mayoría de las organizaciones tienen políticas bien definidas de que tipo de información califica como sensible, por ejemplo la información de contabilidad, financiera, clientes, mercado, ingeniería, y empleados; pero lo que es sensible para una organización, no lo es para otra.

Lo más complicado de identificar la información sensible, es encontrar en que computadoras se encuentran los datos ya que éstos pueden estar distribuidos en varios sistemas. Por ejemplo, en una organización que tiene varias sucursales en diferentes sitios, los cuales a su vez tienen servidores locales, el administrador de la organización debe coordinar con los administradores locales para identificar los servidores que contienen la información que la organización considere sensible.

1.3.2.2 Encontrando los puntos de acceso

El primer punto de acceso de cualquier red de datos es el medio de transporte de la red, ya que es el acceso físico a la red. Los demás puntos de acceso son lógicos y son los que proporcionan los servicios de software como un servidor web, ftp, mail, etc., por lo que encontrar estos puntos de acceso es una tarea compleja que requiere la examinación de cada programa que ofrece un servicio de red (algunas computadoras tienen docenas de esos programas). El nivel de privilegio que esos programas tienen influye en los puntos de acceso a la información sensible que está almacenada en la computadora.

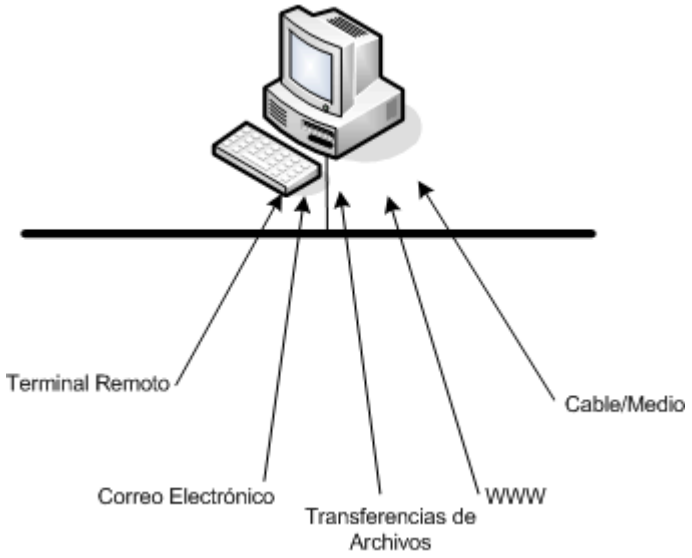


Figura 7: Puntos de Acceso (Ref 1)

Muchos sistemas operativos, en especial Unix, proveen un servicio de terminal remoto, el cual es muy útil, pero que al mismo tiempo se debe configurar correctamente para identificar a los usuarios y restringir sus acciones en el sistema. También hay servicios de transferencia de archivos como por ejemplo el FTP anónimo para el caso de sistemas que posean el protocolo TCP/IP y deseen distribuir software públicamente. Este servicio no requiere clave para identificar a los usuarios sino que da un acceso restringido a ciertos directorios del sistema. Como estos, existen muchos otros servicios como correo electrónico, ejecución remota de programas, servidor de nombres, que representan puntos de acceso en el

sistema que necesitan ser asegurados usando la administración de seguridad.

Además de los sistemas de los que el administrador de la red tiene control absoluto, existen computadoras que los usuarios usan para realizar sus tareas. Algunas ocasiones, sino es siempre, los usuarios instalan programas que pueden ofrecer servicios de red, o peor aún, propagar virus a otros sistemas que están conectados a la red. Para estos casos, los administradores deben usar los procedimientos y herramientas establecidas por la organización para encontrar estos puntos de acceso.

Existen varios procedimientos y herramientas para encontrar los puntos de acceso, entre los cuales están:

- Inspecciones de los procesos del Sistema Operativo: Esta tarea es muy tediosa porque implica realizar inspecciones periódicas a los procesos que están ejecutando en una computadora. Se pueden usar herramientas que pueden ayudar a automatizar esta tarea.

- Registros del Administrador: El administrador debe llevar un registro de todos los procesos que los sistemas deben correr y con las inspecciones de los procesos anteriormente descritos,

hacer una comparación para verificar que procesos que no deben ejecutarse no lo hagan.

- Analizador de Protocolos: Los analizadores de protocolos pueden monitorear todos los paquetes de la red, y son una herramienta excelente para encontrar exactamente que sistemas en la red usan que tipos de protocolos para comunicarse.

- Scanners: Son herramientas que prueban en las computadoras todos los servicios posibles que se puedan ofrecer a la red. Usando estas herramientas, el administrador puede verificar los servicios de cada sistema en la red para llevar un control de los servicios que ofrecen.

En una red se puede utilizar información considerada no sensible para explotar debilidades y así llegar a la información sensible. Esta información no sensible puede ser versiones del sistema operativo, direcciones de red, etc.; si esta información no sensible llega a conocerse por una persona inescrupulosa puede explotar debilidades que tenga esa versión de sistema operativo o alguno de los servicios que ofrece.

Una computadora que contiene información sensible y a la que siempre le debemos dar la importancia debida, es el sistema de administración de la red. Esta computadora en especial contiene información de toda la red, y además es un punto de la red que tiene acceso completo por lo que se le debe dar una consideración especial de seguridad.

1.3.2.3 Asegurando los puntos de acceso

El siguiente paso en la administración de la seguridad es aplicar las técnicas de seguridad necesarias. La seguridad puede ser usada en múltiples capas de la red:

- En la capa de enlaces de datos se puede usar encriptación
- En la capa de red se puede controlar el flujo del trafico a través de filtros de paquetes
- En cada sistema cada punto de acceso de información puede tener un servicio asociado, y cada servicio puede proveer uno o más tipos de seguridades: autenticación de host, autenticación de usuario, y autenticación de llave o esquemas más sofisticados

A continuación trataremos con más detalle cada una de estas técnicas de seguridad.

1.3.2.3.1 Encriptación o cifrado

La encriptación es el proceso de traducir la información a una codificación diferente usando una clave. Este proceso puede ser revertido aplicando la misma clave o una complementaria.

La encriptación es muy útil cuando se envía información sobre enlaces satelitales, microondas, radio, o cualquier medio de transmisión compartido, donde es susceptible de ser interceptada por cualquiera no autorizado. Algunas organizaciones usan encriptación para sus redes locales para asegurarse que el cableado no sea un punto de acceso no autorizado. Lamentablemente esto puede ser muy costoso, porque requiere dispositivos de hardware o software para realizar la encriptación en cada equipo de la red. Además, la encriptación en una red local puede hacer difícil encontrar los problemas y las fallas.

En la encriptación se puede usar dos técnicas para el uso de claves: simétrica y asimétricas. La técnica de clave simétrica consiste en que el emisor y el receptor usan la misma clave para cifrar y descifrar la información. Debido a que la clave puede ser

comprometida(alguien puede averiguar la clave), es necesario cambiar la clave tanto en la fuente como en el destino frecuentemente.

Las ventajas de la clave simétrica son: velocidad del algoritmo de cifrado y la facilidad de cambiar la clave. Las desventajas son: que la clave puede ser interceptada; que las claves pequeñas pueden decodificarse fácilmente, y se requiere una clave por cada par fuente/destino. Ejemplos de algoritmos de clave simétrica son: DES, RC2, RC4[SEG98].

La técnica de claves asimétricas no usa la misma clave para el receptor y el emisor. Este tipo de encriptación usa un algoritmo que tiene una clave dividida en dos. Una parte es privada y la otra es pública. Sabiendo solo la clave pública, se puede cifrar un mensaje que puede ser descifrado sabiendo la clave privada. Si solo se conoce la parte pública de la clave, no es posible cifrar y descifrar un mensaje.

Las ventajas de las claves asimétricas son: mayor seguridad, evita proliferación de claves, posibilidad de implementar firmas digitales. Las desventajas son: se necesita de un administrador que genere o distribuya las claves, las claves son difíciles de revocar y además los algoritmos son computacionalmente costosos [SEG98].

En la actualidad se usan algunos protocolos que utilizan encriptación para sus comunicaciones, entre los cuales están: SSL, IPv6, PPTP, etc.

1.3.2.3.2 Filtrado de paquetes

Muchos dispositivos de red, como bridges, switches y routers, pueden filtrar paquetes basados en la dirección lógica o dirección física de red. El filtrado de paquetes consiste en detener los paquetes que vienen desde o van hacia computadoras inseguras antes de que éstos alcancen un punto de acceso que pueda comprometer la información. Aunque este esquema puede ayudar en la seguridad, también presenta sus problemas:

- Primero, los filtros deben ser configurados para cada uno de los dispositivos de red. Si se añade un nuevo dispositivo, o se cambia la dirección de otro, es necesario cambiar los filtros.

- Segundo, los filtros no funcionan si las computadoras cambian su dirección sin conocimiento del administrador de la red.

Los filtros basados en la dirección mac, o dirección física del dispositivo, son mucho más efectivos, pero más difíciles de

configurar y administrar, en cambio, aquellos basados en la dirección lógica de red son más flexibles y su administración es más fácil que los anteriores, pero en cambio no permiten un control absoluto.

1.3.2.3.3 Autenticación de host

Permite el acceso a los servicios basado en un identificador del sistema fuente, como puede ser la dirección lógica de red, o a dirección mac.

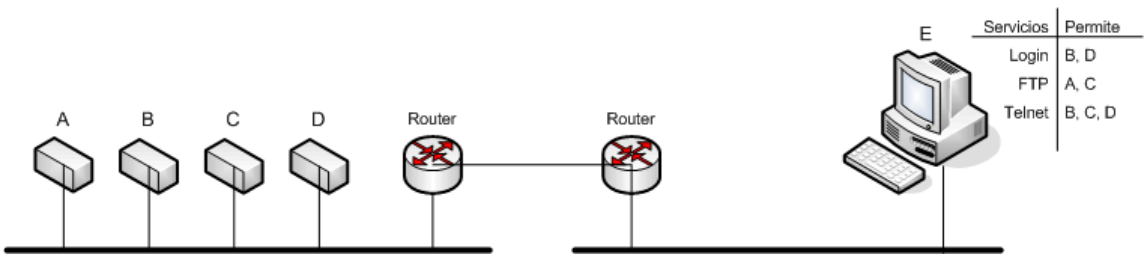


Figura 8: Autenticación de Host (Ref 1)

Debido a que la autenticación de host generalmente está basada en una dirección, muchos dispositivos como bridges, routers o productos de software pueden ayudar a realizar esta tarea y de esta forma liberar de la tarea de autenticación a la computadora destino.

Este esquema de autenticación es útil para proveer seguridad para algunos puntos de acceso, pero no es perfecto. Por ejemplo, si observamos la figura 8, el usuario que trabaja en la computadora A, puede cambiar de computadora, o cambiar la dirección lógica, y de esta forma acceder a la información sensible que se encuentra en la computadora E. Por lo general la autenticación de host se complementa con la autenticación de usuario que se explica a continuación.

1.3.2.3.4 Autenticación de usuario

La autenticación de usuario permite identificar a cada usuario antes de tener acceso al servicio. Provee un mejor control de acceso a los servicios que la autenticación de host porque permite a cada servicio identificar a un individuo en lugar de una computadora.

Un método comúnmente usado para distinguir a los usuarios es la clave, la cual es un código que sólo el usuario sabe y que debe ingresar para acceder a los servicios. Las claves no siempre son tan seguras como uno podría creer. Un problema cuando se usan claves es que algunos servicios de red transmiten las claves tal como se escriben lo cual hace fácil descubrir la clave si se capturan los paquetes de la red. Una solución común es enviar la clave encriptada, usando otros protocolos como SSH, SSL, etc. Otro problema es que los usuarios para recordar sus claves usan palabras comunes que son también fáciles de descubrir por medio de programas que prueban palabras de diccionarios. La alternativa es proveer claves que no sean palabras comunes e incluir en ellas caracteres especiales o números. Existen productos en el mercado, tanto de hardware como de software, que facilitan la administración de claves, y proveen otros mecanismos adicionales como claves que se usan una sola vez, tarjetas magnéticas, etc.

La combinación de autenticación de host y de usuario provee una mayor seguridad en los puntos de acceso que si sólo usáramos uno de los dos.

1.3.2.3.5 Autenticación de llave

Este sistema combina los dos tipos de autenticaciones antes mencionadas, autenticación de host y de usuario. La llave es la combinación de usuario y dirección lógica de red. La autenticación de llave funciona asignando a una computadora en la red la tarea de servidor de llaves. La llave es un mensaje que contiene el nombre de usuario, la dirección de red de la máquina donde se encuentra el usuario, el servicio que el usuario requiere y una firma digital que valida la llave. Este mensaje es encriptado antes de viajar por la red.

Para entender este mecanismo revisemos la figura 9: El servidor de llaves es responsable de entregar las llaves a usuarios autorizados. Cuando un servicio de un computador destino es requerido(1), la computadora fuente consulta al servidor de llaves pidiéndole una llave(2). El servidor de llaves luego pide al usuario que ingrese una clave para autenticación. El servidor de llaves puede identificar tanto a la computadora fuente como al usuario que solicita el servicio. Basado en estos datos y en las reglas de seguridad que residen en el servidor de llaves, éste decidirá si entrega o no una llave válida(3). Con una llave válida la computadora fuente solicita de nuevo el servicio a la computadora destino(4). Este sistema funciona porque la computadora destino

solo dará acceso al servicio si la petición de acceso está acompañado de una llave válida. Algunos servidores de llaves entregan llaves que pueden durar por un período de tiempo. El efecto de esto, es que se termina la sesión del usuario que excede el tiempo, y asegura que una llave no pueda tener un acceso indefinido

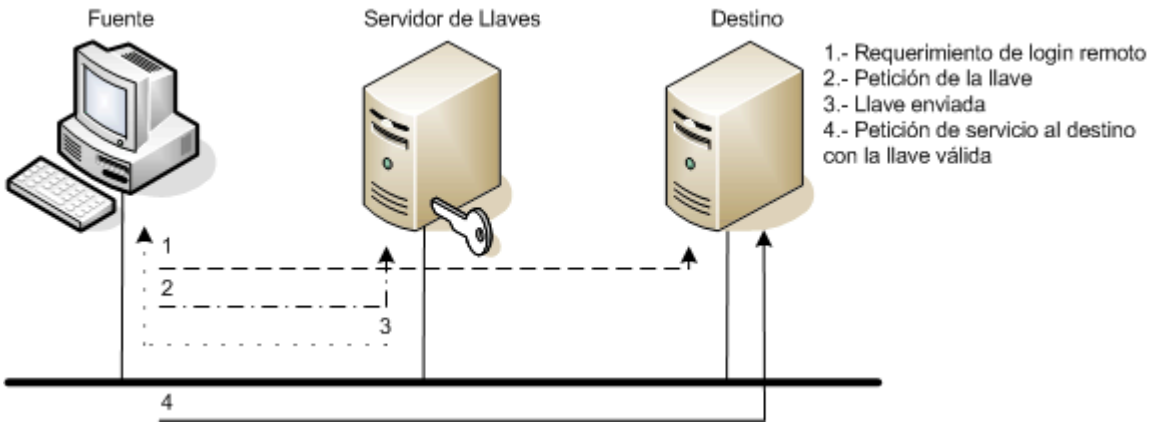


Figura 9: Ejemplo de autenticación de llave (Ref 1)

El servidor de llaves juega un papel importante en este tipo de autenticación, por lo que es crítico para mantener los puntos de acceso a los servicios de red. Es importante que este servidor esté correctamente configurado y administrado. Además, para que este esquema funcione, se requiere que los servicios en la

computadora fuente pidan una llave antes de empezar la transacción y que los servicios en la computadora destino, solo acepten peticiones de servicios si estas están acompañadas de una llave válida, por lo que todas las aplicaciones de las máquinas fuente y destino estén diseñados o adaptados para funcionar de esta manera.

Un ejemplo de autenticación de llave es el esquema de seguridad Kerberos, desarrollado por el Instituto de Tecnología de Massachusetts.

1.3.2.4 Mantener seguros los puntos de acceso

El último paso para asegurar los puntos de acceso es el mantenimiento. Como hemos visto anteriormente, mantener las medidas de seguridad de las redes actualizadas y seguras es una tarea compleja que requiere esfuerzo de la organización tanto en tiempo como en recursos. Una vez establecido un sistema de seguridad, el mantenimiento y modificación del mismo se hace necesario.

La clave del mantenimiento es localizar los huecos de seguridad potenciales o actuales. En algunos casos esto es realizado por los ingenieros responsables de auditar la seguridad de la red. La herramienta básica para auditar la seguridad de la red es la documentación de los potenciales puntos de acceso y la seguridad requerida para cada punto de acceso. Sin embargo, mantener un documento de esa naturaleza con la cantidad de software que existe en la mayoría de organizaciones es una tarea muy complicada por lo que para ayudar a este proceso deben existir y seguirse políticas y guías de seguridad de la organización.

Para vigilar que la seguridad esté bien implementada, se pueden usar programas que intenten explotar errores conocidos de seguridad probando claves al azar, puertos, y errores en los sistemas. Y luego presenten un informe indicando el éxito o fracaso de la prueba. Estos programas son muy útiles porque permiten encontrar huecos de seguridad y realizar las correcciones necesarias antes de que personas no autorizadas puedan acceder a la información a través de estos huecos de seguridad.

Desafortunadamente ningún método garantiza que la seguridad puede ser mantenida apropiadamente. Las auditorías no pueden realizarse todos los días, ni los programas que prueban la

seguridad pueden chequear cada hueco posible. Por lo tanto se deben también establecer puntos de control en la red donde se limita o monitorea los accesos a los servicios que existen a ambos lados del punto de control mediante herramientas de hardware y software comúnmente conocidos como firewalls que se describen más adelante.

1.3.3 Conectándose a una red pública

Los pasos descritos anteriormente son útiles cuando la red de la organización no se conecta a una red pública. Si la organización decide conectar sus redes a una red pública, la implementación de la seguridad no puede sólo basarse en los pasos descritos anteriormente, sino que es necesaria una perspectiva adicional.

Hay tres tipos de acceso posibles desde una red pública a la red de la organización:

- Ningún acceso: Esto puede darse durante períodos de prueba o falta de la conexión y debido a esto no hay necesidad de revisar la seguridad. Por ejemplo, una red privada que no permite login remotos desde la red pública puede usar su conexión para enviar y

recibir correo electrónico. La conexión puede establecerse en ciertas horas a través de un módem para enviar y recibir el correo. Todas las transacciones con la red pública deben ser iniciadas desde la red privada. Con este método no se necesita encontrar los puntos de acceso a la red pública porque no existen.

- Acceso total: Este tipo de acceso implica un acceso similar al de la red interna pero sin la posibilidad de administrar la seguridad de otras redes. Todo el manejo de la seguridad debe residir en cada computadora dentro de la organización lo que resulta demasiado complicado y es altamente peligroso.
- Acceso limitado: Este tipo de acceso, implica un acceso restringido a las redes de la organización. Por ejemplo, en ocasiones las empresas desean poner algunos servicios a disposición de las redes públicas, como páginas web, servicio de correo electrónico pero no la mayoría de servicios internos. Obviamente poner servicios para que sean accedidos a través de redes públicas representa un riesgo de seguridad, pero menor que cuando se permite un acceso total. El limitar el acceso puede ayudar en la administración de la seguridad.

El acceso limitado significa autorizar solo a un grupo de computadoras o usuarios a proveer o utilizar servicios entre la organización y las

redes públicas. Es recomendable que este conjunto de computadoras o usuarios no tengan acceso a los sistemas dentro de la red privada, y si lo tienen, manejarlo con mucha cautela y prestar la debida atención en la configuración de la seguridad. El conjunto de computadoras o dispositivos de red que son colocados entre las redes privadas y las públicas para manejar el acceso limitado y administrar la seguridad, se conocen como firewall, y permite al administrador controlar cada computadora que ofrece servicios a la red pública, limitando los servicios disponibles y mejorando la seguridad de los puntos de acceso a las redes privadas.

Un firewall es una combinación de características de administración de seguridad en una serie de dispositivos de red diseñado para permitir solamente el paso de información de un lado a otro dependiendo de muchas variables como: dirección, usuario, servicio, sentido del requerimiento, etc. Por ejemplo, en la figura 10, el firewall sólo permitirá a las computadoras el acceso al servidor web de la organización, pero impedirá el acceso a los demás servicios que provee la red privada.

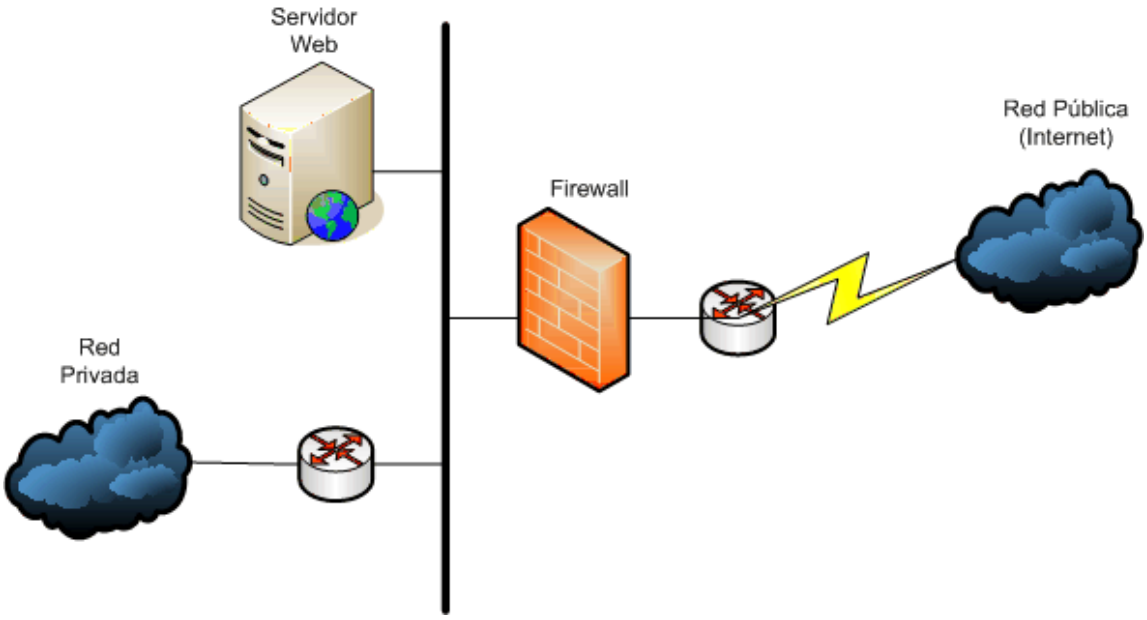


Figura 10: Acceso limitado (Ref 1)

Muchos de los firewalls comerciales tienen funciones adicionales como filtrado de paquetes, registro de eventos, detección de virus, etc. que ayudan en la administración de la seguridad permitiendo un mejor control de las conexiones de la red.

1.3.4 La Seguridad en un Sistema Administrador de Redes

La seguridad en el sistema administrador de la red se maneja a través de herramientas de software, cuyo nivel de sofisticación influye en su efectividad.

Una herramienta de administración simple debe mostrar información pertinente a la seguridad relacionada a los diferentes dispositivos y su estado. Adicionalmente, para permitir localizar todos los dispositivos en la red que restringen a un usuario o dirección de red, esta herramienta debe consultar la base de datos de la configuración y producir un reporte con la información necesaria. Con los datos de la configuración que se encuentran en el sistema de administración de la red se puede resolver problemas complejos de conectividad en la red.

Un problema común es que muchos sistemas administradores no tienen un método estándar para almacenar la información de la configuración de los dispositivos, y muchas veces es difícil determinar que bloques lógicos de la configuración se relacionan con la seguridad de la red. La única solución para los administradores en este caso, es tener herramientas que puedan mostrar porciones de la configuración de los dispositivos para luego realizar una inspección manual de las características de seguridad que están habilitadas. Esto presenta la desventaja de no tener una visión global de la seguridad de la red.

Una herramienta más avanzada puede ser diseñada para incluir una aplicación que monitoree en tiempo real los puntos de acceso a la información sensitiva. Después de detectar un problema potencial de seguridad, esta aplicación cambia el color de los dispositivos afectados en el mapa gráfico de la red, o puede sonar una alarma o mostrar una ventana indicando el evento tal como se muestra en la figura 11. Usando un protocolo de administración, se puede consultar a los dispositivos acerca de los eventos de seguridad que tengan almacenados, por lo menos cada hora, y observar variaciones en los mismos.

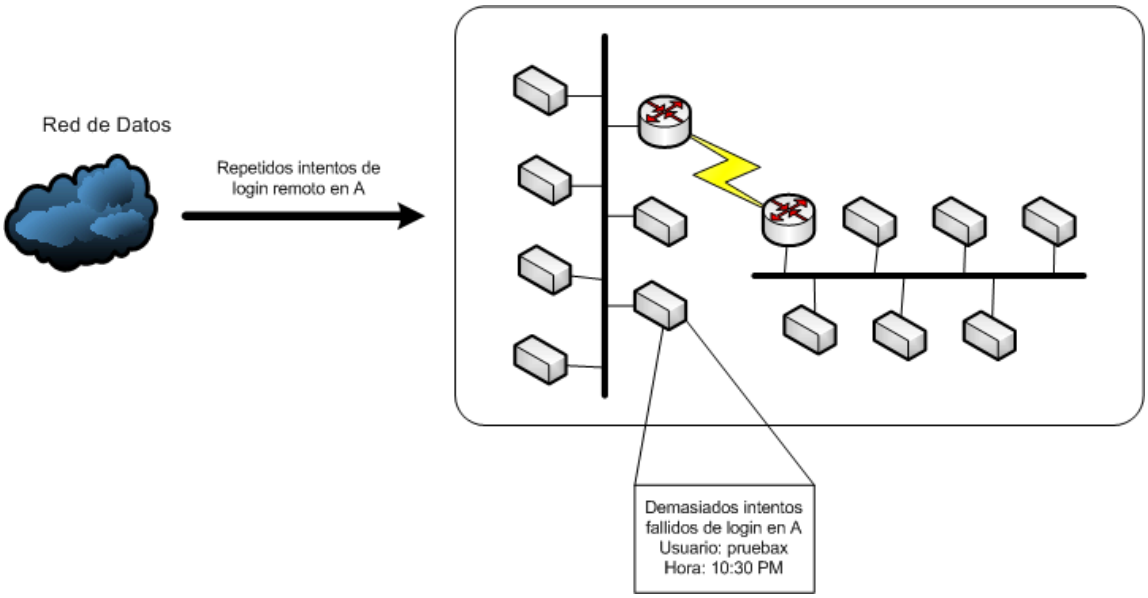


Figura 11: Reporte de una aplicación de administración de seguridad en tiempo real (Ref 1)

Además sería ideal que cada computadora envíe los eventos al sistema administrador de la red acerca de los intentos fallidos de entrar en su sistema. De otra manera, el sistema administrador de la red necesitaría examinar los archivos de registros de la computadora para identificar posibles violaciones de acceso lo cual requeriría que la aplicación de seguridad examine formatos de archivos específicos haciendo el monitoreo muy complejo de implementar.

Una herramienta de administración sofisticada permite automatizar procesos que dan la posibilidad de producir reportes de accesos negados al sistema o archivos que contienen información sensible. También podría permitir usar los datos recolectados acerca de los patrones de tráfico para mostrar al administrador las implicaciones operativas que tiene la imposición del esquema de seguridad. Esta aplicación examinaría el tipo de seguridad que se instalará en una computadora o dispositivo y alertaría al administrador de las repercusiones de ese tipo de instalación. Por ejemplo, supongamos que tenemos dos redes separadas: una con los usuarios y otra con los servidores, y queremos configurar filtros entre las dos redes para limitar el acceso a los servidores. Si la herramienta usada para la administración de la seguridad me informa que la implementación de los filtros puede ocasionar un bloqueo del tráfico entre las redes del 90%, entonces el administrador puede verificar si los filtros son los

adecuados para las redes antes de proceder a la configuración de los mismos.

1.3.5 Maneras de Reportar los Eventos de Seguridad

La clave para asegurar que una red de computadoras está a salvo es llevar un registro de los eventos de seguridad y un proceso que los monitoree. Bajo este esquema la seguridad del registro, que normalmente se encuentra en el sistema de administración, es muy importante. Los reportes deben resumir la información de la seguridad que son críticos para el manejo de ésta.

Con la ayuda de aplicaciones que guardan la información de la seguridad en un archivo de eventos, se pueden determinar patrones de intentos de ingreso a un punto de acceso no autorizado y así detener el acceso. También, el archivo de eventos, puede ayudar a encontrar peticiones no autorizadas que resultan de una mala configuración de la seguridad.

Un ejemplo de un reporte generado a partir del registro de eventos de la seguridad es como se muestra la tabla 1. Se puede notar que la ayuda que presta es fundamental para la seguridad ya que muestra

los intentos fallidos de ingresar a servicios a los cuales los usuarios no tienen acceso.

Transferencias invalidas de archivos:			
Reportado por	Fuente	Usuario	Hora
Espolserver	Bsistema	Teodoro	01:32
Espolten	Asistema	Ursula	05:32
Espolten	Asistema	Ursula	07:32
Espolten	Asistema	Jamil	23:55

Intentos inválidos de login remoto:			
Reportado por:	Fuente	Usuario	Hora
Espolserver	Bsistema	Teodoro	01:32
Espolpuerta	Asistema	Ursula	05:32
Espolman	Dsistema	Maria	07:32
Espolten	Dsistema	Pedro	13:40
Espolten	Csistema		13:53
Espolten	Bsistema		13:54

Tabla 1: Reporte resumido del Registro de Seguridad (Ref 1)

El administrador debe revisar periódicamente los registros para detectar los intentos de violaciones de seguridad y tomar las medidas correspondientes.

1.4 Manejo de la Configuración de la Red

La administración de la configuración es el proceso de obtener datos de la red y usar esos datos para manejar la configuración de todos los

dispositivos de la red [LEIN96]. La administración de la configuración involucra recolectar información acerca de la configuración actual de la red, usar los datos para modificar la configuración de los dispositivos de la red, almacenar la información, mantener un inventario actualizado, y producir reportes basados en los datos recolectados.

El manejo de la configuración mejora el control que tiene el administrador de la red sobre el estado de los dispositivos dando un acceso rápido a los datos vitales de los equipos. Estos datos vitales pueden ser: versión del código, número de interfaces, direcciones lógicas asignadas a las interfaces, números de serie, etc. Teniendo estos datos almacenados en un solo lugar, es posible añadir nuevas interfaces a los equipos, saber que equipo no tiene la última versión de código, etc., sin necesidad de movilizarse al lugar donde se encuentra el dispositivo. Además ayuda al administrador dándole un inventario actual de los componentes de la red con el cual es posible determinar cuantos dispositivos específicos existen en la red y su estado.

La facilidad de inventario que provee la administración de la configuración no se limita a llevar un registro de las configuraciones de los componentes sino también a almacenar información del vendedor, garantías y contratos de mantenimiento.

Los datos del inventario de la red deben ser considerados confidenciales. Por ejemplo si alguien ajeno a la administración de la red sabe que existe un error en una versión de código de cierto componente, fácilmente puede alterar el funcionamiento del equipo llegando a inutilizar la red o a violar la seguridad de la misma.

La administración de la configuración involucra las siguientes acciones:

1. Recolectar información de la red. Una falla en la implementación de este paso puede resultar en desperdicio de tiempo en la resolución de un problema de la red causado por errores en la configuración. La recolección de los datos puede hacerse manual o automáticamente por el sistema.
2. Usar la información recolectada para modificar la configuración de un dispositivo de red. Debido a que la red cambia continuamente, la habilidad de modificar la configuración actual en el momento preciso es esencial. El cambio puede ser manual o automático.
3. Almacenar la información y mantener un inventario actualizado de todos los componentes de red y producir varios reportes.

1.4.1 Recolectando la información de la configuración

La obtención de la información de la red empieza por un esfuerzo manual, que consiste en almacenar el número de serie del equipo, la

dirección lógica del equipo, su ubicación, etc. en una hoja de cálculo, archivo de texto o base de datos. Realizar este paso de esta forma puede dar resultados aceptables, pero mantener todos los datos actualizados en una red que cambia constantemente puede ser un proceso difícil, lleno de errores, monótono y requerir mucho tiempo. Esta forma de hacerlo resultaría en la mayoría de los casos inapropiada si son demasiados componentes, o las redes están distribuidas en regiones geográficas diferentes.

Las fallas de la recolección manual[1] de los datos pueden ser superadas si para esta tarea se usa un método automático como la utilización de un protocolo de administración de red. Por ejemplo para el descubrimiento automático de todos los dispositivos de la red, un método muy usado es enviar una petición de un requerimiento ICMP a todas las direcciones posibles de la red; cuando un dispositivo responde la petición se establece la existencia del dispositivo, y entonces se utiliza un protocolo de administración como SNMP para obtener información más detallada del componente. Esta forma de descubrimiento tiene la ventaja de hallar cada dispositivo funcional de la red, pero tiene la desventaja de enviar peticiones a dispositivos que no existen y consumir los recursos de la red. Otro método, es

[1] Se refiere a dar commandos específicos

encontrar primero un dispositivo en la red y consultarle, por medio de un protocolo de administración de red, con que dispositivos se ha comunicado recientemente. Por cada uno de los dispositivos hallados, la herramienta usa un protocolo de administración de red para consultar y conocer información relevante del equipo. Este mecanismo tiene la ventaja de descubrir rápidamente los componentes, pero tiene la desventaja de usar un protocolo de administración de red que puede no ser soportado por algunos dispositivos además de no poder hallar componentes que no se han comunicado con otro dispositivo en cierto período de tiempo.

El descubrimiento automático de dispositivos y sistemas ayuda a producir un mapa de la red usando un proceso llamado mapeo automático. Es necesario modificar el mapa producido para reflejar la distribución geográfica o funcional de la red, y esta modificación puede ser manual o automática según la funcionalidad del programa.

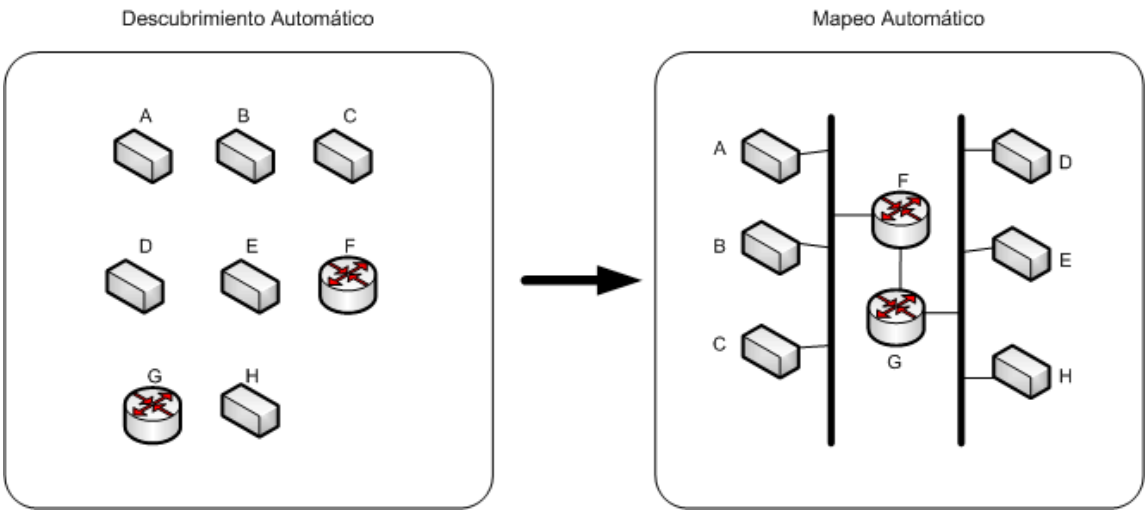


Figura 12: Descubrimiento Automático y Mapeo Automático (Ref 1)

El consumo de los recursos de la red requerido para la recolección de información de la configuración, puede influenciar la decisión de que método usar, aunque los beneficios de la automatización fácilmente justifican los costos. Como en el manejo de las fallas, la frecuencia a la cual los dispositivos son consultados afectará el incremento del tráfico de la red, pero como la configuración de los dispositivos de red cambian muy de vez en cuando, la consulta puede realizarse en períodos distanciados manteniendo bajo el tráfico generado por la configuración de la red.

Una vez que la información de la configuración es obtenida, usualmente será necesario actualizarla (para lo cual se usa los

métodos descritos anteriormente). Consideremos una red con 5000 nodos: si sólo el uno por ciento de esos nodos requieren de un cambio cada semana, el administrador tiene que controlar el cambio de 50 nodos a la semana. Un dispositivo simple puede tener docenas de parámetros configurables que tienen que controlarse lo cual implica que un método manual puede ser ineficiente para este proceso. Además, si no se mantiene un registro de las modificaciones realizadas y los pasos seguidos para realizar los cambios, se puede causar confusión cuando otro administrador examine la configuración de los dispositivos. En cambio, si el sistema de administración realiza el manejo de la configuración del dispositivo, se puede automáticamente mantener el estado anterior de la configuración. Como una ventaja adicional, el sistema de administración de la red podría verificar que los cambios de la configuración sean apropiados para el dispositivo y alertar al administrador antes de configurar incorrectamente el equipo.

Un sistema eficiente de administración debe almacenar la configuración completa de la red en un lugar central de fácil y rápido acceso, sin importar el método usado para el almacenamiento, debe mantener la consistencia y disponibilidad de los datos.

El registro de los datos a través de un archivo de texto es un método muy usado que tiene tres ventajas: facilidad de lectura (tanto por humanos como por programas), facilidad de acceso desde lugares remotos, y una estructura que es fácil entender y administrar. Sin embargo, los archivos de texto usan un considerable espacio de almacenamiento, y la estructura de estos archivos pueden hacer lento el acceso durante los procedimientos de búsqueda. Otra desventaja es que no provee una manera simple y eficiente de representar relaciones complejas de datos.

Una alternativa más eficiente de registro es la de usar un sistema administrador de base de datos (DBMS). Un DBMS ofrece más ventajas que los archivos de texto para el almacenamiento de datos porque:

- Almacena la información eficientemente ocupando un menor espacio y permitiendo que grandes cantidades de datos puedan estar en una sola máquina.
- Almacena los datos en un formato propio, que permite búsquedas rápidas de datos específicos.
- Puede automáticamente ordenar los datos de varias formas.
- Usualmente permite recuperar datos perdidos automáticamente.
- Permite al usuario relacionar datos de manera variable y flexible.

Sin embargo también existen algunas desventajas en el uso de DBMS:

- Involucra un conjunto complicado de procedimientos de administración
- Usa su propio lenguaje, para lo cual se requiere de entrenamiento adicional
- Un DBMS es mucho más complejo y depende de un sistema operativo en particular o plataforma de hardware

Esto hace que los datos almacenados de esta forma sean difíciles de migrar de un sistema a otro. Algunos proveedores de DBMS permiten transformar los datos en archivos de texto para un fácil transporte y uso. Los beneficios que un DBMS provee fácilmente pueden opacar estar desventajas.

1.4.2 La Configuración en un Sistema Administrador de la Red

Las herramientas para la administración de la configuración pueden incrementar la productividad del administrador porque, en algunos

casos, éstas automáticamente recolectan y actualizan los datos de los dispositivos de red, facilitando el almacenamiento temporal de la información de la configuración, permitiendo modificaciones a la red, y facilitando la realización del inventario de la red y la generación de reportes.

Una herramienta simple de configuración por lo menos debe dar un almacenamiento central de toda la información de los dispositivos de la red, como dirección de red, números de serie, localizaciones físicas, y otros datos útiles. Un mecanismo de descubrimiento automático debe existir para encontrar todos los dispositivos de la red para que puedan luego ser consultados por la información relevante y almacenarla automáticamente, si es posible.

La adquisición automática de los datos es muy importante porque asegura que la información obtenida es la actual. El administrador probablemente realizará esta adquisición durante el tiempo en que la utilización de la red sea muy baja para evitar causar un problema de tráfico.

Es realmente importante que los dispositivos usen un protocolo de administración de red para proveer la información de la configuración, y para los dispositivos que no soporten un protocolo de

administración, entonces el administrador debe ingresar los datos manualmente en la bases de datos de la herramienta.

Una herramienta más avanzada podría ser capaz de comparar la configuración actual de un dispositivo con una almacenada en el sistema para realizar comparaciones de los cambios de la configuración. Como con la herramienta simple, la avanzada, proveería de almacenamiento central y fácil recuperación de los datos. Cuando la aplicación compare la configuración actual con una guardada y encuentre discrepancias, ésta preguntará si se cambia la configuración actual del equipo para que concuerde con la almacenada. Esta funcionalidad podría también correr automáticamente, sin realizar cambios, y enviar al administrador las diferencias encontradas ya sea por correo electrónico, buscapersonas (beeper), etc.

Hay que recalcar que no todos los detalles de la configuración son igualmente importantes. La aplicación debe incluir reglas para categorizar las alarmas debidas a las diferencias encontradas en la configuración de los equipos.

Algunas aplicaciones de administración comerciales proveen muchas de las facilidades descritas anteriormente como herramientas avanzadas. También hay aplicaciones que gráficamente representan

la configuración de los dispositivos, con características específicas del proveedor y con diferentes interfaces de usuario; lo que significa que el administrador necesita saber como utilizar estas funciones adicionales que proveen las herramientas.

1.4.3 Reportes de configuración de los dispositivos de la red

Los reportes que la aplicación genera deben permitir al administrador estar bien informado de la configuración de la red. Los reportes de la configuración no son requeridos tan frecuentemente como los de la administración de fallas, pero algunas veces es necesario información actualizada cuando por ejemplo cuando encuentra dos dispositivos con la misma dirección o nombre. En contraste con la administración de fallas que necesita de terminales gráficos para ser completamente funcional, en la administración de la configuración basta con terminales tipo texto para acceder a la mayoría de las facilidades que la herramienta ofrece.

Un reporte detallado de la configuración debe contener: el nombre del dispositivo, la dirección de red, número de serie, fabricante, sistema operativo, y la persona local responsable, información opcional puede ser: el vendedor y la localización física del dispositivo. La frecuencia

con la cual estos reportes son generados puede variar para cada red dependiendo de la frecuencia con que cambia la configuración.

Además del reporte de la configuración actual de la red, se necesitará luego de un uso prolongado un resumen de todas las modificaciones recientes. Este reporte debe listar todos los cambios por categoría, el nombre de la persona que realizó el cambio y cuando se hizo. Las categorías pueden ser: los dispositivos nuevos, cambios en el hardware, software y administración.

Por último, la herramienta debe crear un inventario resumido de la red. Este reporte es crucial para el mantenimiento de la red y debe enfocarse solo en los dispositivos. Por cada equipo, el reporte debe mostrar el número de serie, localización física, fecha de puesta en servicio, tipo y duración de la garantía, y una historia completa de las mejoras hechas. Dependiendo de los equipos, información adicional puede ser requerida.

1.5 Manejo del Rendimiento de la Red

Una red es como una carretera en la cual la información viaja de manera fluida y, como cualquier carretera, puede congestionarse por el

incremento de los requerimientos de los usuarios de la red. Los dispositivos de red pueden sobrecargarse, los enlaces saturarse y bajar del rendimiento.

La administración del rendimiento asegura que la red se mantenga accesible y descongestionada. Esto se logra mediante:

a) El monitoreo de los dispositivos de red y sus enlaces asociados para determinar la utilización y la tasa de errores.
b) La vigilancia de que la red provea un nivel de servicio consistente a los usuarios asegurando que la capacidad de los equipos y enlaces no estén utilizados al tope de su capacidad

La administración del rendimiento consiste de los siguientes pasos:

1. Recolectar información de la utilización actual de los dispositivos y enlaces de la red
2. Analizar la información relevante para determinar las tendencias de utilización elevadas
3. Configurar los límites de utilización
4. Usar simulación para determinar como la red puede ser alterada para maximizar su rendimiento

El beneficio principal del manejo del rendimiento es ayudar al administrador a reducir los congestionamientos de la red y proveer un nivel de servicio consistente a los usuarios. Con la administración del rendimiento, el administrador puede monitorear la utilización de los dispositivos de red y los enlaces. Esta información permite determinar las tendencias de utilización, aislar problemas de rendimiento, y posiblemente resolver los congestionamientos de red antes que impacten al funcionamiento de ésta.

Con los datos obtenidos del monitoreo de la red y sus enlaces, no solamente podemos aislar componentes que están siendo altamente usados, sino también ayudar a resolver otros problemas potenciales. Además analizando la información, podemos predecir las tendencias de utilización y consecuentemente prever un funcionamiento demasiado pobre resultante de una red saturada.

De los datos recogidos por el monitoreo se puede graficar el tráfico versus el tiempo para determinar las horas de menor utilización, con lo cual se puede programar grandes transferencias de información en horas en que la red no es usada.

Los administradores de red pueden determinar la capacidad actual de la red usando un protocolo de administración para recolectar información acerca de todos los bytes enviados y recibidos en las interfaces de red.

El total nos da el tráfico de datos que la red maneja y una vez conocida la capacidad de la red, se puede determinar como el tráfico adicional afectará su funcionamiento.

1.5.1 Recolección de la Información de la Utilización

La determinación de la utilización de los dispositivos no es usualmente una tarea sencilla, ya que cada dispositivo tiene diferentes características relacionadas a la utilización del mismo. Por ejemplo, la utilización de un servidor de archivos puede ser medido por la utilización del procesador, velocidad de acceso al disco, y la utilización de la interface de red.

En un router o bridge la utilización puede ser medida por la velocidad en el reenvío de paquetes, la carga del procesador, porcentaje de los frames descartados en cada interface, y el número de paquetes que están en la cola.

La utilización de un enlace de red es más fácil de determinar. Hay que tener presente que un enlace no puede ser 100 por ciento usado y que un porcentaje bajo también puede evidenciar problemas de rendimiento porque puede implicar que el equipo no puede procesar la

información para transferir los datos por problemas en el mismo o por una mala configuración. La utilización de un enlace es la cantidad de bits enviados y bits recibidos por segundo dividido para el ancho de banda.

Utilización%= (bits enviados + bits recibidos)/ ancho de banda

Esta fórmula no funciona en enlaces seriales full duplex, ya que éstos pueden enviar y recibir al mismo tiempo. Para calcular la utilización se toma el máximo de bits enviados y recibidos por segundo y se usa ese número para el cálculo de la utilización.

Utilización%= MAX (bits enviados + bits recibidos)/ ancho de banda

Una manifestación significativa de la sobreutilización de dispositivos y enlaces de red es que los usuarios reciben un bajo nivel de servicio. Para medir este nivel de servicio, se necesita determinar lo siguiente:

- Tiempo total de respuesta
- Tasa de rechazo
- Disponibilidad

El tiempo total de respuesta es la cantidad de tiempo que se requiere para que un requerimiento de un dispositivo entre a la red, sea

procesado y se reciba una respuesta. Por ejemplo, el tiempo total de respuesta para una sesión de login remoto puede ser medido en el momento en que un usuario tipea un carácter en el teclado sumando el tiempo que toma el dato viajar por la red a la máquina destino y regresar, hasta que es mostrado en el terminal local.

Muchos protocolos de transporte, como el TCP, miden el tiempo medio de viaje (round-trip time) en milisegundos para cada dato enviado desde la fuente al destino para propósitos de control de flujo. Se podría usar este número para obtener una buena aproximación del tiempo total de respuesta.

La tasa de rechazo es el porcentaje del tiempo que la red no puede transferir información debido a carencia de recursos o problemas de rendimiento. Para medir la tasa de rechazo, muchos dispositivos registran el número de intentos para iniciar una conexión y el número total de conexiones hechas. Dividiendo el número de conexiones hechas para el número de intentos nos da el porcentaje de rechazo.

La disponibilidad es el porcentaje de tiempo que la red está accesible para el uso y en algunos casos es medido como tiempo medio entre fallas, o MTBF[2]. Muchos vendedores de dispositivos proveen de una

[2] Median Time Between Failures (Tiempo Medio entre Fallas)

medida teórica de la disponibilidad de sus dispositivos. En la práctica algunos dispositivos tienen variables que indican el tiempo que han estado en operación. Este valor comparado con el tiempo total desde que el equipo está en servicio puede dar una medida aproximada de la disponibilidad del dispositivo.

Se puede usar un protocolo de administración de red para recolectar los datos que tienen relación con el manejo del rendimiento como: bytes transmitidos, bytes recibidos, tiempo en operación de un dispositivo, etc. Esta información, también puede ser importante para resolver los problemas de la red y para un análisis de las tendencias de utilización de la misma.

1.5.2 Análisis de la información

Los datos recolectados para la administración del rendimiento de la red deben ser mostrados de una manera gráfica para mostrar la utilización de un dispositivo de red o un enlace medido por ejemplo en tiempo real o históricamente. Pueden usarse gráficos de líneas o de barras, los cuales con muy útiles para el análisis del rendimiento ya que muestran las tendencias de uso de un recurso o dispositivo de red.

El análisis de gráficos en tiempo real puede mostrar la utilización actual de la red y sus dispositivos y puede ayudar a diagnosticar problemas de rendimiento de la red. Entre los gráficos más comunes para el análisis están:

- Información de dispositivos: uso de memoria, uso del procesador, velocidad de acceso al disco, número de sesiones, etc.
- Información de enlaces: utilización, porcentaje de errores, etc.

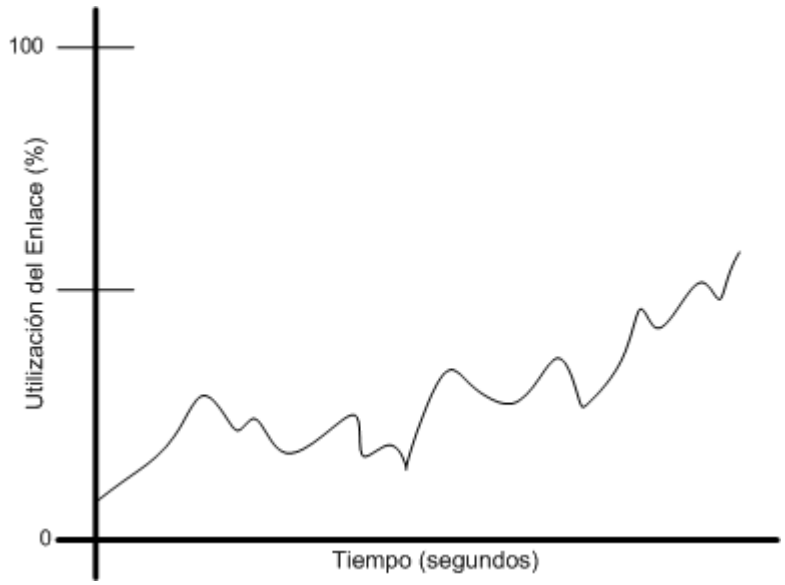


Figura 13: Gráfico del rendimiento en tiempo real (Ref 1)

Los gráficos que muestran datos históricos no indican el estado actual de la red pero son útiles para ilustrar las tendencias. Estos gráficos ayudan a predecir cuando la demanda de la red excede la capacidad

de un dispositivo o enlace y así prever el aumento de la capacidad de la red y rediseñar de la misma.

Otros gráficos usados son los que muestran la tasa de errores en un enlace o el uso del procesador en un equipo, permitiendo predecir tendencias en el aumento de errores o aumento en el uso de procesadores.

1.5.3 Configuración de los límites de utilización

La configuración de los límites de utilización es el proceso de observar los patrones de uso de red y sus dispositivos para establecer valores que permitan un funcionamiento óptimo de los mismos. Lo cual puede ser complicado y requiere analizar la información histórica de la red. Por ejemplo si analizando la información histórica se determina que cuando el porcentaje de error excede el 10% se presentan problemas en la red, un buen límite sería 8% permitiendo al ingeniero arreglar el problema antes de que afecte a la red.

Una vez que los límites han sido configurados, la herramienta para el manejo del rendimiento puede reportar cuando un límite ha sido

alcanzado o sobrepasado permitiendo al administrador localizar y reparar un problema antes de que éste afecte a la red.

La combinación de la representación gráfica de la utilización de la red, con la administración de los límites de uso, es una técnica poderosa para la administración del rendimiento.

1.5.4 La simulación de la Red

La simulación de la red es el proceso que permite conocer el comportamiento de la red bajo patrones de tráfico, configurados por el administrador, similares a los que se presentan cuando la red está completamente operativa, o cuando se añade una aplicación nueva que use los recursos de la red.

La simulación es usada para determinar como alterar una red para un uso más eficiente y un mayor rendimiento. Por ejemplo, considere una red remota en donde los usuarios están usando una nueva aplicación que tiene un tiempo de respuesta alto. Investigando el enlace entre la red central y la remota se descubre que la utilización es mayor al 80%, por lo que el administrador decide aumentar la capacidad del enlace. Aún después de aumentar la capacidad, el usuario sigue

experimentando un tiempo de respuesta alto. Analizando más a fondo, se descubre que el protocolo de transporte usado por la nueva aplicación usa el protocolo pare-y-espere(tamaño de la ventana de un paquete). Este protocolo usa un mecanismo de control de flujo que envía un paquete a la vez, y espera que el paquete sea reconocido antes de transmitir el próximo. Mayor capacidad de transmisión en el enlace permite que más paquetes sean enviados en un tiempo dado, pero no ayuda a disminuir la demora de los usuarios de la red causada por el control de flujo.

Usando una aplicación sofisticada de simulación de red antes de aumentar la capacidad del enlace, probablemente se hubiera podido detectar la deficiencia relativa a la capa de transporte de red a un costo menor y en menos tiempo. La solución sería hacer que la aplicación que los usuarios usan, utilice una cola a nivel de protocolo que maneje el control de flujo con una ventana mayor a uno diseñada para redes WAN.

1.5.5 Manejo del rendimiento en un sistema administrador de red

El manejo del rendimiento tiene que ver con el uso de herramientas inteligentes que puedan examinar el estado de la red ya sea en

tiempo real o históricamente para tomar acciones de acuerdo a los patrones de uso. La efectividad en el manejo del rendimiento depende del nivel de sofisticación de la herramienta usada por el administrador.

Una herramienta simple para la administración del rendimiento debe por lo menos proveer información en tiempo real acerca de los dispositivos de red y enlaces, preferiblemente en forma de gráficos de línea o de barra. Esta herramienta puede ayudar a encontrar los congestionamientos de la red y aislar los problemas de rendimiento, permitiendo graficar cualquier estadística de la red usando un protocolo de administración de red.

Una herramienta simple provee una forma fácil de recolectar información acerca de la utilización del procesador y memoria del equipo. Una alta utilización del procesador quiere decir que el dispositivo no puede manejar el tráfico de la red, una utilización excesiva de la memoria significa que el equipo está encolando grandes cantidades de información. Un gráfico en tiempo real de estas variables pueden alertar de un problema potencial con el dispositivo.

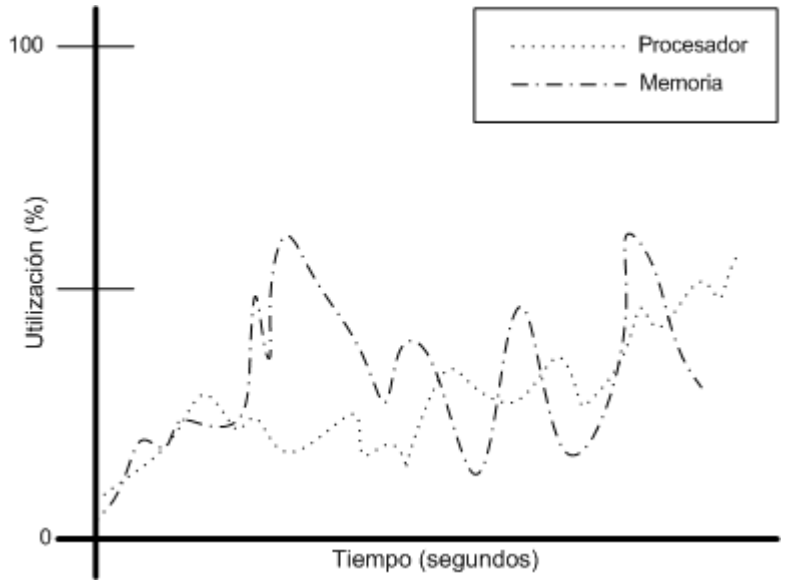


Figura 14: Funcionalidad de una herramienta simple para la administración del rendimiento (Ref 1)

Una herramienta simple también debe mostrar la utilización actual y la tasa de errores de los enlaces de red. Por ejemplo, un gráfico de tráfico en paquetes por segundo y bits por segundo del enlace es útil para mostrar el rendimiento general. En una situación de problema sería necesario examinar el número de errores en un enlace contra la cantidad de información enviada correctamente.

Una herramienta más sofisticada, a más de tener las funcionalidades descritas anteriormente, permite configurar límites para la utilización y la tasa de errores. Si la red excede esos límites, la herramienta puede realizar una acción especificada por el administrador. También puede

recolectar información en tiempo real y almacenarla en un sistema de administración de base de datos, el cual puede ser usado para realizar estudios históricos y producir gráficos que muestren el estado anterior del rendimiento de la red.

El configurar límites que puedan activar una acción le permite al administrador manejar una funcionalidad que ahorra mucho trabajo. La herramienta permite especificar la acción, la cual puede ser tan simple como hacer sonar una campana o prender y apagar un foco, o más avanzada como habilitar un enlace de respaldo, enviar un mensaje al administrador a través de un buscapersonas (beeper) o un correo electrónico.

Los límites que se configuran pueden tener niveles de prioridades para la notificación, los cuales pueden ser por lo menos: bajo, medio, alto. Cada nivel puede tener un color diferente en el mapa de la red o en el despliegue del registro de eventos en el sistema administrador de la red.

La herramienta debe ser capaz de manejar los límites de una manera estable. Esto se puede ver afectado por la frecuencia de muestreo que la herramienta realiza para la recolección de los datos de monitoreo. Por ejemplo, supongamos que una alarma suena cuando la utilización de un enlace supera el 60% de uso, luego deja de sonar

cuando baja al 58% y vuelve a sonar cuando llega al 61%. Para prevenir esto, la herramienta debe ser capaz de manejar límites superiores e inferiores de tal manera que, cuando el límite superior es alcanzado la primera vez activa la alarma, la cual se desactiva solo cuando se alcanza un límite inferior programado.

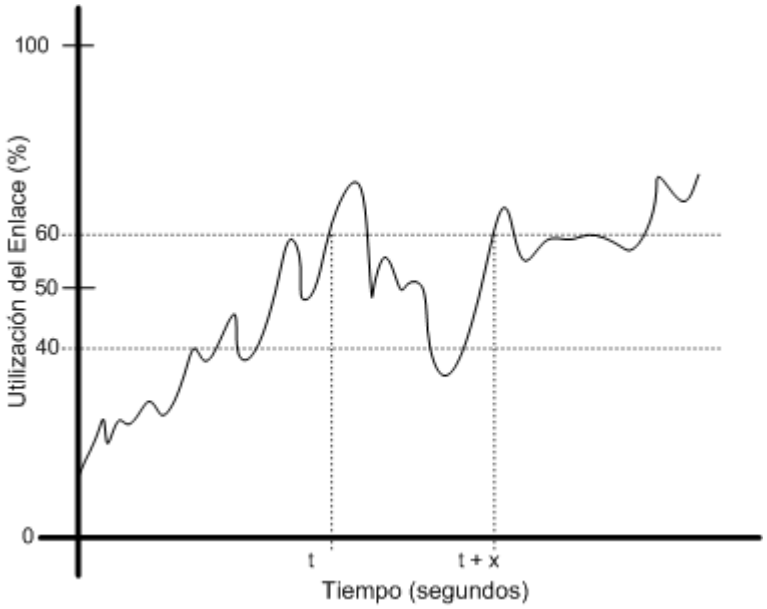


Figura 15: Ejemplo de la utilización de límites en una herramienta de administración del rendimiento (Ref 1)

En la mayoría de plataformas para la administración de redes se pueden configurar límites y sus prioridades, y especificar las acciones si un límite es alcanzado.

Otra de las funcionalidades de una herramienta más avanzada para la administración del rendimiento es graficar los datos históricos del rendimiento que fueron almacenados en el sistema de base de datos.

Los tipos de gráficos que debe poder presentar una herramienta de este nivel son gráficos de línea, barra y pastel. Los gráficos de línea son muy útiles para examinar las tendencias de los datos, como por ejemplo la utilización, mientras que los gráficos de barra son efectivos para comparar valores como cuanta memoria se consume en un dispositivo o el número de paquetes manejados a lo largo del tiempo. Los gráficos de pastel se usan para mostrar porcentajes de valores, como el tipo de tráfico que viaja en una red según el protocolo o cualquier otro criterio aplicable.

Una herramienta avanzada para la administración del rendimiento, además de proveer las funcionalidades antes mencionadas, debe ser capaz de explorar los datos históricos del estado de la red, hacer simulaciones de la red para predecir tiempos de respuesta, tiempos de rechazo y disponibilidad de la misma.

1.5.6 Reportes del rendimiento de la red

Muchos son los métodos disponibles para reportar los datos del rendimiento de una red. El formato más común es un reporte de texto, el cual puede ser mostrado en cualquier terminal y puede ser impreso sin mayor complicación. Típicamente estos reportes muestran la utilización y porcentajes de errores para los dispositivos y enlaces de red tal como lo muestra la tabla 2.

	% Utilización		***% Errores***	
Dispositivo	*Promedio*	*Máximo*	*Promedio*	*Máximo*
Diamante	30	97	5	10
Esmeralda	20	88	3	8
Zafiro	20	90	3	7

Tabla 2: Reporte de utilización de dispositivos de red

Un reporte de texto contiene información muy valiosa. Sin embargo también es necesario un método de representar los datos de una forma gráfica porque son más fáciles de comprender, como por ejemplo un reporte gráfico que muestre la tendencia de uso de un enlace o un gráfico tipo pastel que muestre la utilización de la red por protocolo.

1.6 Registros de Eventos y Administración de Usuarios

El registro de eventos es el proceso de recolectar información de las estadísticas de la red para ayudar al administrador a tomar decisiones acerca de la localización y utilización de los recursos. Esta información es útil para controlar parámetros como el espacio en disco, poder de procesamiento, realización de respaldos, aunque estas tareas no son necesariamente parte de la administración de la red, son complementarias para una administración total.

El registro de eventos permite al administrador medir y reportar información de la utilización de los recursos por parte de los usuarios y facturar (de ser necesario) los recursos utilizados individualmente o en grupo. Además incrementa el entendimiento que tiene el administrador de lo que ocurre en la red, lo cual ayuda a obtener una red mucho más productiva y mejor administrada.

El registro de eventos involucra la medición del uso de los recursos de la red para establecer referencias, examinar cuotas, determinar costos, y facturar el uso de los recursos de la red a los usuarios. Esta administración comprende los siguientes pasos:

- Recolectar datos acerca de la utilización de los recursos de la red
- Establecer cuotas de uso usando las referencias establecidas

- Determinar costos por el uso de la red

La facturación a los usuarios puede ser útil en ciertos casos para recuperar los costos destinados a las redes y sistemas, y para el mantenimiento de la misma. En otros casos ayuda a establecer centros de costos para ayudar a estimar los costos de instalación, operación y mantenimiento de la red de datos.

Examinando las referencias y cuotas, el administrador puede asegurar que cada usuario tiene los recursos suficientes para realizar sus tareas. También se pueden usar las estadísticas de administración de usuarios para hacer un seguimiento del uso de varios recursos de red, como servidores de aplicaciones, servidores de archivos, servidores de impresión. Por ejemplo, un grupo de documentadores pueden utilizar la red para acceder a una aplicación de procesamiento de palabras ubicada en un servidor que se encuentra en un sitio remoto. Usando la información de la administración de usuarios, se encuentra que la mayoría del tráfico, que hay entre las redes donde se encuentran los documentadores y el servidor, es debido al acceso del grupo de documentadores a la aplicación en el servidor remoto. Con esta información se puede sugerir la adquisición de un servidor de aplicaciones para el grupo de documentación.

El uso de la administración de usuarios permite que el administrador entienda mejor que es lo que sus usuarios realizan en la red y permite influenciar su comportamiento para utilizar los recursos de una forma más óptima.

Debido a que la tecnología de redes cambia rápidamente, las técnicas administración de usuarios son también usadas para determinar donde los recursos de redes son colocados para optimizar el rendimiento costo-beneficio que las diferentes tecnologías existentes ofrecen.

1.6.1 Pasos para realizar el Registro de Eventos y Administración de Usuarios

El registro de eventos, como se mencionó anteriormente, consiste de los siguientes pasos:

1. Recolectar datos acerca de la utilización de los recursos
2. Usar medidas para ayudar a configurar cuotas
3. Determinar costos por el uso de la red

1.6.1.1 Recolección de los datos

La recolección de los eventos de la red puede realizarse a través de protocolos de administración de redes como SNMP, o en el caso de los dispositivos no tengan esta capacidad, será necesario la implementación o adquisición de programas que ayuden a recolectar esta información.

La recolección puede ser tan infrecuente como una vez al día o muy frecuente como una vez cada treinta minutos. El tiempo medio entre cada recolección depende en gran medida de la capacidad de los dispositivos para almacenar la información y además del uso de la red cuando se realizan las consultas.

1.6.1.2 Uso de medidas para establecer cuotas

Realizar mediciones es una ayuda para saber el nivel de utilización de los recursos por parte de los usuarios en la red. Por ejemplo, una medida puede ser el número de conexiones hechas a un servidor de terminales, otra el número de transacciones hechas a través de la red a una base de datos en particular, otra el tiempo total de una sesión de un usuario en una computadora, etc. Como parte de la administración de usuarios, es necesario que el

administrador establezca los recursos que serán monitoreados para luego empezar a recolectar las mediciones acerca de su uso.

Las mediciones permiten ajustar las cuotas para permitir que cada usuario obtenga un "recurso justo" de la red. El recurso justo es el recurso que el usuario necesita para realizar su tarea adecuadamente. De acuerdo con diferentes políticas, se pueden establecer las cuotas y luego penalizar a los usuarios que excedan las cuotas establecidas negándoles el uso del recurso en mención, o cobrarles un valor adicional si es que se le factura al usuario.

1.6.1.3 Determinación de costos por el uso de la red

La determinación de costos por el uso de la red puede ser hecha en base a dos escenarios: recargos por instalación y mensualidades, o recargos por los recursos de red consumidos.

1.6.1.3.1 Recargos por instalación y mensualidades

Bajo este esquema, los recargos son por la instalación de la conexión a la red y luego por un valor fijo por cada mes de uso. El

registro de eventos no es necesario para la facturación, porque ninguna información de la red es requerida.

Este sistema es el más fácil de implementar, pero resulta difícil justificar porque un usuario que continuamente usa los recursos de la red paga la misma cantidad que un usuario casual.

1.6.1.3.2 Recargos basados en la cantidad de recursos de red consumidos

La implementación de esta técnica requiere de estadísticas de la utilización de la red por parte de los usuarios. Los siguientes criterios, usados individualmente o en conjunto, pueden ser medidos para determinar el uso de los recursos de la red: número total de transacciones, paquetes totales, y bytes totales. La medida de los paquetes totales o bytes totales puede ser basada en los datos enviados por el usuario a la red, o los datos recibidos por el usuario a través de la red o ambos. Algunas organizaciones usan este esquema conjuntamente con recargos pequeños por instalación y mensualidades.

Con el número total de transacciones por usuario, una organización puede medir diferentes criterios, incluyendo el

número de sesiones establecidas con un servidor, mensajes de correo electrónico enviados, etc. Este esquema es fácil de implementar, pero cada transacción hecha tiene el mismo recargo sin importar el tiempo o los recursos de red usados. Por ejemplo si un usuario hace una transacción que envía 500Mbytes de información, éste tiene el mismo recargo que un usuario que envía un mail de 100bytes; por lo que los usuarios podrían reclamar que la forma de facturación es injusta e inapropiada.

Otro método es contar el número de paquetes que son transmitidos o recibidos, reflejando de esta manera la utilización de la red. Pero este método tiene un problema que el recargo por un número determinado de paquetes es el mismo sin importar la cantidad de información que el paquete transporte.

Las desventajas de los primeros dos métodos pueden ser contrarrestadas utilizando un tercer método: recargos (cobro) por los bytes transmitidos. Según este método, el usuario es facturado de acuerdo con la cantidad de recursos usados, ya que los recursos de red, como ancho de banda o el acceso al disco de un servidor, son consumidos byte por byte (o bit por bit), y no por paquetes transmitidos. Para manejar este método de medición, es necesario decidir si facturar los bytes totales enviados o recibidos.

La mayoría de compañías escogen facturar los bytes recibidos de la red, bytes enviados a la red, o alguna combinación de los dos esquemas. Hay claras ventajas en cada uno de ellas y desventajas como se describirá más adelante.

El recargo por los bytes enviados a la red tiene sentido, ya que cuando un usuario envía algo a través de la red aumenta el uso de la ésta (por lo que su factura aumenta también). Desafortunadamente en los esquemas actuales de computación cliente/servidor, este método tiene desventajas como por ejemplo la de reprimir que los usuarios ofrezcan, desde sus computadoras, servicios a la red para no aumentar sus costos.

El recargo por los bytes recibidos de la red elimina el problema antes mencionado. No se factura por los datos enviados a la red, sino solo por los recibidos. Como todo método, este también tiene falencias. El primer problema es de que muchos de los protocolos envían paquetes de reconocimientos que es información no solicitada o inútil para el usuario final y que por lo tanto no debería se facturada. Otro problema es que el usuario puede recibir información no solicitada de la red, como correo electrónico no solicitado. Un problema adicional son los paquetes enviados para las tareas de monitoreo de la red desde la estación de

administración a las distintas computadoras. Todo tráfico recibido por los usuarios por este concepto no debería ser facturado.

Implementar un esquema de facturación basado en el consumo de un recurso requiere del uso de la administración de usuarios para recolectar las estadísticas necesarias, procesar los datos recolectados y producir las facturas basadas en los recursos consumidos. No hay un método que sea aplicable a toda organización para el recargo por el uso de la red. Muchas organizaciones tienen políticas establecidas para facturar por el uso de los recursos para lo cual el administrador debe analizar el método que convenga a la organización de acuerdo a sus políticas.

1.6.2 Herramientas para registros de eventos y administración de usuarios

La efectividad de las tres tareas antes mencionadas para el registro de depende mucho de la herramienta usada por el administrador.

Antes de realizar la tarea de registro de eventos hay que determinar las herramientas que tenemos a nuestra disposición y examinar para

qué servirían. Una herramienta simple debe permitir monitorear los recursos en base a cualquier medida (uso de espacio en disco, bytes transferidos, etc.). Esa medida debe ser almacenada en un sistema de base de datos, que es parte del sistema de administración. Para determinar si una cuota ha sido sobrepasada, se puede consultar a la base de datos a través de una sentencia SQL(Lenguaje Estructurado de Consulta) y mostrar los datos en una ventana o almacenarlos en la misma base para su posterior revisión. Por ejemplo, si es necesario monitorear el número de usuarios que están usando un servidor de aplicaciones, se puede configurar a la herramienta para que consulte al servidor de aplicaciones cada hora, determine el número de usuarios, y luego ponga la información en la base de datos.

La funcionalidad de la herramienta depende en mucho de las características que posea la base de datos, como la automatización de las consultas, formato de los resultados, y los triggers(procedimientos que se activan de acuerdo a un evento).

Una herramienta simple permite consultar las medidas y cuotas, mientras que una herramienta avanzada hace mucho más que eso, como por ejemplo llevar la facturación de los usuarios de la red. Para esto la herramienta puede tomar como entrada la topología de la red de datos y los dominios de facturación y luego realizar los cálculos

para las facturas de los usuarios. Este tipo de herramienta necesita además información del sistema administrador y del administrador de la red. Con estos datos, la herramienta puede determinar que dispositivos tiene que ser consultados, y que datos, utilizar para realizar el recargo por el uso de los recursos de la red.

Adicionalmente, esta herramienta, permite predecir la necesidad de adquirir recursos de red o aumentar la capacidad de algún dispositivo. Con el análisis de tendencias hecha por este tipo de herramientas, el administrador puede establecer referencias y cuotas para los distintos recursos de la red.

1.6.3 Maneras de reportar los registro de eventos y administración de usuarios

Los reportes, para el registro de eventos y administración de usuarios, pueden ser mensajes en tiempo real y reportes de textos. Los mensajes en tiempo real pueden informar el valor actual de una medida; los reportes de texto proveen información histórica e información para la facturación.

Por ejemplo, se puede hacer que la herramienta para el registro de eventos muestre un mensaje en tiempo real indicando una medida para un dispositivo, como se ve en la figura 16.

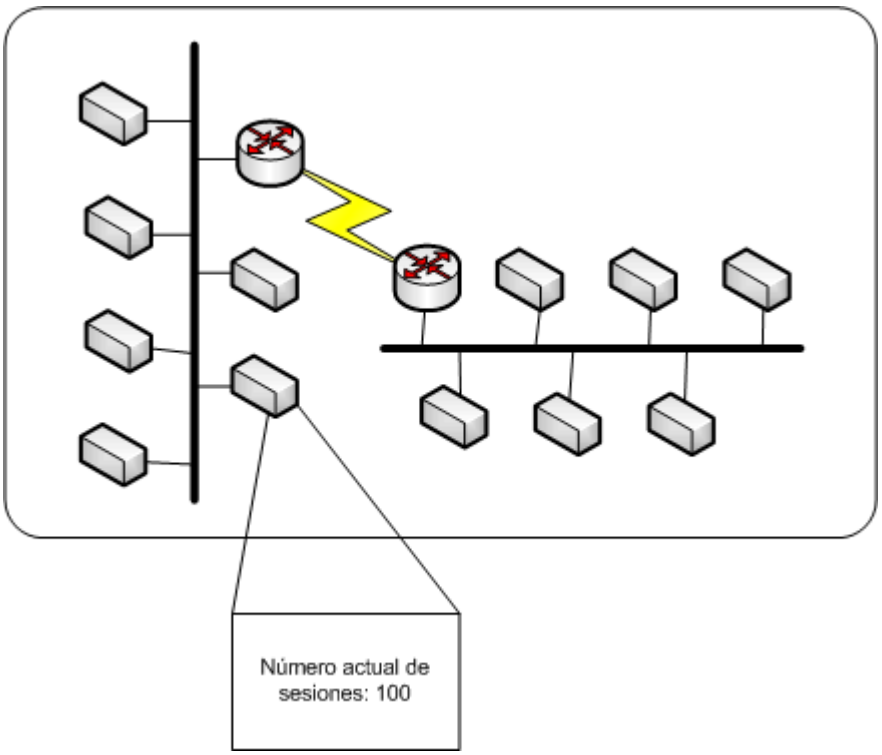


Figura 16: Mensaje en tiempo real para el registro de eventos y administración de usuario (Ref 1)

La herramienta del registro de eventos y administración de usuarios puede generar un reporte de texto de estadísticas del uso de los recursos de la red por parte de los usuarios. Estos reportes pueden

ser una recopilación histórica de las medidas o tendencias del uso futuro de un recurso de red. Esta información puede ser usada para planear cuotas para los recursos de la red.

Otros reportes pueden ser los usados para la facturación a los usuarios. Estos reportes muestran la utilización de los recursos de la red por parte de los usuarios y además pueden incluir cual podría ser el valor de la siguiente factura. Este tipo de reporte se muestra en la tabla 3.

Período de Facturación	Noviembre 2003
Número de bytes recibidos	500 Megabytes
Dispositivo consultado para determinar el consumo	Diamante
Precio por Megabyte	$ 0.05
Precio Total	$ 25
Factura Anterior	$ 20
Factura siguiente (calculado por extrapolación)	$ 28
Porcentaje de aumento	12%

Tabla 3: Factura por la utilización de un recurso

Capítulo II

Protocolos para la Administración de Redes

2.1 Introducción

Los protocolos para la administración de redes proveen las bases para la comunicación entre el sistema administrador de la red y los dispositivos administrados, permitiéndole al administrador consultar, monitorear, configurar, y realizar otras operaciones sobre los dispositivos.

Este capítulo presenta un breve resumen del desarrollo de los protocolos de administración y revisa con mayor detalle el protocolo SNMP que es el estándar más utilizado en la actualidad.

2.2 Protocolos para la administración de redes

Antiguamente, los ingenieros encargados de administrar una red necesitaban aprender y aplicar una gran variedad de métodos para recolectar información de los dispositivos que administraban. Así, si nuevos equipos de redes eran desarrollados, los fabricantes instalaban en éstos aplicaciones propietarias para permitir la recolección de información y configuración de sus productos. La consecuencia era que si

dos dispositivos tenían la misma funcionalidad pero venían de fabricantes diferentes, tenían también diferentes formas de administración. Debido a estos problemas, los protocolos de administración de redes tomaron vigencia como una necesidad para proveer una forma de acceder a un conjunto estándar de información y acciones aplicables a cualquier dispositivo sin importar que este fuere hecho por diferentes fabricantes. Los ejemplos de valores a estandarizar pueden ser: nombre del equipo, versión del software, número de interfaces, dirección de una interfaz de un dispositivo, etc.

Los protocolos estándares de administración tienen el beneficio adicional de proveer una apariencia uniforme a los datos enviados y retornados por un dispositivo.

El conjunto estándar de información que los dispositivos almacenan sin importar el fabricante se denomina MIB (Management Information Base).

2.3 MIB

Para poder establecer un protocolo de administración de redes, es necesario también establecer cuáles son los datos que caracterizan a los diferentes dispositivos y cómo se relacionan estos datos entre sí.

El MIB (Management Information Base) es un estándar que define la información disponible para un dispositivo administrable a través de la red.

El RFC[3] 1065 describe la sintaxis y el tipo de información disponible en el MIB para el manejo de redes TCP/IP, titulado “Structure and Identification of Management Information for TCP/IP based Internets (SMI)”, este RFC define reglas simples para la denominación y creación de tipos de información. En la tabla 4 se describen algunos tipos de información permitidos por el. EL RFC 1065 luego fue adoptado por el IAB como un estándar de Internet e incorporó modificaciones que se encuentran expresadas en el RFC 1155.

Tipo Definido	Significado
NetworkAddress	Una dirección de red de cualquier protocolo. Solamente existe para el protocolo IP
IpAddress	Dirección IP(32 bits)
Counter	Entero no negativo que se incrementa desde 0 hasta un valor máximo de 2^{32}-1.
Gauge	Entero no negativo que puede incrementar o decrementar y tiene un valor máximo de 2^{32}-1.
TimeTicks	Entero no negativo que cuenta el tiempo en centésimas de segundo.
Opaque	Sintaxis arbitraria, usada para texto.

Tabla 4: Tipos de Datos definidos por RFC1155 (SMI) (Ref 1)

[3]RFC (Request For Comment) son una serie de documentos (desde 1969), acerca de internet. estos documentos tratan sobre muchos aspectos de comunicación de computadoras, centrándose en los protocolos de red, procedimientos, programas y conceptos.

Antes de eso, tomando las reglas del SMI (RFC 1065), se define el RFC 1066, el cual presenta la primera versión del MIB para usar con los protocolos TCP/IP. Este estándar, ahora conocido como MIB-I, explica y define la información base necesaria para monitorear y controlar dispositivos de hardware y software bajo el protocolo TCP/IP. El RFC 1066 fue aceptado por el IAB[4] como un estándar completo en el RFC 1156.

El RFC 1158 propone una segunda versión del estándar MIB, denominada MIB-II, para usarse con el protocolo TCP/IP. Esta propuesta, formalizada como un estándar y aprobada por el IAB en el RFC 1213, extiende la información base definida en el MIB-I expandiendo el conjunto de objetos definidos en el MIB.

Para facilitar la migración de los protocolos específicos de los fabricantes de equipos de redes a un protocolo estándar de administración, el RFC 1156 permite definiciones dentro del estándar MIB, a través de las cuales se expanden los datos a manejar para así incluir información propietaria que puedan añadir los fabricantes.

[4] AB Internet Architecture Board es un comité del IETF (Internet Engineering Task Force) que supervisa el trabajo en tecnologías de red y protocolo para la comunidad TCP/IP

Hasta ahora hemos visto que el MIB solo soporta el protocolo TCP/IP pero se han hecho esfuerzos para tener definiciones del MIB que no tengan dependencia con protocolos de la capa de red sino que permita trabajar con objetos a nivel de la capa de enlace de datos relacionados a protocolos como FDDI, Token Ring, Ethernet, etc. Este desarrollo significativo permite ubicar al MIB como un estándar aplicable a una arquitectura completa (TCP/IP) o a protocolos de las capas de enlace de datos y física (como ethernet).

Nombre del MIB	Estándar propuesto
Interface IEEE 802.5 Token Ring	RFC 1743
Monitoreo de redes remotas (RMON)	RFC 1757
Interface FDDI	RFC 1512
Brigdes	RFC 1493

Tabla 5: Algunos MIBs específicos propuestos (Ref 1)

2.3.1 Sintaxis del MIB

Un subconjunto de la ISO[5] "Abstract Syntax Notation One (ISO ASN.1)" define la sintaxis del MIB. Cada MIB usa la arquitectura de

[5] ISO Internacional Organization for Standardization es una organización internacional que desarrolla, sugiere y nombra estándares para protocolos de redes

árbol definida en el ASN.1 para organizar toda la información. Cada pieza de información en el árbol es un “nodo etiquetado”. Cada nodo etiquetado contiene un identificador de objeto y un texto breve que describe el objeto. El identificador de objeto (OID por su sigla en inglés) es una serie de enteros, separados por puntos que indican el camino completo en el árbol ASN.1.

Un nodo etiquetado puede tener subárboles que contienen otros nodos. Cada nodo en un subárbol es numerado en orden ascendente. Este orden lexicográfico provee un esquema para numerar todos los objetos en el árbol MIB.

Si un nodo no tiene subárboles, u hojas, éste contiene un valor y es conocido como un objeto. Los nodos hojas también son numerados en orden ascendente.

La figura 17 muestra un árbol MIB con los correspondientes números ASN.1, el orden lexicográfico del árbol es: 1, 1.1, 1.1.1, 1.2, 1.2.1, 1.2.1.1, 1.2.2, 2. Este orden del árbol permite descubrir, sin mayor conocimiento de su estructura, todos los identificadores de objeto que un dispositivo de red dado puede soportar.

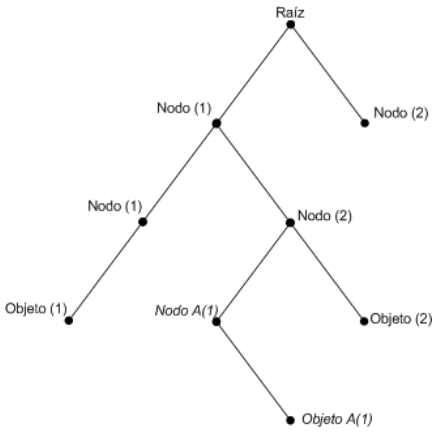


Figura 17: Ejemplo de un árbol ASN.1 (Ref 1)

El nodo raíz del árbol MIB no tiene un nombre o número, pero tiene tres subárboles que son:

- ccitt(0), administrado por la CCITT[6]
- iso(1), administrado por la ISO
- join-iso-ccitt(2), administrado conjuntamente por la ISO y CCITT

[6] CCITT Internacional Telephone and Telegraph Consultitative Committee Organización internacional que define estándares y recomendaciones para conexión de equipos telefónicos. Conocida ahora como ITU-T (Internacional Telecommunication Union Telecommunication Standarditazation Sector).

La sintaxis, como ccitt(0), denota que el nodo ccitt tiene un identificador de objeto número 0 en este nivel del árbol MIB.

Adicionalmente, muchos otros subárboles existen bajo el nodo iso(1), incluyendo el subárbol definido por la ISO para otras organizaciones, org(3). Bajo el subárbol org(3), un nodo particular de interés es el usado por el Departamento de Defensa de Estados Unidos (DOD): dod(6). Toda la información recolectada de los dispositivos de comunicación que tengan que ver con los protocolos de la DOD, como TCP/IP, residen en el subárbol que tiene el identificador de objeto 1.3.6.1. Este identificador es conocido como Internet. La descripción de texto para este identificador es {iso org(3) dod(6), 1}.

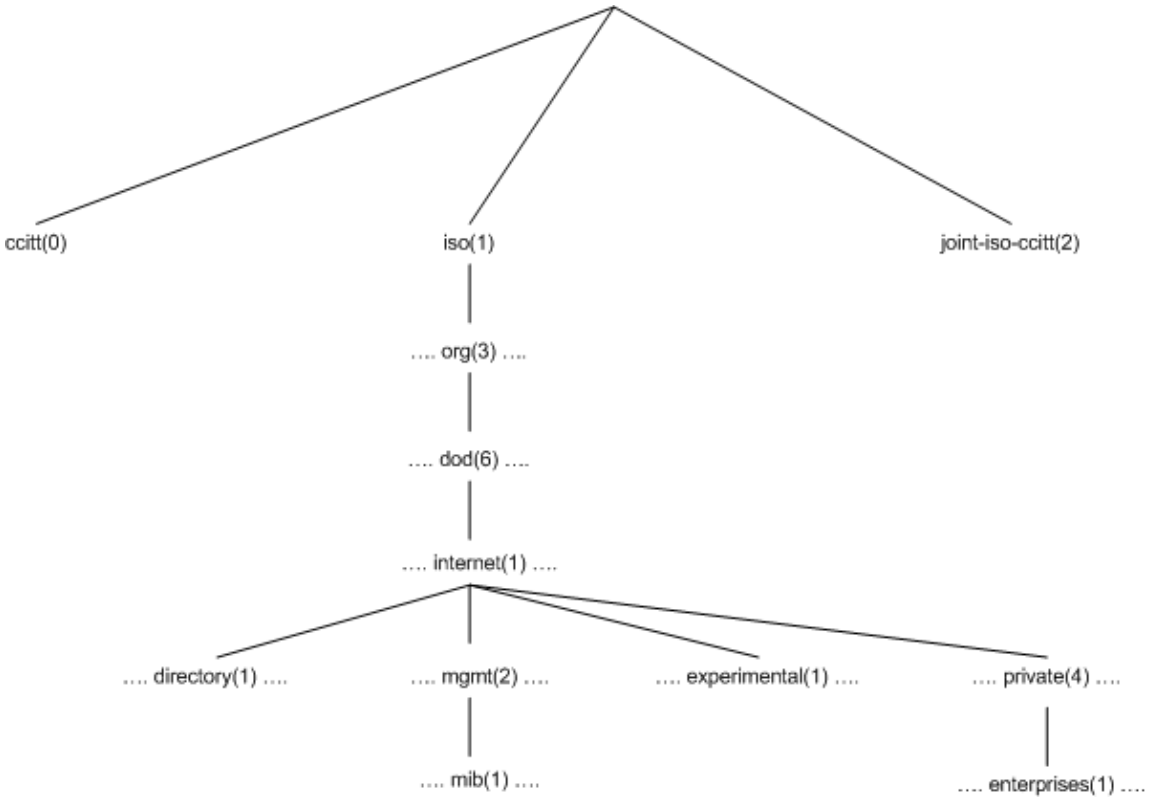


Figura 18: Estructura superior del árbol MIB (Ref 1)

Cuatro subárboles definidos están bajo el identificador de objeto de Internet:

- directorio(1)
- mgmt(2)
- experimental(3)
- privado(4)

Subárbol Directorio(1)

El subárbol directorio(1) está reservado para un futuro uso. Este subárbol contendrá información acerca del servicio OSI de directorio (X.500[7]).

Subárbol Mgmt(2)

Hoy en día los objetos de este subárbol están ampliamente implementados. La definición MIB-I (RFC 1156), originalmente identificaba a este subárbol como el objeto 1.3.6.1.2.1, o {mib 1}, el cual ha sido reemplazado por la definición MIB-II (RFC 1213). Conservando el mismo identificador de objeto que establece el MIB-I.

[7] X.500 Estándar de la ISO e ITU que define como directorios globales deben ser estructurados. Los directorios X.500 son estructurados con diferentes niveles para cada categoría de información, como país, estado/provincia, cuidad, etc.

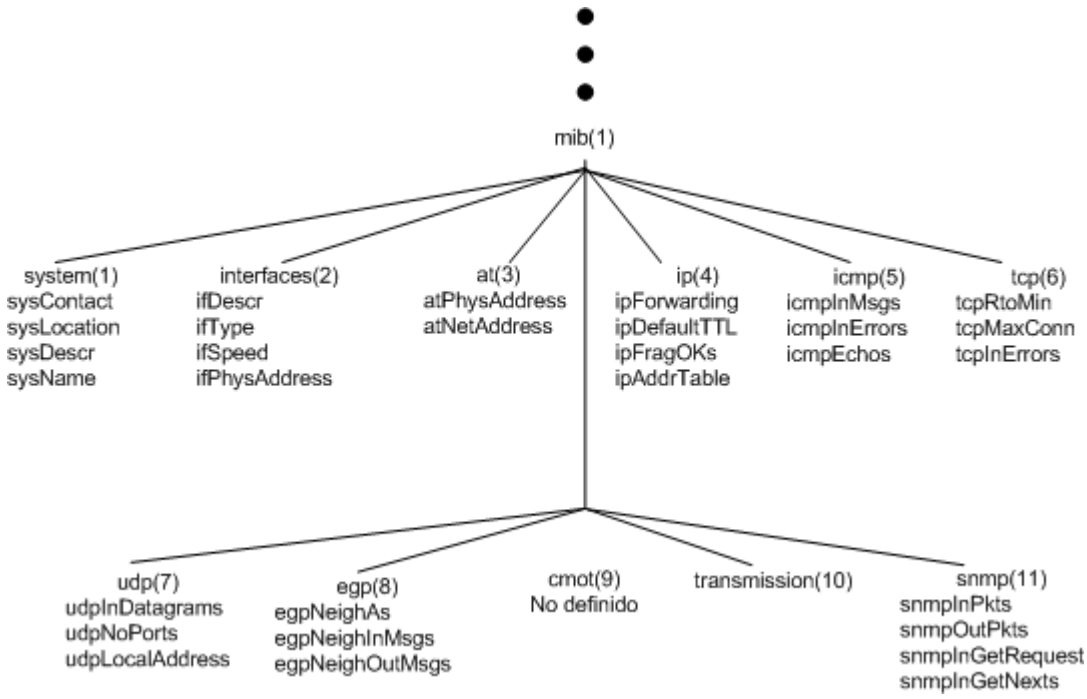


Figura 19: Estructura del subárbol mgmt(2) y algunos objetos MIB-II (Ref 1)

Dentro del subárbol mgmt(2) están los objetos usados para obtener información específica de los dispositivos de redes. Esos objetos pueden dividirse en once categorías que se muestran en la tabla 6.

Categoría	Información
system(1)	Provee una descripción textual del dispositivo. Esta descripción incluye: descripción del sistema, identificador de objeto(OID), el lapso de tiempo desde la última reinicialización del equipo y otros detalles administrativos. Esta categoría es mandatoria.
interfaces(2)	Provee información acerca de las interfaces del equipo. Esta información está en una tabla. El primer objeto(ifNumber) indica el número de interfaces, por cada interface hay una fila en la tabla que tiene 22 columnas que proveen información como la tasa de transmisión, dirección física, estado operacional actual, y estadísticas de paquetes.
address translation(3)	Esta categoría se la incluyó en el MIB-I, pero fue descartada en el MIB-II. Provee una tabla con las equivalencias entre direcciones IP y direcciones físicas. En el MIB-II y versiones futuras, cada grupo de protocolo debe contener sus propias tablas de equivalencias. Cada fila de esta tabla contiene tres columnas (dirección lógica, dirección física en interfaz)
ip(4)	Este grupo es mandatorio y provee información acerca del uso del protocolo IP. Esta categoría incluye objetos que proveen estadísticas de los datagramas IP y además tiene tres tablas: una tabla de direcciones(ipAddrTable); una tabla de equivalencias de direcciones IP a direcciones físicas (ipNetToMediaTable); y una tabla de ruteo ip (ipForwardTable).
icmp(5)	Este grupo es un componente mandatorio del protocolo IP. Contiene 26 objetos que mantienen las estadísiticas acerca de los mensajes ICMP, como el número de mensajes ICMP Echo Request recibidos, etc.
tcp(6)	Grupo mandatorio. Provee información acerca de las operaciones y conexiones TCP. Contiene 14 objetos y una tabla. Los objetos escalares registran varios parámetros y estadísticas TCP, como el número de conexiones que el dispositivo soporta, o el número total de segmentos TCP transmitidos. La tabla, tcpConnTable, contiene información particular de las conexiones TCP.
udp(7)	Grupo mandatorio. Provee información concerniente a las operaciones UDP. Debido a que UDP no es orientado a conexión, este grupo es mucho más pequeño que el TCP. Contiene cuatro objetos escalares y una tabla. Los objetos escalares mantienen estadísticas de los datagramas UDP. La tabla, udpTable, contiene la información de la dirección el puerto.
egp(8)	Grupo mandatorio para todos los sistemas que tengan implementado el protocolo EGP.El grupo EGP incluye cinco objetos escalares y una tabla. Los objetos escalares mantienen estadísticas de los mensajes del protocolo EGP. La tabla, egpNeighTable, contiene información de los vecinos EGP.
cmot(9)	Especificaciones del servicio de información de administración común en tcp
transmission(10)	Especificaciones del medio de transmisión
snmp(11)	Especificaciones del protocolo SNMP

Tabla 6: Categorías del subárbol Mgmt(2) (Ref 1)

Subárbol Experimental(3)

En esta rama se encuentran protocolos experimentales y desarrollos de MIB. Todos los objetos bajo esta rama tienen identificadores que empiezan con 1.3.6.1.3.

Subárbol Privado(4)

Esta rama es usada para especificar objetos definidos unilateralmente por parte de fabricantes de hardware y software. Para muchos sistemas administradores de redes la rama más accedida de este subárbol es enterprise(1). Enterprise representa a la empresa u organización que ha registrado sus propias extensiones al MIB. Cada subárbol bajo este nodo es asignado a una única empresa. La empresa luego puede crear atributos bajo este subárbol que son específicos a sus productos. Por ejemplo, un dispositivo de red de un fabricante tiene su propia extensión del subárbol enterprise, en la cual están atributos propios del equipo que complementan al mib estándar. Los atributos que se encuentran en esta extensión, proveen funcionalidad necesaria para la administración de características propias del equipo no incluidas dentro del estándar MIB.

2.4 Protocolo SNMP y SNMPv2

El protocolo de para la administración de redes más usado en la actualidad es, el Simple Network Management Protocol (SNMP). Este protocolo fue primero propuesto en el RFC 1067 en el cual se define como la información es intercambiada entre los sistemas de administración y los agentes (elementos de hardware/software). Luego fue propuesto el RFC 1098 que hizo obsoleto el RFC 1067. Después, con el RFC 1157 fue aceptado el RFC 1098, reconociendo al protocolo SNMP como un estándar.

El RFC 1157 describe el modelo agente/estación usado en SNMP. Un agente SNMP es un sistema capaz de responder a solicitudes hechas desde una estación SNMP, conocida como sistema administrador de la red, acerca de información definida en el MIB. Los agentes son módulos de programas que corren en los dispositivos administrados y tienen acceso a la información del equipo facilitándola al sistema administrador. Tal esquema se muestra en la figura 20.

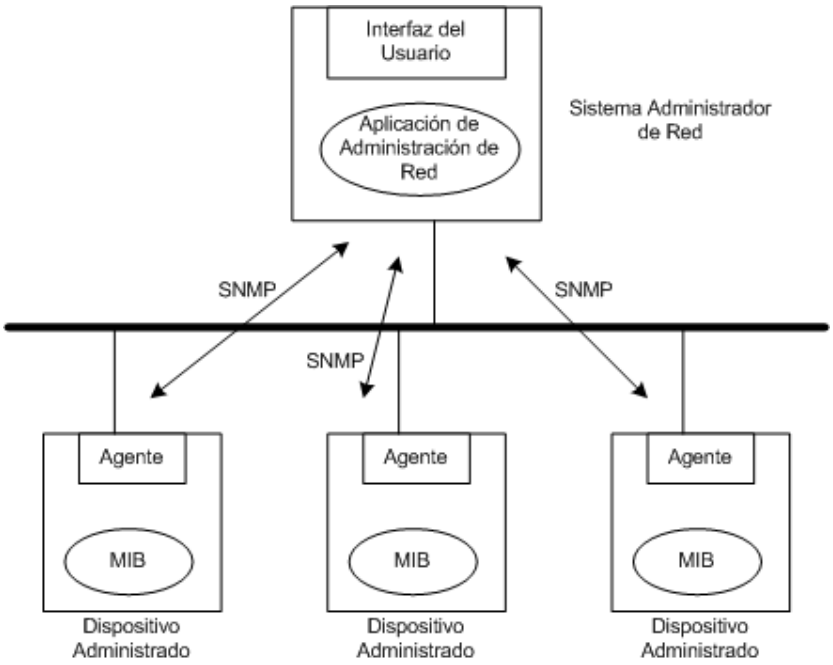


Figura 20: Modelo de Agente/Estación del protocolo SNMP (Ref 1)

Los agentes y estaciones SNMP se comunican a través de mensajes comunes. Un mensaje puede ser enviado en un solo paquete entre la estación y el agente. Cada paquete, que puede contener todo o parte de un intercambio de mensaje, es llamado PDU (Unidad de datos de protocolo).

SNMP usa el protocolo UDP como capa de transporte. UDP provee un servicio no orientado a conexión, por lo que SNMP no tiene que mantener una conexión entre un agente y una estación para transmitir un mensaje; además UDP provee un servicio rápido de transporte con una

mínima cantidad de recursos usados, pero con ciertas desventajas cuando es necesario intercambiar mensajes muy grandes entre un agente y una estación, el diagrama del protocolo SNMP se muestra en la figura 21.

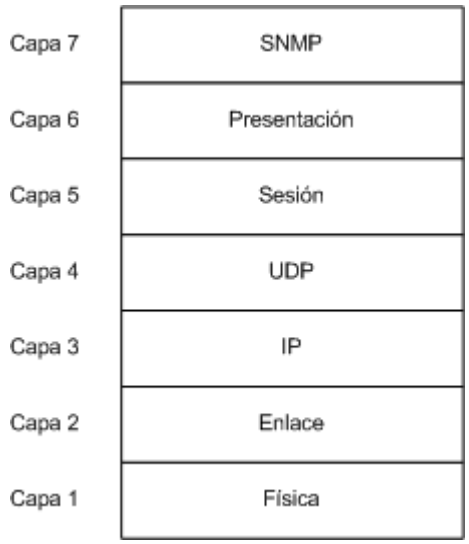


Figura 21: Protocolo SNMP en el modelo de referencia de la ISO (Ref 1)

2.4.1 Tipos de Mensajes

SNMP para la comunicación entre una estación y un agente utiliza mensajes. Estos mensajes pueden ser:

- GetRequest
- GetResponse

- GetNextRequest
- SetRequest
- Trap

Una estación SNMP usa el mensaje GetRequest para obtener información de un dispositivo de red que tiene un agente SNMP. El agente SNMP responde al mensaje GetRequest con un GetResponse. La información retornada incluye el nombre del sistema, cuanto tiempo el sistema ha estado prendido, y el número de interfaces en el sistema.

El mensaje GetNextRequest es usado junto con GetRequest para obtener una tabla de objetos. Primero, GetRequest obtiene un objeto específico, luego GetNextRequest pregunta por el siguiente objeto en la tabla. El agente responde al mensaje GetNextRequest con un mensaje GetResponse, y así sucesivamente hasta llegar al fin de la tabla, el cual es indicado por un mensaje de error que el agente envía.

El mensaje SetRequest permite configurar remotamente parámetros de los dispositivos tales como, nombre del equipo, dirección de una interface, velocidad de una interface, etc.

Un Trap es un mensaje no solicitado que un agente SNMP envía a una estación (ver figura 22). Tiene por objeto informar al servidor sobre la ocurrencia de un evento específico. Por ejemplo, un mensaje Trap puede ser usado para informar al sistema administrador de la red que un circuito acaba de fallar, que un disco está alcanzando su límite de capacidad, o que un usuario ha entrado a un sistema. Para enviar este mensaje, el agente debe conocer la dirección de la estación a la cual debe enviar los traps.

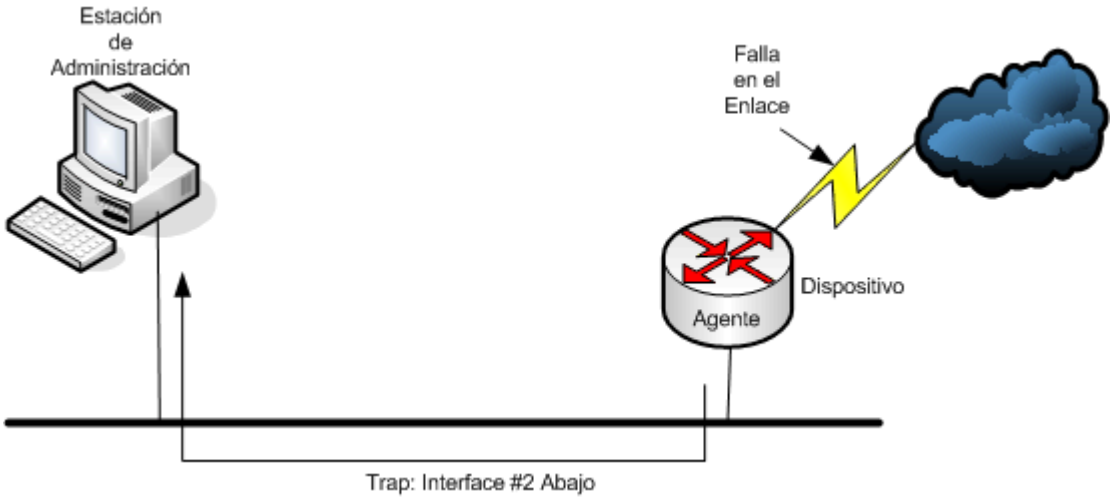


Figura 22: Mensaje Trap (Ref 1)

Como parte del MIB-II, se definen siete tipos de SNMP traps, los cuales se muestran en la tabla siguiente:

Trap	Nombre del Trap	Información
1	Arranque en frío	Enviado cuando el equipo se enciende
2	Arranque en caliente	Enviado cuando un equipo se reinicia
3	Interface abajo	Enviado cuando un enlace específico en el equipo ha fallado
4	Interface arriba	Enviado cuando un enlace en el equipo se ha reestablecido
5	Falla en la autenticación	Enviado al sistema de administración de la red si el agente SNMP determina que una petición de un sistema administrador no provee una autenticación adecuada(comunidad SNMP errónea)
6	Pérdida del vecino EGP[8]	Usado por el agente SNMP para reportar la pérdida de un vecino EGP
7	Específico del fabricante	Implementado por el vendedor del equipo para proveer funcionalidad adicional que complementa a los traps genéricos

Tabla 7: Clases de Traps existentes (Ref 1)

Algunas compañías han implementado traps basados en el uso del disco de una estación, número máximo de usuarios, alta utilización del procesador, etc. Algunos dispositivos de red pueden enviar traps

[8] EGP Exterior Gateway Protocol Documentado en RFC 904, es un protocolo de enrutamiento para enviar información de disponibilidad entre sistemas autonomos.

basados en la utilización de un enlace, tasas de errores, fallas de fuentes de poder, etc.

2.4.2 Formatos de los mensajes

Los mensajes SNMP tienen el formato mostrado en la figura 23.

Figura 23: Formato de un mensaje SNMP (Ref 1)

- Versión.- Número de versión del protocolo SNMP. El agente y el administrador tienen que usar el mismo número de versión.

- Comunidad.- Es un nombre usado para autenticar a una estación antes de permitir el acceso al agente.

- PDU.- Porción del mensaje que contiene el tipo de mensaje y las variables asociadas. Como se explicó anteriormente, hay cinco tipos diferentes de mensajes: GetRequest, GetNextRequest, GetResponse, SetRequest, y Trap.

Formato de los PDU

El formato de los diferentes PDU se muestra en la figura 24.

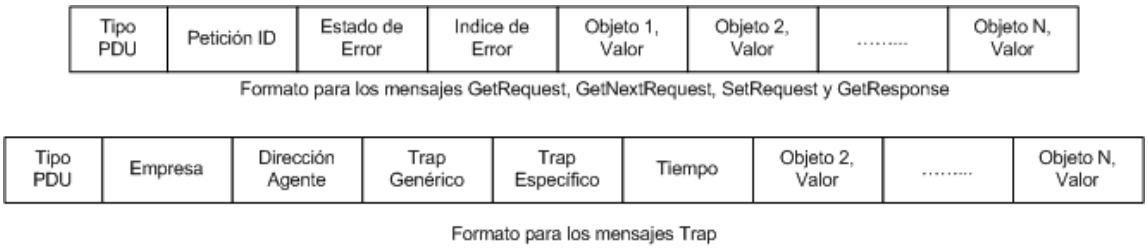


Figura 24: Formatos de los diferentes PDU (Ref 1)

El campo tipo PDU que se indica en la figura 24 especifica el tipo de PDU y puede tener los siguientes valores:

Valor	Tipo de Mensaje
0	GetRequest
1	GetNextRequest
2	GetResponse
3	SetRequest
4	Trap

Tabla 8: Valores y tipos de mensajes PDU (Ref 1)

Formato del paquete tipo GetRequest, GetNextRequest, SetRequest y GetResponse

A continuación se indica el significado de cada uno de los campos:

- Petición ID.- Número que sirve para relacionar la petición de una estación con la respuesta del agente.

- Estado de Error.- Indica una operación normal o una de las cinco condiciones de error. Los posibles valores son:

Valor	Estatus	Descripción
0	noError	Operación libre de errores
1	tooBig	El tamaño de la respuesta GetResponse excede la limitación local
2	noSuchName	El objeto solicitado no corresponde a un objeto del MIB del agente
3	badValue	El SetRequest contiene un tipo inconsistente, longitud y valor para la variable
4	readOnly	No definido en el RFC1157
5	genErr	Otro error no definido ha ocurrido

Tabla 9: Tipos de error en los mensajes de SNMP (Ref 1)

- Indice de Error.- Identifica la variable que causa el error.

- Objeto/Valor.- Asociación entre una variable y su valor. Existe un campo de este tipo por cada objeto sobre el que se esté ejecutando la operación.

Formato del Trap

- Empresa.- Identifica la empresa bajo la cual se ha registrado el trap, este campo solo se especifica cuando el trap es un trap específico.

- Dirección del agente.- Dirección IP del agente, usado para identificación futura.

- Trap Genérico.- Campo que describe el evento reportado. Descrito anteriormente.

- Trap específico.- Usado para identificar un trap definido por el fabricante.

- Tiempo.- Representa la cantidad de tiempo transcurrido entre el último reinicio del equipo y la generación del trap.

- Objeto/Valor.- Tupla que contiene el identificador de una variable y su valor.

2.4.3 Problemas del protocolo SNMP versión 1

El protocolo SNMP es muy poderoso para la administración de redes, pero tiene algunos inconvenientes como los que se mencionan a continuación:

- Es un estándar definido solo para redes que usan el protocolo IP. Una posible solución aplicable a este problema, en caso de necesitar manejar otros protocolos de red, es la de usar los llamados agentes proxy SNMP, los cuales pueden recolectar información de dispositivos de red que no usan IP y enviar la información a las estaciones SNMP, permitiendo que los dispositivos que no usan IP puedan ser administrados a través de SNMP.

- Es ineficiente para transmitir grandes tablas de datos. La recolección de información de las tablas se la realiza con el comando GetNextRequest (explicado anteriormente). Por ejemplo si tuviéramos una tabla con 2000 filas, para cada una de las filas se enviaría una petición.

- Usa únicamente una cadena de texto simple no cifrada para identificar a los miembros de una comunidad, haciendo su autenticación bastante insegura. Alguien que pueda leer paquetes del segmento de la red puede encontrar fácilmente el identificador

de la comunidad. Si eso ocurre, la persona tiene acceso a leer y cambiar la configuración de los dispositivos de red que usan SNMP.

2.4.4 SNMP versión 2

Esta versión tiene la misma funcionalidad básica que la versión 1, esto es, adquirir y cambiar datos del MIB en los dispositivos de red. Esta versión fue desarrollada por los miembros de la comunidad de Internet para solucionar los diferentes problemas que presenta la versión 1.

La versión dos como se la conoce, no fue el primer intento de mejorar la antigua versión. La primera propuesta que pretendió añadir seguridad a SNMP fue llamada SNMP seguro, la cual fue definida en un conjunto de RCFs (RFC1321 y RFC1351-RFC1353) en Julio de 1992. Sin embargo, esta propuesta no resolvió los problemas de recuperación ineficiente de las tablas, y el uso de IP como el único protocolo de la capa de red, entre otras deficiencias. Para trabajar en estos problemas, un protocolo llamado SMP(Protocolo de administración simple) fue desarrollado e introducido en Julio de 1992 como un conjunto de documentos (no RFCs) enviados a la comunidad de Internet.

Después de esta propuesta, los miembros de la comunidad de Internet comenzaron a fusionar las propiedades de SNMP seguro y SMP en una nueva versión del protocolo SNMP, conocida ahora como SNMP versión 2.

SNMPv2 tiene características que mejoran algunas falencias de la versión previa, incluyendo adiciones al SMI, nuevo tipos de mensajes, soporte multiprotocolo, mejoras significativas en la seguridad, nuevos objetos MIB, y compatibilidad para coexistir con implementaciones de la versión 1.

2.4.4.1 Mejoras a la Estructura de la información de administración(SMI)

El estándar SMI para la versión 1 tiene deficiencias que incluyen la de tener un número máximo de 2^{32}-1 para los datos enteros del MIB, y la de solo poder representar direcciones para redes IP.

SNMPv2 permite tener datos enteros de 64 bits, identificados por el tipo de datos Counter64. Los números de 32bits, llamados Counter y Gauge en el SMI para la versión 1, fueron renombrados a Counter32 y Gauge32. El Counter64 permite manejar valor es

suficientemente grandes para características que requieren conteo que de otra forma causarían un problema al estar limitadas a números de 32 bits.

Una de las consideraciones en el uso de variables enteras de 64 bits para todos los objetos enteros de un MIB en un agente, es la memoria necesaria para mantener esos valores que puede llegar a ser el doble de la usada por enteros de 32 bits. Para dispositivos con memoria limitada (modems, tarjetas de red), esto puede ser un problema, por eso muchos MIBs estándares tienen un número limitado de objetos representados como enteros de 64 bits.

Otra ventaja del SMI de la versión 2, es la de poder representar números con signo. El bit de signo en los números binarios puede ser usado para representar ya sea un número negativo o positivo, o puede ser usado como un bit adicional para la representación del número.

Finalmente, el SMI para la versión 2 también añade un nuevo tipo de datos para representar la dirección del OSI NSAP(punto de servicio de acceso a la red), llamado NsapAddress. El NSAP es una dirección de red jerárquica usada por la capa de red de la OSI.

2.4.4.2 Soporte multiprotocolo

El SNMPv1 está estandarizado para trabajar solamente con redes IP. Así para redes que sólo usen AppleTalk o Novell IPX, no es posible su administración vía SNMPv1 sin añadir el protocolo IP. Por otro lado SNMPv2, está estandarizado para funcionar en diferentes protocolos tales como: IP, Apple AppleTalk, Novell IPX, y OSI Connectionless Network Service (CLNS). Por lo tanto los mensajes SNMPv2 pueden ser trasnmitidos por la mayoría de protocolos usados en la actualidad (ver figura 25).

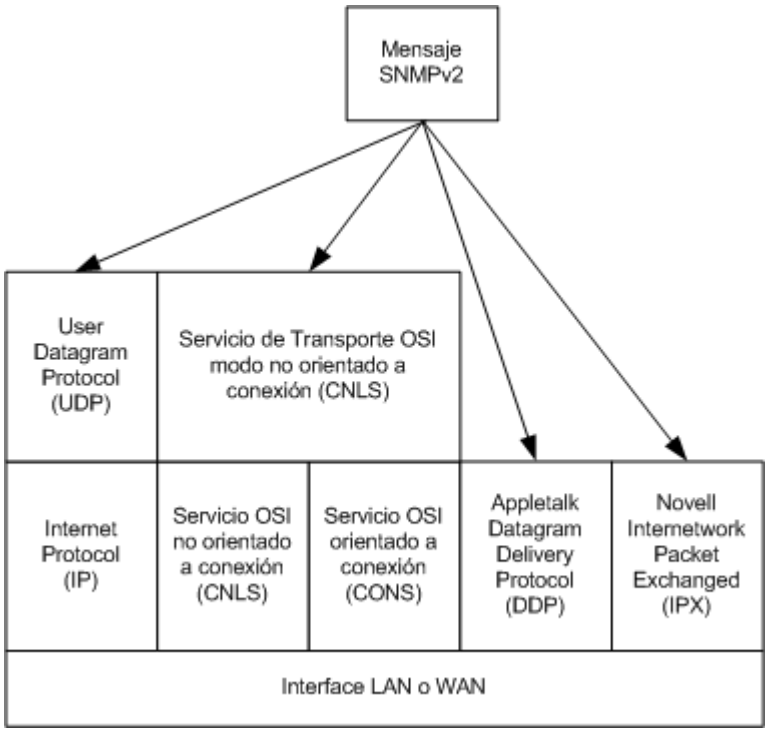


Figura 25: Soporte multiprotocolo del SNMPv2 (Ref 1)

2.4.4.3 Tipos de mensajes

Todo el conjunto de mensaje que existe en la versión 1 (GetRequest, GetNextRequest, GetResponse, SetRequest, y Trap) son incorporados en el SNMPv2. La diferencia principal es que la versión 1 usa el mismo PDU para todos los mensajes, excepto para el Trap, mientras que la versión 2 usa un mismo formato para todos los mensajes excepto para GetBulkRequest.

Hay dos nuevos tipos de mensajes en esta versión: InformRequest y GetBulkRequest. Cada uno provee de funcionalidad adicional tanto al agente como al manager(o estación).

El mensaje InformRequest es enviado por un manager a otro manager permitiendo a una aplicación enviar información a otra para llevar a cabo una comunicación manager-manager. Este nuevo tipo de mensaje puede ser usado como un sistema de comunicación jerárquico o distribuido para la administración.

El mensaje GetBulkRequest ayuda a optimizar la recuperación de grandes cantidades de información de administración, el cual era uno de los grandes problemas del SNMPv1. Con el mensaje

GetBulkRequest se pueden solicitar múltiples datos de un objeto con un solo mensaje en lugar de solicitarlos uno a uno. Es decir, para una tabla, se pueden pedir los siguientes X valores en la tabla (donde X puede ser cualquier valor entero) en un solo requerimiento de este tipo.

Para entender mejor el mensaje GetBulkRequest definamos lo siguiente:

L= el número total de nombres de variables(nombres de objetos MIB) en una petición o solicitud

N= el número de nombres de variables(comenzando desde el principio de la lista de nombres) que no son repetidos, para lo cual se necesita un solo valor.

R= el número de variables después de las primeras N que son repetidas, para las cuales se requieren múltiples peticiones

M= el número de veces que se quiere recorrer el árbol MIB para cada variable R

Usando las definiciones anteriores, se puede ver que R= L – N. El número total de variables que requieren solicitudes repetidas al MIB es igual al número total de variables en una petición, menos

las variables que requieren solo una petición. También se nota que el número total de valores de variables solicitadas en un mensaje GetBulkRequest es N+(M*R). Este es número total de peticiones simples más el número total de peticiones repetitivas por el número de peticiones.

Cuando se recibe un mensaje GetBulkRequest, un agente debe determinar cuantos objetos están siendo solicitados (L), encontrar el número de objetos que requieren solo una petición (N), encontrar el número de objetos que requerirán solicitudes repetitivas (R), y determinar cuantas peticiones el mensaje solicita para cada objeto R (M). Luego el agente puede procesar la solicitud e intentar retornar los valores de los objetos MIB eficientemente.

Por ejemplo, si un agente recibe una petición GetBulkRequest con los valores L=5, N=2, R=3, y M=10, hay cinco objetos (L) siendo solicitados en la petición GetBulkRequest. Los primeros dos objetos no son repetidos (N); los tres objetos siguientes son repetidos (R). El agente tratará de hacer 10 (M) peticiones para cada objeto especificado en R.

2.4.4.4 Formato de los mensajes

En esta sección se explicará el detalle del mensaje que se conoce como PDU y más adelante se mostrará en que consiste la cabecera del mensaje SNMPv2 (elemento que no existe en la definición de SNMPv1).

Figura 26: Formato del mensaje de SNMPv2 (Ref 1)

Para simplificar el procesamiento de los PDU, todos los mensajes excepto GetBulkRequest usan el mismo formato que se expone a continuación:

Formato de los PDU

El formato de los diferentes PDU se muestra en la figura 27

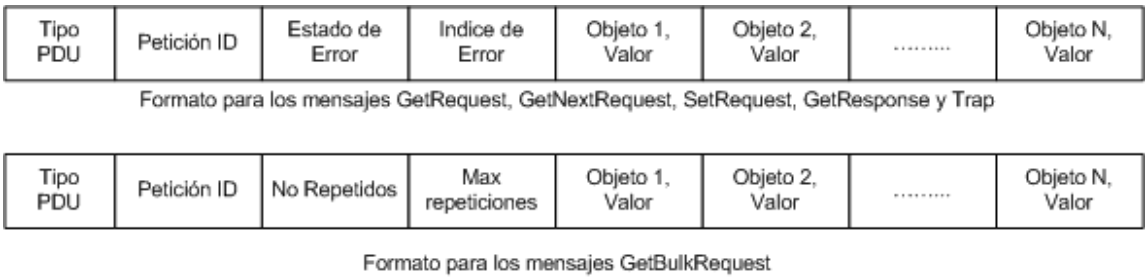


Figura 27: Formatos de los diferentes PDU (Ref 1)

A continuación se explica el significado de cada ara el formato que corresponde a los mensajes de GetRequest, GetNextRequest, SetRequest, GetResponse y Trap

- Tipo PDU.- Puede tener uno de los siguientes valores que fundamentalmente indican el tipo de petición que puede realizar una estación a un agente, a excepción de los tipos 6 (de estación a estación) y 7 (del agente a la estación)

Valor	Tipo de Mensaje
0	GetRequest
1	GetNextRequest
2	GetResponse
3	SetRequest
5	GetBulkRequest
6	InformRequest
7	SNMPv2 Trap

Tabla 10: Valores y tipos de mensajes PDU (Ref 1)

- Petición ID.- Es un número aleatorio que relaciona la petición de una estación con la respectiva respuesta del agente.

- Estado de Error.- Indica la operación normal o una de las 17 condiciones de error. Los posibles valores son:

Valor	Estatus
0	NoError
1	TooBig
2	NoSuchName
3	BadValue
4	ReadOnly
5	GenErr
6	NoAccess
7	WrongType
8	WrongLength
9	WrongEncoding
10	WrongValue
11	NoCreation
12	InconsistentValue
13	ResourceUnavaliable
14	CommitFailed
15	UndoFailed
16	AuthorizationError
17	NotWritable
18	InconsistentName

Tabla 11: Tipos de error en los mensajes de SNMP (Ref 1)

- Indice de Error.- Identifica la variable que causa el error.

- Objeto/Valor.- Tupla que contiene el identificador de una variable y su valor.

El mensaje GetBulkRequest, tiene un formato diferente a los anteriores y el significado de cada uno de sus campos se explica a continuación:

- Tipo PDU.- Especifica el tipo de PDU. Para el mensaje GetBulkRequest, el valor es 5.

- Petición ID.- Es un número aleatorio que relaciona la petición de una estación con la respectiva respuesta del agente.

- No Repetidos.- Número de variables solicitadas que no serán procesadas repetidamente, por ejemplo, una sola instancia de una variable.

- Max-Repeticiones.- Número máximo de ejecuciones repetidas para requerir una variable específica.

- Objeto/Valor.- Tupla que contiene el identificador de una variable y su valor.

La cabecera del paquete SNMPv2 se describirá más adelante junto con la explicación de la seguridad que soporta el protocolo SNMPv2.

2.4.4.5 Arquitectura de Administración

SNMPv2 soporta la misma administración centralizada que SNMPv1 , así como también estrategias distribuidas basadas en el MIB manager-manager. En una arquitectura distribuida, existen sistemas que operan de ambas formas: como administradores o como agentes. Cuando actúan como agentes, aceptan comandos de un sistema de administración superior en la jerarquía. Estos comandos pueden requerir información almacenada localmente, o pueden requerir que el sistema de administración que actúa como agente (tomando un rol diferente) a su vez provea información de agentes subordinados (para los cuales es visto como un administrador). Adicionalmente, un administrador intermediario puede enviar un trap a un administrador superior.

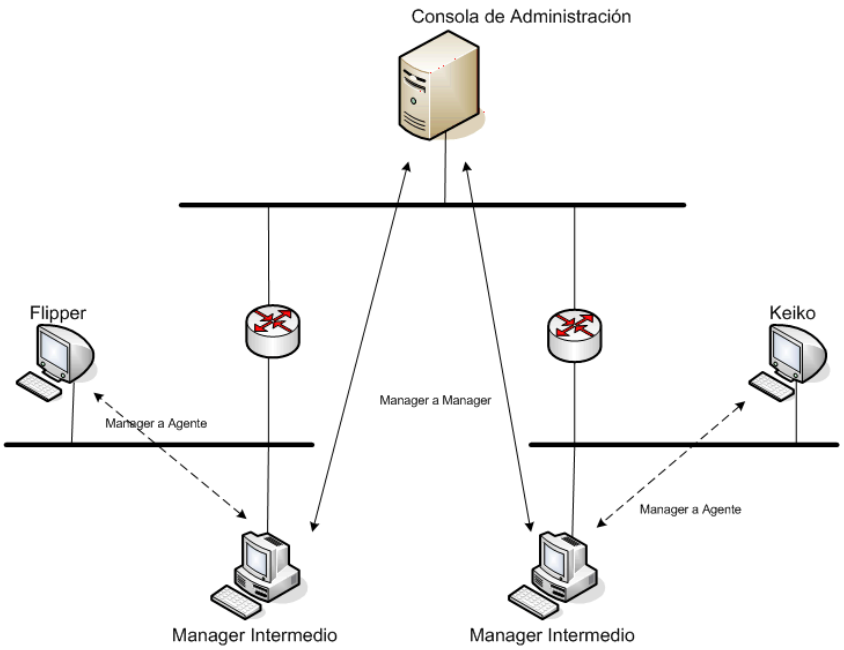


Figura 28: Arquitectura de administración del SNMPv2 (Ref 1)

2.4.4.6 Seguridad en SNMPv2

El mecanismo de seguridad del SNMPv1 consiste en el nombre de la comunidad que es configurado en cada uno de los dispositivos o sistemas. Aunque este mecanismo previene el acceso indiscriminado para monitorear y cambiar objetos del MIB; un ingeniero que conoce de redes, teniendo el equipo o programa apropiado, puede determinar la comunidad usada en una red y a partir de allí tener completo control de su monitoreo y configuración.

SNMPv2 tiene un mecanismo de seguridad más refinado que provee tanto la autenticación como el cifrado de los mensajes SNMP. La información de seguridad se encuentra en la cabecera del mensaje SNMPv2 (revisar figura 26), en tres modalidades diferentes tal como la indica la figura 29. En la primera, el mensaje se transmite sin ninguna seguridad, por lo que no se utilizan algunos campos de la cabecera. En la segunda se puede autenticar el mensaje pero no se realiza ningún proceso de cifrado. En la tercera (que es la más segura de las tres), se autentica el mensaje y se cifra el PDU y parte de la cabecera.

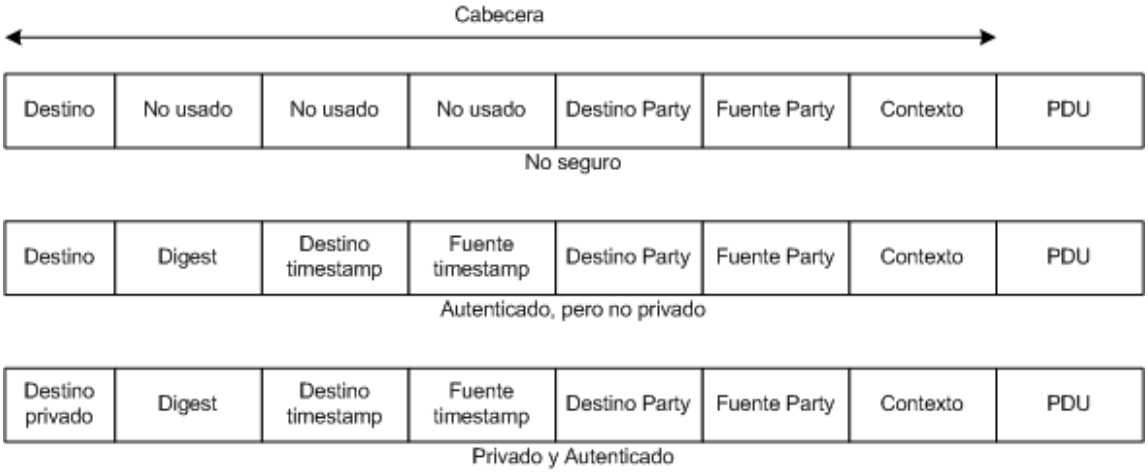


Figura 29: Formato del mensaje SNMPv2 (Cabecera detallada) (Ref 1)

En el formato de la cabecera del mensaje SNMPv2, el campo “Fuente Party” identifica la estación o el agente que envía el mensaje. El campo “Destino Party” identifica al manager o agente a quien va dirigido el mensaje, que también se repite en el campo “Destino Privado”. Esta repetición es necesaria porque el mensaje puede estar encriptado en todos los campos siguientes a partir del campo Destino Privado, el cual se mantiene en texto normal para identificar fácilmente el destino. Si el mensaje requiere autenticación, también contiene un “Digest” (que se explicará más adelante), “Fuente timestaamp” que es el valor del reloj del sistema que envía el mensaje, y “Destino timestamp” que es el valor de reloj del sistema destino que el sistema origen obtuvo de un intercambio previo de mensajes. Esta información es necesaria para que las partes involucradas en el intercambio de mensajes puedan identificarse entre ellas y verificar la integridad del mensaje cuando es enviado o recibido.

Las partes involucradas en una comunicación SNMPv2 son un grupo de entidades que intercambian información de administración. Cada una tiene un conjunto de propiedades que gobiernan accesos y privilegios a la información del MIB. Esas propiedades incluyen autenticación, encriptación, Vistas del MIB, y

contextos (los dos últimos tienen que ver más con el acceso que con la seguridad).

Autenticación en SNMPv2

El protocolo SNMPv2 provee un método seguro para autenticar la comunicación entre las partes; este método es llamado digest authentication protocol. El digest es una huella que se obtiene aplicando un algoritmo de codificación sobre el mensaje SNMP para poder luego asegurar que el mensaje recibido fue el mensaje enviado y que la fuente del mensaje ha sido autenticada.

Para esto, la fuente calcula el digest del mensaje (usando un algoritmo llamado MD5), usando una llave de autenticación. El digest del mensaje es un número de 128 bits relacionado al mensaje y su contenido. El mensaje y el digest son enviados a la red. Cabe anotar que la llave de autenticación no es enviada con el mensaje.

Cuando el destino recibe el mensaje, recalcula el digest del mensaje, usando su propia copia local de la llave de autenticación. Si el nuevo digest calculado concuerda con el enviado en el mensaje, el mensaje es autenticado. El uso del digest confirma

que el mensaje recibido es el mensaje que se envió, y que se utilizó la misma llave.

Como medida adicional de autenticación, el MD5 usa valores de reloj (timestamps) en el mensaje para asegurar que el mensaje no fue capturado y reenviado para obtener acceso no autorizado. Cada parte tiene un tiempo máximo para enviar cada mensaje (lifetime), y los valores de reloj en el mensaje aseguran que el tiempo máximo del mensaje no haya sido excedido.

Otro de los problemas que tienen que ser manejados por el protocolo de autenticación, es la distribución que la llave de autenticación a los diferentes sistemas. Para esto, este protocolo usa un algoritmo de clave pública para distribuir las llaves de autenticación necesarias para ejecutar el algoritmo que calcula el digest en los mensajes.

Encriptación en SNMPv2

Si una parte especifica que un mensaje requiere encriptación, se utiliza el algoritmo estándar de encriptación de datos (DES).

Cuando se especifica que la comunicación entre entidades debe ser encriptada, todo el mensaje, excepto el campo “Destino

Privado", es encriptado antes de la transmisión. El destinatario, o receptor, desencripta el mensaje usando el mismo algoritmo, usando la misma llave enviada al inicio de la comunicación.

2.4.4.7 MIBs en SNMPv2

Tres MIBs son definidos para ayudar en la administración de SNMPv2: MIB SNMPv2, Manager-Manager MIB, y party MIB. Esos MIBs proveen información vital para la administración, configuración y monitoreo del protocolo SNMPv2.

2.4.4.7.1 MIB SNMPv2

Definido en el RFC1450, titulado "Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2)", define objetos que describen el funcionamiento del SNMPv2. Este MIB está dividido en cinco grupos:

1) Estadísticas de SNMPv2.- Provee objetos que dan estadísticas acerca del SNMPv2 manager o agente, en particular, información de mensajes que no pueden ser procesados. Este grupo es similar en función, que le grupo SNMP en el MIB-II.

2) Estadísticas de SNMPv1.- Provee objetos que dan información estadística acerca del manager o agente compatibles con agentes SNMPv1.

3) Recursos de objetos.- Provee información de los objetos que un agente SNMPv2 puede definir dinámicamente. Cada recurso de objeto tiene un identificador de objeto(OID) y una descripción. Consultando esta tabla, un manager puede determinar que parámetros SNMPv2 puede configurar dinámicamente en un agente. Uno de esos objetos puede indicar que un manager y un agente en particular necesitan encriptación o autenticación para el envío de mensajes.

4) Traps. – Contiene una tabla de información acerca de cada uno de los traps que un agente puede envíar. Cada trap tiene un OID asociado y un contador que indica el número de veces que un trap ha sido enviado. Por ejemplo, esta tabla puede contener información diciendo que un agente puede enviar un trap indicando de una congestión en un circuito WAN. El contador asociado a este trap, muestra el número de veces que el agente ha eviado el trap al manager.

5) Set.- Provee un solo objeto que permite a múltiples managers enviar un mensaje Set a un agente sin problemas de

sincronización. Este objeto, conocido como número de serie set, ayuda a prevenir que dos managers traten de cambiar un mismo objeto MIB simultáneamente incrementando el valor de este objeto para cada mensaje set procesado.

2.4.4.7.2 MIB Manager-Manager

Definido en el RFC1451, “Manager-to-Manager Management Information Base”, este conjunto de objetos provee información de cómo funciona una entidad, ya que en SNMPv2 una entidad puede funcionar como un manager o un agente (ver figura 28).

Este MIB contiene dos grupos:

1) Alarmas.- Proveen información descriptiva y configuración de alarmas. El mecanismo de alarma periódicamente toma muestras de variables SNMPv2 y las compara con limites preestablecidos. La tabla de alarmas almacena información de la variable, periodo de muestreo, y limites. Cuando un límite es excedido, la alarma genera un evento.

2) Eventos.- Proveen información de la configuración de los eventos en una entidad SNMPv2. La tabla de eventos asocia

un tipo de evento con el método de notificación y parámetros asociados. Este grupo contiene también una tabla de notificación que define las notificaciones que deben ocurrir cuando un evento asociado se activa.

2.4.4.7.3 Party MIB

Este MIB está definido en el RFC1447, y contiene objetos que describen y configuran los sistemas involucrados con una entidad SNMPv2. Los cuatro grupos de este mib son: grupo de base de datos de las partes, grupo de base de datos de contextos, grupo de base de datos de privilegios de acceso, grupo de base de datos de vistas del MIB.

El grupo de base de datos de los sistemas involucrados contiene información, la cual es almacenada en el dispositivo, acerca de todos los agentes o sistemas administradores remotos o locales. Algunos de los datos de este grupo indican si una parte es local o remota, cual el protocolo de autenticación a usar, cual es el tiempo de vida, y como la información es almacenada.

Los otros grupos tienen que ver con los privilegios entre un manager y un agente. Estos grupos permiten el control de contextos locales o remotos en una entidad SNMPv2, las políticas de acceso que el agente o el manager implementen, y las vistas MIBs definidas así como las partes que tienen acceso a ellas.

Vistas MIBs

Una parte puede tener el acceso controlado a porciones del MIB en un agente. La porción del MIB que es accesible al manager es llamada vista del MIB. Un agente puede permitir solo a ciertos managers acceder a ciertas partes de la base de datos del MIB local.

Contextos

Un contexto SNMPv2 es un conjunto de objetos manejados que un manager o agente pueden accesar.

2.4.4.8 Coexistencia con SNMPv1

El protocolo SNMPV2 fue creado como un sucesor del SNMPv1 por lo que, la migración de la versión 1 a la versión 2 no es muy

complicada. Para la transición, los managers y agentes de la versión 1, tienen que manejar los cambios en la información SMI y los cambios en el formato de los mensajes de la versión 2.

Los cambios en el SMI hacen necesario que tanto agentes como managers sean mejorados para que entiendan los nuevos tipos de datos y mensajes. Pero como el SMI para la versión 2 es un superconjunto de la versión 1, todas las definiciones MIB del SNMPv1 son compatibles con los agentes y managers que implementen SNMPv2.

Los mensajes en SNMPv2 son muy similares a la versión anterior. Los dos protocolos tienen mensajes GetRequest, GetNextRequest, SetRequest, GetResponse y Trap. Adicionalmente, el mensaje GetBulkRequest de la versión 2, puede ser interpretado como una serie de mensajes GetNextRequest de la versión 1. El fácil traslado entre los tipos de mensajes de los dos protocolos permite dos posibles estrategias para la comunicación entre entidades SNMPv1 y SNMPv2. La primera es tener un agente proxy que realice la traducción de mensajes. La segunda es tener un manager que entienda los dos protocolos, SNMPv1 y SNMPv2.

Hoy en día la segunda opción es más aceptada porque el manager puede tomar la decisión acerca de que protocolo es necesario para comunicarse con cada uno de sus agentes.

2.4.4.9 Problemas con SNMPv2

El SNMPv2 es un protocolo de administración de redes bien definido y funcionalmente completo que tiene mejores características que la versión 1. El problema que existe, es que como el SNMPv1 todavía es muy usado, el SNMPv2 carece de implementación, además de que el uso de la encriptación y autenticación implica que los agentes deben tener mayor poder de procesamiento y almacenamiento para las tareas de administración

2.5 Estructura del protocolo de la OSI para la administración de redes

SNMPv1 tiene una mejor aceptación en la industria de las comunicaciones, y SNMPv2 tiene muchas características que los ingenieros de redes desean, pero muchas personas sienten que el protocolo de administración que mejor satisface las necesidades de administración de redes es el protocolo de administración de la OSI:

Common Management Information Services/Common Management Information Protocol(CMIS/CMIP). CMIS/CMIP tiene muchas características útiles para las tareas de administración.

CMIS define los servicios generales que provee cada componente de red para la administración; CMIP es el protocolo que implementa los servicios CMIS. El conjunto de protocolos OSI provee una arquitectura común para cada una de las capas del modelo de referencia OSI, de la misma manera, CMIS/CMIP provee un conjunto completo de protocolos de administración para usarse con cualquier dispositivo de red.

La diferencia básica entre CMIS/CMIP y cualquier versión de SNMP es que para el protocolo CMIS/CMIP, los manager (puede haber varios) pueden delegar a los dispositivos administrados para que realicen muchas más tareas de manera activa, mientras que para el protocolo SNMP se realiza toda la administración en el administrador, permitiendo que el agente sea muy simple. CMIS/CMIP distribuye la carga de la administración más equitativamente, requiriendo más recursos y capacidades de los dispositivos administrados.

Para entender el protocolo CMIS/CMIP es necesario entender la terminología de OSI respecto a la administración de redes. Los términos más usados son:

- Sistema abierto.- Es un sistema, como un router o una estación de trabajo, que usa el protocolo de redes OSI.

- Sistemas abiertos iguales.- Dos dispositivos que se comunican usando los protocolo de una misma capa del Modelo de Referencia OSI

La estructura del protocolo de OSI para la administración de la red sigue el modelo de referencia de la OSI. Los procesos de administración de la red interactúan con la capa de aplicación del modelo de referencia de OSI; concretamente con el protocolo Common Management Information Service Element (CMISE), especificado en el documento ISO 9595 y 9596, el cual define como se provee la información a las aplicaiones que usan CMIP. CMISE, a su vez, se vale de otros dos protocolos definidos en la capa de aplicación: el Association Control Service Element(ACSE, especificado en el documento ISO 8649 y 8650, y el Remote Operation Service Element(ROSE), especificado en el documento ISO 9072-1 y 9072-2. La figura 30 muestra la relación entre los protocolos. El protocolo ACSE define como se establece y termina las asociaciones entre las aplicaciones; mientras que el protocolo ROSE maneja las interacciones de requerimiento/respuestas entre las aplicaciones. De acuerdo con el modelo OSI, ACSE y ROSE usan el OSI presentation service y los demás protocolos del modelo de referencia OSI.

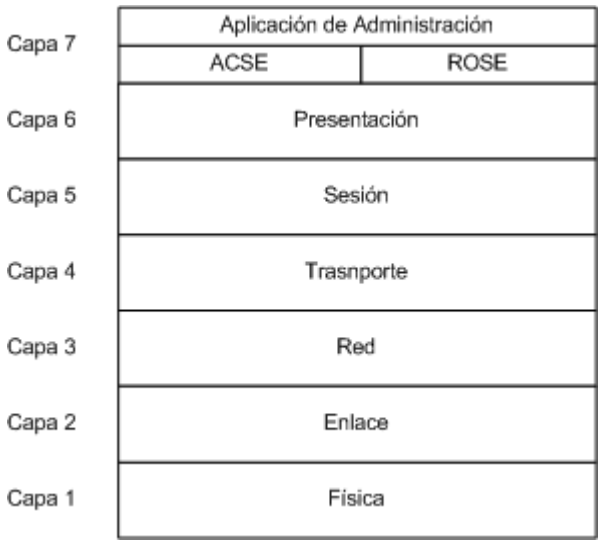


Figura 30: El protocolo CMIP en el modelo de referencia de la OSI (Ref 1)

2.5.1 CMIS

El conjunto de servicios CMIS provee el bloque de funcionalidad básica para un sistema de administración de redes. Cada servicio CMIS es una simple operación que una aplicación de administración de red puede realizar. Cualquier aplicación que funcione como administrador de red es un CMISE-service-user. CMIS ha definido tres clases de servicios para el uso del CMIS-service-users: management association, management notification, y management operation. Cada una de estas clases de servicios está diseñada para diferentes aspectos de la administración de redes.

Management Association Services

La primera clase de servicio, management association services, controla la comunicación entre sistemas abiertos iguales. Esos servicios, usados principalmente para establecer y terminar conecciones entre sistemas, controlan la inicialización, terminación normal, y terminación anormal de una conección con los siguientes servicios:

- M-INITIALIZE: el cual crea una conexión con un sistema CMISE-service-user para administración

- M-TERMINATE: el cual termina una conexión entre sistemas CMISE-service-users

- M-ABORT: usado cuando una conexión entre CMISE-service-users termina bruscamente.

Estos servicios de asociaciones asumen el uso de los servicios de ACSE para su operación, debido a que ACSE es usado para iniciar y terminar una conexión entre aplicaciones. Otros servicios CMIS, que usan una conexión existente para la información de administración, operan con ROSE.

Management Notification Services

Así como los traps SNMP proveen información acerca de eventos en la red, managemente notification services proveen una funcionalidad similar para CMIS. El servicio M-EVENT-REPORT comunica a otro sistema CMISE-service-user acerca de un evento que ha ocurrido en otro CMISE-service-user. Si el CMISE-service-user de un sistema nota el cambio de un valor, le notifica al sistema administrador con el servicio M-EVENT-REPORT.

Los traps de SNMP están perfectamente definidos, los eventos del management notification service no están estrictamente definidos, ellos son específicos al sistema que genera la notificación.

Management Operation Services

Este tercer grupo de servicios proveen las operaciones básicas para la administración. Este tercer grupo, comprende las siguientes operaciones:

- M-GET, usado por un CMISE-service-user para obtener información de administración de otro sistema CMISE-service-user. Es análogo al mensaje SNMP GetRequest.

- M-CANCEL-GET, usado para cancelar una petición M-GET. Si un CMISE-service-user envía una petición M-GET y decide, antes de recibir la respuesta, que no requiere esta información, puede cancelar la petición con un M-CANCEL-GET.

- M-SET, permite a un CMISE-service-user modificar información de administración de otro CMISE-service-user. Este servicio es similar al mensage SNMP SetRequest.

- M-ACTION, servicio invocado por un CMISE-service-user para indicar a otro CMISE-service-user que realice una acción deseada. La acción realizada es relativa a cada dispositivo específico, por ejemplo, un sistema abierto podría pedir a otro sistema abierto enviar mensajes ICMP Echo (pings) a otros dispositivos para probar la conectividad IP.

- M-CREATE, es usado por un CMISE-service-user para pedir a otro CMISE-service-user que cree otra instancia de un objeto administrado. En CMIS, cada objeto que es administrado tiene una instancia asociada. CMIS permite muchas instancias de del mismo objeto, pero sólo una definición del objeto. Esto es similar al concepto de programación orientada a objetos, en el que cada

objeto tiene una definición llamada clase, y cada uso de la definición es llamada instancia de la clase.

- M-DELETE, es el servicio contrario a M-CREATE. Este servicio es usado por un CMISE-service-user pedir a otro CMISE-service-user que borre una instancia de un objeto administrado.

Management Associations

Es la conexión entre dos sistemas abiertos para administración de la red. El proceso de conexión se basa en CMISE que hace de interfaz con otros protocolos del modelo OSI. Hay cuatro tipos de asociaciones pueden existir entre los sitemas abiertos:

- Event.- Una asociación event permite a dos sitemas abiertos enviar mensages M-EVENT-REPORT. Dos sistemas abiertos pueden tener una asociación de eventos cuando necesitan enviarse entre ellos solo eventos de administración. Los sistemas abiertos pueden usar los servicios management association y management notification para una asociación de eventos.

- Event/Monitor.- Esta asociación es la misma que la anterior, excepto que cada sistema también pueden recibir o enviar un

mensaje M-GET. Este tipo de asociación permite que los sistemas abiertos consultar información de administración y recibir eventos de la red.

- Monitor/Control.- Esta asociación permite los mensajes M-GET, M-CANCEL-GET, M-SET, M-CREATE, M-DELETE y M-ACTION entre sistemas abiertos, pero no permite la recepción de eventos de la red. Un CMISE-service-user puede usar una asociación monitor/control para cambiar la configuración de un sistema. En este caso la recepción de los servicios de notificación no son muy importantes, ya que la única tarea es configurar el sistema.

- Full Manager/Agent.- Esta asociación incluye todos los servicios CMIS.

- Access List.- De la misma manera que el SNMPv1 usa comunidades para verificar que un sistema pueda accesar a la información de administración, CMIS usa listas de acceso o access list.

2.5.2 CMIP

El protocolo que implementa el CMIS es el Common Management Information Protocol (CMIP). LA especificación de este protocolo explica en detalle la forma en la cual el protocolo debe realizar un servicio CMIS.

El protocolo CMIP requiere una máquina CMIP, o CMIPM (maquina CMIP), para funcionar de acuerdo a la especificación definida. A CMIPM es un software que realiza dos funciones: Primero, acepta las operaciones enviadas por un CMISE-service-user e inicia los procedimientos apropiados para realizar la operación asociada; segundo, la máquina CMIP usa ROSE para enviar mensajes a través de la red. La figura 31 muestra el camino de una petición de un servicio CMIS entre dos CMISE-service-user.

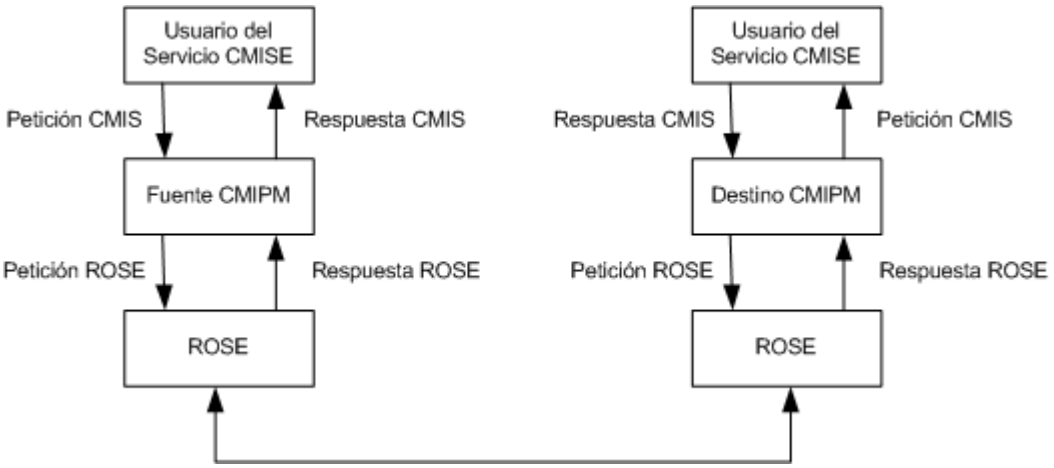


Figura 31: Flujo de una petición de servicio CMIS entre doe CMISE-service-user (Ref 1)

CMIPM usa un conjunto de unidad de datos bien definidos para implementar los servicios CMIS. Cada servicio CMIS usa una serie de esas unidades de datos. Como un ejemplo, dos CMIPM son m-GET y m-Linked-Reply; m-GET obtiene una pieza específica de datos del MIB de un CMISE-service-user, mientras que m-Linked-Reply es usado para responder al m-GET. La tabla 12 muestra los servicios CMIS y sus correspondientes unidades de datos CMIP.

Servicios CMIS	Unidad de datos CMIP
M-EVENT-REPORT	m-EventReport, m-EventReport-Confirmed
M-GET	m-Get, m-Linked-Reply
M-CANCEL-GET	m-Cancel-Get-Confirmed
M-SET	m-Set-Confirmed, m-Linked-Reply
M-ACTION	m-Action, m-Action-Confirmed, m-Linked-Reply
M-CREATE	m-Create
M-DELETE	m-Delete

Tabla 12: Servicios CMIS y sus correspondientes unidades de datos (Ref 1)

CMIP solo define como leer la información en un paquete de datos; no tiene conocimiento que es lo que el CMISE-service-user hará con la información solicitada de un objeto administrado.

2.5.3 Problemas con CMIS/CMIP

Aunque el conjunto de protocolos CMIS/CMIP proveen a los administradores un protocolo capaz de realizar muchas tareas de administración de redes, hay dos problemas críticos: el primero, CMIS/CMIP requiere de una gran cantidad de recursos. Segundo, es difícil de implementar.

Ambos problemas resultan del hecho que el CMIS/CMIP está diseñado para ejecutarse en un protocolo que sigue completamente el

modelo del protocolo de red de la OSI. Muchos dispositivos de red no tienen la memoria o el poder de procesamiento adecuado para soportar el modelo OSI.

2.5.4 CMOT

El Common Management Information Services and Protocol over TCP/IP (CMOT) propone implementar los servicios CMIS encima del conjunto de protocolos TCP/IP como una solución temporal hasta que se desarrolle una solución al conjunto OSI. El RFC 1189 define el protocolo CMOT. La figura 32 muestra el protocolo CMOT en el modelo de referencia de la OSI.

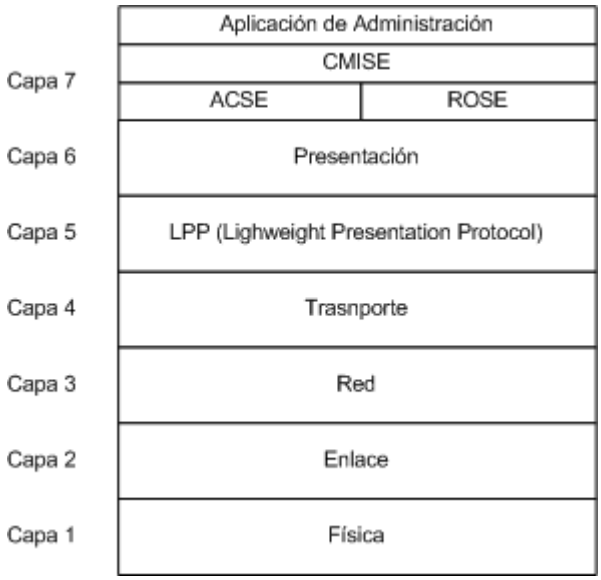


Figura 32: El protocolo CMOT en el modelo de referencia de la OSI (Ref 1)

El protocolo de aplicación usado por el CMIS no cambia en la implementación del CMOT. El CMOT se basa en los protocolos CMISE, ACSE y ROSE. Pero, en vez de esperar por la implementación de un protocolo de la capa de presentación de la ISO, CMOT requiere el uso de otro protocolo en la misma capa del modelo de referencia de la OSI, el Lightweight Presentation Protocol (LPP), el cual es definido en el documento RFC 1085. Este protocolo provee la interfaz para cualquiera de los dos protocolos de la capa de transporte más comunes hoy en día, UDP y TCP, los cuales usan IP como capa de red.

Un sistema que cumple con la especificación CMOT tiene que tener la funcionalidad de establecer una de las asociaciones reconocidas: event, event/monitor, monitor/control, o full manager/agent con un sistema abierto.

Un problema potencial en el uso de CMOT es que muchos de los fabricantes de dispositivos de redes no quieren perder tiempo implementando soluciones intermedias como CMOT. De hecho, aunque la definición de CMOT existe, no se ha hecho un trabajo práctico significativo en el protocolo en algún tiempo.

Capítulo III

Arquitectura de los Sistemas para la Administración de Redes

3.1 Introducción

Este capítulo examina los sistemas para la administración de redes en detalle. Primero presenta los dos elementos principales: la plataforma para la administración de redes y las aplicaciones adicionales que la acompañan. Y luego, presenta las posibles arquitecturas de las plataformas, que pueden ser: centralizada, jerárquica y distribuida.

El objetivo de este capítulo es el de establecer los criterios para elegir la plataforma y aplicaciones apropiadas para un ambiente de red en particular.

3.2 Plataformas para la administración de redes

El manejo de la red históricamente abarcaba muchos sistemas, que separadamente manejaban un conjunto específico de componentes. Un caso típico era la utilización de diversos sistemas para controlar hubs, ruteadores, bridges, y otro tipo de componentes de red. Sin embargo en la actualidad, debido a las restricciones de dinero, espacio físico y

experiencia técnica, es mucho mejor que la red pueda ser controlada por un solo sistema que sea capaz de mostrar las conexiones a través de un mapa de red, y administrar sus componentes de manera integrada.

La plataforma de administración de la red es un software que provee las funcionalidades básicas para el manejo de los diferentes componentes. El objetivo es proveer una funcionalidad genérica para el manejo de una variedad de dispositivos. Estas funcionalidades básicas son:

- Interfaz Gráfica
- Mapa de Red
- Método estándar de interrogación de dispositivos
- Base de Datos
- Sistema de menús flexible
- Registro de eventos

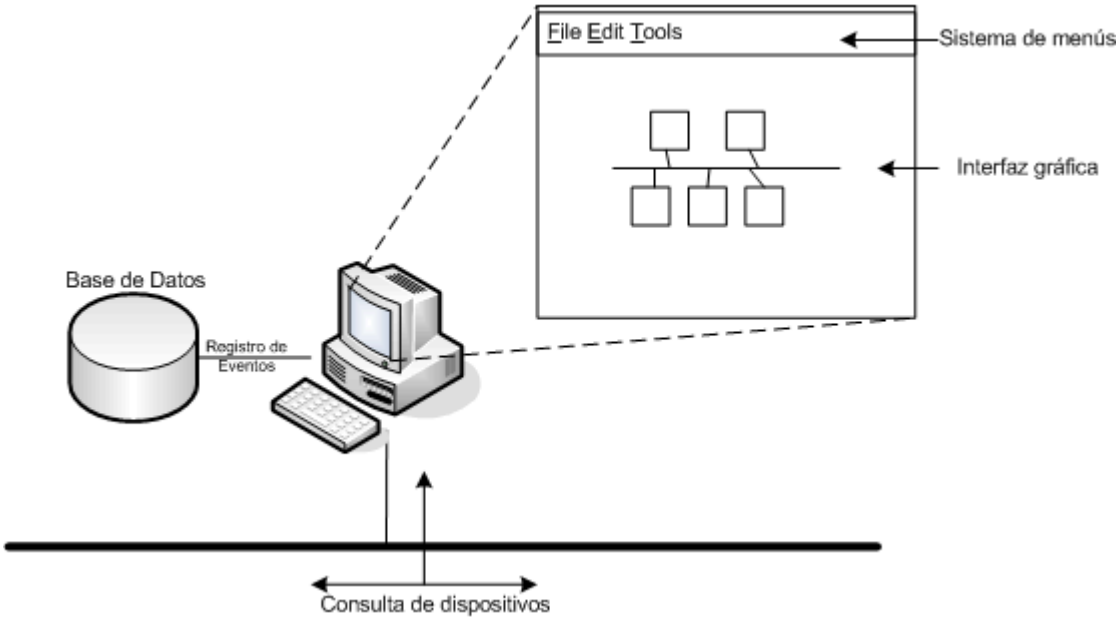


Figura 33: Componentes Básicos de una Plataforma de Manejo de Redes (Ref 1)

La interfaz gráfica es útil para dar un acceso rápido a los usuarios de las funciones de la plataforma. Además, la interfaz gráfica debe ser consistente siguiendo un estándar como los de: Microsoft Windows, Sun Microsystems Openlook, etc. El hecho de usar una interfaz gráfica estándar hace que el sistema sea fácil de usar y manipular.

El mapa es útil para casi todas las áreas del manejo de redes. Las herramientas de manejo de fallas pueden ayudar a aislar la causa de la falla usando colores en el mapa. Las herramientas de configuración pueden mostrar las configuraciones físicas y lógicas de la red por medio

de un gráfico. Las herramientas para el manejo del rendimiento pueden gráficamente mostrar el rendimiento actual de los dispositivos y enlaces por medio de colores o gráficos diferentes. Un beneficio adicional sería que la plataforma provea un mecanismo automático de descubrir los dispositivos de la red (llamado autodiscovery) y luego dibujarlos en la red gráficamente (llamado automapping).

Para que la herramienta de administración puede consultar diferentes dispositivos de diferentes fabricantes, es necesario un método de consulta común. Este método de consulta común puede ser el soporte a un protocolo estándar de administración, como los explicados en el capítulo anterior. La información consultada, debe ser almacenada en el sistema de administración, ya sea usando un sistema propietario del vendedor de la herramienta, o usando una base de datos relacional.

La base de datos es útil para muchas de las tareas de administración. Las aplicaciones pueden usar la base de datos para almacenar su información. Además se pueden establecer relaciones entre los datos lo cual ayuda en el diagnóstico y mantenimiento de la red. Muchas bases de datos permiten a los usuarios generar reportes propios y automatizan la realización de respaldos periódicos.

El sistema de menús flexible es necesario, porque además de facilitar el uso de la herramienta, permiten que diferentes fabricantes añadan

opciones adicionales específicas para los dispositivos que fabrican, y los usuarios pueden incluir opciones que activen programas que han creado.

La última característica esencial de una plataforma de administración de redes es el registro de eventos. Este registro mantiene los eventos de la red de una manera cronológica en un formato legible. La plataforma escribe información a este registro sobre cualquier evento que ocurriese en la red, y puede generar sus propios eventos. También los dispositivos pueden enviar mensajes no solicitados por la plataforma que pueden ser interpretados como eventos de la red. Sin importar de donde provienen los eventos, estos mantienen al administrador de redes al tanto de los sucesos que ocurren en la red.

Adicional, a las funcionalidades básicas, la plataforma para el manejo de red debe tener las siguientes características:

- Herramientas gráficas para visualización de datos
- Interfaz de programación de aplicaciones (API)
- Seguridad

La plataforma debe proveer la habilidad de producir gráficos, como líneas, barras, etc., de la información que administra el sistema. También es útil , la posibilidad de insertar gráficos en los reportes, debido a que en

gran medida los usuarios prefieren ver la información representada en gráficos en vez de texto ya que es más fácil visualizar información útil. Gráficos del tráfico actual de la red pueden ayudar en el manejo de fallas y rendimiento, y gráficos de la información histórica ayudan a aislar las tendencias de la red.

El API es una librería de procedimientos y funciones de programación que permiten el acceso a la información mantenida dentro de la plataforma de administración de redes. Solo a través del API los programas externos usan o se integran al mapa de la red, se pueden integrar en el sistema de menú, pueden almacenar y recuperar información de la base de datos, enviar mensajes al registro de eventos, etc. Además el API es importante por dos razones: permite la integración de las aplicaciones de terceros, y ayuda a los administradores de la red escribir programas acordes con las redes que manejan. Sin el API, la plataforma para el manejo de la red sería esencialmente una “caja negra”. Es importante que este API sea estándar entre diferentes plataformas.

Otra característica importante es que la plataforma de administración debe integrar un mecanismo de seguridad para proteger la información importante de la red y el acceso a la configuración de los componentes, además de proveer facilidades de implementación de políticas de seguridad.

Las funciones básicas permiten al administrador cumplir con todas las áreas funcionales del manejo de la red. Por ejemplo, el administrador de red puede trabajar en el rendimiento y el manejo de fallas consultando todos los dispositivos de la red a través de la plataforma y luego usando esta información para producir un gráfico de la utilización del enlace serial para todos los dispositivos de la red que lo posean. Los pasos que antiguamente (antes del uso de estándares como SNMP) se habrían realizado para llevar a cabo esta tarea serían los siguientes:

1. Decidir cuál es la información necesaria de cada componente de red (usualmente bytes enviados o recibidos por interfaz)
2. Recolectar la información apropiada usando la plataforma de administración
3. Poner los datos en una hoja de cálculo o un paquete similar para producir los gráficos deseados

De todos ellos, el primer paso era el más difícil de realizar debido a que la plataforma para la administración de red usaba un método estándar propietario para consultar cada componente de red, mientras que cada componente podía tener la información almacenada de una manera particularizada para su marca. En los ambientes actuales, las piezas de información están almacenadas en un formato estándar de manejo de

información (MIB) por lo que la plataforma de administración obtiene la información útil independientemente de la marca y modelo de los dispositivos.

En la actualidad existen muchas plataformas para el manejo de redes, como HP OpenView, IBM Tivoli, CiscoWorks, etc. Cada una de ellas provee las características básicas, así como también funcionalidades adicionales que las diferencian una de la otra. Sin embargo, lo más importante de una plataforma de administración de red es el proveer una funcionalidad genérica para todo el manejo de la red.

3.3 Arquitecturas para la Administración de Redes

La plataforma para la administración de redes generalmente utiliza varias arquitecturas para proveer mayor flexibilidad en el manejo de los dispositivos y las redes que estos conforman. Las arquitecturas más comunes para la administración de redes son:

- Centralizada
- Jerárquica
- Distribuida

No hay una arquitectura que sea la mejor, cada tipo tiene características específicas que trabajan bien en ciertos ambientes. Es una buena regla el elegir la arquitectura para la administración de redes que se asemeje más a la estructura organizacional de la compañía.

3.3.1 Arquitectura Centralizada

Una arquitectura centralizada tiene la plataforma para la administración de la red en un solo computador, el cual es el responsable por todo el manejo de la red, como lo muestra la figura 34. Este sistema usa una base de datos centralizada. Para redundancia, el sistema es respaldado a otro sistema en períodos regulares. Además el sistema central, que es el punto focal para el manejo de la red, puede permitir el acceso y enviar eventos a otras consolas de la red. La computadora de administración en una arquitectura centralizada es la encargada de manejar:

- Todas las alertas y eventos de la red
- Toda la información de la red
- Todas las aplicaciones de administración

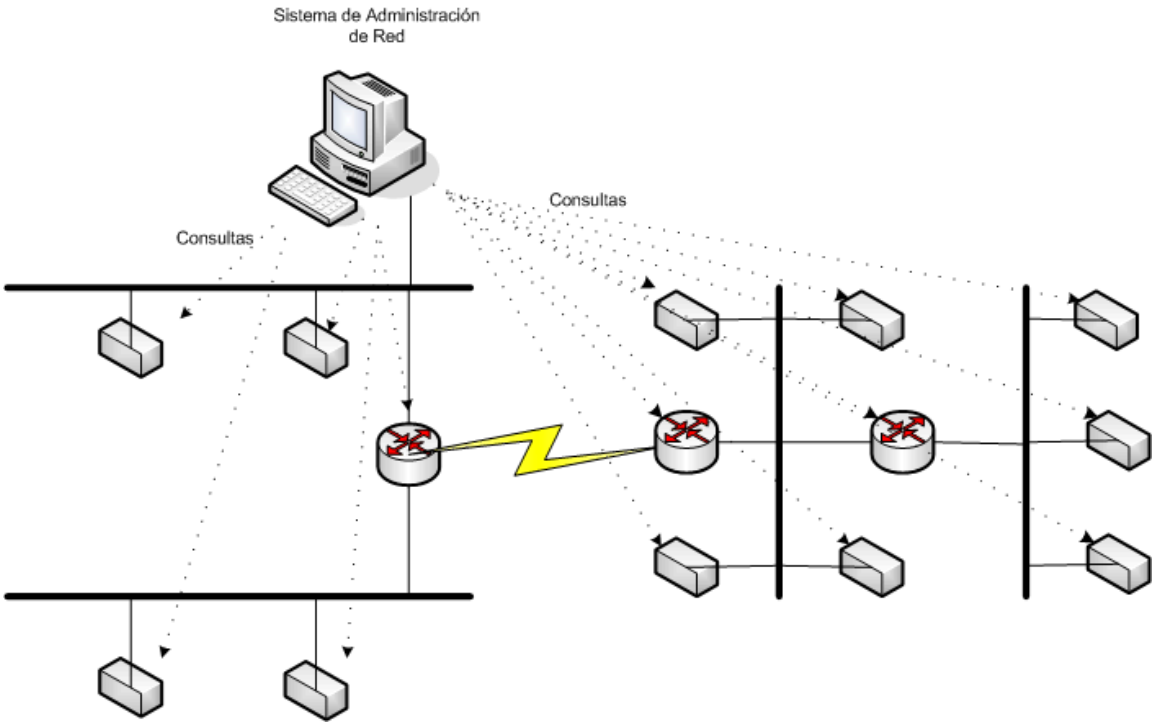


Figura 34: Arquitectura Centralizada (Ref 1)

Usando un esquema centralizado, el administrador de red tiene un solo repositorio del cual obtener todas las alertas y eventos de la red, lo cual es útil para la determinación de problemas y la correlación de los mismos. El tener un solo sistema para acceder a todas las aplicaciones e información de la red es conveniente, de fácil acceso y seguro para el administrador. Por el hecho de tener una sola localidad de administración, la seguridad es más fácil de mantener poniendo el sistema en un área segura y restringida, que además puede ser configurado para permitir el acceso solamente a ciertos usuarios.

Sin embargo el tener todas las funciones de administración dependiendo de un solo sistema, presenta las desventajas de no exhibir redundancia o ser tolerante a fallas. Para contrarrestar esto, respaldos totales deben ser mantenidos idealmente en otra localización física. Otra desventaja es que a medida que nuevos elementos son añadidos a la red, puede ser difícil y caro mejorar el sistema (hardware y software) para que maneje la carga extra. Una última desventaja de esta arquitectura es la de tener que consultar a todos los dispositivos desde una sola localidad. Esto pone carga adicional a todos los enlaces conectados al sistema, ya que si por algún motivo la conexión del sistema de administración a la red se pierde o ocurre alguna falla, todas las tareas de administración también se perderán. Localizar la estación en un punto central en la red podría ayudar a este problema, pero la localización ideal para la plataforma puede no ser un lugar idóneo para el administrador.

3.3.2 Arquitectura Jerárquica

Una arquitectura jerárquica usa múltiples sistemas, en el cual uno actúa como servidor central y los otros funcionan como clientes, según se muestra la figura 35. Algunas de las funciones de administración residen en el servidor y otras en componentes

distribuidos. Por ejemplo, el administrador puede configurar componentes separados para monitorear diferentes porciones de la red y hacer que las alertas en cambio sean transmitidas solamente al administrador principal.

A pesar de su naturaleza, la arquitectura jerárquica usa tecnología cliente/servidor para mantener la base de datos en el administrador central. Los componentes distribuidos de administración no tienen base de datos separadas y usan el administrador central para interactuar con la base de datos. Debido a la importancia del sistema central en la jerarquía, éste requiere de respaldos periódicos para redundancia.

La arquitectura jerárquica para la plataforma de administración de la red tiene las siguientes características principales:

- No depende de un solo sistema para todas las funciones
- Distribución de las tareas de administración de la red
- Monitoreo distribuido a través de la red
- Almacenamiento central de la información

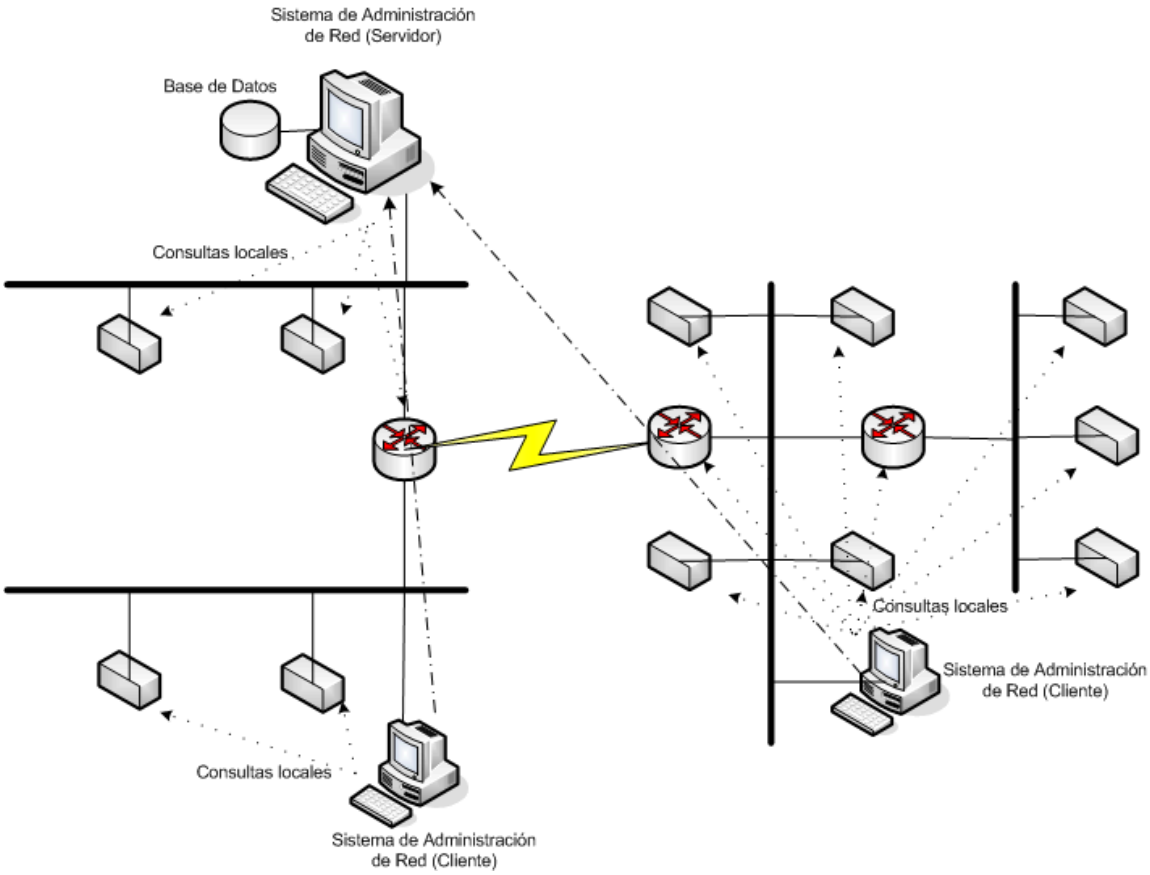


Figura 35: Arquitectura Jerárquica (Ref 1)

El esquema jerárquico ayuda a solucionar algunos de los problemas de la arquitectura centralizada distribuyendo las tareas de administración entre el sistema central y los clientes. Los administradores pueden distribuir el monitoreo en los componentes, aliviando la carga adicional que representa que el servidor central monitoree todos los dispositivos. Además los componentes

distribuidos no necesitan implementar toda la funcionalidad del servidor central.

Una desventaja de la arquitectura jerárquica es que el uso de múltiples sistemas cliente para manejar la red, lo que dificulta la configuración para la recolección de la información para el administrador. Otra desventaja de este esquema es de que la lista de los dispositivos manejados por cada cliente necesita ser lógicamente predeterminado y configurado manualmente. A menos que esto se realice con cuidado, se puede tener la posibilidad de que tanto el sistema central como varios componentes monitoreen el mismo dispositivo lo que puede ocasionar que se consuma hasta el doble de ancho de banda de la red de lo que se hubiera utilizado para propósitos de administración.

3.3.3 Arquitectura Distribuida

La arquitectura distribuida combina los esquemas centralizados y distribuidos, como lo muestra la figura 36, en vez de tener una plataforma centralizada o una plataforma jerárquica de plataformas centrales/clientes, el esquema distribuido usa múltiples plataformas en la cual una actúa como líder de un conjunto de ellas. Cada plataforma

tiene una base de datos completa de los dispositivos de la red; esto permite ejecutar varias tareas y reportar los resultados al sistema central.

Debido a que la arquitectura distribuida combina el esquema centralizado y jerárquico, además de tener las ventajas de ambos, tiene las siguientes:

- Localización única para toda la información de la red, alertas y eventos
- Localización única para acceder todas las aplicaciones de administración
- No depende de un solo sistema
- Distribución de las tareas de administración
- Distribución del monitoreo de la red a través de la misma

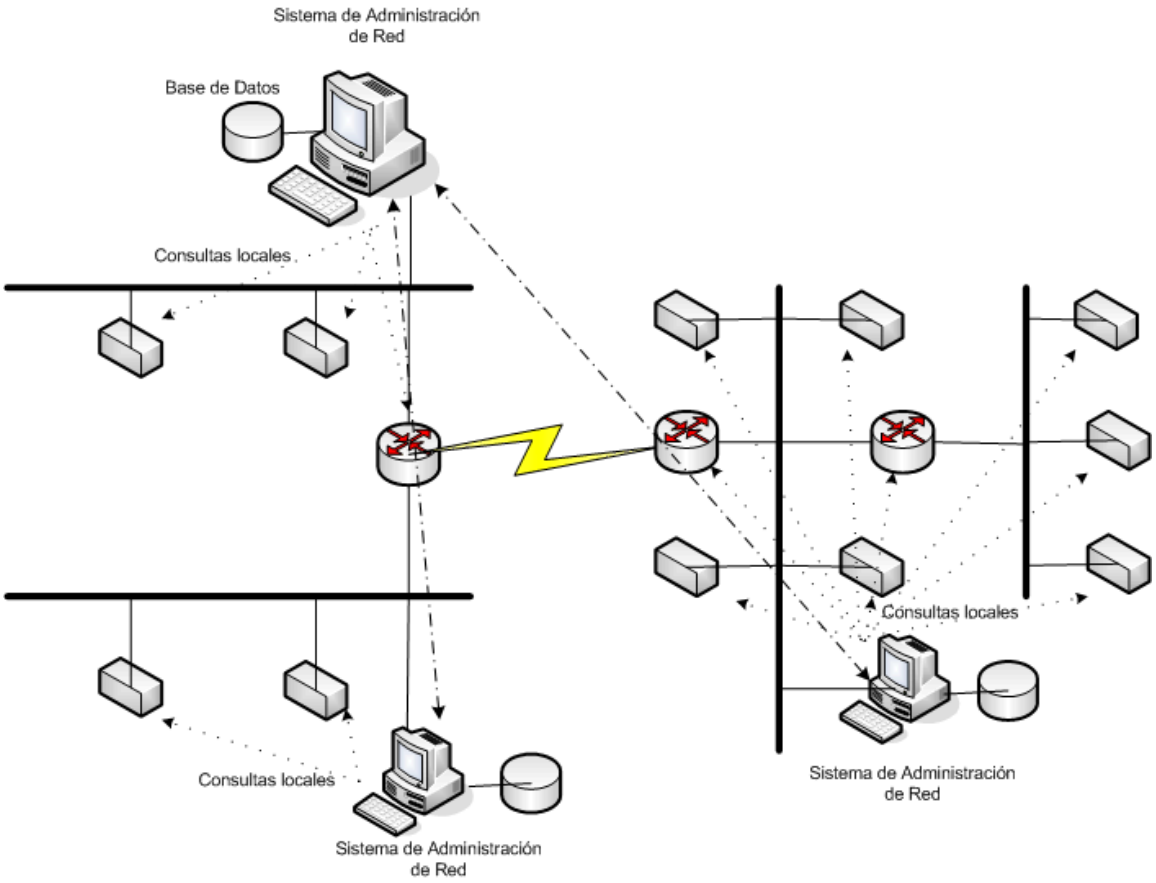


Figura 36: Arquitectura Distribuida (Ref 1)

La tecnología de replicación de bases de datos es muy útil para la implementación de esta arquitectura. El servidor de replicación mantiene múltiples copias de la base en diferentes sistemas completamente sincronizadas lo que no es un trabajo fácil. Esta tecnología es muy compleja, de hecho, la sobrecarga asociada con la sincronización consume muchos más recursos que la tecnología cliente/servidor.

3.4 Aplicaciones para la Administración de Redes

Las plataformas para el manejo de la red proveen funcionalidades generales para todos los dispositivos manejados. En cambio el diseño de aplicaciones para la administración de la red es ayudar al administrador a manejar un conjunto específico de dispositivos o servicios. La relación entre la plataforma para el manejo de la red y las aplicaciones se muestra en la figura 37. Muchas aplicaciones son desarrolladas por los vendedores de dispositivos para ayudar a sus clientes el manejar sus dispositivos en particular.

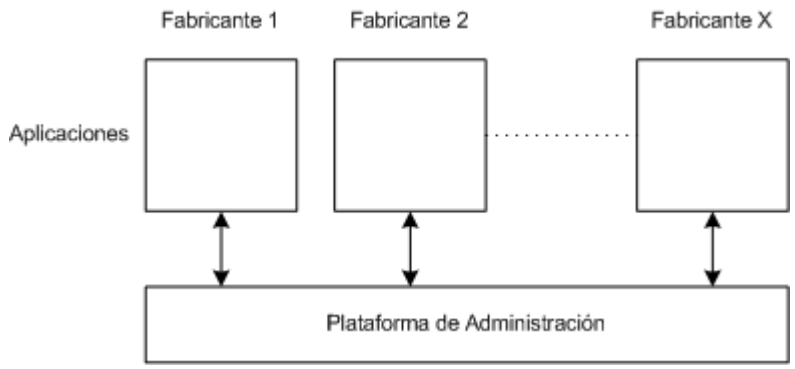


Figura 37: Relación entre la Plataforma de Administración de la Red y las Aplicaciones (Ref 1)

Por ejemplo, un fabricante de un dispositivo de red, como modems, hubs o bridges, puede implementar un conjunto de aplicaciones que trabajen junto con la plataforma de administración así provee una serie de

herramientas para el administrador. Usando esta estrategia, un administrador necesitaría comprar solo las herramientas necesarias para manejar un conjunto específico de dispositivos. Si un administrador trata manejar hubs y servidores de archivos, por ejemplo, solo se necesitaría comprar las aplicaciones que trabajen junto con la plataforma, así se tendría un sistema de administración de red que provea toda la funcionalidad general de la plataforma y aplicaciones para manejar características específicas de los hubs y de los servidores de archivos usando el mismo mapa, teniendo la misma interface, accesible a través del mismo sistema de menús e interrogando el mismo sistema de bases de datos.

Las aplicaciones de administración de red tienen los siguientes objetivos;

- Manejo efectivo de un conjunto específico de dispositivos
- Funcionalidad adicional a la plataforma
- Integración con la plataforma a través del API y el sistema de menús
- Independiente de la plataforma

Las aplicaciones tratan de manejar efectivamente un conjunto específico de dispositivos, por ejemplo una aplicación que maneje hubs, permitiría al usuario configurar las características del hub, habilitar puertos o

monitorear cantidad de errores y velocidad de transmisión; además ayudaría en las tareas de manejo de la configuración y rendimiento del dispositivo.

Las aplicaciones no deben incorporar características que ya posea la plataforma, el resultado sería que algunas tareas se puedan realizar de varias formas con lo cual produce una confusión al usuario, también es una pérdida de esfuerzo de desarrollo de funciones que ya incluye la plataforma. La excepción sería cuando la plataforma no provea una característica que la aplicación requiera.

Uno de los objetivos de la aplicación es la integración con la plataforma, esto permite al usuario ver tanto a las aplicaciones y a la plataforma como un sistema uniforme de administración de redes. El API provee la interface de comunicación con la plataforma y el sistema de menús permite la invocación de las aplicaciones a través del mismo menú de la plataforma. En muchas plataformas, la integración de una aplicación con el sistema de menús solo requiere la edición de un archivo de texto.

Una aplicación que solo está disponible en una plataforma fuerza al administrador a usar esa plataforma para las tareas de administración de la red, lo que no siempre es conveniente por diferentes motivos como que la plataforma no posea las características o soporte necesario para las demás aplicaciones que se necesitan.

3.5 Consideraciones para la elección del sistema administrador de la red

El sistema administrador está compuesto de dos componentes básicos: la plataforma y las aplicaciones adicionales. Eligiendo bien los componentes, se puede obtener un sistema que realmente ayude al administrador realizar las funciones de administración de la red. Los pasos para la elección de un sistema administrador son:

1. Realizar un inventario de los dispositivos
2. Priorizar las áreas funcionales del administrador de la red
3. Examinar las aplicaciones administradoras de la red
4. Elegir la plataforma para la administración de la red

Primero hay que identificar cada dispositivo en la red, como estaciones, computadoras personales, controladores, gateways, routers, switches, bridges, hubs, impresoras y modems; averiguar si cada uno de los dispositivos pueden ser manejados por cualquier protocolo de administración, ya sea estándar o propietario. Si un dispositivo no puede ser manejado por los protocolos estándares, hay software que pueden hacer la traslación entre el protocolo propietario y el estándar. Teniendo

la lista, es necesario priorizar los dispositivos críticos, como servidores, routers, etc.

Como la administración comprende cinco áreas funcionales, hay que priorizar cual es la más importante para la organización. En muchos casos el área más importante es el manejo de fallas. Este paso es crucial porque determina la elección de las aplicaciones para los dispositivos.

El tercer paso es encontrar las aplicaciones de administración de la red que ayuden a administrar las áreas claves de la red. Usar aplicaciones diseñadas para manejar los dispositivos importantes permite concentrar todos los recursos en la administración en vez de crear aplicaciones.

Por último hay que seleccionar la plataforma para la administración de la red. Lo ideal sería que las aplicaciones seleccionadas trabajen en una plataforma común, pero en la realidad esto no es así, con las aplicaciones, se tienen una serie de plataformas para elegir, entonces se debe seleccionar la plataforma cuya arquitectura se acerque más a la forma en como la organización tiene planeado manejar la red. Por ejemplo, si la organización planea administrar centralmente la red, una plataforma con arquitectura centralizada o jerárquica es lo recomendable. En cambio si se requiere de administradores de respaldo diseminados en la red, una plataforma distribuida es lo correcto.

Otro criterio cuando se selecciona una plataforma es en que tipo de computadora el sistema puede funcionar. Por ejemplo, si la plataforma ideal para la red corre en un tipo de computadora que la organización no tiene o no sabe nada acerca de ella, una plataforma menos ideal pero que corra en la computadora que la organización conoce es lo mejor.

En muchos casos las organizaciones realizan estos pasos en desorden, seleccionando primero la plataforma, luego descubren que no hay aplicaciones que ejecuten en ella para administrar los dispositivos, por lo que la secuencia de los pasos antes descritos es muy importante.

Capítulo IV

Situación y necesidades de las redes de la ESPOL

4.1 Introducción

Las redes de computadoras en la actualidad son de mucha importancia porque nos permiten compartir recursos de almacenamiento, impresión, información; y es evidente que su integración a la Internet las hace imprescindibles para cualquier organización que quiere ser competitiva en la actualidad.

La ESPOL, como organización líder en educación superior, ha implementado un backbone que posibilita la interconexión entre sus distintas unidades, permitiendo la integración de los servicios computacionales. Entre los servicios más usados que presta esta red están: acceso a Internet, acceso a las aplicaciones de la universidad, correo electrónico, etc. Para entender mejor donde y como se ofrecen los servicios del backbone es necesario describir los recursos disponibles y las necesidades actuales de los dos campus de la ESPOL donde se ha implementado el backbone: Gustavo Galindo y Las Peñas.

Campus Gustavo Galindo

Es el principal campus de la universidad, en el cual se encuentran la mayoría de las carreras que ofrece la ESPOL. Como este campus ocupa una gran extensión, las diferentes unidades se encuentran dispersas dentro de él, haciendo que la implementación de las redes sea mucho más costosa debido a la necesidad de usar fibra óptica para la interconexión por las grandes distancias que hay entre las unidades. Dentro de este campus, se encuentran dos grandes localidades denominadas: Ingenierías y Tecnologías. En las Ingenierías, la mayoría de los edificios de las unidades ya se encuentran interconectadas (la forma de interconexión se explicará más adelante), mientras que en las Tecnologías se encuentran interconectadas solamente algunos de sus edificios (la forma de interconexión se explicará más adelante). Entre las Tecnologías y las Ingenierías hay un enlace de fibra que permite la interconexión de estas dos grandes localidades dentro del campus Gustavo Galindo. En la figura 38 se muestra el campus Gustavo Galindo y sus dos localidades: Ingenierías y Tecnologías. Como se puede apreciar la localidad de las Ingenierías es más extensa y más dispersa que la de las Tecnologías.

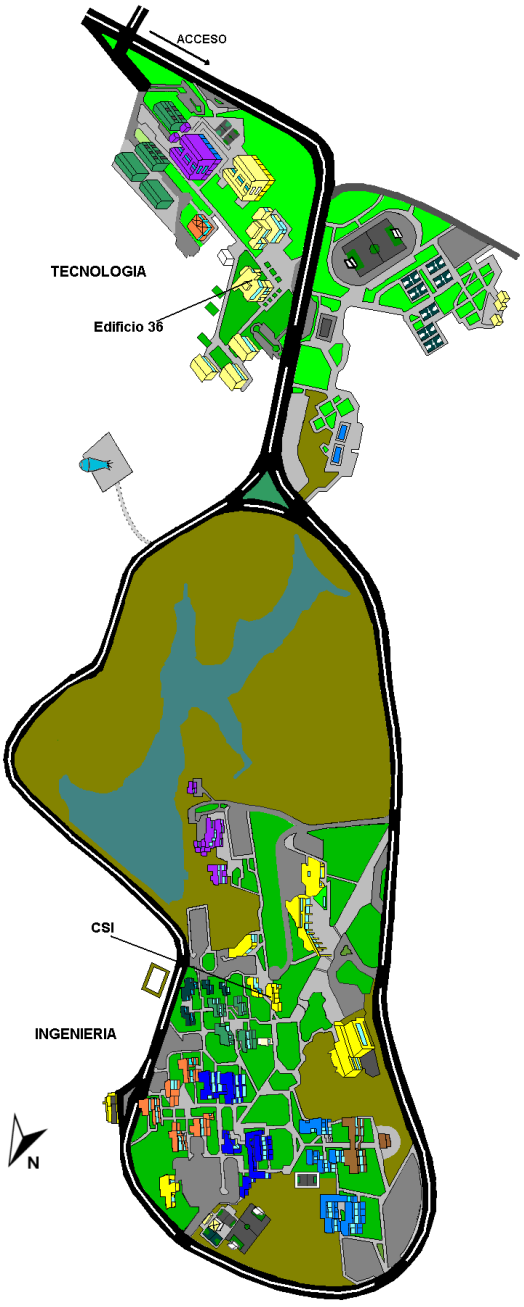


Figura 38: Campus Gustavo Galindo

Campus Las Peñas

En este campus se encuentran la mayoría de las maestrías y algunas carreras de tecnología. Aquí, a diferencia del campus Gustavo Galindo, no se ocupa una gran extensión de terreno, lo que hace mucho más simple la implementación de las redes pudiendo usarse cable de cobre (y no fibra óptica) para la interconexión de las unidades. La figura 39 muestra el diagrama del campus Las Peñas.

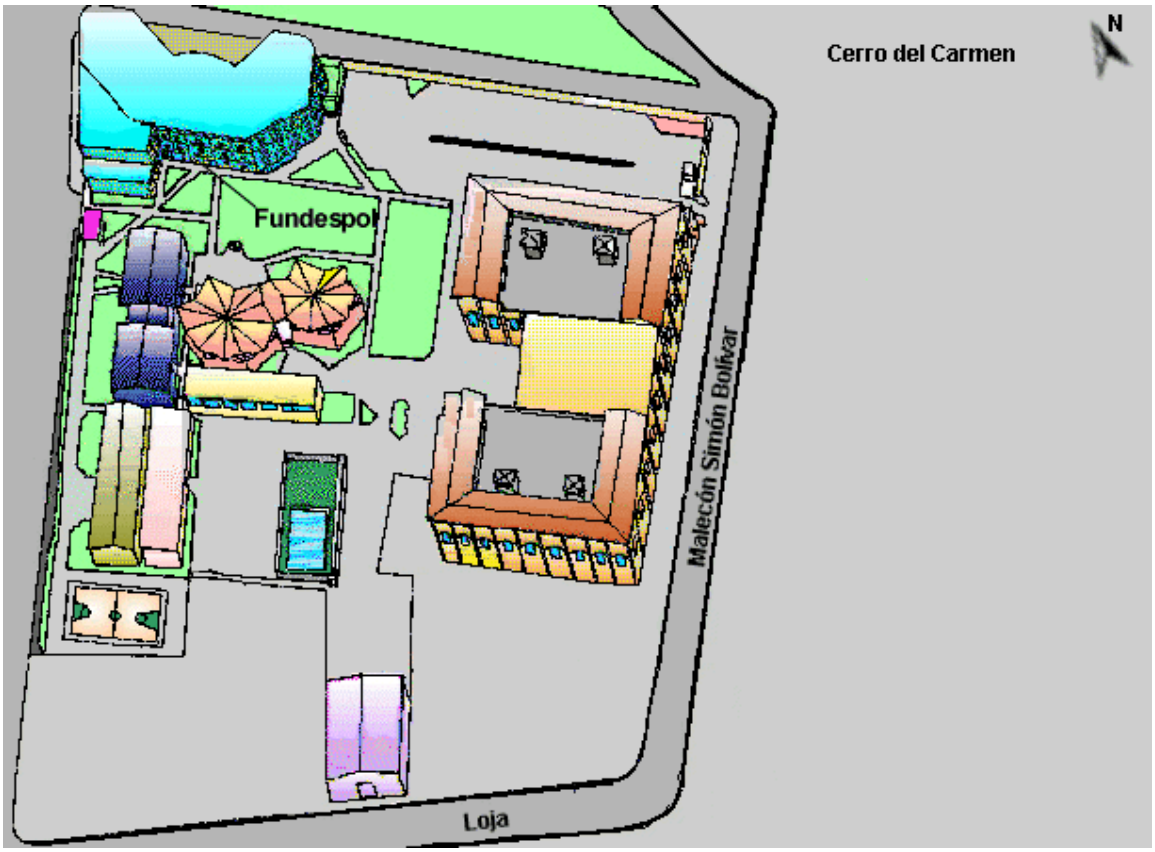


Figura 39: Campus Las Peñas

La forma en la que están conectadas las redes de ambos campus se la conoce como topología en estrella, en la cual todas las redes se conectan a un lugar específico, que para el caso del campus Gustavo Galindo es CSI (Centro de Servicios Informáticos) para el sector de Ingenierías, y el Edificio 36 (donde se encuentra la central telefónica) para el sector de Tecnologías. Para el Campus Las Peñas el centro de la estrella es el edificio de Fundespol. Estos lugares son los sitios en los que se encuentran la mayoría de los equipos concentradores y de interconexión de redes, por lo que cuentan con una infraestructura adecuada para el alojamiento de los componentes. Esta infraestructura consiste de unidades de respaldo de energía (UPS), unidades de acondicionadores de aire y espacio físico suficiente. Además estas instalaciones cuentan con el apoyo de personal técnico que administra estos equipos.

CSI es la unidad encargada del planeamiento y revisión de nuevas redes ha implementar en la ESPOL y de vigilar el cumplimiento de las normas técnicas necesarias para la conexión de estas al backbone. Las unidades que deseen conectarse a la red principal de la ESPOL deben solicitar a CSI que realice los estudios necesarios, técnicos y económicos. Para mayor información al respecto, las unidades deben solicitar a CSI los estándares necesarios e informarse de los beneficios y responsabilidades que se deriven de la conexión al backbone.

4.2 Descripción general de las redes

Las redes de los campus Gustavo Galindo y Las Peñas se encuentran interconectadas a través de un enlace provisto por un proveedor externo que además provee el servicio de Internet. Esta interconexión se detalla en la Figura 40 en la que además se muestran tres computadoras para usarlas de referencia en la explicación:

- A, que se encuentra en el Campus Gustavo Galindo
- B, que se encuentra en el Campus Las Peñas, y
- C, que se encuentra en Internet.

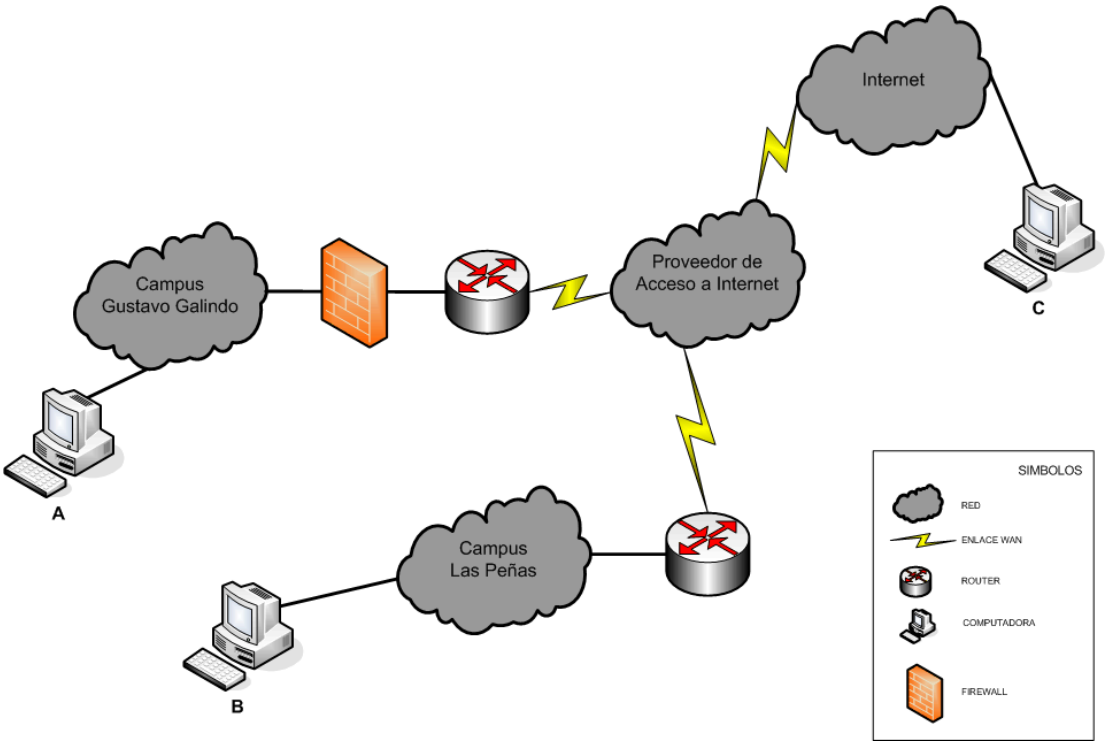


Figura 40: Diagrama Lógico de conexión entre Gustavo Galindo, Peñas e Internet

Si la *computadora A* desea acceder a la computadora B, la computadora A debe estar conectada a una de las redes del campus Gustavo Galindo utilizando el protocolo TCP/IP siguiendo los parámetros de acuerdo a al red a la cual está conectada. Estas mismas condiciones se aplican para la *computadora B que se encuentra conectada al Campus Las Peñas*. El requerimiento de comunicación va desde la computadora A hacia el ruteador más cercano, este lo redirige hacia la red principal de distribución, para luego redirigirla hacia el ruteador que se interconecta

con el proveedor. El proveedor luego redirige la comunicación hacia el ruteador del campus Las Peñas, el cual a su vez, envía la comunicación hacia el ruteador que se encuentra en la misma red que la *computadora B*.

Lo explicado anteriormente también se aplica para el caso de la comunicación de la *computadora A*, hacia la *computadora C*, que está en Internet. La diferencia es que el proveedor redirecciona la comunicación hacia su enlace de Internet para alcanzar a la *computadora C*.

4.2.1 Descripción general de las redes del Campus Gustavo Galindo

El detalle de la constitución interna de la red del Campus Gustavo Galindo se muestra en la figura 41, en la cual podemos apreciar que hay un segmento de red principal que permite la comunicación de todas las redes instaladas en la Espol. Este segmento de comunicación es el que se conoce como la red de distribución o Backbone, el cual utiliza tecnología gigabit ethernet. Esta red principal tiene una gran concentración de equipos de comunicaciones, y a ella se conecta el equipo que nos proporciona la comunicación con el proveedor de acceso a Internet. Además de este segmento de red, existe otro segmento en el cual se encuentran los servidores

principales de Internet de la Espol (servidor de correo electrónico, servidor de nombres, servidor web, servidor proxy).

Como se detalla en la figura 41, cada unidad, facultad o instituto, tiene un ruteador que le provee la comunicación hacia el backbone que la redirige según sea el destino, hacia los servidores locales o hacia Internet.

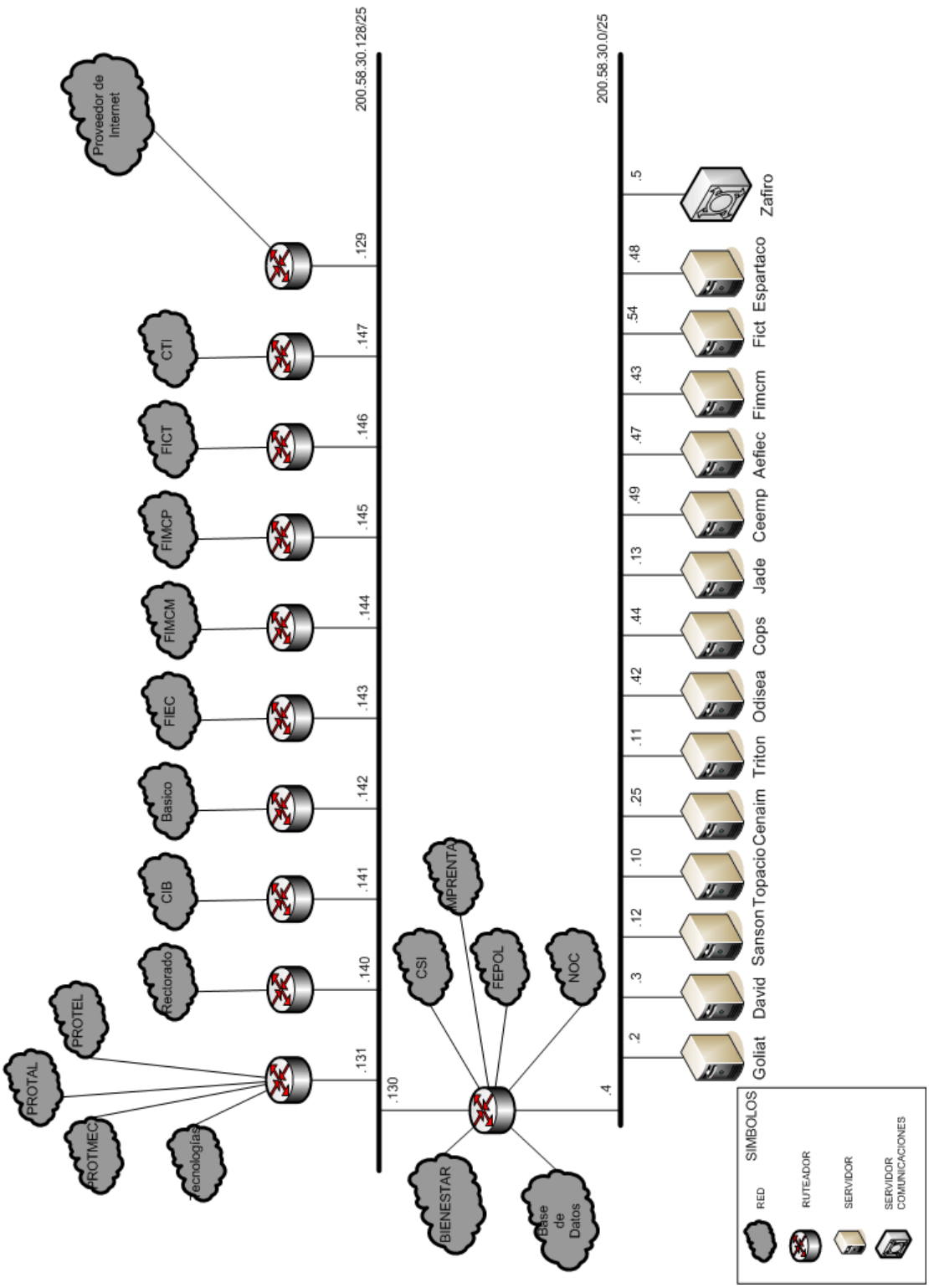


Figura 41: Diagrama del Backbone del Campus Gustavo Galindo

Las figuras anteriores muestran en forma general como están constituidas las dos redes principales del Campus Gustavo Galindo, ahora vamos a explicar como se establece la comunicación tomando los siguientes casos:

1. Acceso a la base de Datos de la Espol, desde una máquina que se encuentra en un instituto o facultad

2. Acceso a Internet, desde una máquina que se encuentra en un instituto o facultad

3. Acceso al correo del dominio espol.edu.ec, desde una máquina que se encuentra en un instituto o facultad

Estos son casos generales de la forma en que los diferentes componentes de la red interactúan para establecer las comunicaciones, y basados en la figura 42 en la que se muestran los componentes de la red.

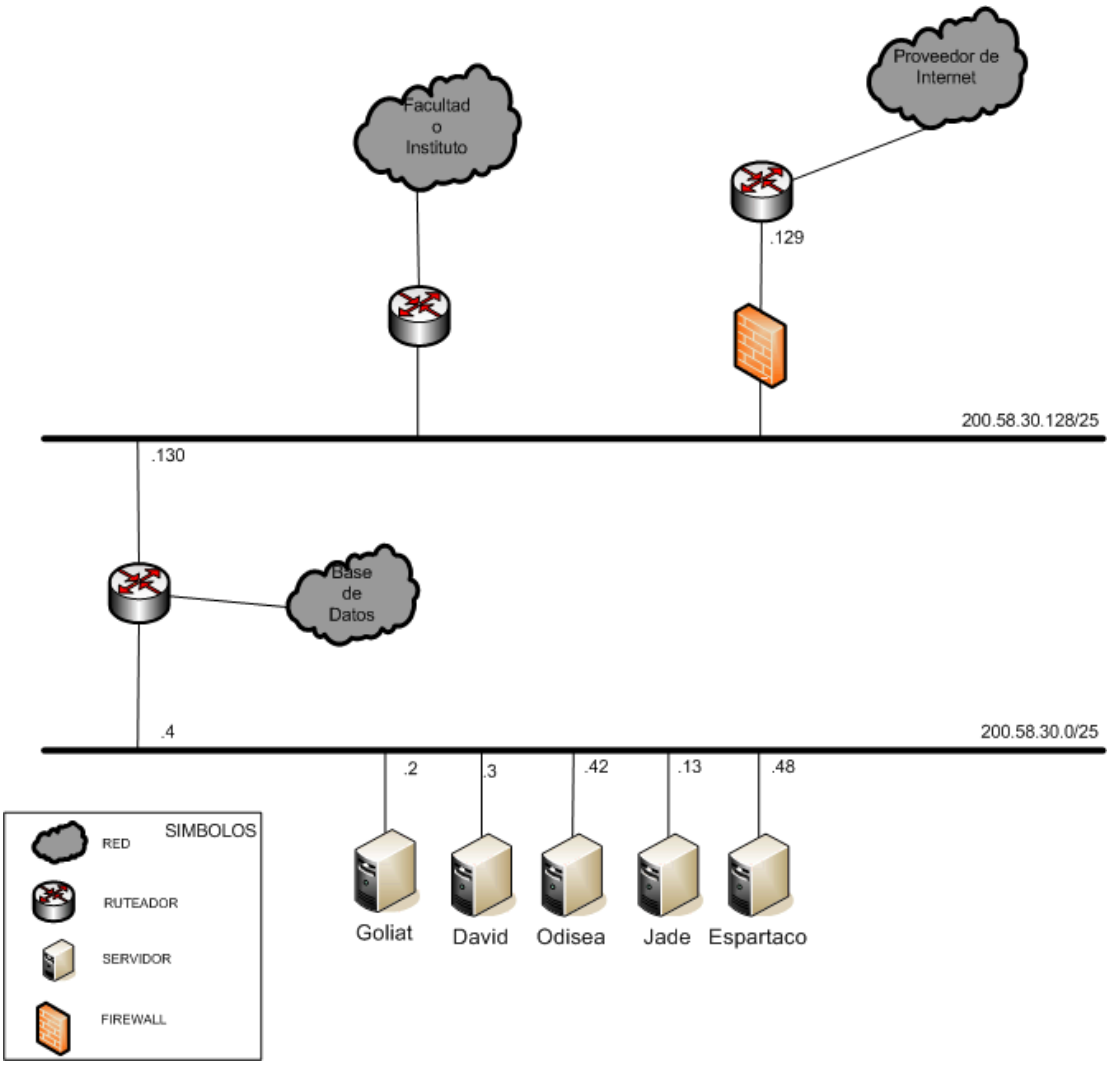


Figura 42: Interconexión de los componentes de Red en el Campus Gustavo Galindo

Desde una estación que se encuentra en un instituto o facultad, primero la comunicación pasa por el ruteador de ese instituto o facultad, luego este dispositivo, envía el requerimiento al ruteador principal (ver Figura 42) que se encuentra en el backbone. Este

ruteador luego redirecciona la comunicación a la red donde se encuentra el servidor de base de datos.

Acceso a Internet

Para este caso es necesaria la interacción de varios servicios, adicionales a los componentes de comunicación, estos servicios son:

DNS (Servidor de Nombres).- Este servicio transforma las direcciones de nombres, como hotmail.com, a direcciones lógicas IP. Este servicio corre en el Servidor GOLIAT.

Proxy.- Este servicio sirve de intermediario entre un cliente web y ftp, y el correspondiente servidor, para que la próxima vez que otro cliente requiera la misma página, o archivo, el servidor proxy le envía el requerimiento desde su almacenamiento local. Los servidores proxies son: DAVID y ESPARTACO.

Firewall.- Este servicio permite o no el paso de la información de acuerdo a las reglas configuradas. El firewall actualmente está en una configuración brigde, y el servidor es: APOLO.

Tendiendo en cuenta estos servicios, para que un cliente pueda visualizar una página web, primero el sistema tratará de saber la dirección IP de la página que se quiere visualizar, para lo cual envía el

requerimiento a GOLIAT, a través del ruteador del instituto o facultad, quien lo dirige a través del backbone, al ruteador principal, el cual luego envía el requerimiento a GOLIAT quien a su vez responde la dirección IP siguiendo el mismo camino pero de regreso.

Con la dirección IP, el cliente envía nuevamente el requerimiento hacia el ruteador del instituto y este a su vez al backbone para que finalmente sea enviado al ruteador que tiene la conexión con el proveedor de Internet. Pero antes, de que llegue a este ruteador, este requerimiento es examinado por el firewall, el cual, de acuerdo con sus reglas, deja pasar o no el requerimiento. Si el requerimiento es permitido por el firewall, entonces este lo pasa a su vez al ruteador principal quien a su vez redirecciona el requerimiento web a uno de los proxies (ya sea DAVID o ESPARTACO). Si esta página ha sido accedida anteriormente, entonces se la envía directamente al cliente, de lo contrario, se hace la petición hacia Internet. Cabe anotar que en estos proxies existen reglas de navegación, las cuales establecen que sitios son permitidos acceder.

El protocolo que se usa para la comunicación entre el ruteador y los proxies, es el Protocolo de Control de Comunicación Web (WCCP).

Acceso al correo del dominio espol.edu.ec

El correo del dominio espol.edu.ec se encuentra en el servidor GOLIAT, para la cual si una máquina desea acceder a una cuenta configurada en este servidor, el requerimiento pasa por el ruteador del instituto o facultad al cual está conectado, luego éste lo transfiere al backbone, y luego al ruteador principal, el cual envía el requerimiento al servidor GOLIAT. En este servidor corren los servicios POP3 o IMAP que se encargan de procesar el requerimiento y devolver la respuesta por el mismo camino pero de regreso.

4.2.2 Descripción general de las redes del Campus Las Peñas

En el Campus Las Peñas no existen tantas redes como las hay en el Campus Gustavo Galindo y en la figura 43 podemos apreciar que hay un segmento de red, que permite la comunicación de todas las redes instaladas en este Campus. Este segmento es el que se conoce como la red de distribución o Backbone, el cual utiliza tecnología ethernet. Esta red principal tiene una gran concentración de equipos de comunicaciones, y a ella se conectan los servidores principales de Internet del Campus Las Peñas (servidor de nombres, servidor web,

servidor proxy) y del equipo que proporciona la comunicación con el proveedor de acceso a Internet.

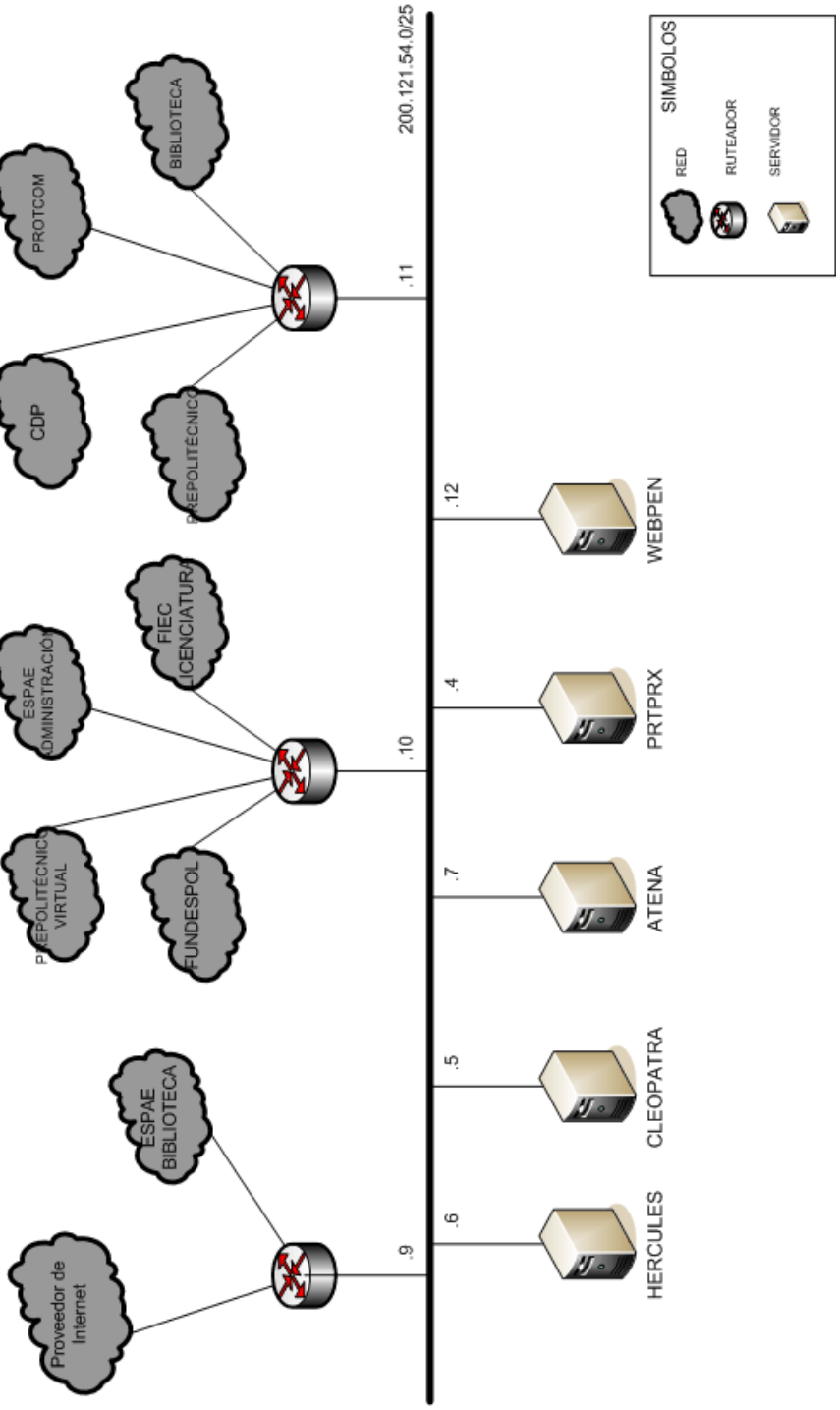


Figura 43: Diagrama de las Redes en Peñas

Como se puede apreciar en la figura 43, muchos de los departamentos comparten los ruteadores para conectarse al backbone de este campus. En este campus no existe un firewall como en el campus principal, por lo cual se han establecido ciertas restricciones a través de listas de acceso configuradas en el ruteador de comunicación con el proveedor de Internet.

Si alguna máquina requiere acceder a los datos de los servidores que se encuentran en el campus principal, la comunicación se dirige hacia el ruteador en el cual se encuentra la máquina, luego éste envía el requerimiento hacia el backbone de las Peñas, llegando al ruteador de comunicación externa. Este equipo envía el requerimiento hacia la red del proveedor del servicio, el cual a su vez la envía hacia el ruteador del campus Gustavo Galindo que se encarga de redireccionar la comunicación hacia la red del servidor de bases de datos.

El servicio de proxy, es proporcionado por 2 servidores: CLEOPATRA y PRTPRX que solo usa el Protcom. Además el servicio de DNS lo proporciona CLEOPATRA. Estos servicios son usados para la comunicación con Internet, pero en este campus, la configuración de los proxies debe hacerse en cada uno de los navegadores de las

máquinas, ya que en esta red el ruteador de comunicación con el proveedor no soporta el protocolo WCCP.

Si una persona desea acceder a una página de Internet, entonces el requerimiento se hace directamente al proxy a través del ruteador en el cual está la máquina. El proxy se encarga de resolver la dirección IP, de acuerdo con el nombre, de la página que se requiere. Luego este servidor verifica que la página no se encuentra en su almacenamiento local, con lo cual hace el requerimiento a Internet a través del ruteador.

4.2.3 Descripción de los Equipos usados

En la figura 44 se muestran los diferentes equipos que conforman el backbone del campus Gustavo Galindo. Para el backbone, existe actualmente un switch cisco 4500 con 12 puertos de fibra óptica gigabit ethernet y 48 puertos 10/100/1000Base-T en el cual se conectan todos los componentes que deben estar en el backbone, tales como el firewall que se conecta luego al ruteador de Internet y al campus Las Peñas, los servidores principales y los switches que proveen conectividad a cada unidad o instituto.

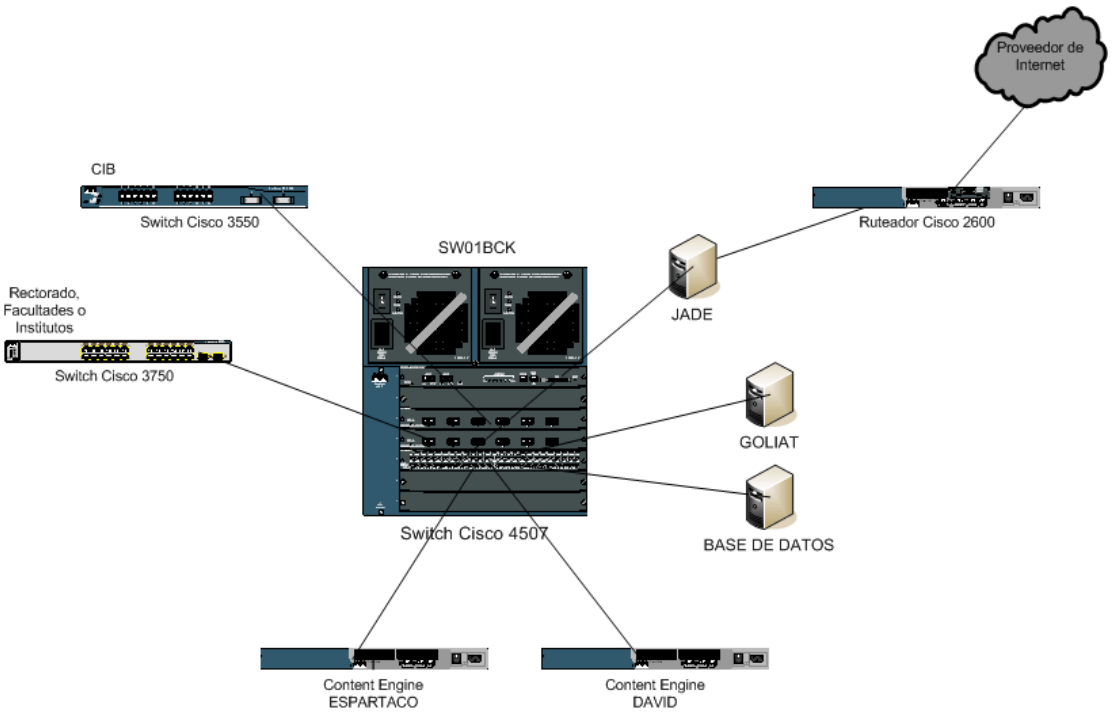


Figura 44: Equipos que componen el Backbone de la ESPOL

El switch Cisco 4500 es un equipo que puede alojar diferentes tipos de tarjetas (7 tarjetas en el modelo 4507 y 6 en el modelo 4506), actualmente tiene tarjetas de fibra para la interconexión con las diferentes unidades y una tarjeta para la conexión con los servidores principales. Hay 2 de estos equipos instalados, uno en CSI y el otro en Tecnologías. Cada uno de ellos tiene las características descritas en la siguiente tabla.

Unidad	Cant	Tipo de Tarjeta	Descripción
CSI	Cisco 4507		
	2	PWR-C45-1300AC	Fuente de poder
	1	WS-X4515	Tarjeta controladora, configura y administra el equipo. Soporte a los protocolos TCP/IP, Ruteo, e IPv6
	2	WS-X4306-GB	Tarjeta de Acceso 6 puerto para módulos Gigabit
	1	WS-X4448-GB-RJ45	Tarjeta de Acceso 48 puertos 10/100/1000BaseT
	1	WS-F4531	Tarjeta Netflow para monitoreo
Tecnologías	Cisco 4506		
	2	PWR-C45-1300AC	Fuente de poder
	1	WS-X4515	Tarjeta controladora, configura y administra el equipo. Soporte a los protocolos TCP/IP, Ruteo, e IPv6
	1	WS-X4306-GB	Tarjeta de Acceso 6 puerto para módulos Gigabit
	1	WS-X4424-GB-RJ45	Tarjeta de Acceso 24 puertos 10/100/1000BaseT

Tabla 13: Características del Switch Cisco 4500

Entre los equipos que ofrecen servicios relacionados a la Internet tenemos: GOLIAT, que es el servidor DNS y de Correo, JADE que es el firewall, Diamante, que el ruteador que nos da el acceso a Internet y es un Cisco 2600, además de los proxy que son DAVID y ESPARTACO que son Cisco Content Engine 570.

Además de los componentes descritos, cada unidad posee su propio equipo para la conectividad con el backbone, este equipo es un switch Cisco 3750.

Todos los equipos de redes descritos soportan el protocolo de administración SNMP, por lo que han sido configurados para poder ser administrados remotamente desde una estación de administración. Esta estación de administración, se encuentra en CSI, y tiene un software que permite monitorear las redes de la Espol a través del protocolo mencionado.

4.2.4 Administración y configuración de los equipos

Administrar redes TCP/IP en la actualidad para un campus del tamaño del de la ESPOL, con la cantidad de máquinas y equipos interconectados es una tarea muy compleja y requiere de políticas para llevar un control de la administración. Dentro de las políticas que se deben implementar están:

- Política de asignación de direcciones IP: Esta política tiene que ver con la asignación de direcciones IP, tanto para los equipos, como para las computadoras conectadas a las redes. Por ejemplo se puede implementar una política que establezca que las direcciones IP sean asignadas por protocolos como DHCP (Protocolo para la configuración automática de direcciones IP) o una política que establezca que las direcciones IP sean direcciones permanentes y

asignadas por el administrador. Otra política que tiene ver con la asignación de bloques de direcciones IP para diferentes tipos de equipos. Por ejemplo se podrían, fijar rangos de direcciones para equipos de red (routers, hubs, etc), para servidores, y para estaciones de trabajo.

- Política de enrutamiento: Esta política tiene que ver con las diferentes configuraciones posibles para la asignación de rutas en los equipos. Por ejemplo, estas rutas pueden ser estáticas o dinámicas. Además debe especificarse que protocolo debe usarse para la propagación de rutas y que medidas de seguridad deben establecerse para certificar que las rutas son las correctas.

- Política de protocolo de administración: Esta política tiene que ver con los diferentes protocolos existentes para la administración de los equipos de redes. Por ejemplo, si el protocolo utilizado es el SNMP, hay que establecer la comunidad y los accesos para los sistemas de administración.

- Políticas de acceso: Para esta política se deben establecer las redes y nodos que puedan tener acceso a los servicios. Estos accesos pueden establecerse con filtros de acceso en los ruteadores, pero lo ideal sería la implementación de mecanismos

más completos y sofisticados como los que provee un firewall, además de la utilización de sistemas de detecciones de intrusos que permitan establecer detalladamente los accesos desde y hacia las redes internas y determinar los intentos de acceso no autorizado a los diferentes recursos.

- Políticas de claves de equipos de redes: Esta política es similar a la política de uso de claves en general, pero además, en muchos equipos existen claves predeterminadas, las cuales deben ser borradas para una mayor seguridad.

Estas políticas, son un subconjunto de las políticas generales de administración. En la siguiente sección se evalúan los componentes involucrados en las diferentes redes y se determinan los componentes que son críticos para las redes.

4.3 Evaluación y determinación de los recursos críticos de las redes

La implementación de las redes de la ESPOL, en un principio, fue pensada para mejorar la administración académica y operativa/financiera de la institución. Por este motivo, es importante para la ESPOL el acceso a los servicios académicos y financieros manejados por los servidores de

Bases de Datos. También un recurso muy usado en la universidad es Internet, por lo que este servicio debe estar operativo siempre.

Como el recurso más crítico para la operatividad de la Universidad es el acceso al servidor de base de datos, es necesario que los componentes que establecen este enlace con el backbone, estén en funcionamiento constante. Los equipos involucrados (de acuerdo a la figura 42) en este enlace son:

Servidor de Base de Datos → Tarjeta de Acceso al medio GigaEthernet → Switch Principal

El funcionamiento de cada uno se ha descrito anteriormente, y si cualquiera de ellos falla, la comunicación se pierde y hay que establecer un mecanismo de respaldo para restablecer esta comunicación. Lo ideal sería que existan componentes de respaldo para poder cambiar cualquiera de ellos que presenten fallas, pero como no existen recursos disponibles para esto, se ha establecido un plan de contingencia que permita reestablecer el enlace entre el servidor de base de datos y el backbone.

Para la comunicación con Internet, a más de los equipos involucrados en el enlace, también es necesario que los diferentes servicios necesarios estén operativos, por lo cual se ha divido el problema en dos partes: Una

tiene que ver con la conectividad y otra que tiene que ver con la falla de algún servicio.

Los equipos involucrados en la conectividad (de acuerdo con la figura 44) con Internet son:

Switch de Instituto o Facultad → Switch Principal → Firewall → Ruteador Internet → Enlace proveedor

Si llegara a fallar el switch del Instituto o Facultad, solo ellos no tendrían conectividad hacia Internet. Pero, si llegara a fallar el switch principal, entonces todas las comunicaciones de la Espol fallarían. Si llegara a fallar el firewall o el ruteador, entonces la Universidad no tendría acceso a Internet. Actualmente, hay equipos que ya han sido reemplazados por nuevos equipos que tienen mejores características, y que si en algún momento llegan a fallar, se puede reestablecer el servicio instalando los equipos anteriores. No existe reemplazo para el firewall, pero hay un equipo que puede usarse para sustituir a un switch de una unidad o facultad.

Los servicios usados para la comunicación a Internet la proveen los siguientes equipos:

- GOLIAT: DNS, Correo

- TRITON: Servidor Web

- DAVID, ESPARTACO: Servidores proxies

Existen computadoras con los respectivos programas que pueden usarse para reestablecer cualquier servicio en caso de cualquiera de los dispositivos que prestan los servicios de Internet fallen.

Cualquiera que sea el caso de falla en la red, es necesario contar con procedimientos de respaldo que permitan reestablecer la conectividad. Estos procedimientos pueden incluirse en planes de contingencia que se deben elaborar para todos los casos posibles.

A más de los equipos de redes, también existen los enlaces que permiten la interconexión entre estos equipos. En la figura 46, se muestra los muchos enlaces que existen en el campus Gustavo Galindo, estos enlaces están constituidos principalmente por conexiones de fibra óptica, a excepción de Bienestar y Dirección de Tecnologías, cuya conexión es a través de cable par trenzado (CAT5).

Sin los enlaces no habría conectividad entre las diferentes redes y por lo tanto son un componente vital para el correcto funcionamiento. Si bien han sido instalados de acuerdo a normas y estándares, esto no garantiza que por causas externas, puedan presentar problemas.

Como se puede apreciar en la figura 45, existen enlaces que dependen de otros para proveer conectividad entre la unidad y la red principal que se encuentra en CSI. En base a esta conectividad se ha determinado los diferentes enlaces involucrados, los cuales se detallan en la tabla 14, en la cual se indican los departamentos o unidades que dependen de ese enlace para conectarse a la red.

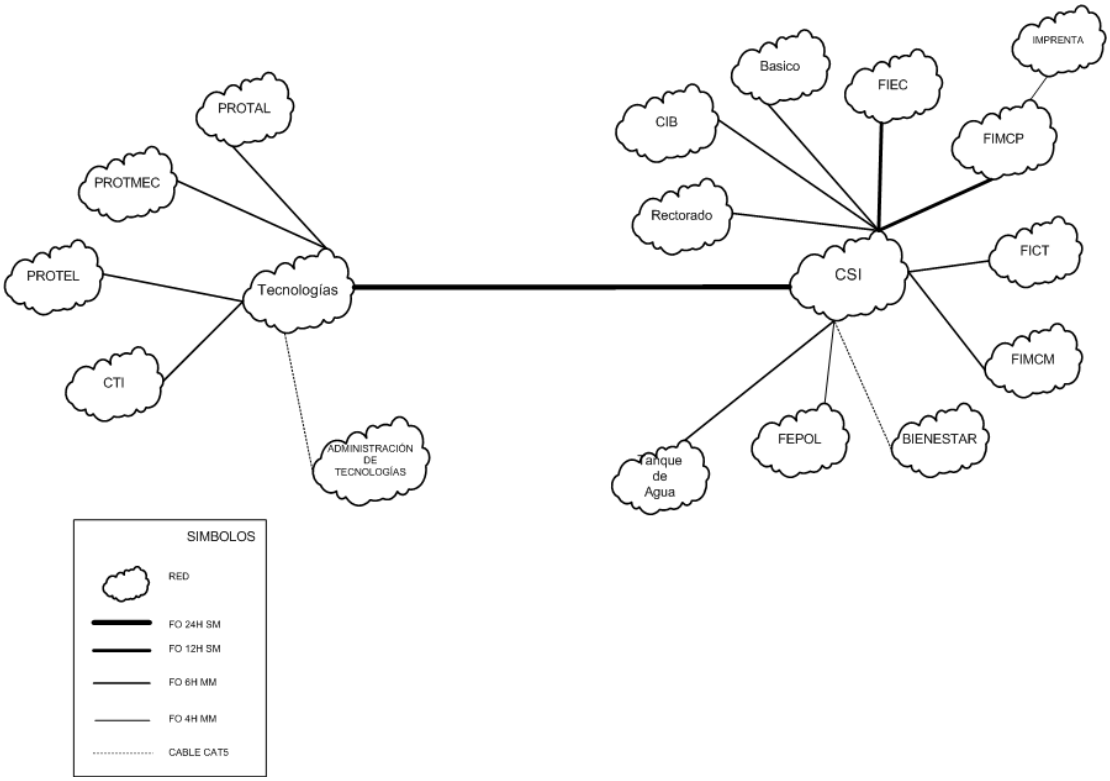


Figura 45: Enlaces de datos en el Campus Gustavo Galindo

Existen varios enlaces entre CSI y FIEC, CSI y FIMCP, CSI y Tecnologías, pero solo se está usando uno. Es necesario entonces plantear el uso de estos enlaces para poder tener redundancia entre las dependencias mencionadas. De acuerdo a la tabla 14, el enlace más crítico para la ESPOL, es el que une a la Administración Central y CSI, ya que de él depende no solo la conexión entre estas unidades, sino también Internet. Los demás enlaces no son críticos para el normal funcionamiento de la red de la ESPOL, pero si son críticos para la unidad a la cual interconectan.

Enlace	Tipo de Enlace	Provee conexión	Observaciones
CSI-Administración Central	Fibra Óptica 12 Hilos MM	Administración Central	
CSI-Tanque de Agua	Fibra Óptica 6 Hilos MM	Internet	
CSI-Biblioteca Central	Fibra Óptica 6 Hilos MM	Biblioteca Central	
CSI-Básico	Fibra Óptica 6 Hilos MM	ICM ICF ICQ ICHE	El enlace principal llega a ICQ, desde el cual se distribuye la comunicación hacia las demás unidades del Básico
CSI-FIEC	Fibra Óptica 6 Hilos MM	FIEC	Fibra Anterior (llega al decanato)
CSI-FIEC	Fibra Óptica 12 Hilos SM	FIEC	Nueva Fibra (llega al Laboratorio de Computación)
CSI-FIMCP	Fibra Óptica 6 Hilos MM	FIMCP Imprenta	El enlace llega al laboratorio de FIMCP, y de ahí se ha instalado una fibra que va hacia la Imprenta. Fibra Anterior (llega al decanato)
CSI-FIMCP	Fibra Óptica 12 Hilos SM	FIMCP	Nueva Fibra (llega al Laboratorio de Computación)
CSI-FICT	Fibra Óptica 6 Hilos MM	FICT	
CSI-FIMCM	Fibra Óptica 6 Hilos MM	FIMCM	
CSI-FEPOL	Fibra Óptica 4 Hilos MM	FEPOL	
CSI-Bienestar	Cable Categoría 5	Bienestar	
Administración Central-Tecnologías	Fibra Óptica 6 Hilos MM	Tecnologías	Fibra Anterior
CSI-Tecnologías	Fibra Óptica 24 Hilos SM	Tecnologías Administración de Tecnologías PROTEL CTI SEBIOCA PROTEC	Nueva Fibra
Tecnologías-CTI	Fibra Óptica 6 Hilos MM	CTI CENAIM CIBE CYCIT Biblioteca Tecnologías	
Tecnologías-PROTEL	Fibra Óptica 4 Hilos MM	PROTEL	
Tecnologías-Administración Tecnologías	Cable Categoría 5	Administración de Tecnologías	
Tecnologías-SEBIOCA	Fibra Óptica 4 Hilos MM	SEBIOCA PROTAL PROTMEC	
SEBIOCA-PROTMEC	Fibra Óptica 4 Hilos MM	PROTMEC	

Tabla 14: Enlaces instalados en el Campus Gustavo Galindo

Todos los equipos, servidores y enlaces instalados son importantes y cada uno cumple con una función específica dentro de la red, pero hay equipos que son críticos para el funcionamiento de la red. Como se ha descrito anteriormente, la tabla 15 hace un resumen de los componentes de las redes de acuerdo al servicio que prestan y su importancia[9].

Equipo	Disponibilidad
Switch Principal	Alta
Router Internet	Alta
Firewall	Alta
Goliat	Alta
David	Moderada
Espartaco	Moderada
Sanson	Alta
Triton	Moderada
Servidor de Base de Datos	Alta
Switch de cada unidad	Moderada

Tabla 15: Importancia de cada componente en la red

La tabla anterior presenta un detalle de los principales componentes de la red y su importancia global dentro de la ESPOL. Cada administrador local, debe determinar los componentes críticos que tiene en su red.

[9] Datos basados en la Tesis: "Seguridad de redes de computadoras frente a Internet: Estudio, diagnóstico e implementación de firewalls

4.4 Problemas en la administración de redes

La ESPOL es una institución educativa y de investigación donde se requiere un nivel de libertad en el uso del recurso de la Internet pero que sin embargo requiere mecanismos de control para evitar un crecimiento anárquico y riesgos de seguridad. En la actualidad en la ESPOL cada unidad es responsable por sus propias redes, lo que implica que deben establecerse políticas que deben seguir todos y cada uno de los administradores de las redes locales que cada unidad posee. Dichas políticas no existen actualmente.

Es necesario que el organismo central encargado del backbone posea todas las herramientas necesarias para determinar la procedencia de fallas dentro de la red porque en una arquitectura de redes como TCP/IP que tiene varias capas, descubrir y resolver problemas es algo complejo que necesariamente conlleva la utilización de equipos y software específicos para esta tarea.

Entre los equipos necesarios para una correcta administración están:

- Probador de cableado
- Probador de enlaces de fibra
- Analizador de protocolo para la capa de red y de aplicación

- Capturador de paquetes de red

- Sistema de administración de redes (como el descrito en el Capítulo Uno)

Además de los equipos, la mayoría de los problemas en la administración de redes son los mismos usuarios que utilizan la red, los cuales desconfiguran los parámetros de redes de sus estaciones de trabajo. Este problema puede ser resuelto con software para restringir el acceso a la configuración de los protocolos y tarjetas de red de las estaciones.

En la administración de redes existen varios problemas, los cuales pueden ser:

- Crecimiento en extensión y en tráfico

- Seguridad interna

- Seguridad externa

- Contingencia

- Redundancia

- Capacitación

- Documentación

Crecimiento en Extensión y en tráfico

A medida que se añaden nuevas redes o nuevos servicios dentro de la red de la ESPOL, es necesario realizar un estudio de impacto para determinar la conveniencia o no de esta instalación. Este estudio determinaría los requerimientos adicionales necesarios para llevar a cabo la nueva instalación.

Como la ESPOL es una institución educativa, uno de los recursos más utilizados por los estudiantes es Internet. Supongamos que se instala un nuevo laboratorio para los estudiantes con 500 computadoras, y no se aumenta el ancho de banda a Internet. A medida que vaya aumentando el número de máquinas usadas en el laboratorio, Internet se volverá más lento, impidiendo que otras redes puedan acceder a Internet.

Es necesario realizar estudios de tendencias de tráfico, para determinar nuevas mejoras de equipos y/o enlaces.

Seguridad Interna

Muchas veces cuando se habla de seguridad de una red, solo se visualiza el elemento externo que puede penetrar a la red a través de Internet, y no se considera las posibles amenazas internas. Como es necesario dar libertad en el uso de las redes internas, también es necesario establecer medidas de seguridad que permitan proteger los sistemas de las amenazas dentro de la institución.

Para llegar a establecer una seguridad interna, se requiere establecer los tipos de usuarios, y los niveles de privilegio que cada uno de ellos tienen respecto a los recursos (detalles en el capítulo 5), y configurar todos los recursos de acuerdo a estos niveles de privilegio en todos y cada uno de los componentes de las redes.

Seguridad Externa

No solo se recomienda la instalación de firewalls que protejan a la ESPOL de amenazas externas, también se recomienda herramientas adicionales para prever amenazas externas, una de las cuales puede ser un Sistema de Detección de Intrusos que se usa principalmente para determinar las posibles amenazas a las cuales se expone una red que se conecta a Internet.

Es necesario configurar los equipos y servicios de acuerdo a las políticas de seguridad que se establecerán.

Contingencia

Para todos y cada una de las conexiones o servicios de red, se hace necesario establecer planes de contingencia que permitan, en caso de fallas, reestablecerlos en el menor tiempo posible. Estos planes de contingencia deberán incluir el procedimiento paso a paso a seguir, también deben incluir todos los componentes necesarios.

A medida que nuevos equipos se incluyan en las redes, es necesario realizar un plan de contingencia para dicho equipo o enlace.

Redundancia

Para los componentes críticos de las redes, se recomienda la adquisición de equipos de redundancia que permitan establecer los servicios que proveen los equipos principales en caso de falla.

Capacitación

Cada red dentro de la ESPOL es administrada por una persona diferente, las cuales muchas veces no poseen el conocimiento técnico necesario para determinar la causa de algún problema. Por lo tanto es necesario

brindarle capacitación básica para que los administradores locales puedan resolver problemas dentro de su unidad.

Documentación

La documentación de la red debe incluir un inventario actualizado de todos los componentes de la red, diagrama de conexión (físico y lógico), y un registro con los detalles de todos los problemas que han surgido.

La administración central deberá tener la documentación de la configuración de todos los equipos conectados al backbone de la ESPOL, para efectos de coordinar y controlar de mejor manera la operación del backbone y planificar su crecimiento.

Todos los usuarios y administradores locales deben seguir las políticas establecidas y los requerimientos necesarios para la administración de la red

Capítulo V

Modelo de Administración de las redes de la ESPOL

5.1 Introducción

El tamaño y complejidad de las redes y los sistemas que se encuentran instalados en la ESPOL y la diversidad de componentes de hardware y software dificultan su administración por lo que es necesario la implementación de un sistema de administración automatizado que libere a los administradores de la carga excesiva que implica la operación diaria de las redes y sistemas existentes.

Para la implementación de un sistema de administración es necesario que la organización tenga establecidas políticas generales de acuerdo a los objetivos de la institución en base a los sistemas y servicios instalados. Las políticas son usadas para definir metas, responsabilidades y autoridad de los diferentes usuarios y para que el funcionamiento de las redes y sistemas sean coherentes y aplicables a todos y cada uno de los servicios computacionales ofrecidos. Las políticas expresan la forma de hacer las cosas y como los sistemas deben reaccionar a situaciones dadas.

Actualmente, no existe un documento formal en el cual se detallen las políticas que controlen los sistemas de información en la Universidad. Solo existe el documento “Reglamento 2113[10] para la asignación y uso de cuentas electrónicas”. Este reglamento define el procedimiento para la asignación de cuentas y las obligaciones de los dueños de las mismas. A más de este reglamento, sólo existen ciertos documentos que indican que hay que elaborar políticas para los sistemas de información, pero sin detallar las políticas que gobiernan los sistemas.

En este capítulo se diseñarán los diferentes esquemas para cada una de las áreas de la administración de redes que fueron descritos en el capítulo 1, así como el esquema de administración, el cual fue descrito en el capítulo 3. Por último se establecerán las políticas generales que serán aplicables a las redes de la ESPOL.

5.2 Arquitectura del Sistema de Administración

Como se describió en el capitulo 3, existen 3 arquitecturas para los sistemas de administración:

10

- Centralizada
- Jerárquica
- Distribuida

Cada una de las cuales tiene sus ventajas y desventajas, lo cual hace que cada una de ellas sea apropiada dependiendo de la distribución de la red y de los recursos. Para detallar las ventajas y desventajas de cada una, se ha elaborado un cuadro considerando los factores más importantes para la implementación de los sistemas de administración. Estos factores son:

- Tiempo de Implementación.- El tiempo que toma el implementar un sistema de administración usando una de las diferentes arquitecturas

- Tráfico.- Tráfico adicional que se genera para las tareas de administración de la red dependiendo de la arquitectura usada

- Respaldo.- Si es que la arquitectura usada tiene respaldo tanto del sistema de administración como de la información de administración.

- Recursos.- Los recursos tanto de hardware, software y personal usados dependiendo de la arquitectura de administración.

---

http://www.intranet.espol.edu.ec/webDoc/Reglamen.nsf/0e42bd246b15d458052563c1007db7ec/a91d71a1d46836e305256783006c796b?OpenDocument6836e305256783006c796b?OpenDocument

Arquitectura	Tiempo de Implementación	Tráfico	Respaldo	Recursos
Centralizado	Corto	Alto	No	Pocos
Jerárquico	Medio	Moderado	No	Moderados
Distribuido	Largo	Poco	Si	Muchos

Tabla 16: Factores para la evaluación de la arquitectura de administración

Cada una de las arquitecturas tiene sus ventajas para la implementación en diferentes ambientes de redes. En la ESPOL, los dos factores más importantes a considerar para la implementación de una arquitectura son la disponibilidad de recursos y la forma de la administración de los sistemas, que en la ESPOL, se realiza de manera centralizada por el departamento que administra la red y los sistemas de la institución. Por estos motivos, la arquitectura considerada para la administración es la Centralizada, en la cual, el servidor de administración de la red se ubicará en el backbone de la red del Campus Gustavo Galindo.

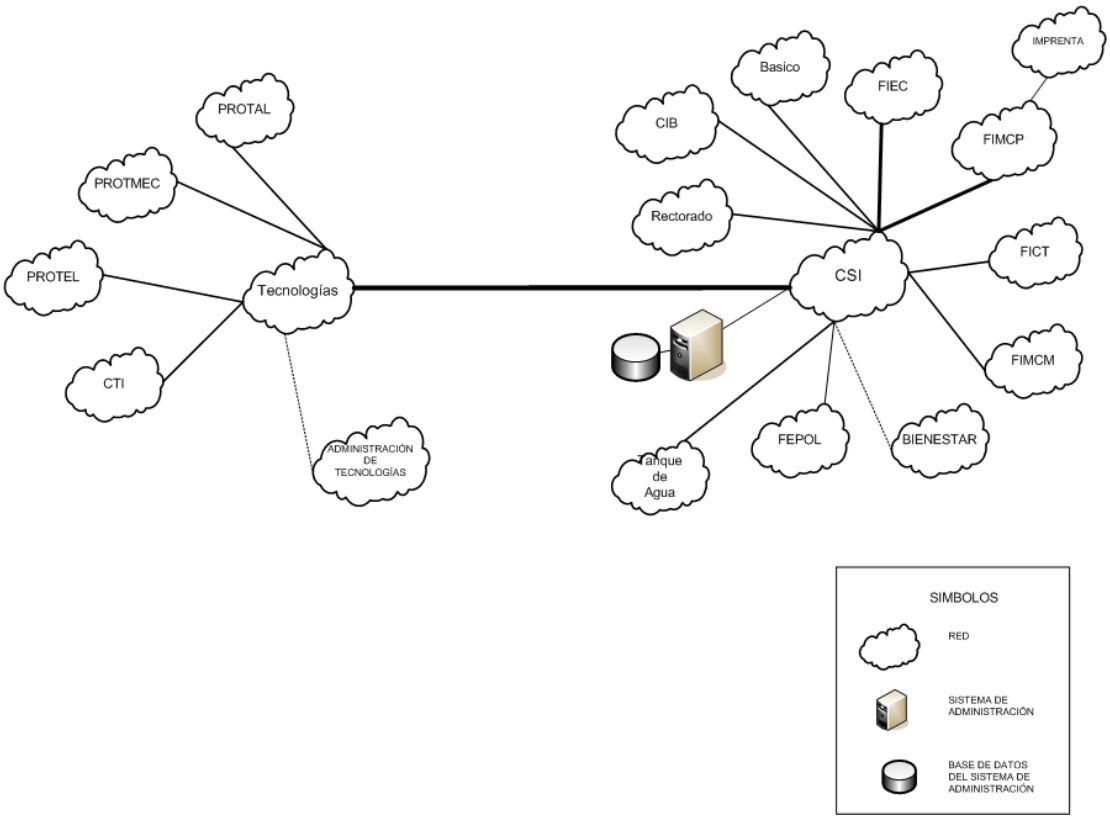


Figura 46: Arquitectura de Administración para la ESPOL

Como se describió en el Capítulo 3, la arquitectura Centralizada tiene algunas desventajas, para lo cual se ha considerado establecer procedimientos que ayuden a disminuir el impacto de estas en la administración de la red. Estas desventajas son: tráfico adicional y respaldo, para lo cual en el diseño de los diferentes esquemas de administración se considerarán esto puntos para minimizar su impacto.

5.3 Diseño del Esquema para el Manejo de Fallas

El manejo de fallas en una red es una tarea crítica y compleja dentro de la administración. Es crítica, porque si existe alguna falla, la red no funciona correctamente, y compleja porque determinar el origen de la falla es complicado ya que pueden ser varias las causas que la originan. Por estas razones, el diseño del esquema para el manejo de fallas es crucial en la administración de una red.

Como se describió en el Capítulo 1, existen varias etapas para el manejo de las fallas, y son:

- Localización de las fallas,
- Elección de las fallas,
- Aislamiento de las fallas, y
- Corrección de las fallas.

Para la localización de las fallas, se considera el uso de los dos mecanismos comunes: recolección de los eventos de la red, y verificación periódica de los dispositivos. Para la recolección de los eventos, es necesario que los dispositivos de red estén configurados apropiadamente para enviar las notificaciones al sistema administrador, para la cual, todos los dispositivos de red deben ser administrables. Para la verificación periódica de los dispositivos, es necesario determinar el período de tiempo ideal de verificación de los dispositivos para que así el

administrador no sature la red pero al mismo tiempo obtenga información crítica en tiempos razonables, este tiempo puede ser de alrededor de 5 minutos inicialmente, y luego realizar medidas que indiquen el ancho de banda requerido para determinar si es factible disminuir este tiempo. Para el cálculo del ancho de banda potencialmente consumido en el control de fallas en el backbone de la ESPOL, tenemos lo siguiente:

Intervalo de consulta: 5 minutos

Dispositivos: 40

Tamaño de consulta/respuesta: 100 bytes

(100 bytes + 100 bytes) * 40 dispositivos = 8000 bytes

(8000 bytes * 8 bits/byte) = 64000 bits

O sea que por cada consulta se generaría 64000 bits de tráfico. Si la consulta se realiza cada 5 minutos, en promedio serían:

64000 bits / 300 segundos = 213 bps (bits/segundo)

Este uso del ancho de banda es poco comparado con los 1000Mbps que tienen las redes de la ESPOL. Además, a esto habrá que sumar el tráfico generado por las demás áreas de la administración, lo cual veremos más adelante.

En la Figura 47 se muestra el diagrama del Backbone del Campus Gustavo Galindo, en la cual se muestran los equipos de comunicaciones y los principales servidores de la ESPOL, como cada Unidad, Facultad e Instituto es responsable de su propia red, la Administración Central debe garantizar la comunicación hasta el dispositivo que le da conectividad a cada una de las redes (este tipo de alcance debe ser una política de la administración central), por lo que el manejo de fallas debe incluir los equipos centrales y los equipos que dan conectividad a cada una de las redes, no las redes locales individuales. Sin embargo se podría también monitorear dispositivos dentro de las redes de cada unidad bajo requerimiento explícita de esta para informarle indirectamente de alguna falla.

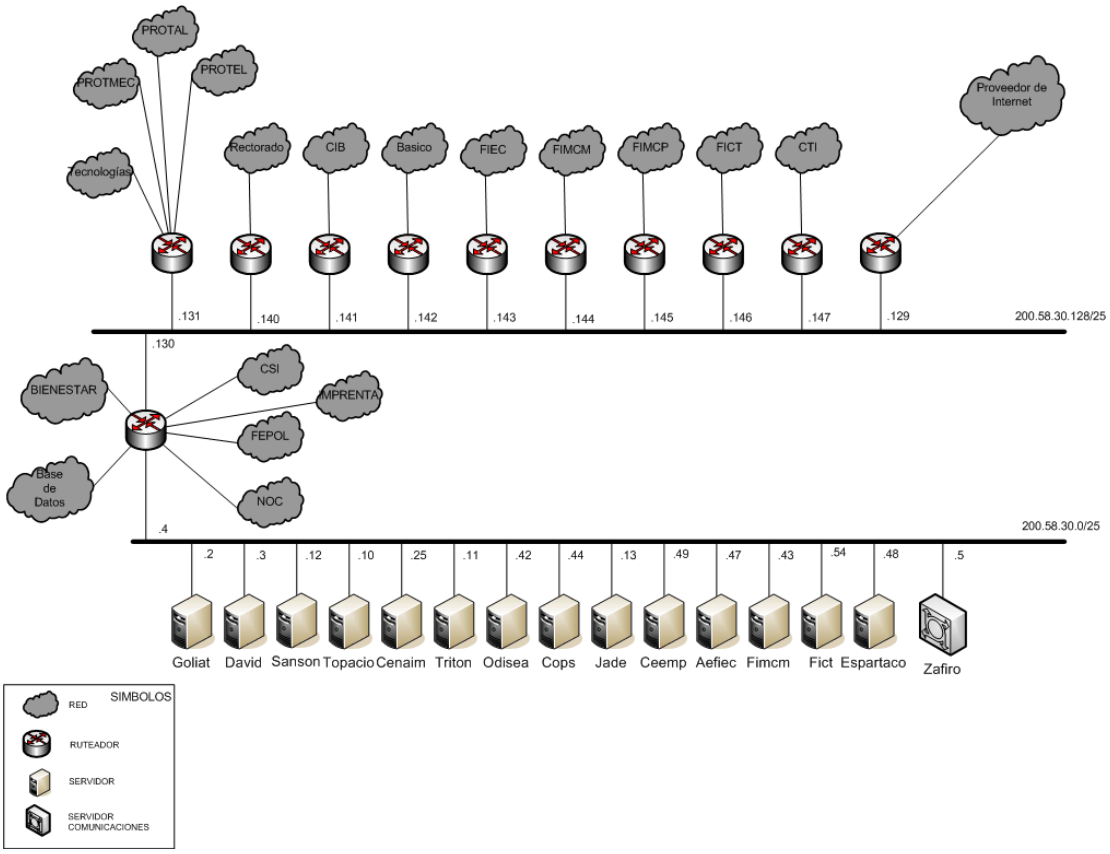


Figura 47: Diagrama del Backbone de la ESPOL, Campus Gustavo Galindo

El manejo de fallas debe incluir:

Las fallas de los componentes principales de la red.- Estos componentes incluyen los dispositivos centrales de comunicación, servidores principales y los dispositivos que proveen comunicación a cada red individual. Estos equipos y servidores se muestran en la

figura 47. Igualmente se puede incluir a equipos que posean las unidades, previa solicitud de la misma.

Los eventos críticos de los dispositivos administrados.- Los eventos críticos como pérdida de un enlace de comunicaciones, falla en los servicios de los servidores principales (smtp, pop, http), utilización más allá del 90% en ciertos recursos como procesador y memoria en los dispositivos de redes y servidores (incluyendo el uso del almacenamiento).

Para el aislamiento de las fallas, es necesario relacionar los diferentes eventos que pueden ocurrir debido a una misma causa, para evitar que en la herramienta de administración aparezcan mensajes innecesarios que pueden ocultar la verdadera causa del problema.

Para la corrección de las fallas, es necesario establecer procedimientos que conlleven a la solución de la falla, para lo cual será necesario catalogar las fallas de acuerdo a la siguiente tabla:

	Falla	Descripción
1	Enlace de Comunicaciones	Falla causada por la perdida de un enlace físico
2	Componente de Comunicaciones	Falla causada por daño del equipo
3	Pérdida de configuración	Falla causada por la pérdida de configuración o mala configuración del equipo
4	Caída de un servicio	Falla causada por la caída de un servicio en un servidor o componente de red
5	Sobreutilización de Recursos	Falla causada cuando un recurso de un componente es sobreutilizado por un período de tiempo determinado como: procesador, memoria, disco, enlace

Tabla 17: Categorización de las Fallas

La herramienta de administración puede notificar al administrador sobre la caída de un enlace, pero esta falla puede deberse a 1, 2 ó 3, para lo cual se debe descartar cada una en el siguiente orden: Primero se debe determinar si el equipo se ha dañado, luego se debe determinar si el equipo ha perdido la configuración, y por último, se debe determinar si el enlace ha sufrido algún daño.

Para las fallas 4 y 5 se debe examinar los eventos que tiene la herramienta de administración, estos eventos pueden mostrar la sobreutilización de un recurso de un componente o la caída de un servicio.

Las fallas deben ser notificadas por varios medios a los administradores de la red. Estos medios los da la herramienta de administración que la institución posea. El principal medio de notificación, es mostrando en la pantalla de eventos la ocurrencia de una falla, esto se puede complementar con un sonido, para llamar la atención del administrador, y también la falla puede ser notificada a través de un mensaje a un beeper o teléfono celular (la notificación a través de beeper o teléfono, será posible siempre y cuando la ESPOL contrate este servicio).

Para el esquema del manejo de fallas tenemos que:

- La arquitectura de administración es centralizada, en la cual el sistema administrador deberá estar en una red exclusiva para la administración.
- La recolección de eventos por parte del sistema administrador se realizará cada 5 minutos.
- Todos los componentes principales de la red, incluyendo los servidores, deben ser configurados para enviar las alertas al sistema administrador.
- Es necesario catalogar las fallas, de acuerdo a la Tabla 17 para que la administración sea coherente.

- La forma de reportar las fallas es mediante mensajes de texto que se presentarán en la pantalla del sistema de administración y un sonido para alertar al administrador de la red. Si es posible, también el sistema administrador podrá enviar una alarma a través de un beeper al administrador o administradores de la red (siempre y cuando se contrate este servicio).

Además de la notificación de la falla, es necesario establecer un mecanismo para manejarlas, el cual establece el procedimiento para resolver las fallas. Este procedimiento conlleva los siguientes pasos:

Notificación.- El administrador de la red es notificado de la falla, ya sea por medio de la herramienta de administración, los administradores locales, etc. Una vez que el administrador es notificado, es necesario abrir un caso para la falla que ha sido notificada, este caso deberá llevar un número único y deberá ser catalogada de acuerdo a la Tabla 17.

Asignación.- El administrador de la red asignará el caso a una persona para que maneje la falla. Esta persona deberá revisar la falla, para determinar el origen de la misma.

Resolución.- Luego de que la persona ha determinado el origen de la falla, deberá corregirla.

Finalización.- Una vez que la falla ha sido corregida, la persona asignada, documentará todo el proceso que hizo para resolver la misma.

Todo el proceso de resolución de fallas, puede ser documentado a través de un documento de texto, o usando una base de datos de Lotus Notes. Esta base de datos deberá tener un registro de todas las fallas.

5.4 Diseño del Esquema de Configuración

Como se mencionó en el Capítulo 1: “La administración de la configuración involucra recolectar información acerca de la configuración actual de la red, usar los datos para modificar la configuración de los dispositivos de la red, almacenar la información, mantener un inventario actualizado, y producir reportes basados en los datos recolectados.”

Pero para realizar la tarea de administrar la configuración de dispositivos, es necesario tener pautas que permitan estandarizar, lo siguiente:

La compra de equipos de comunicaciones.- Para que los datos de la configuración sean fáciles de recolectar y procesar, es necesario que todos los equipos de comunicaciones posean una misma base de datos (MIB) para poder recolectar información acerca de: Modelo, Número de Serie, Versión del Código, Detalle de Componentes.

Direccionamiento de red y nombre de equipos.- Será necesario establecer un direccionamiento de red que sea común a todos los componentes de red, es decir establecer rangos de direcciones para la configuración. Así, para los equipos de comunicaciones, como switches y otros equipos de comunicaciones utilizaremos los 10 primeros números de la red, para los ruteadores usaremos los 10 números siguientes, para las impresoras los 5 siguientes, para los servidores los 20 siguientes, y para las demás computadoras el rango que queda. Por ejemplo, en una red común con 2 switches, 1 ruteador, 2 servidores, 1 impresora y 10 computadoras, el direccionamiento sería el siguiente: Switch 1 192.168.1.1, Switch 2 192.168.1.2, Ruteador 1 192.168.1.11, Impresora 1 192.168.1.21, Servidor 1 192.168.1.31, Servidor 2 192.168.1.32, Computadora 1 192.168.1.51, Computadora 2 192.168.1.52 y así sucesivamente.

Para los nombres de los componentes, se usará la siguiente nomenclatura: sw para switches o hubs, gw para ruteadores, srv para servidores, wrks para computadoras, prt para impresoras, ap para puntos de acceso, br para puentes, asrv para servidores de acceso. Luego del tipo de componente, se colocará el último número de la dirección de red que el equipo tenga, seguido por la ubicación del equipo. Por ejemplo, si tenemos un switch que está en la red del CSI con dirección de red 192.168.1.1, el nombre del equipo sería: sw01csi.

Configuración básica del dispositivo.- Es necesario que en todo dispositivo de red se configure lo básico para poder tener acceso a las funcionalidades del mismo. La configuración básica es: nombre del equipo, dirección de red, clave de acceso y protocolo de administración. Para el nombre del equipo y dirección de red, es necesario usar el esquema planteado anteriormente. Para la configuración del protocolo de administración de red, es necesario establecer los parámetros que el protocolo requiera; por ejemplo, para el caso de SNMP es necesario establecer la comunidad de lectura, escritura y servidor desde el cual se puede hacer consultas y al que se van enviar las alertas del equipo.

Otro de los aspectos que se deben considerar para la configuración, es los diferentes protocolos que se van a usar dentro de la red, como TCP/IP, NetBios, Appletalk o IPX; y dentro de los protocolos, que tipo de rutas se usarán, si serán rutas estáticas o dinámicas, y el protocolo de intercambio de rutas.

Si bien el manejo de la configuración debe tener un alcance al igual que el manejo de fallas, es necesario que una sola unidad sea la encargada de la configuración de los diferentes componentes de red, para así evitar problemas de compatibilidad de configuración entre dispositivos.

El manejo de la configuración conlleva 3 pasos principales, que son: recolección, modificación y almacenamiento de la información de la

configuración. Para la recolección de la información, será necesario recolectar lo siguiente:

- Nombre del Equipo,
- Dirección de red,
- Marca y modelo,
- Número de serie y número de parte,
- Componentes adicionales,
- Fecha de compra,
- Tiempo de Garantía,
- Versión del software y/o sistema operativo,
- Localización del equipo,
- Persona Responsable del equipo.

Esta información recolectada, será almacenada en una base de datos de Lotus Notes que ya posee el Centro de Cómputo de la ESPOL.

Como en la red actual de la ESPOL (ver Capítulo 4) no hay una marca común para los componentes de comunicaciones, no es posible usar una misma herramienta para configurar los equipos. Los cambios son hechos manualmente, lo que debe establecer que cada cambio que se haga en la configuración de algún equipo debe contener lo siguiente:

- Fecha y hora del Cambio,
- Nombre del Equipo,
- Dirección de red,
- Número de serie,
- Persona que hizo el cambio,
- Cambios realizados,
- Observaciones.

Este registro de cambio es de suma importancia, ya que permite llevar un control de todos los cambios de la configuración de los diferentes componentes de comunicaciones, estos cambios, también deberán ser almacenados en la misma base de datos de Lotus Notes del Centro de Cómputo de la ESPOL.

Además de la documentación de la configuración de los equipos, es necesario llevar la documentación de las configuraciones del cableado de red y fibra óptica de cada unidad. El CSI, será el responsable mantener la documentación de toda la fibra instalada en el Campus Gustavo Galindo, y de la documentación del cableado de la red de la Administración Central y CSI. Cada unidad será responsable de mantener la documentación del cableado de la red propio

5.5 Diseño del Esquema para el Manejo del Rendimiento

El manejo del rendimiento, al igual que el manejo de fallas, es fundamental en la administración de la red ya que asegura que la red se mantenga accesible y descongestionada. Como se vio en el Capítulo 1, el manejo del rendimiento conlleva cuatro pasos fundamentales, que son:

1. Recolectar información de la utilización actual de los dispositivos y enlaces de la red
2. Analizar la información relevante para determinar las tendencias de utilización elevadas
3. Configurar los límites de utilización
4. Usar simulación para determinar como la red puede ser alterada para maximizar su rendimiento

Primero, el recolectar información de la red, es el paso principal y fundamental para la administración del rendimiento. Al igual que en el manejo de fallas, la recolección de los datos para la medición del rendimiento se hará cada 5 minutos. Los datos a recolectar serán de los dispositivos que están en la Figura 48, y además de los servidores de Internet y de bases de datos, los cuales se detallan en el Capítulo 4. Los datos a recolectar de cada equipo será:

- Para dispositivos de comunicaciones.- Uso del procesador, uso de memoria, uso de cada una de las interfaces de comunicaciones (Información transmitida, Información recibida, Porcentaje de Error, porcentaje de rechazo).

- Para servidores.- Uso del procesador(es), uso de memoria (física, paginamiento), uso cada una de las interfaces de comunicaciones (Información transmitida, Información recibida, Porcentaje de Error, porcentaje de rechazo), porcentaje de uso de cada partición del servidor.

Segundo, la información recolectada a través del sistema de administración, se analizará para determinar la tendencia de uso de los diferentes componentes de un equipo. Esta tendencia de uso permitirá determinar la necesidad de mejoras o reemplazo de componentes para permitir a la red operar de manera adecuada.

Tercero, del análisis de las tendencias de uso, junto con el manejo de fallas, permitirá establecer los valores de los umbrales de cada componente de los diferentes equipos de comunicaciones. Estos umbrales deben establecerse en base a los datos históricos. Por ejemplo, si el uso de la memoria de un enrutador supera el 90%, y este uso produce una falla en las comunicaciones que maneja ese enrutador,

entonces se deberá establecer un umbral del 85% para este componente del dispositivo. El umbral deberá ser menor al valor que produjo la falla, con lo cual permitirá realizar las medidas adecuadas para evitar una falla en la red.

Cuarto, la simulación, dentro del manejo del rendimiento, permite identificar posibles mejoras en el rendimiento general de una red, pero modelar una red simple en una herramienta de simulación es una tarea demasiado compleja y su costo es alto. Por esta razón, la simulación debe hacerse en base a la información histórica para poder determinar las tendencias del uso en el futuro.

La información del uso de procesador, memoria y disco se mostrará en porcentaje, al igual que el uso de las interfaces de comunicaciones y de los porcentajes de error y de rechazo. La información recolectada deberá mostrarse a través de un gráfico de línea. La información del rendimiento se revisará cada semestre para analizar las tendencias de uso de los diferentes equipos, de comunicaciones y servidores, para determinar la necesidad de ampliación o actualización de los mismos. Esta tendencia también deberá ser analizada para los enlaces externos que la ESPOL posea, principalmente el enlace hacia Internet.

Para el reporte del rendimiento de la red, es necesario establecer alertas que indiquen que se ha alcanzado un umbral de alguno de los

componentes y/o servidores de la red. La manera de reportar estos eventos será de la misma forma en la que se reportan los eventos del manejo de fallas.

Los reportes del rendimiento deben ser mostrados a través de una interfaz adicional a la que muestra el sistema de administración, esta interfaz puede ser una página web, en donde se muestre los gráficos de todos los equipos que la administración central administra. Se deberán permitir el acceso a todos los administradores de las redes locales para que vean el uso de sus recursos de red y de Internet.

Adicionalmente, el manejo del rendimiento permite verificar que los proveedores externos cumplan con los acuerdos de nivel de servicio ofrecidos en los contratos preestablecidos.

5.6 Diseño del Esquema de Seguridad

El manejo de la seguridad, como se describió en el Capítulo 1, envuelve los siguientes pasos:

- Identificar la información sensible
- Encontrar los puntos de acceso
- Asegurar los puntos de acceso
- Mantener la seguridad de los puntos de acceso

Como toda institución, la ESPOL posee aplicaciones que le permiten funcionar día a día, y todas ellas almacenan la información en una base de datos central. Además de esta base de datos central, puede considerarse información sensible a los proyectos claves para la institución y también a los exámenes que los profesores elaboran. De acuerdo a esto, hay que tener dos esquemas diferentes para el manejo de la seguridad, uno para asegurar la base de datos central, y el otro, para asegurar las computadoras personales de los profesores y administrativos.

Seguridad de la Base de Datos Central

Todas las máquinas de la Espol pueden tener acceso, mediante la red, a la base de datos central, por lo que asegurar el acceso a la base de datos central, deberá contemplar lo siguiente:

- *Ninguna máquina usada por los estudiantes deberá tener acceso directamente a la base de datos central.*

- *Todos los servicios no necesarios en el servidor de base de datos, deberán ser deshabilitados.*

- *Será necesario la actualización periódica del sistema operativo y del software de base de datos.*

- *Se configurará filtros de acceso para el servidor de base de datos.*

Seguridad de las computadoras personales de profesores y administrativos

Para la protección de las computadoras de profesores y administrativos, se realizará lo siguiente:

- *Las computadoras de profesores y administrativos no estarán en redes donde existan computadoras para estudiantes.*

- *Deberán establecerse filtros de acceso para que las redes de los estudiantes no tengan acceso a las redes de profesores y administrativos.*

- *Se instalarán programas antivirus y se configurarán todas las seguridades a nivel de sistema operativo en las computadoras de profesores y administrativos.*

- *Se actualizarán los parches del sistema operativo y las bases de datos del software antivirus.*

Para el acceso a Internet, será necesaria la instalación de un firewall, en el cual se permitirá el ingreso a los servicios que la ESPOL ofrece solamente, los demás servicios serán restringidos. La información de usuario y contraseña, deberán ser encriptadas para transmitirlas por Internet. También será necesaria la instalación de un Sistema de Detección de Intrusos[11] y un sistema antivirus. Este esquema de conexión se muestra en la siguiente figura:

[11] Un sistema de detección de intrusos (IDS) permite, en base a reglas establecidas, examinar el tráfico de la red y alerta al administrador de posibles intentos de violación de los servicios disponibles

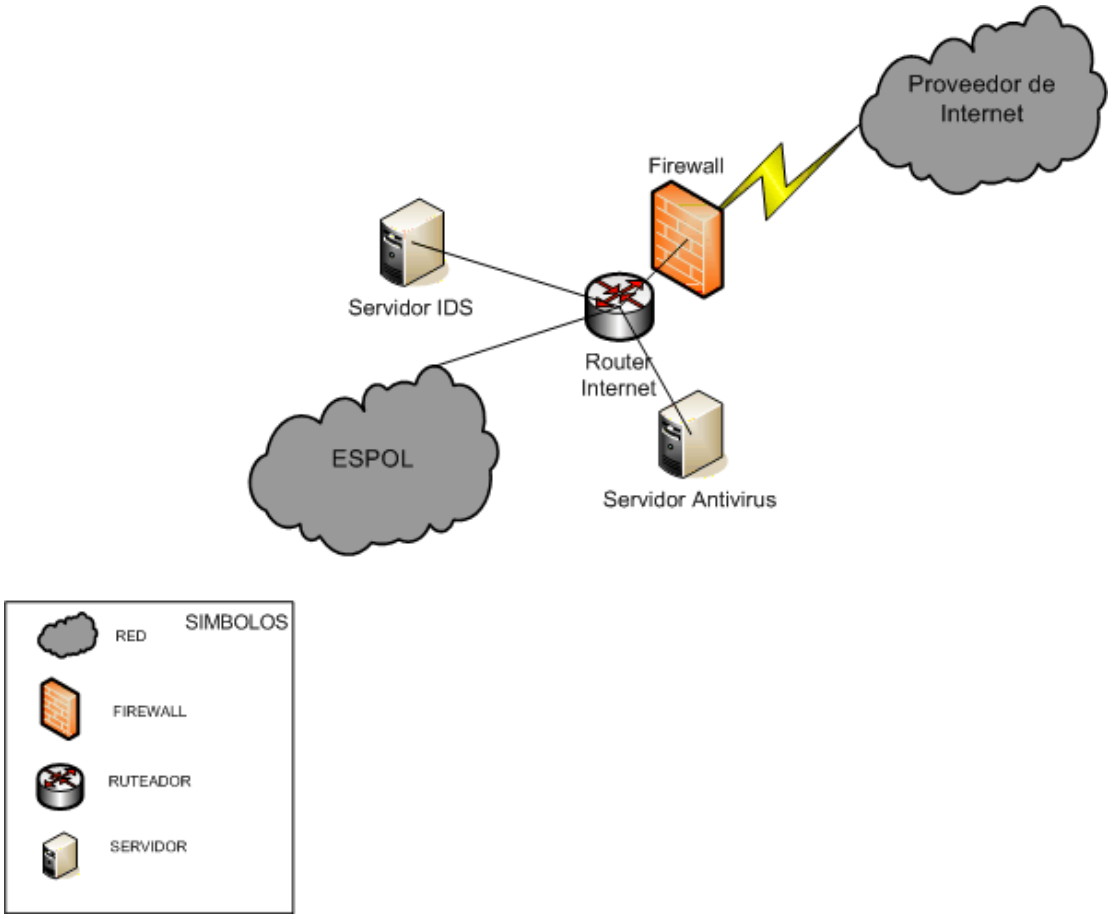


Figura 48: Esquema de conexión a Internet

El balance entre la seguridad de la red y el uso de la misma es algo que se debe tener en consideración al momento de establecer las políticas que definan la forma y acceso a la información.

5.7 Diseño de las Políticas Generales

Las políticas generales son un conjunto de reglas simples que gobiernan la administración de los recursos de las redes. Estas políticas tienen que ser congruentes con los objetivos de la institución y permitir el acceso a la información de una forma coherente, simple y segura.

Las políticas generales se pueden dividir en:

- Compra de equipos.- Políticas para establecer estándares para la compra de equipos de comunicaciones.
- Acceso a la red.- Son las políticas que permiten acceder a la red de la institución.
- Usuarios y claves.- Políticas que permiten crear cuentas para el acceso a la información y definir diferentes perfiles de acceso.
- Seguridad.- Políticas que permiten especificar las seguridades a implementarse en la ESPOL.
- Varias.- Políticas adicionales que complementan las anteriores.

Compra de Equipos

Los equipos de comunicaciones deben ser compatibles con los ya existentes y deben tener las características que el Centro de Cómputo defina, estas características deberán permitir cumplir con las políticas descritas.

Cada unidad es responsable del equipo que le provee conectividad, si son varias las unidades que se conectan a través de un mismo equipo, entonces la responsabilidad será compartida.

Todos los equipos deben ser inventariados y deben mantenerse con un contrato de mantenimiento.

La ubicación de los equipos de comunicaciones deberá ser en cuartos especiales destinados para el efecto, los cuales deberán tener seguridad de acceso, temperatura adecuada y contar con un UPS para respaldo eléctrico.

Acceso a la Red

La unidad que requiera conectarse a la red de la institución, deberá pedir un estudio al Centro de Cómputo para la determinación de presupuestos y características técnicas necesarias para la conexión.

Para enlaces externos, ninguna unidad podrá contratar enlaces sin coordinar previamente con el Centro de Cómputo.

La comunicación hacia Internet se hará únicamente a través del enlace contratado por la Administración Central, y es responsabilidad de la misma mantener un enlace adecuado para las necesidades de la institución.

El Centro de Cómputo asesorará en la configuración y reemplazo de componentes existentes o nuevos.

El Centro de Cómputo realizará mediciones del rendimiento de la red, para determinar el uso de la red.

El esquema de direccionamiento de red y protocolos de comunicaciones serán determinados por el Centro de Cómputo.

Las redes de laboratorios no deberán contener ninguna otra máquina que no sea destinada para el uso de estudiantes.

Usuarios y Claves

La Administración Central mantendrá una sola base de usuarios y claves que será común a todos los servicios que se implementen.

Todos los estudiantes, profesores y personal administrativo de la ESPOL tendrán una cuenta electrónica que le servirá para acceder a los servicios implementados y futuros, y los lineamientos de uso de la cuenta serán de acuerdo al Reglamento de Asignación y Uso de Cuentas Electrónicas.

Los usuarios deberán tener un máximo de ocho letras y las claves un mínimo de seis letras.

Las cuentas electrónicas serán bloqueadas si existen tres intentos seguidos no autorizados de acceso a la cuenta.

Seguridad

Todas las computadoras deberán ser protegidas con sistemas antivirus para poder tener acceso a la red.

Es responsabilidad de los diferentes administradores de las unidades instalar los últimos parches necesarios para todos los programas y sistemas.

El Centro de Cómputo será el encargado de determinar los diferentes sistemas de seguridad a implementarse de acuerdo a las nuevas tecnologías que se vayan desarrollando.

Todas las comunicaciones que involucren el intercambio de usuario y contraseña deben ser encriptadas.

El acceso inicial a la red de la ESPOL deberá ser a través de un usuario y contraseña provisto por la institución.

La implementación de cualquier nueva tecnología, deberá ser configurada adecuadamente, asegurando la seguridad de información y del acceso a la red.

Varias

Cada unidad deberá contar con un administrador local, y deberá notificar al Centro de Cómputo de la persona responsable.

El Centro de Cómputo será la encargada de medir las tendencias de uso de los recursos de comunicaciones.

Capítulo VI

Implementación de la Administración de las redes de la ESPOL

6.1 Introducción

Para la administración de la red de la ESPOL, se usan las siguientes herramientas:

- Tivoli Netview versión 5.1.- Para la visualización de la red a nivel lógico. Provee un mapa de conectividad de las diferentes redes, y además recolecta información de los diferentes dispositivos administrados.

- CiscoWorks 2000.- Para la configuración de los equipos cisco que están instalados en el campus Gustavo Galindo. Esta herramienta también provee un mapa de conectividad, pero a nivel de capa 2 de los dispositivos cisco.

- Flowscan.- Herramienta usada para mostrar el uso del enlace de Internet, por aplicación y por red. Esta herramienta procesa la información del router Cisco que provee la conectividad a Internet.

- Active Directory.- Usada para la administración de los usuarios que hay en la ESPOL.

- SunScreen 3.1.- Firewall que se conecta al router de Internet.

Todas estas herramientas sirven para cada una de las áreas de administración, como se muestra en la tabla siguiente:

	Áreas de Administración				
Herramienta	**Fallas**	**Seguridad**	**Configuración**	**Rendimiento**	**Usuarios**
Tivoli Netview	X		X	X	
CiscoWorks	X		X		
Flowscan				X	
Active Directory		X			X
SunScreen		X			

Tabla 18: Herramientas usadas en las diferentes áreas de administración

Si bien todas las herramientas descritas anteriormente, se usan en la ESPOL, las configuraciones e implementación se harán sobre Tivoli Netview, CiscoWorks y Flowscan.

6.2 Características de las Herramientas de Administración con que cuenta la ESPOL

Si bien se mencionó cinco herramientas usadas para la administración de la red de la ESPOL, solo se tratarán tres. Las herramientas usadas principalmente para la administración, son: Tivoli Netview, CiscoWorks y Flowscan.

6.2.1 Tivoli Netview

La ESPOL posee licencia de Tivoli Netview 5.1.1. Según la documentación del producto, “Tivoli Netview es una herramienta de administración de redes TCP/IP para dispositivos de comunicaciones de diferentes marcas. El programa Tivoli Netview provee manejo de la configuración, fallas, rendimiento, y otras características que lo hacen fácil de usar e instalar”.

Para el manejo de fallas incluye funciones que ayudan a identificar y recuperar a la red de los problemas rápidamente. Las funcionalidades que incluye son:

- Monitoreo del mapa de la red para detectar problemas
- Monitoreo de eventos para detectar problemas
- Localización de los problemas en la red
- Resolución de problemas en la red

Estas funcionalidades pueden verse en las siguientes figuras:

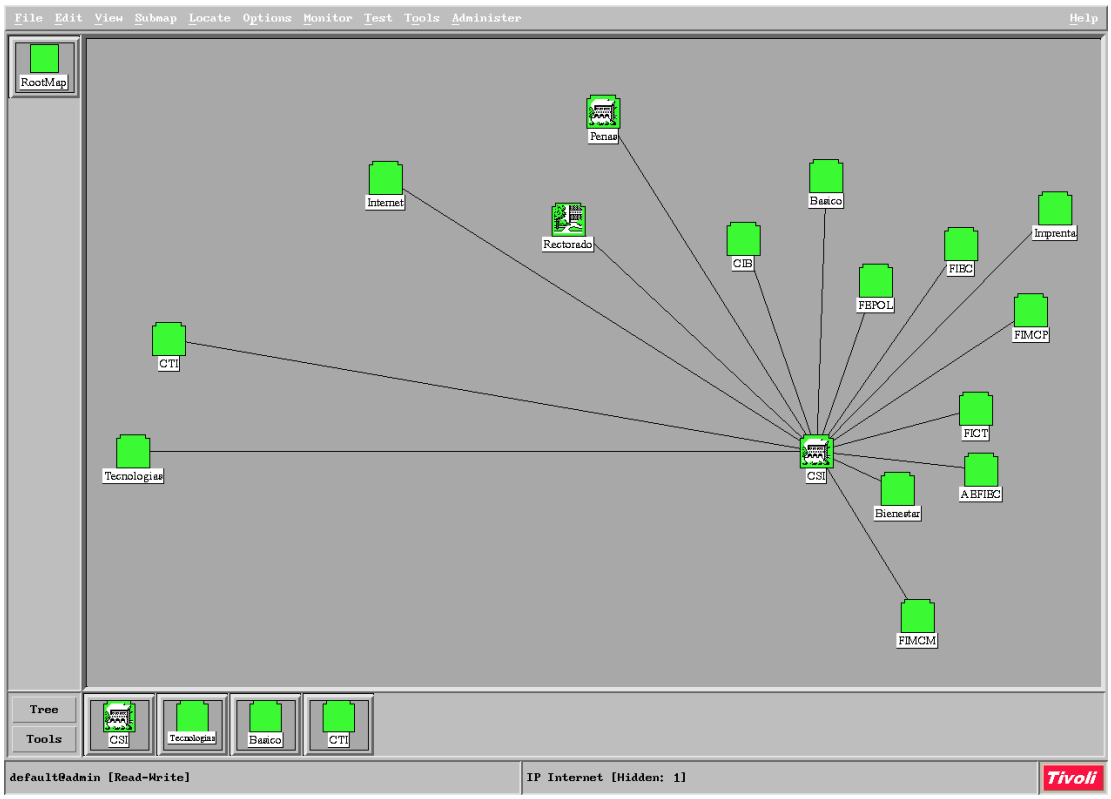


Figura 49: Interfaz gráfica principal de Tivoli Netview

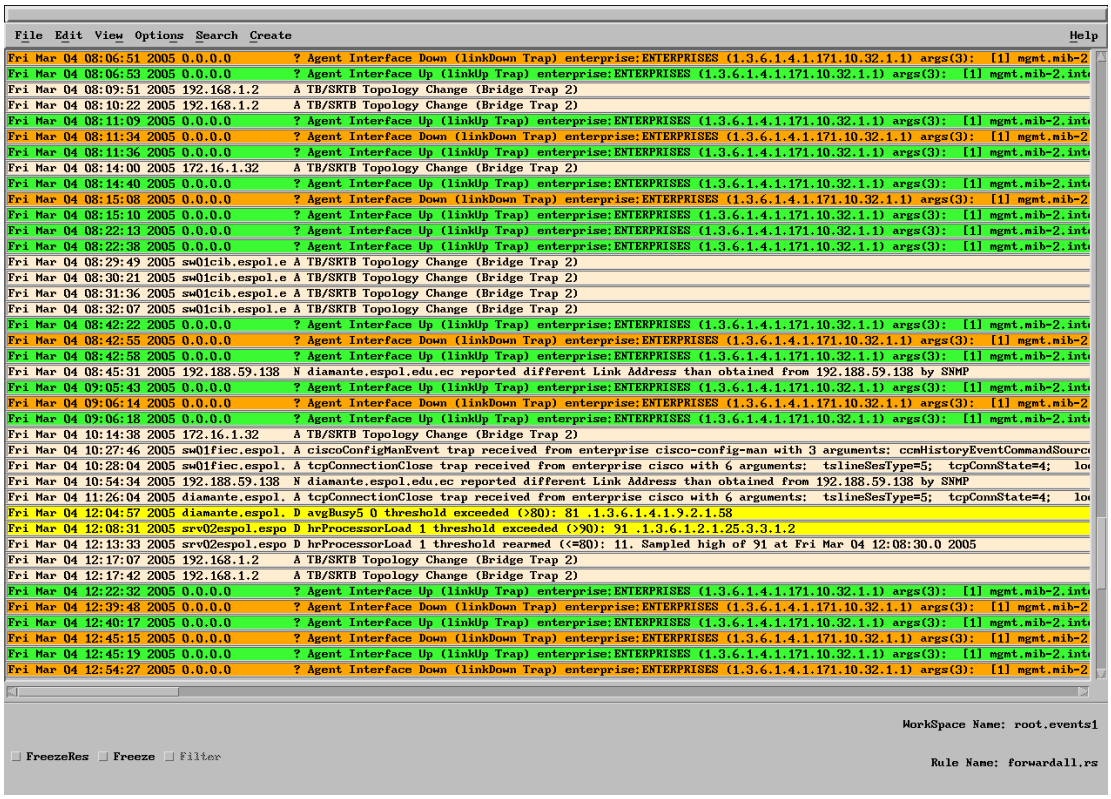


Figura 50: Interfaz Gráfica para el manejo de Eventos

Para le manejo del rendimiento, incluye funciones para monitorear cualquier variable MIB de cualquier dispositivo administrado. Estas funcionalidades son:

- Compilar MIBs
- Recolectar información de variables del MIB
- Establecer umbrales para variables del MIB

Estas funcionalidades, pueden verse en las siguientes figuras:

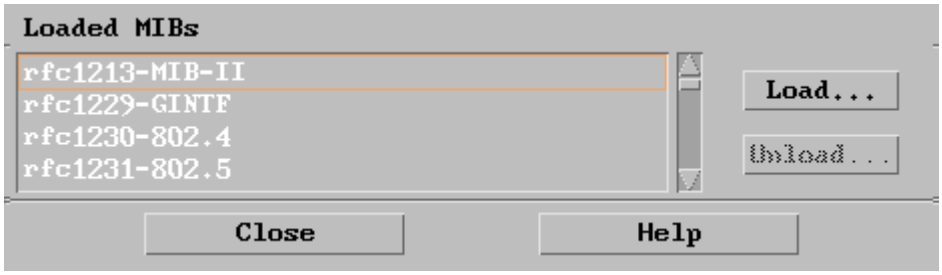


Figura 51: Interfaz Gráfica para compilar MIBs

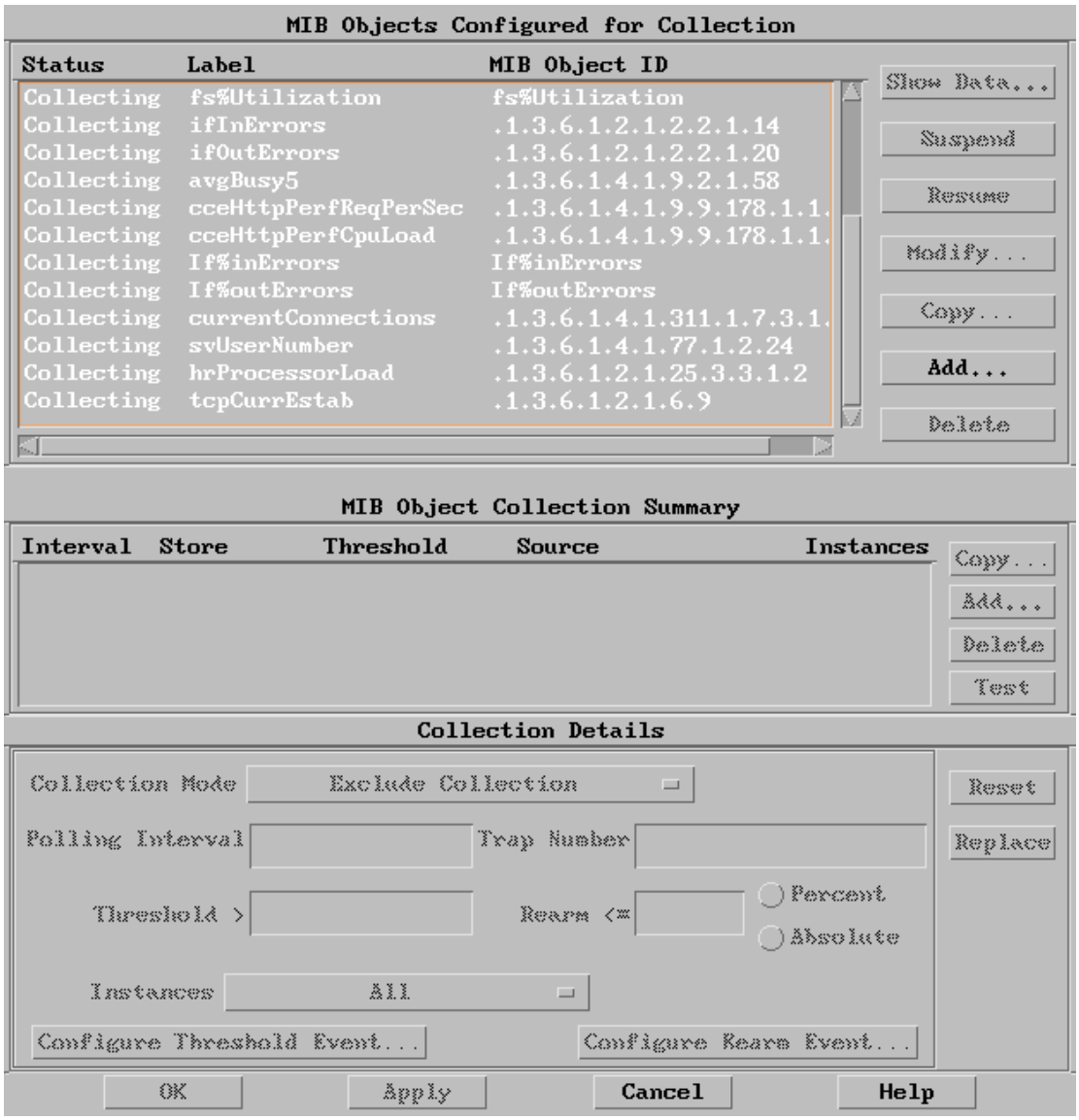


Figura 52: Interfaz Gráfica para recolectar variables dentro del MIB

Para el manejo de la configuración, incluye funciones que permiten recolectar información de los dispositivos administrados, tales como configuración ip, información de contacto, ubicación, tipo de dispositivo.

También muestra la configuración de la red a nivel de capa 3, ver Figura 49.

Figura 53: Interfaz Gráfica que muestra la descripción de un dispositivo administrado

Si bien Tivoli Netview tiene muchas ventajas, la versión instalada en la ESPOL solo soporta recolección de variables SNMPv1.

6.2.2 CiscoWorks

La ESPOL posee licencia de CiscoWorks 2000 LAN Management versión 2.2. Este software de Cisco, comprende varias aplicaciones que trabajan de manera conjunta para administrar redes que tengan equipos cisco. Las aplicaciones que tiene, según el sitio web de Cisco[12], son:

CiscoWorks Management Server.- Provee servicio de administración común a todas las aplicaciones de la familia CiscoWorks. También provee integración con herramientas Cisco y de terceros.

CiscoView.- Es una herramienta Web que provee, de una manera gráfica, el estado actual de los dispositivos Cisco. Esta herramienta también puede mostrar información de uso de las interfaces y las funciones de configuración de acceso.

Campus Manager.- Es un conjunto de aplicaciones basadas en el Web diseñada para administrar redes conformada por switches Cisco. Incluye descubrimiento de conectividad de capa 2, vista de topologías detalladas, configuración de ATM y VLAN/LANE[13], seguimiento de estaciones finales, herramientas de análisis de caminos para capa 2/3, e información de usuarios de teléfonos IP.

[12] http://www.cisco.com/en/US/products/sw/cscowork/ps2425/products_data_sheet09186a00800a9e97.html
[13] VLAN Redes Virtuales, LANE Emulación LAN para redes ATM

Device Fault Manager.- Provee análisis de falla para dispositivos Cisco. Genera “Traps inteligentes Cisco” a través de una colección de información y de técnicas de análisis. Estos traps pueden ser mostrados localmente, envíados por correo electrónico o a otros sistemas de manejo de eventos.

Real-Time Monitoring.- Es una herramienta web, multiusuario para administrar el tráfico. Provee acceso a la información RMON para monitorear, solucionar problemas y mantener operativa la red. Esta aplicación reporta y analiza: dispositivos, enlaces y puertos gráficamente, a partir de la información RMON proporcionada por los dispositivos Cisco que tienen esta funcionalidad.

Resource Management Essentials.- Provee las herramientas necesaria para manejar equipos Cisco. Incluye inventario, manejo de cambios, configuración de red y administración de imagen de software, disponibilidad y análisis syslog[14].

Si bien CiscoWorks tiene muchas ventajas, no es un administrador genérico como los es Tivoli Netview, ya que solo permite administrar dispositivos de la marca Cisco.

[14] Recibe los eventos de los equipos Cisco.

6.2.3 Flowscan

Según el sitio web[15] “FlowScan analiza y reporta la información del protocolo IP que es suministrada por los routers. Consiste de programas y módulos perl, Flowscan une un motor de colección de flows (versión alterada de clfowd[16]), una base de datos (RRD[17]) y una herramienta de visualización. Flowscan produce gráficos que proveen una vista casi actual del tráfico de borde[18]”.

Flowscan analiza y reporta en base al formato Netflow[19](proporcionado por los routers Cisco) de la información recolectada por cflow de CAIDA. Flowscan examina la información y mantiene contadores que reflejan lo hallado. Los valores de los contadores son almacenados usando RRDTool, un sistema de base de datos para información de series de tiempo. Finalmente, Flowscan, usa las capacidades de RRDTool y otras herramientas para reportar la información flow procesada.

[15] http://www.caida.org/tools/utilities/flowscan/
[16] http://www.caida.org/tools/measurement/cflowd/
[17] http://www.caida.org/tools/utilities/rrdtool/
[18] Información que va desde una empresa a otra, por ejemplo desde el Campus Gustavo Galindo a Internet
[19] Para más información: http://www.cisco.com/en/US/tech/tk812/tech_brief0900aecd80173f71.html

6.3 Configuración de las Herramientas de Administración con que cuenta la ESPOL

Antes de empezar con la configuración de las diferentes herramientas, es necesario establecer lo siguiente:

- Se mostrarán direcciones ip y comunidad snmp, diferentes a las configuradas en los sistemas de administración.

- Las claves de administradores y diversos sistemas, no serán mostrados.

- No se explicarán los procesos de instalación de cada herramienta

Para la instalación de las diferentes herramientas de administración, será necesario que los componentes a administrar tengan configurado los siguientes parámetros:

- Comunidad SNMP, deberá quitarse la configuración de la comunidad “public” y configurarse solamente la comunidad a usar dentro del Campus Gustavo Galindo.

- Ubicación, ubicación física del equipo de la siguiente manera: Ubicación fisica, Departamento, Campus de la ESPOL, ESPOL.

- Contacto, información de la persona encargada del equipo, de la siguiente manera: Nombre Apellido, <dirección de correo electrónico>

- Envío de eventos, configurar los diferentes dispositivos para que envíen los eventos al sistema de administración.

Por seguridad, se recomienda deshabilitar la administración a través del web de los diferentes componentes de comunicaciones y que el acceso snmp sea solo permitido para los servidores de administración.

6.3.1 Configuración de Tivoli Netview 5.1

En la ESPOL, el software Tivoli Netview 5.1.1 corre en AIX 4.2.1 que está instalado en un IBM RS/6000 modelo E30, con 768MB de RAM y 12GB de disco duro.

Para la configuración de Tivoli Netview serán necesarios los siguientes pasos:

1. Configuración snmp en el servidor. Esta configuración se detalla en el Apéndice A.

2. Instalación de Tivoli Netview. La instalación se hará en base a los manuales de instalación que vienen con el producto.

3. Respaldo del sistema

4. Configuración de los diferentes componentes:

    a. Interfaz Gráfica: Incluirá la forma de presentación de la herramienta.

    b. Configuración SNMP de Tivoli Netview: Incluirá la configuración SNMP para monitoreo de los componentes de la red, y también incluirá las redes a monitorear.

    c. Carga de los diferentes MIBs soportados por los equipos de comunicaciones: Será necesario cargar todos los MIBs que soporten los diferentes equipos de comunicaciones.

    d. Reestructuración del Mapa de la Red: Como será estructurado el mapa de la red del Campus Gustavo Galindo.

    e. Configuración de recolección de las diferentes variables MIBs de los diferentes equipos: Todas las variables a recolectar de los dispositivos administrados.

    f. Configuración de seguridades de Tivoli Netview: Seguridad de la aplicación, que incluirá un usuario administrador y otro para acceso de lectura.

6.3.2 Configuración de Ciscoworks 2000

En la ESPOL, el software CiscoWorks 2000 versión 2.2 corre en Solaris 8 que está instalado en un SUN Blade 1500, con 1GB de RAM y 80GB de disco duro.

Para la configuración de CiscoWorks 2000 serán necesarios los siguientes pasos:

1. Configuración snmp en el servidor. Esta configuración se detalla en el Apéndice B.

2. Instalación de CiscoWorks 2000. La instalación se hará en base a los manuales de instalación que vienen con el producto.

3. Respaldo del sistema

4. Configuración de los diferentes componentes:

    a. Configuración de CiscoWorks Common Services: Configuración de usuario para acceso al sistema.

    b. Configuración de Campus Manager: Configuración de SNMP y dirección ip base, para monitoreo y descubrimiento de los dispositivos de la red.

c. Configuración de Resource Manager Essentials: Ingreso de los dispositivos para la administración de la configuración.

d. Configuración de Device Fault Manager: Ingreso de los dispositivos para la administración de fallas y envío de alertas a Tivoli Netview.

e. Integración de CiscoWorks y Netview: Integración de estas herramientas para poder ver los diferentes eventos e iconos de los dispositivos Cisco que administra CiscoWorks.

6.3.3 Configuración de FlowScan

En la ESPOL, el software FlowScan versión 1.006, corre en Linux Redhat 9 que está instalado en un servidor Dell 2550 con 1GB de RAM y 100GB de disco duro.

Para la configuración de FlowScan serán necesarios los siguientes pasos:

1. Instalación de cflowd: Este programa es el que recolecta los flows que envía el router. La instalación se la hará de acuerdo a las indicaciones del programa.

2. Configuración de cflowd: Configuración de puerto de acceso y comunidad snmp para recolectar la información del router.

3. Instalación de FlowScan: La instalación se la hará de acuerdo a las indicaciones del programa.

4. Configuración del router de borde: Comprende la configuración del router de borde para el envío de los flows al programa flowscan.

6.4 Implementación de los diferentes esquemas de los Modelos de Administración

Para empezar con la implementación de los diferentes modelos de administración, es necesario que los dispositivos administrables[20] de la red estén configurados para permitir que los diferentes sistemas administradores pueden monitorearlos. En la red del Campus Gustavo Galindo, existen diferentes dispositivos de comunicaciones y diferentes servidores con varios sistemas operativos, la configuración para cada uno se puede ver en el Apéndice D.

Como existen tres sistemas para administrar la red, cada uno realizará tareas específicas:

Tivoli Netview.- Este sistema mantendrá el estado de la red a nivel lógico, será el repositorio de eventos de todos los dispositivos y también guardará los datos de la recolección de las variables MIB.

CiscoWorks.- Este sistema mantendrá las configuraciones, rendimiento y manejo de fallas de los equipos Cisco. Todas las alertas generadas, deberán ser enviadas al sistema Tivoli Netview. Además todos los dispositivos Cisco deberán enviar sus registros a este servidor.

FlowScan.- Este sistema mantendrá el estado del enlace de Internet. Solo generará gráficas de uso de Internet.

La interacción de los sistemas de administración con los dispositivos de comunicaciones se puede ver en la siguiente figura.

[20] En la Figura 47 se detalla los dispositivos a administrar

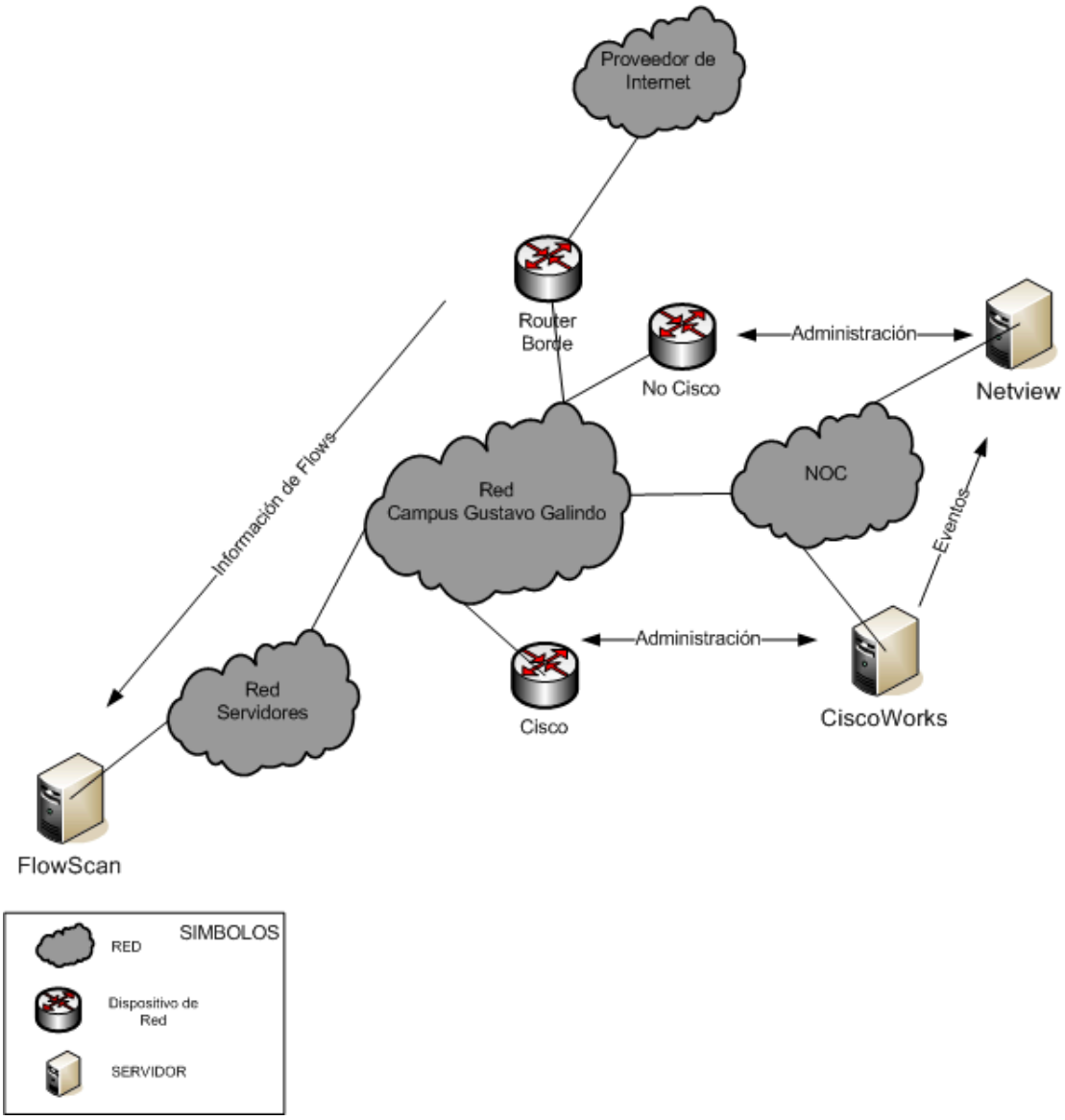


Figura 54: Esquema de Administración del Campus Gustavo Galindo

Los servidores Tivoli Netview y CiscoWorks, están en una red especial conocida como NOC[21]. Al no contar con un servidor específico para FlowScan, se ha usado un servidor que está en la red de servidores.

Como existen varios servidores para la administración, es necesario sincronizar los relojes para poder relacionar los diferentes eventos de la red, esta sincronización se la hace a través del protocolo NTP[22], y también se usa para sincronizar las estaciones de trabajo y los demás servidores.

6.4.1 Implementación del Modelo para el Manejo de Fallas

El esquema de fallas establece lo siguiente: El sistema administrador deberá recolectar los eventos cada 5 minutos, y debe incluir todos los dispositivos administrables (ver Figura 47).

En la siguiente tabla se muestran los equipos a administrar inicialmente, pero esto no excluye a equipos que las diferentes unidades del Campus Gustavo Galindo quieran que el CSI administre.

[21] NOC Centro de Operaciones de la Red (Network Operations Center)
[22] NTP Network Time Protocol, Servicio usado para sincronizar los relojes de los dispositivos de red

Tipo	Nombre	Características	Administrado por:
Switch	Sw01bck	Switch principal del Campus Gustavo Galindo	CiscoWorks
	Sw02bck	Switch principal de Tecnologías	
	Sw01adm	Switch de Administración Central	
	Sw01cib	Switch de Biblioteca de Ingenierías	
	Sw01cib	Switch de Biblioteca de Ingenierías	
	Sw01bas	Switch del Básico	
	Sw01fiec	Switch de la FIEC	
	Sw01fimcp	Switch de la FIMCP	
	Sw01fict	Switch de la FICT	
	Sw01fimcm	Switch de FIMCM	
	Sw01cti	Switch del CTI	
Router	Diamante[23]	Router de Internet	
Proxy	David	Proxy de Internet	
	Espartaco	Proxy de Internet	
Servidor de Acceso	Zafiro	Servidor de Acceso remoto para administración	Tivoli Netview
Servidor	Goliat	Servidor principal de Internet	
	Odisea	Servidor secundario de Internet	
	Sanson	Servidor Web CSI	
	Topacio	Servidor Web Académico	
	Cenaim	Servidor de Internet del Cenaim	
	Triton	Servidor Web principal	
	Cops	Servidor de Logs	
	Jade	Servidor principal de Seguridad	
	Ceemp	Servidor Web del CEEMP	
	Aefiec	Servidor Web de la AEFIEC	
	Fimcm	Servidor Web de la FIMCM	
	Base de Datos	Servidor Principal de Base de Datos	

Tabla 19: Dispositivos de Red y Sistema Administrador

En Tivoli Netview y CiscoWorks, debe configurarse para que examinen los dispositivos cada 5 minutos (ver Apéndice A y B). En Tivoli Netview además hay que configurar a los dispositivos de la tabla anterior como administrados. Los equipos y servidores deberán configurarse de acuerdo a lo descrito en el Apéndice D.

[23] Este dispositivo además, enviará la información de flows al servidor de Flowscan

En Tivoli Netview también se han creado colecciones para los diferentes servicios de los servidores que hay en la red. Estas colecciones permiten monitorear la disponibilidad de un servicio.

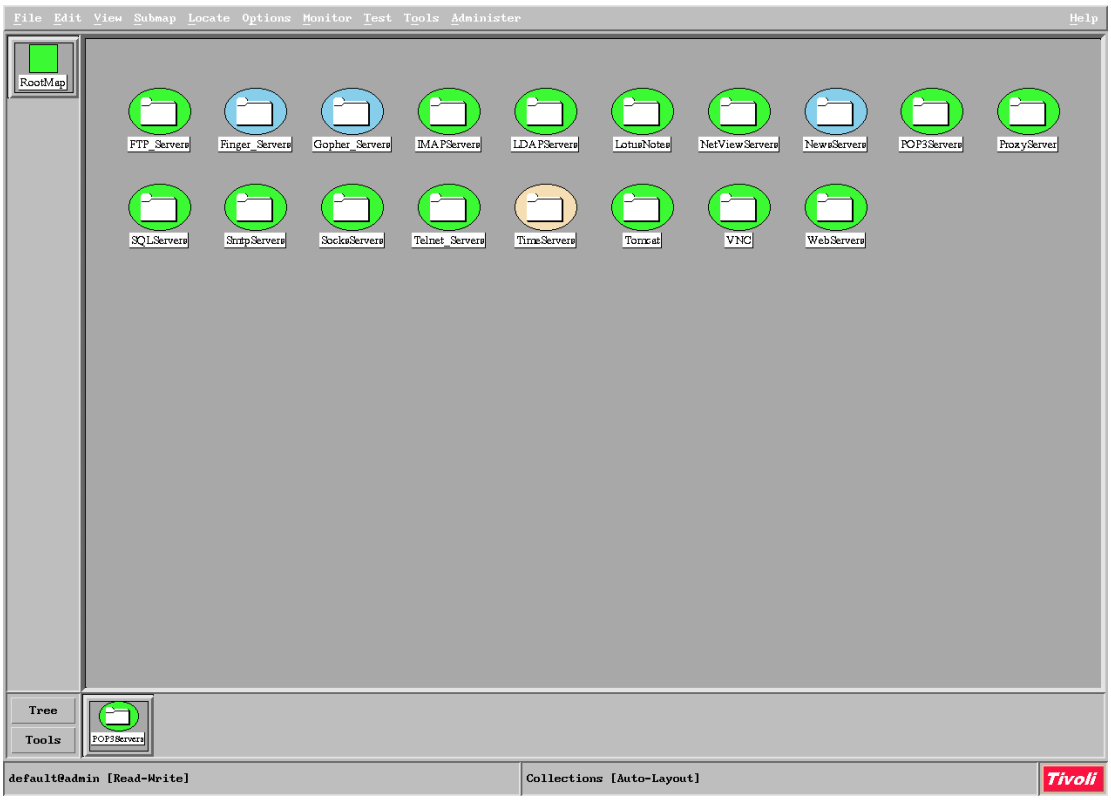


Figura 55: Gráfico de las colecciones para el Manejo de Fallas

6.4.2 Implementación del Modelo para el Manejo de la Configuración

Para el manejo de la configuración, se ingresará la información de todos los equipos de comunicaciones en la base de datos “Inventario de Equipos de Redes de la ESPOL” de Lotus que existe en el CSI, en

la cual los administradores locales tienen permiso para ingresar la información correspondiente a su unidad.

Para tener una idea clara de la configuración, será necesario cambiar el mapa de Tivoli Netview, para que sea por unidad, de la siguiente forma

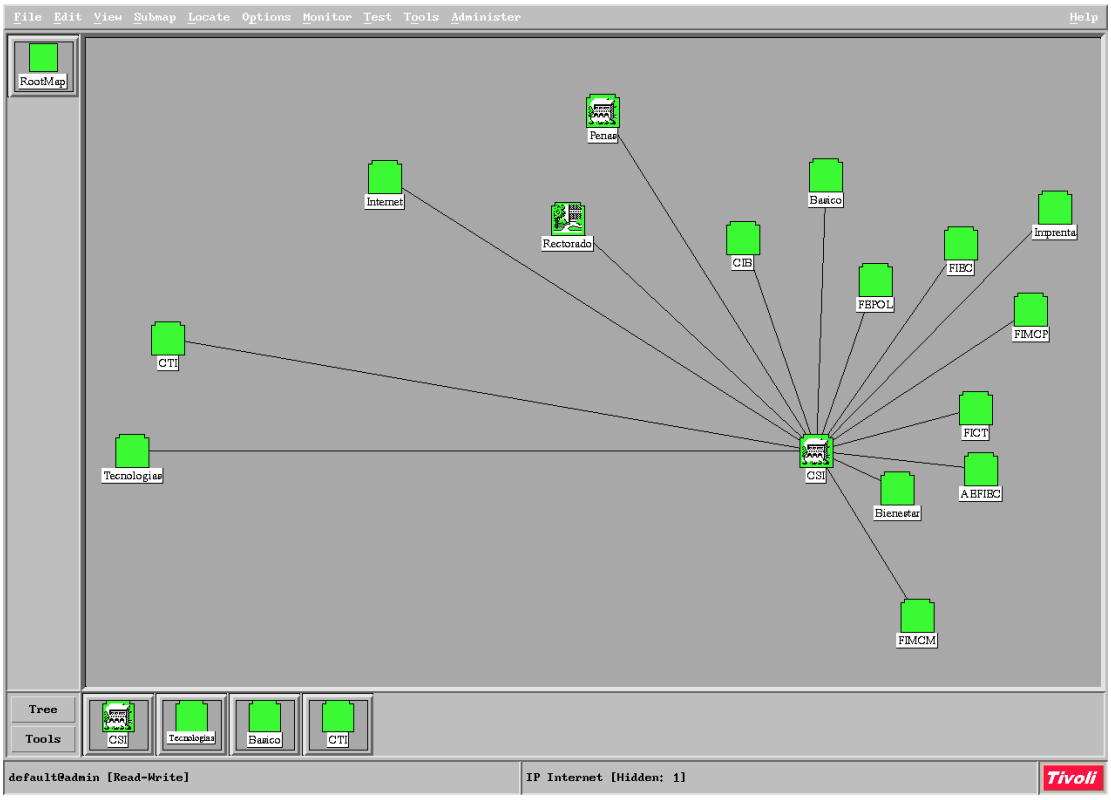


Figura 56: Configuración del Gráfico para el manejo de la Configuración

En CiscoWorks, hay que ingresar uno a uno todos los equipos Cisco, junto con la clave dentro de “Resource Manager Essentials->Administration->Inventory->Add Devices”. Esta aplicación permite tener toda la configuración de los equipos cisco, junto con las modificaciones hechas.

6.4.3 Implementación del Modelo para el Manejo del Rendimiento

Para el manejo del rendimiento, se ha considera monitorear las siguientes variables mib de los dispositivos:

Equipo	Variable MIB		Características
Diamante Sw01bck Sw02bck Sw01adm Sw01cib Sw01bas Sw01fiec Sw01fimcp Sw01fict Sw01fimcm Sw01cti	ifInOctects	.1.3.6.1.2.1.2.2.1.10	Tráfico de entrada de la interfaz
	ifOutOctects	1.3.6.1.2.1.2.2.1.16	Tráfico de salida de la interfaz
	ifInErrors	.1.3.6.1.2.1.2.2.1.14	Errores de entrada de la interfaz
	ifOutErrors	.1.3.6.1.2.1.2.2.1.20	Errores de salida de la interfaz
	avgBusy5	.1.3.6.1.4.1.9.2.1.58	Porcentaje de uso del procesador
Srv01espol Srv02espol Srv03espol	hrProcessorLoad[24]	.1.3.6.1.2.1.25.3.3.1.2	Porcentaje de uso del procesador para servidores
Goliat Srv01espol Srv02espol Srv03espol Odisea Sanson	ifInOctects	.1.3.6.1.2.1.2.2.1.10	Tráfico de entrada de la interfaz
	ifOutOctects	1.3.6.1.2.1.2.2.1.16	Tráfico de salida de la interfaz
	ifInErrors	.1.3.6.1.2.1.2.2.1.14	Errores de entrada de la interfaz
	ifOutErrors	.1.3.6.1.2.1.2.2.1.20	Errores de salida de la interfaz
	fs%Utilization		Expresión MIB[25] para medir porcentaje de uso del disco
David Espartaco	cceHttpPerfCpuLoad	.1.3.6.1.4.1.9.9.178.1.1 .2.8	Porcentaje de uso del procesador de los proxies
	cceHttpPerfReqPerSec	.1.3.6.1.4.1.9.9.178.1.1 .2.2	Número de requerimientos para los proxies
Base de Datos Goliat Odisea Srv01espol Srv03espol Sanson	tcpCurrEstab	.1.3.6.1.2.1.6.9	Número de sesiones tcp

Tabla 20: Variables Recolectadas por Tivoli Netview para el Manejo del Rendimiento

Además CiscoWorks tiene un monitoreo predefinido para todos los equipos Cisco, este monitoreo incluye CPU, Memoria, Porcentaje de Uso de los puertos de comunicaciones. También tiene configurado umbrales de uso para esas variables, de acuerdo a los estándares de Cisco. Además hay que configurar CiscoWorks, para que envíe las

[24] El agente MIB de los servidores Linux no tienen una variable para el uso del procesador para la versión SNMPv1

alertas de traspaso de los umbrales de los dispositivos administrados a Tivoli Netview(ver Figura 54). Estas alertas pueden verse en la siguiente figura:

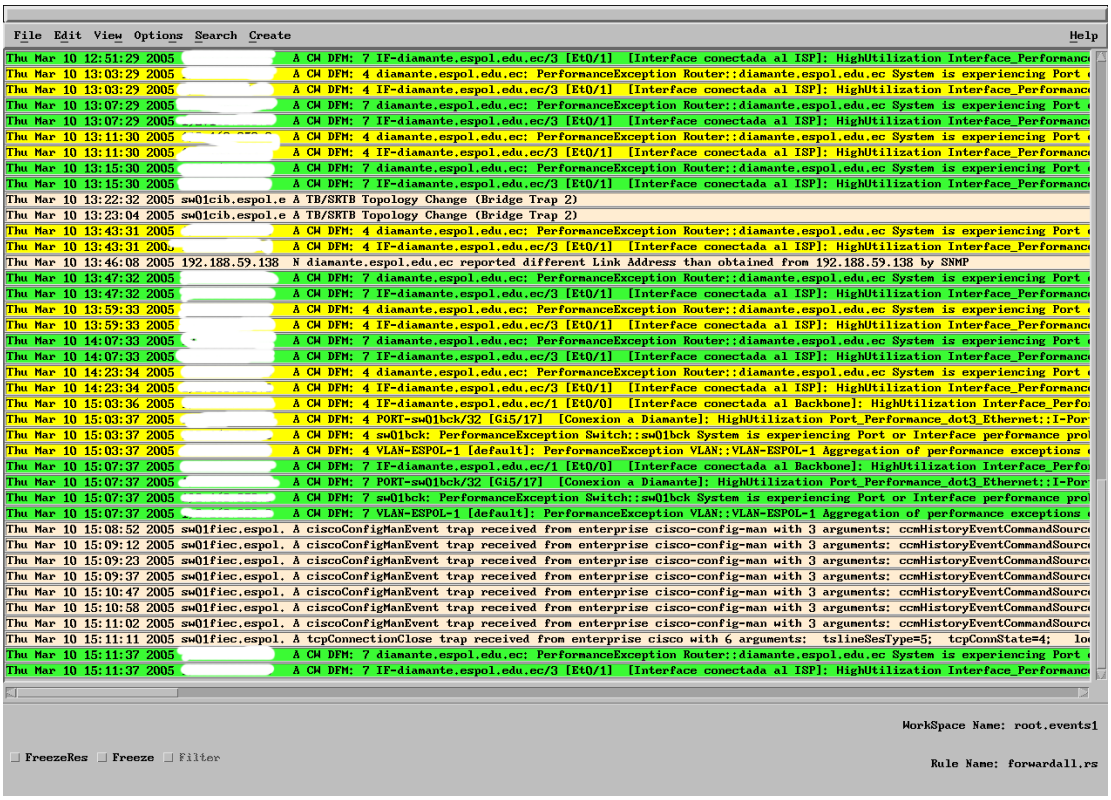


Figura 57: Envío de Eventos de CiscoWorks a Tivoli Netview

Con FlowScan, obtenemos el uso del enlace de Internet, por aplicación y por Red, este uso se lo puede observar en las siguientes figuras:

[25] El detale de esta expresión MIB puede verse en el Apéndice A

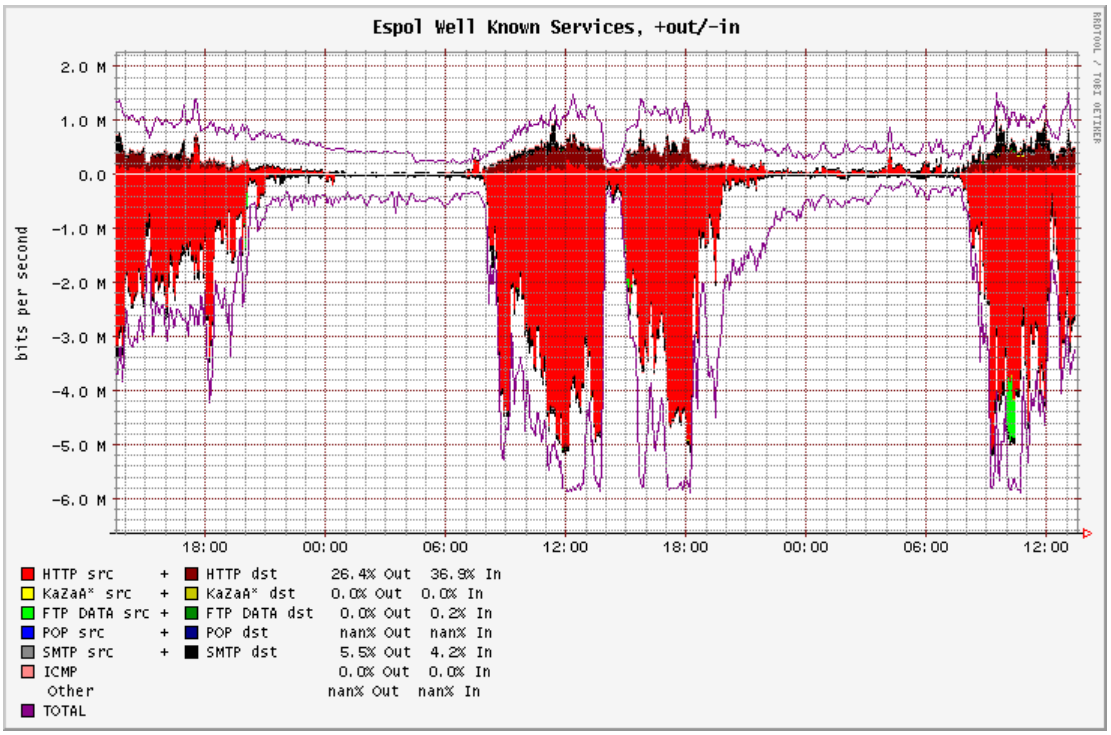


Figura 58: Gráfico de Uso de Internet por aplicación (FlowScan)

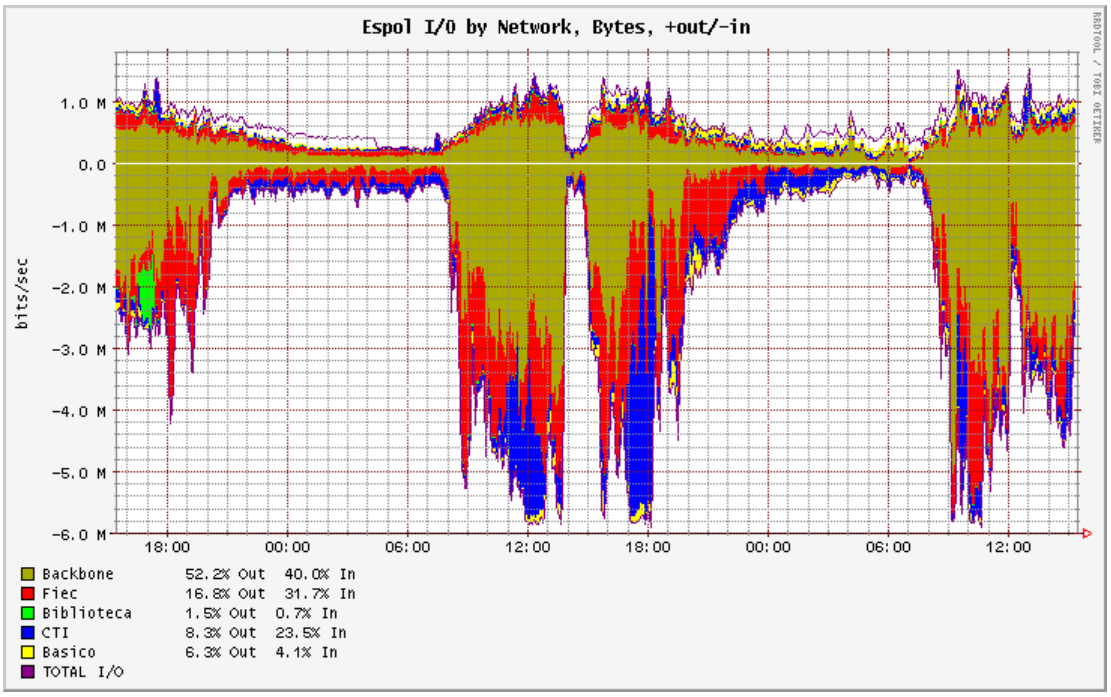


Figura 59: Gráfico de Uso de Internet por Red (FlowScan)

Para la generación de los gráficos anteriores, se puede ver al Apéndice C y E, que contiene la configuración y el código usado para generarlos.

CONCLUSIONES Y RECOMENDACIONES

El administrar una red local, compuesta de unos pocos dispositivos, es algo sencillo que no demanda sistemas especializados. Pero a medida que se unieron varias redes, para compartir información y servicios, esta tarea sencilla se complicó y entonces se produjo la necesidad de crear protocolos y sistemas para la administración de redes.

Las políticas propuestas deben ser ratificadas por las autoridades de la ESPOL, para que sean seguidas por todas las unidades. Además, estas políticas, deberán revisarse cada año para su actualización y apego a los objetivos de la ESPOL y a la tecnología existente.

Es necesario diferenciar la administración de redes, de la administración de sistemas. La función principal de la administración de redes es de garantizar la disponibilidad de los servicios de comunicaciones, mientras que la función de la administración de sistemas es la de garantizar la disponibilidad de las aplicaciones y datos empresariales. Estas dos funcionalidades se complementan.

Se recomienda que a futuro se considere tener toda la información de administración de redes en una base de datos común, para poder tener un registro más amplio de la información de administración y además poder

integrar nuevas aplicaciones de administración y de seguridad a las ya existentes.

Si bien Tivoli Netview versión 5.1 es usada en la administración de las redes del Campus Gustavo Galindo, esta versión no soporta completamente SNMPv2, lo cual impide recolectar información de variables que están bajo el árbol SNMPv2, como los agentes Linux, y los usos de los enlaces de alta velocidad que existen en el Campus Gustavo Galindo. Por este motivo se hace necesaria una actualización de la versión.

Las experiencias obtenidas en la administración de redes del Campus Gustavo Galindo, pueden extenderse a los demás Campus de la ESPOL y así mejorar la disponibilidad de las redes de éstos.

Para la administración de redes no basta con una solo herramienta de administración, cada situación es distinta y es necesario el uso de varias herramientas que se complementan, las cuales permiten manejar diferentes aspectos de la administración. Tal es así que muchos fabricantes crean complementos a herramientas existentes para administrar sus equipos de comunicaciones.

APÉNDICE A – CONFIGURACIÓN DE TIVOLI NETVIEW

Archivo de configuración de SNMP de Tivoli Netview (/usr/OV/conf(ovsnmp.conf). Las XXXXX representan la comunidad usada en la ESPOL

```
#
#  TME 10 NetView SNMP Configuration File
#  WARNING: DO NOT EDIT THIS FILE DIRECTLY
#
#  TME 10 NetView V5R1 SNMP Platform now uses an ndbm database
#  to maintain the snmp configuration to improve performance.  This
#  ovsnmp.conf(4) formatted file is maintained as a mirror image of the
#  snmp configuration database for backward compatibility of TME 10 NetView
#  based snmp applications running on the V5R1 SNMP Platform.
#
#  To modify entries in this file, use the SNMP Configuration commands
#  which are accessible through the X11/Motif graphical user interface
#  (OVW menu: Options->SNMP Configuration) or through the xnmsnmpconf
#  command line options.
#
#  See xnmsnmpconf(1) and ovsnmp.conf(4) for more information.
#
192.188.59.*:XXXXX:*:20:3:300:::::::1:1:1
200.10.148.*:XXXXX:*:20:3:300:::::::1:1:1
200.10.149.*:XXXXX:*:20:3:300:::::::1:1:1
200.10.150.*:XXXXX:*:20:3:300:::::::1:1:1
200.10.151.*:XXXXX:*:20:3:300:::::::1:1:1
200.9.176.*:XXXXXX:*:20:3:300:::::::1:1:1
172.16.*.*:XXXXXX:*:20:3:300:::::::1:1:1
192.168.*.*:XXXXXX:*:20:3:300:::::::1:1:1
*.*.*.*:public:*:20:3:300:::::::1:1:1
```

Archivo necesario para poder recolectar información de variables MIB (/usrOV/conf/mibExpr.conf)

```
#
# $Revision: 1.12 $ $Date: 92/08/10 12:54:02 $
#
# /usr/lib/OV/conf/mibExpr.conf: List of MIB expressions
#
# The form of this file is:
#                  name "description" expression
#
# The description may be separated into "lines" using the carriage-return
#   character (\015, ctrl-M) as the line separator.  50 characters per
#   line is the normal maximum.
#
# Expression may be any combination of values or operators in postfix
#   notation. "A / (B + C)" would be A B C + /
#
# Operators can be any of the following: + - * /
#
# Value must be a MIB variable or number.  MIB variables always
```

```
#  start with a '.'.  If a MIB variable ends in '.', an instance is
#  to be appended
#
# Name must be 13 characters or less, else this may fail on a short filename
#  system
#
# NOTE: Incorrect modifications to this file may cause data collection to
#       terminate and cause all expressions to fail
#
################################################################################
# This was event IPIE_EV      0058720261    Interface Percent Input Errors
If%inErrors \
"input errors/total packets received" \
                           .1.3.6.1.2.1.2.2.1.14. \
                           .1.3.6.1.2.1.2.2.1.11. \
                           .1.3.6.1.2.1.2.2.1.12. \
                           + / 100 *

# This was event IPOE_EV      0058720262    Interface Percent Output Errors
If%outErrors \
"output errors/total packets transmitted" \
                           .1.3.6.1.2.1.2.2.1.20. \
                           .1.3.6.1.2.1.2.2.1.17. \
                           .1.3.6.1.2.1.2.2.1.18. \
                           + / 100 *

Trapgend_Diskutil \
"% utilization, as reported by trapgend subagent,^M\
(blocks - bfree) / (blocks)" \
                           .1.3.6.1.4.1.2.6.4.4.2.1.2.   \
                           .1.3.6.1.4.1.2.6.4.4.2.1.3. - \
                           .1.3.6.1.4.1.2.6.4.4.2.1.2. / 100 *

If%EtherStat \
"% utilization ethernet segment, including errors" \
                           .1.3.6.1.2.1.16.1.1.1.4. \
                           8 * \
                           10000000 / \
                           100 *

If%Util \
"Percent of available banwidth utilized on a interface,^M\
computed by:^M\
    (Received byte rate + transmitted byte rate) * 8^M\
    ------------------------------------------------^M\
                  interface link speed^M\
then converted to a percentaje." \
                           .1.3.6.1.2.1.2.2.1.10. \
                           .1.3.6.1.2.1.2.2.1.16. \
                           + 8 * \
                           .1.3.6.1.2.1.2.2.1.5. \
                           / 100 *

If%Collisions \
"% collisions ethernet segment" \
                           .1.3.6.1.2.1.16.1.1.1.5. \
                           .1.3.6.1.2.1.16.1.1.1.13. \
                           / 100 *

fs%Utilization \
"% Utilization of a Filesystem on Computers" \
                           .1.3.6.1.2.1.25.2.3.1.6. \
                           .1.3.6.1.2.1.25.2.3.1.5. \
                           / 100 *
```

Archivo para la recolección de variables MIBS de los diferentes dispositivos de red (/usr/OV/con/snmpCol.conf)

```
#
# snmpCollect configuration file
# created by xnmcollect on Wed Apr 06 11:21:55.3 2005
#       pid: 31888 uid:0 euid:0
#
# File format:
#       MIB <MIB object ID> <MIB alias name> <units> <data type> <S|R>
#               (S=suspend R=resume)
#
#       <C|X|W> <node name|IP range> <interval> <threshval> <resetVal>
#               <A|%|xA> <s|d> <specific trap #> <REGEXP|LIST|ALL> <instances>
#               (C=collect X=don't collect W=wildcard)
#               (A=absolute reset %=% reset x[A%]=no threshold)
#               (s=store data d=don't store data)
#               (REGEXP=as regular expression, LIST=as list, ALL=all)
#
###############################################
MIB Trapgend_Diskutil diskutil percent EXPRESSION S
W *.*.*.* 43200 97.000000 95.000000 A s 58720263 ALL -
MIB .1.3.6.1.2.1.2.2.1.11 ifInUcastPkts pkts/sec COUNTER S
W *.*.*.* 3600 10000.000000 50.000000 % s 58720263 ALL -
MIB .1.3.6.1.2.1.2.2.1.17 ifOutUcastPkt units/sec COUNTER S
W *.*.*.* 3600 10000.000000 50.000000 % s 58720263 ALL -
MIB .1.3.6.1.2.1.11.1 snmpInPkts units/sec COUNTER S
W *.*.*.* 3600 10.000000 70.000000 % s 58720263 LIST 0
MIB .1.3.6.1.2.1.2.2.1.10 ifInOctets units/sec COUNTER R
C diamante.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 1,2
C sw01bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,6,7,10,11,12,13
C sw01fiec.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw02bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,7,10,11,12,13
C sw01cti.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01adm.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01bas.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fimcp.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fict.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fimcm.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01cib.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 25,26
C goliat.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C triton.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
C srv02espol.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
C odisea.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C sanson.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C topacio.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
MIB .1.3.6.1.2.1.2.2.1.16 ifOutOctets units/sec COUNTER R
C diamante.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 1,2
C sw01bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,6,7,10,11,12,13
C sw01fiec.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw02bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,7,10,11,12,13
C sw01cti.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01adm.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01bas.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fimcp.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fict.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fimcm.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01cib.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 25,26
C goliat.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C triton.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
C srv02espol.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
C odisea.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C sanson.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C topacio.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
MIB If%EtherStat If%EtherStat percent EXPRESSION S
C 192.188.59.1 300 0.000000 0.000000 xA s 58720263 LIST 1
C 192.188.59.23 300 0.000000 0.000000 xA s 58720263 LIST 203,205,206,207,208,209,210
C 192.188.59.20 300 0.000000 0.000000 xA s 58720263 LIST 202,203
C 192.188.59.21 300 0.000000 0.000000 xA s 58720263 LIST 201,203
C 192.188.59.37 300 0.000000 0.000000 xA s 58720263 LIST 2,3,4,5,6
C 172.16.1.2 300 0.000000 0.000000 xA s 58720263 LIST 3
C 192.168.1.213 300 0.000000 0.000000 xA s 58720263 LIST 3
C 200.10.148.10 300 0.000000 0.000000 xA s 58720263 ALL -
C 200.10.148.11 300 0.000000 0.000000 xA s 58720263 ALL -
MIB fs%Utilization fs%Utilization units EXPRESSION R
```

```
C goliat.espol.edu.ec 3600 80.000000 75.000000 % s 58720263 LIST 1,7,8
C srv02espol.espol.edu.ec 3600 80.000000 75.000000 % s 58720263 LIST 2,3
C triton.espol.edu.ec 3600 80.000000 75.000000 % s 58720263 LIST 2,3,4
C odisea.espol.edu.ec 3600 80.000000 75.000000 % s 58720263 LIST 1,6,7
C sanson.espol.edu.ec 3600 80.000000 75.000000 % s 58720263 LIST 1,5,6
C topacio.espol.edu.ec 3600 80.000000 75.000000 % s 58720263 LIST 2,4
MIB .1.3.6.1.2.1.2.2.1.14 ifInErrors units/sec COUNTER R
C diamante.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 1,2
C sw01bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,6,7,10,11,12,13
C sw01fiec.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw02bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,7,10,11,12,13
C sw01cti.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01adm.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01bas.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fimcp.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fict.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fimcm.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01cib.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 25,26
C goliat.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C triton.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
C srv02espol.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
C odisea.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C sanson.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C topacio.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
MIB .1.3.6.1.2.1.2.2.1.20 ifOutErrors units/sec COUNTER R
C diamante.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 1,2
C sw01bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,6,7,10,11,12,13
C sw01fiec.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw02bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,7,10,11,12,13
C sw01cti.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01adm.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01bas.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fimcp.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fict.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01fimcm.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 10101,10102
C sw01cib.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 25,26
C goliat.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C triton.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
C srv02espol.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
C odisea.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C sanson.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 2,3
C topacio.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 16777219
MIB .1.3.6.1.4.1.9.2.1.58 avgBusy5 units INTEGER R
C sw01bck.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw02bck.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw01cib.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw01adm.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw01bas.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw01fiec.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw01fimcp.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw01fict.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw01fimcm.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C sw01cti.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
C diamante.espol.edu.ec 300 75.000000 60.000000 A s 58720263 LIST 0
MIB .1.3.6.1.4.1.9.9.178.1.1.2.2 cceHttpPerfReqPerSec units GAUGE R
C david.espol.edu.ec 300 240.000000 230.000000 A s 58720263 LIST 0
C espartaco.espol.edu.ec 300 240.000000 230.000000 A s 58720263 LIST 0
MIB .1.3.6.1.4.1.9.9.178.1.1.2.8 cceHttpPerfCpuLoad units GAUGE R
C david.espol.edu.ec 300 80.000000 70.000000 A s 58720263 LIST 0
C espartaco.espol.edu.ec 300 80.000000 70.000000 A s 58720263 LIST 0
MIB If%inErrors If%inErrors units EXPRESSION R
C sw01bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,6,7,10,11,12,13
MIB If%outErrors If%outErrors units EXPRESSION R
C sw01bck.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 4,5,6,7,10,11,12,13
MIB .1.3.6.1.4.1.311.1.7.3.1.13 currentConnections units INTEGER R
C topacio.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C triton.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
MIB .1.3.6.1.4.1.77.1.2.24 svUserNumber units INTEGER R
C srv02espol.espol.edu.ec 43200 0.000000 0.000000 xA s 58720263 LIST 0
MIB .1.3.6.1.2.1.25.3.3.1.2 hrProcessorLoad units INTEGER R
C srv02espol.espol.edu.ec 300 90.000000 80.000000 A s 58720263 ALL -
C triton.espol.edu.ec 300 90.000000 80.000000 A s 58720263 ALL -
```

```
C topacio.espol.edu.ec 300 90.000000 80.000000 A s 58720263 ALL -
MIB .1.3.6.1.2.1.6.9 tcpCurrEstab units GAUGE R
C 192.168.254.25 300 0.000000 0.000000 xA s 58720263 LIST 0
C goliat.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C topacio.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C triton.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C sanson.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C odisea.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
MIB .1.3.6.1.2.1.1.3 sysUpTime units TIMETICKS R
C diamante.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01bck.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw02bck.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01adm.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01cib.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01bas.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01fiec.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01fimcp.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01fict.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01fimcm.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
C sw01cti.espol.edu.ec 3600 0.000000 0.000000 xA s 58720263 LIST 0
MIB .1.3.6.1.4.1.9.9.13.1.3.1.3 ciscoEnvMonTemperatureStatusValue units GAUGE R
C sw01bck.espol.edu.ec 1800 70.000000 50.000000 A s 58720263 LIST 1
C sw02bck.espol.edu.ec 1800 70.000000 50.000000 A s 58720263 LIST 1
MIB .1.3.6.1.2.1.25.1.6 hrSystemProcesses units GAUGE R
C goliat.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C triton.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C topacio.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C sanson.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C odisea.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
C srv02espol.espol.edu.ec 300 0.000000 0.000000 xA s 58720263 LIST 0
```

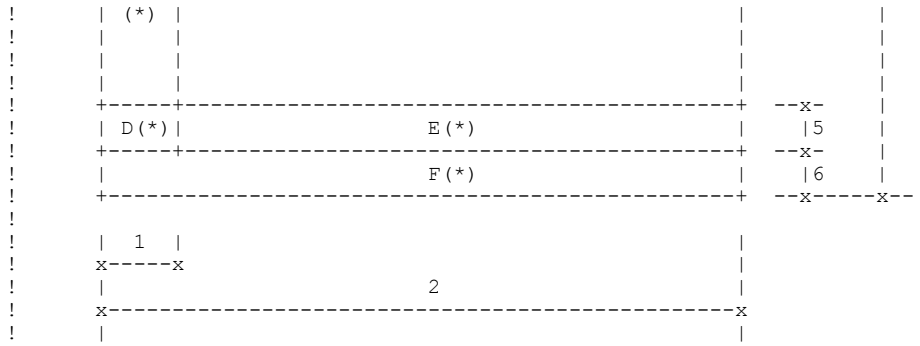
Archivo de configuración de la interfaz gráfica (/usr/OV/app-defauls/OVw)

```
!############################################################################
!
!   Module Name: OVw
!
!   Description: Defines the defaults resources associated with the
!                product's user interface window server. It includes the
!                original OVW server and the extensions. The resources
!                are divided and described in two groups. The first one
!                related to the original OVW server and the second one
!                related to the extensions.
!
!   Status Information:
!      Version:    $Revision$ - $Date$
!      Compiler Options:
!      Makefile:
!      Library:
!
!   Includes:
!
!   Filename: $File$
!
!   Change Activity:
!    $Log$
!
! Licensed Program Product: TME 10 NetView V5R1
!
!     "(C) COPYRIGHT International Business Machines Corp. 1992,1994",
!     "(C) COPYRIGHT Hewlett-Packard Co. 1992",
!     "    All Rights Reserved",
! US Government Users Restricted Rights - Use, duplication or
```

```
! disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
!
!#############################################################################
!
!

!
!*****************************************************************************
!
!   OVW  - defaults resources associated with the original OVW server.
!
!*****************************************************************************
!

!
! Colors for the symbols and lines connecting those
! symbols on various maps you see in your product(OVw).
!

OVw*unknownStatusColor:                 SkyBlue
OVw*downStatusColor:                    red
OVw*marginalStatusColor:                yellow
OVw*upStatusColor:                      green
OVw*unmanagedStatusColor:               wheat
OVw*acknowledgedStatusColor:            LimeGreen
OVw*user1StatusColor:                   #ffdc6969b4b4
OVw*user2StatusColor:                   #94940000d3d3

OVw*unknownStatusLineColor:             SkyBlue
OVw*downStatusLineColor:                red
OVw*marginalStatusLineColor:            yellow
OVw*upStatusLineColor:                  black
OVw*unmanagedStatusLineColor:           wheat
OVw*acknowledgedStatusLineColor:        LimeGreen
OVw*user1StatusLineColor:               black
OVw*user2StatusLineColor:               black

OVw*selectColor:                        grey49
OVw*mapBackground:                      grey66
OVw*mapWidth:                           600
OVw*mapHeight:                          500

*logoName:                              /usr/OV/bitmaps/C/hp.bitmap
*logoForeground:                        black
*logoBackground:                        grey66

!*********************************************************************
!
! Defines the time, in seconds, ovw waits for applications to call
! OVwAckMapClose() in response to the ovwMapClose event.
!
! If you set this value to a small value (such as 0) ovw will close
! the current map promptly, and applications may not have an
! opportunity to respond to the ovwMapClose event.
!
! The value of this resource should be greater than 30.
!
!*********************************************************************
*closeTimeout:                          120

*queryTimeout:                          120

OVw*keyboardFocusPolicy:                explicit
*availsnapsDialog*ok*labelString:       Close
*availmapsDialog*ok*labelString:        Close
*submapDialog*ok*labelString:           Close

!*********************************************************************
!
! To make the symbol label font sizes larger, simply comment out the
! *size20Font, *size10Font, and *smallFont lines.  This will force
! symbol labels to be displayed with the largest possible size (size30).
```

```
!
!*********************************************************************

*size30Font:    -monotype-times new roman-medium-r-normal--13-100-100-100-p-73-ibm-850
*size20Font:    -monotype-times new roman-medium-r-normal--13-100-100-100-p-73-ibm-850
*size10Font:    -monotype-times new roman-medium-r-normal--13-100-100-100-p-73-ibm-850
*smallFont:     -monotype-times new roman-medium-r-normal--11-80-100-100-p-57-ibm-850

! OVw*submapDialog*XmList*FontList:       -ibm--medium-r-medium--14-10-100-100-c-80-
ibm-850
OVw*submapDialog*XmList*FontList:       Rom10

!*********************************************************************
!
! Defines the map to be open when starting ovw.
! This resource should be set in the user resource file, not here, so
! that only his application is affected. When this resource is set, it
! overrides the option Set User Default Map of the EUI.
!
!*********************************************************************

/* OVw*mapName:                          default  */

!*********************************************************************
!
! Requests a file to be touched indicating ovw is ready to process user
! requests.
!
!*********************************************************************

/* OVw*initFile:                                  */

/* OVw*defaultRadius:                  15       */
/* OVw*maxRadius:                      25       */
/* OVw*readOnly:                       False    */
/* OVw*minimumSubmapBorder:            30       */
/* OVw*uiTest:                         False    */

!*********************************************************************
!
! Setting background Color to gray for items. (dialogs etc... ).
!
!*********************************************************************

OVw*background:                           gray

!*********************************************************************
!
! Resources to turn the labels Off/On.  When the radius of the symbol
! on the map is more than this value, labels will be displayed.
!
! ( Note :- These are approximate values. )
!
!*********************************************************************

*busSymbolRadius:                         8
*ringSymbolRadius:                        8
*connSymbolRadius:                        8
*meshSymbolRadius:                        8
*starSymbolRadius:                        8

!*********************************************************************
!
! Tired of clicking in OK button for Copyright information window ?
! Then change this to False.
!
!*********************************************************************

*displayCopyRightWindow:                False

!*********************************************************************
!
```

```
! If the *overrideAcknowledge resource is set to False, when
! an Acknowledged node comes up (changes to ovwNormalStatus), the
! Acknowledged status will *not* be replaced with the
! ovwNormalStatus.  The node will remain Acknowledged.
!
! If the *overrideAcknowledge resource is set to True, then if
! an Acknowledged node comes up (changes to ovwNormalStatus), then
! the Acknowledged status will be replaced with ovwNormalStatus.
!
! In other words, if you want Acknowledged nodes to have their
! Acknowledged status turned off when they come back up, set the
! resource to True.
!
!*********************************************************************

*overrideAcknowledge:                 True

!*********************************************************************
!
! There is a massive use of memory when a submap is displayed with
! background graphics and zoom presentation (because the entire
! virtual display has to be buffered, allowing user to navigate
! using sliders). The following resource is to control whether to
! prompt a warning dialog every time user makes a zoom in a submap
! with background graphics. Zoom ratios greater than this cause
! a warning dialog to prompt. If no warning is needed, set the
! resource to 10. Setting resource to 0 prompts the warning dialog
! for any zoom ratio.
!
!*********************************************************************

OVw*checkZoomAllocation:              10

!
!*************************************************************************
!
!   EUI  - defaults resources associated with the IBM extensions.
!          The resources are grouped in the classes listed bellow.
!
!*************************************************************************
!
!
! Default X resources for:
!
!     EUIShell
!     EUIToolShell
!     EUINaveTreeShell
!     EUIToolPalette
!     EUITool
!     EUIMainWindow
!     EUIViewArea
!     EUIControlDesk
!     EUINavigationTree
!     EUIScrolledBox
!     EUIApplShell
!     EUIApplTool
!     EUIFonts
!
!     and other EUI general use.
!


!              MainWindow with ViewArea geometry resources
!
!      +-------------------------------------------------+  --x-----x--
!      |                          A                      |    |3    |
!      +-----+-------------------------------------------+  --x-    |
!      |     |                                           |          |
!      |     |                                           |          |
!      |     |                                           |          |
!      |     |                                           |          |
!      |  C  |                  B                        |          |4
```



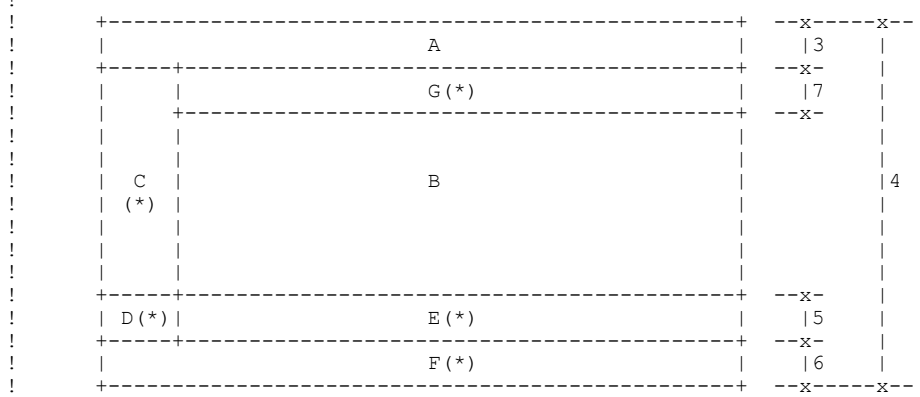
A - MenuBar D - Control
B - ViewArea E - ViewChildren
C - ViewStack F - MessageArea

(*) - these elements may be present or not, depending on the boolean value defined in the appropriate resource

The following size resources are defined in pixels
They can be changed as desired.

1 - scrolledBoxBoxWidth
2 - shellWidth
3 - this size is defined by Motif; it is dependable on the font choosen for the menu items.
4 - shellHeight
5 - scrolledBoxBoxHeight
6 - mainWindowMessageHeight

MainWindow with ControlDesk geometry resources



A - MenuBar E - ViewChildren
B - ViewArea F - MessageArea
C - ViewStack G - ControlDesk box
D - Control

(*) - these elements may be present or not, depending on the boolean value defined in the appropriate resource

The following size resources are defined in pixels
They can be changed as desired.

```
!      1 - scrolledBoxBoxWidth
!      2 - shellWidth
!      3 - this size is defined by Motif; it is dependable on the font
!          choosen for the menu items.
!      4 - shellHeight
!      5 - scrolledBoxBoxHeight
!      6 - mainWindowMessageHeight
!      7 - controlDeskBoxHeight
!
!
!            MainWindow, ToolPalette and NavTree layout when integrated
!
!
!             |                 1                   |
!             x-------------------------------------x
!             |                                     |
!
!   --x--     +-------------------------------------+-----------+    --x---
!     |       |                                     |           |      |
!     |       |                                     |           |      |
!     |       |                                     |           |      |
!     |       |                                     |           |      |
!     |       |                 A                   |           |      |
!   2 |       |                                     |           |      |
!     |       |                                     |           |      |
!     |       |                                     |     C     |      | 6
!     |       |                                     |           |      |
!   --x--     +-------------------------------------+           |      |
!     |       |                                     |           |      |
!     |       |                                     |           |      |
!   4 |       |                 B                   |           |      |
!     |       |                                     |           |      |
!     |       |                                     |           |      |
!   --x--     +-------------------------------------+-----------+    --x---
!
!             |                 3                   |     5     |
!             x-------------------------------------x-----------x
!             |                                     |           |
!
!      A - MainWindow
!      B - NavTree
!      C - ToolPalette
!
!      The following size resources are defined as a percentage of the
!      size of the shell in which the element is located.
!      They can be changed as desired.
!
!      1 - mainWindowWidth
!      2 - mainWindowHeight
!      3 - navTreeWidth
!      4 - navTreeHeight
!      5 - toolPaletteWidth
!      6 - toolPaletteHeight
!
!      Obs : the layout depicted above can also be modified through resources
!            in terms of relative position inside the shell. For example :
!            the ToolPalette can be positioned on the left side of the shell,
!            the NavTree can be at the top of the shell, and so on.
!            The resources that define these positioning are :
!            toolPaletteHReference, toolPaletteVReference,
!            navTreeHReference, navTreeVReference,
!            mainWindowHReference, mainWindowVReference.
!            Look at their explanation for valid values definition.
!

!-----------------------------------------------------------------------------
!
! EUIShell resources
! It refers to the top level windows that are created by EUI.
!
!-----------------------------------------------------------------------------
```

```
!
! Determines whether the EUI shell is created iconified or not
!
OVw*shellIconify:               False
!
! Defines the EUI shell x and y coordinates used in the creation. Not used
! when the creation is related with a drag/drop operation. If this resource
! is not set ( omitted ) the shells are open in cascade mode.
! The unit is number of pixels.
!
OVw*shellX:                     0
OVw*shellY:                     0
!
! Defines the EUI shell width and height to be used in the creation.
! The unit is number of pixels.
!
OVw*shellWidth:                 1268
OVw*shellHeight:                907

!-----------------------------------------------------------------------
!
! EUIToolShell resources
! It refers to the particular top level window created by EUI that
! houses the ToolPalette. The other resources are those defined in the
! EUIShell.
!
!-----------------------------------------------------------------------

!
! Determines whether the EUI ToolPalette shell is created iconified or not
!
OVw*toolShellIconify:           True
!
! Defines the EUI shell x and y coordinates used in the creation. Not used
! when the creation is related with a drag/drop operation. If this resource
! is not set ( omitted ) the mwm default is used (cascade).
! The unit is number of pixels.
!
OVw*toolShellX:                 721
OVw*toolShellY:                 0
!
! Defines the EUI shell width and height to be used in the creation.
! The unit is number of pixels.
!
OVw*toolShellWidth:             150
OVw*toolShellHeight:            1000

!-----------------------------------------------------------------------
!
! EUINavTreeShell resources
! It refers to the particular top level window created by EUI that
! houses the Navigation Tree (NavTree). The other resources are those
! defined in the EUIShell.
!
!-----------------------------------------------------------------------

!
! Determines whether the EUI NavTree shell is created iconified or not
!
OVw*navTreeShellIconify:        True
!
! Defines the EUI shell x and y coordinates used in the creation. Not used
! when the creation is related with a drag/drop operation. If this resource
! is not set ( omitted ) the mwm default is used (cascade).
! The unit is number of pixels.
!
OVw*navTreeShellX:              892
OVw*navTreeShellY:              0
!
! Defines the EUI shell width and height to be used in the creation.
! The unit is number of pixels.
!
```

```
OVw*navTreeShellWidth:          366
OVw*navTreeShellHeight:         400

!-------------------------------------------------------------------------
!
! EUIToolPalette resources
! It refers to the palette used to house boxes (icons) that represent
! the available tools
!
!-------------------------------------------------------------------------

!
! Defines the EUI ToolPalette color (background).
!
OVw*toolPaletteColor:           gray
!
! Determines whether the EUI ToolPalette is present or not
!
OVw*toolPalettePresent:         True
!
! Determines whether the EUI ToolPalette box (handle) is present or not
!
OVw*toolPaletteHasBox:          False
!
! Defines the EUI ToolPalette box color (handle).
!
OVw*toolPaletteBoxColor:        gray
!
! Determines whether the EUI ToolPalette is integrated (coupled to the
! EUI primary shell) when created or not
!
OVw*toolPaletteIntegrated:      False
!
! Defines the EUI ToolPalette shell width and height proportions used
! in the creation when they are integrated or used when the ToolPalette
! is dropped inside a shell that contains a MainWindow. The values are
! percentage of the shell width and height. These resources together with
! the References resources and integrated/box resources define the couple
! and uncouple behaviour of the ToolPalette.
!
OVw*toolPaletteWidth:           25
OVw*toolPaletteHeight:          100
!
! Defines the EUI ToolPalette horizontal and vertical location
! It is valid only when the ToolPalette is created integrated or if the
! ToolPalette has a box (handle).
! The valid values are :
!         Vertical Reference   ->   1 - top location       2 - down location
!         Horizontal Reference ->   3 - left location      4 - right location
!
OVw*toolPaletteVReference:      1
OVw*toolPaletteHReference:      4

!-------------------------------------------------------------------------
!
! EUINavigationTree resources
! It refers to the EUI object that enable the user to navigate throughout
! the view tree in a clean and easy way.
!
!-------------------------------------------------------------------------

!
! Defines the EUI NavTree background color
!
OVw*navTreeColor:               gray
!
! Determines whether the EUI NavTree is present or not
!
OVw*navTreePresent:             True
!
! Determines whether the EUI NavTree box (handle) is present or not
!
```

```
OVw*navTreeHasBox:              False
!
! Defines the EUI NavTree box (handle) color.
!
OVw*navTreeBoxColor:            gray
!
! Determines whether the EUI NavTree is integrated (coupled to the
! EUI primary shell) or not when created.
!
OVw*navTreeIntegrated:          False
!
! Defines the EUI NavTree shell width and height proportions used
! in the creation when they are integrated or used when the NavTree
! is dropped inside a shell that contains a MainWindow. The values are
! percentage of the shell width and height. These resources together with
! the References resources and integrated/box resources define the couple
! and uncouple behaviour of the NavTree.
!
OVw*navTreeWidth:               75
OVw*navTreeHeight:              30
!
! Defines the EUI NavTree horizontal and vertical location.
! It is valid only when the NavTree is created integrated or if the
! NavTree has a box (handle).
! The valid values are :
!         Vertical Reference   ->   1 - top location       2 - down location
!         Horizontal Reference ->   3 - left location      4 - right location
!
OVw*navTreeVReference:          2
OVw*navTreeHReference:          3
!
! Defines the Eui NavTree maximum depth (number of levels)
OVw*navTreeDepth:               100

!-----------------------------------------------------------------------------
!
! EUITool resources
! It refers to the boxes (icons) that represent the available tools inside
! the ToolPalette
!
!-----------------------------------------------------------------------------

!
! Defines the height of the box that represents a tool inside the ToolPalette.
! The width is determined by the width of the ToolPalette Shell.
! The unit is number of pixels.
!
OVw*toolBoxHeight:              95

!
! Defines the height of the label area of the box that represents a tool
! inside the ToolPalette.
! The unit is a percentage (0-100) of the total box height (toolBoxHeight)
!
OVw*toolBoxLabelHeight:         35

!-----------------------------------------------------------------------------
!
! EUIMainWindow resources
! It refers to the EUI object that may contain a menu, contains a ViewArea
! and/or a ControlDesk and may contain a MessageArea
!
!-----------------------------------------------------------------------------

!
! Defines the EUI MainWindow background color
!
OVw*mainWindowColor:            gray
!
! Defines the EUI MainWindow width (as a proportion of the shell width)
! and height (as a proportion of the shell height). These proportions
! are used when a ToolPalette and/or NavTree are dropped inside the shell
```

```
! that contains this MainWindow or if at least one of these elements are
! created integrated with the primary MainWindow.
!
OVw*mainWindowWidth:            75
OVw*mainWindowHeight:           70
!
! Defines the EUI MainWindow ControlDesk height used in the Creation.
! The unit is number of pixels.
!
OVw*mainWindowControlHeight:    450
!
! Determines whether the EUI MainWindow has menubar or not
!
OVw*mainWindowHasMenu:          True
!
! Determines whether the EUI MainWindow has message area or not
!
OVw*mainWindowHasMessage:       True
!
! Defines the EUI MainWindow message height
! The unit is number of pixels.
!
OVw*mainWindowMessageHeight:    40
!
! Defines the EUI MainWindow horizontal and vertical location
! It is valid only when the ToolPalette or NavTree are integrated to
! the MainWindow.
! The valid values are :
!         Vertical Reference   ->   1 - top location       2 - down location
!         Horizontal Reference ->   3 - left location      4 - right location
!
OVw*mainWindowVReference:       1
OVw*mainWindowHReference:       3

!
! Defines the EUI MainWindow menu background color
!
OVw*mainWindowMenuColor:        gray
!
! Defines the EUI MainWindow status1 and status2 color. The message area is
! composed of two separate parts: status1 and status2 message areas, used
! for displaying messages to the user. Different type of messages are displayed
! in each status.
!
OVw*mainWindowStatus1Color:     gray
OVw*mainWindowStatus2Color:     gray
!
! Defines the EUI MainWindow Message area frame color
!
OVw*mainWindowFrameColor:       gray

!-----------------------------------------------------------------------------
!
! EUIViewArea resources
! It refers to the EUI object where the views (Submaps) are displayed. It
! contains a ViewStack and ViewChildren which are scrolled windows elements that
! allow navigation throughout the view tree.
!
!-----------------------------------------------------------------------------

!
! Defines the EUI ViewArea background color
!
OVw*viewAreaColor:              gray
!
! Determines whether the EUI ViewArea has ViewStack and ViewChildren
! (areas that contain the representation of the view ancestors and first
! level descendants) or not
!
OVw*viewAreaHasScrolled:        True
!
! Determines whether the EUI ViewArea has Control (buttons that raises
```

```
! the ToolPalette and NavTree)
!
OVw*viewAreaHasControl:         True

!-----------------------------------------------------------------------------
!
! EUIControlDesk resources
! It refers to the EUI object that works as a repository of applications
! associated width the views. It contains a ApplCache and ApplPark which are
! scrolled windows elements that allow access to the applications inside
! the Control Desk.
!
!-----------------------------------------------------------------------------

!
! Defines the EUI ControlDesk background color
!
OVw*controlDeskColor:           gray
!
! Determines whether the EUI ControlDesk has box or not
!
OVw*controlDeskHasBox:          True
!
! Determines whether the EUI ControlDesk box has menu
!
OVw*controlDeskHasBoxMenu:      True
!
! Defines the EUI ControlDesk box color
!
OVw*controlDeskBoxColor:        gray
!
! Defines the EUI ControlDesk dialog color. This dialog is used in
! rename of the ControlDesk.
!
OVw*controlDeskDialogColor:     gray
!
! Defines the EUI ControlDesk box height
! The unit is number of pixels.
!
OVw*controlDeskBoxHeight:       20
!
! Determines whether the EUI ControlDesk has ApplCache and ApplPark (areas
! that contain representations of the applications inside the ControlDesk
! and the applications that were open from this ControlDesk)
!
OVw*controlDeskHasScrolled:     True
!
! Determines whether the EUI ControlDesk has Control (buttons that raises
! the ToolPalette and NavTree)
!
OVw*controlDeskHasControl:      False
!
! Determines the type of box used to represent the control desk in tool palette
! 0 - indicates picture
! 1 - indicates color
! 2 - indicates bitmap
!
OVw*controlDeskIconType:        0
!
! Determines the color used to represent the label of control desk in tool palette
!
OVw*controlDeskIconLabelColor:  black
!
! Determines the figure (picture/color/bitmap) used to represent the control desk
! in tool palette
!
OVw*controlDeskIconName:        /usr/OV/icons/gifs/control_desk.gif

!
! Determines the bitmap used to represent the control desk when the icon is
! beeing dragged
!
```

```
OVw*controlDeskDragIconName:            /usr/OV/icons/drag-bitmaps/control_desk.xbm

! Defines minimun width and height of the applications inside the control desk
! (if control desk is resized smaller than these values, scrollbars are mapped
! so that work window matches these values)
OVw*controlDeskApplMinWidth:    480
OVw*controlDeskApplMinHeight:   330


!----------------------------------------------------------------------------
!
! EUIScrolledBox resources
! It refers to common resources related to the ViewStack, ViewChildren,
! ApplCache and ApplPark.
!
!----------------------------------------------------------------------------

!
! Defines the width and height of the boxes located inside the ViewStack,
! ViewChildren, ApplCache and ApplPark
! The unit is number of pixels.
!
OVw*scrolledBoxBoxWidth:        80
OVw*scrolledBoxBoxHeight:       70

!
! Defines the height of the label area of the boxes located inside the
! ControlDesk boxes (ApplCache and ApplPark)
! The unit is a percentage of total height of the boxes (scrolledBoxBoxHeight)
!
OVw*scrolledBoxBoxLabelHeight:  35

!----------------------------------------------------------------------------
!
! EUIApplShell resources
! It refers to the EUI object that creates a EUI shell to house an application
!
!----------------------------------------------------------------------------

!
! Defines the EUI ApplShell label color
!
OVw*applShellLabelColor:        black
!
! Defines the EUI ApplShell icon color
!
OVw*applShellIconColor:         white

!----------------------------------------------------------------------------
!
! EUIApplTool resources
! It refers to the EUI object that creates a Tool box to house an application
!
!----------------------------------------------------------------------------

!
! Defines the EUI ApplTool label color
!
OVw*applToolLabelColor:         black
!
! Defines the EUI ApplTool icon color
!
OVw*applToolIconColor:          white

!----------------------------------------------------------------------------
!
! EUIViewNode  resources
! It refers to the EUI object that contains information related to a view.
! The ViewNode is the central element in terms of view navigation because
! it points to its parent node, so that it is always possible to access
! the whole tree from a given node
!
```

```
!-----------------------------------------------------------------------------

!
! Defines the width and height of the boxes that represent views inside
! the NavTree
! The unit is number of pixels.
!
OVw*viewNodeWidth:              60
OVw*viewNodeHeight:             60

!-----------------------------------------------------------------------------
!
! EUITree  resources
! It refers to the EUI object responsible to draw the view tree inside the
! NavTree.
!
!-----------------------------------------------------------------------------

!
! Defines the distance between boxes of the same hierarchical level
! inside the NavTree
! The unit is number of pixels.
!
OVw*treeSiblingSeparation:      10
!
! Defines the distance between hierarchical levels inside the NavTree
! The unit is number of pixels.
!
OVw*treeLevelSeparation:        20
!
! Defines the maximum number of levels supported by the NavTree
!
OVw*treeMaxLevel:               25
!
! Defines the color of the lines connecting the boxes inside the NavTree
!
OVw*treeLineColor:              black

!-----------------------------------------------------------------------------
!
! EUIFonts resources
! It refers to the fonts used by the EUI
!
!-----------------------------------------------------------------------------

!
! Defines the font of the tree button
!
! OVw*treeButtonFont:                -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*treeButtonFont:     Rom10
!
! Defines the font of the tools button
!
! OVw*toolsButtonFont:               -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*toolsButtonFont:    Rom10
!
! Defines the font of the status line 1
!
! OVw*statusLine1Font:               -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*statusLine1Font:    Rom10
!
! Defines the font of the status line 2
!
! OVw*statusLine2Font:               -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*statusLine2Font:    Rom10
!
! Defines the font of the boxes of the tool palette
!
OVw*toolPaletteBoxesFont:          -monotype-times new roman-medium-r-normal--13-100-
100-100-p-73-ibm-850
!
! Defines the font of the title and of the popup menu of the box of the
```

```
! control desk
!
! OVw*controlDeskTitleFont:          -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*controlDeskTitleFont:       Rom10
!
! Defines the font of the menuBar
!
! OVw*menuBarFont:                   -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*menuBarFont:      Rom10
!
! Defines the font of the popup menu of the boxes of the nav tree
!
! OVw*navTreeBoxesPopupMenuFont:     -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*navTreeBoxesPopupMenuFont:  Rom10
!
! Defines the font of the popup menu of the submap background
!
! OVw*submapPopupMenuFont:           -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*submapPopupMenuFont:        Rom10
!
! Defines the font of the popup menu of the symbol in the submap
!
! OVw*symbolPopupMenuFont:           -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*symbolPopupMenuFont:        Rom10
!
! Defines the font of the title of the application in the application box
! when the application is inside of the Control Desk
!
! OVw*applicationBoxFont:            -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*applicationBoxFont: Rom10
!
! Defines the font of the box in the application cache and in the application
! park in the Control Desk
!
OVw*applicationCacheAndParkFont:   -monotype-times new roman-medium-r-normal--13-100-
100-100-p-73-ibm-850
!
! Defines the font of the dialogs in the OVW
!
! OVw*FontList:                      -ibm--medium-r-medium--14-10-100-100-c-80-ibm-850
OVw*FontList:   Rom10

!-----------------------------------------------------------------------------
!
! EUI general use resources
! It refers to general commom resources shared by the various EUI objects
!
!-----------------------------------------------------------------------------

!
! Defines the application box height in the EUI
! The unit is number of pixels.
!
OVw*applicationBoxHeight:       20
!
! Determines whether the action of making a view (Submap) visible (by selecting or
! double clicking a box or symbol) should be done in the the shell where
! the action was performed (True) or in a new shell (False)
!
OVw*replaceView:                True
!
! Defines the shape of the cursor when a box is dragged
! See the file "cursorfont.h" for the valid options
!
OVw*dragCursorShape:            52
!
! Defines the color for various graphical elements of EUI
!
OVw*defaultColor:               gray
!
! Defines the color of the dialog that is poppedup when the user attempts
! to close a ControlDesk
```

```
!
OVw*dialogDisplayShellColor:    gray
!
! Defines the color for various labels of EUI
!
OVw*defaultLabelColor:          black
!
! Determines whether the internal tools (NavTool and CDTool) are to be
! created or not
!
OVw*thereAreInternalTools:      True
!
! Determines the color that represents displayed views in the Navigation
! Tree and ViewStack/ViewChildren areas
!
OVw*viewShownColor:             blue
!
! Determines the color that represents views that are opened but not
! displayed in the Navigation Tree and ViewStack/ViewChildren areas
!
OVw*viewNotShownColor:          gray

!
! Overriding keys for "Add Object : Palette" dialog, when adding objects
! to the maps.  Now inorder to add a subclass, click the middle mouse on
! the type of the subClass and drag it to the window, you want to copy.
!
! The following are the remaining translation keys.  Change these at your
! own RISK!!!
!
!      <Btn1Down>:                     Arm()\n\
!      <Btn1Motion>:                   RubberBand()\n\
!      <Btn1Up>:                       Disarm()\n\
!      Ctrl<Btn2Down>:                 MoveStart()\n\
!      Ctrl<Btn2Motion>:               Moving()\n\
!      Ctrl<Btn2Up>:                   MoveEnd()\n\
!      <Btn2Down>, <Btn2Motion>:       EuiMove()\n\
!      <Btn3Down>:                     Popup()\n\
!      <Key>osfSelect:                 ArmAndActivate()\n\
!      <Key>osfActivate:               ArmAndActivate()\n\
!      ~s ~m ~a <Key>Return:           ArmAndActivate()\n\
!      ~s ~m ~a <Key>space:            ArmAndActivate()
!

OVw*subClasses.translations: #override \n\
<Btn2Down>:      MoveStart()\n\
<Btn2Motion>:    Moving()   \n\
<Btn2Up>:        MoveEnd()

! this sets horizontal width for Help->Indexes->Applications
OVw*applications.text.columns:     35

! Resources for Background FileSelectionBox
OVw*backgroundFileDialog_popup.Width:   500
OVw*backgroundFileDialog_popup.Height:  700
OVw*backgroundFileDialog_popup.allowResize: True
OVw*backgroundFileDialog_popup*title:   Select Background Picture

! This resource is used to determine the default filter for background
! images. When setting the background for a map the background list will
! give an initial set of choices from the files in the directory specified
! here.

OVw*imageDirectory:     /usr/OV/backgrounds

!
! Defines the size of Attributes dialog
!
OVw*attributesWidth             : 680
OVw*attributesHeight            : 310

!-----------------------------------------------------------------------------
```

```
!
! Resources used to define the offsets between symbols in the Legend and
! Add Object dialogs, respectively.
!
! Warning: In order for these settings to work, they should be kept around
! the default values, which are:
!    OVw*legendXOffset:              75
!    OVw*legendYOffset:              75
!    OVw*addObjectXOffset:           70
!    OVw*addObjectYOffset:           70
!
!-----------------------------------------------------------------------------

OVw*legendXOffset:              75
OVw*legendYOffset:              75
OVw*addObjectXOffset:           70
OVw*addObjectYOffset:           70

!-----------------------------------------------------------------------------
!
! Resources for nmpolling's window
!
!-----------------------------------------------------------------------------
!
! Dialog labels and titles
!
nmpolling.titlebar:             Topology/Status Polling Configuration
nmpolling.masterontitle:        Enable Polling and Discovery Settings
nmpolling.statustitle:          Poll for Status
nmpolling.deleteintervaltitle:  Delete Nodes Down for
nmpolling.topolabel:            Topology Update
nmpolling.newnodetitle:         Discover New Nodes
nmpolling.newnodetogglelabel:   Use Auto-Adjusting Polling Interval
nmpolling.newnodelabel:         Fixed Polling Interval
nmpolling.configchecktitle:     Poll for Configuration Changes
nmpolling.configchecklabel:     Polling Interval
!
! Button labels
!
nmpolling.ok:                   OK
nmpolling.cancel:               Cancel
nmpolling.defaults:             Defaults
nmpolling.help:                 Help
!
!-----------------------------------------------------------------------------
!
! Resources for GIF generator program (ovw_webconfig) for web interface
!
!-----------------------------------------------------------------------------
*ovw_webconfig*overrideRedirect:true
```

Archivo de configuración gráfica de la Ventana de Eventos (/usr/OV/app-defaults/Nvela)

```
!
! defines maximum number of concurrent opened workspaces
!
nvela.maxNumWS             : 20
!
! defines maximum number of events to be loaded from ovevent.log
!
nvela.maxLoadEvents        : 500
```

```
!
! defines initial presentation style (card or list)
!
nvela.initialPresCard      : True
!
! defines if workspace name is located at right or left of the window
!
nvela.posRightName         : True
!
! directory used when saving the last active filters
!
nvela.profileDir           : $HOME
!
! directory used when reading filters for filling the filter control window
!
nvela.filterDir            : /usr/OV/filters/filter.samples
!
! directory used when creating reports
!
nvela.reportDir            : $HOME
!
! directory used when saving workspaces
!
nvela.saveDir              : $HOME
!
! size of nvela windows
!
nvela.widthMain       : 800
!
! size of the nvela windows
!
nvela.heightMain           : 500
!
! normal card color
!
nvela.cardColor            : #ffdcedf3d2d2
!
! color of the card when it is selected
!
nvela.cardColorSelect      : #ffdab9
!
! initial position of scroll bar in the card deck
!
nvela.scrollBarUp          : False
!
! defines the order of events presentation (increasing or decreasing)
!
nvela.normalPresent        : True
!
! defines number of cards to "card deck" appearance
!
nvela.numberFillCards      : 4
!
! controls double click interval in selecting items in the list
!
nvela.doubleClickInterval  : 350
!
! color used as background in nvela application
!
nvela*background           : gray
!
! type of icon used in the nvela (0: gif)
!
nvela.iconType             : 0
!
! color to be used when writing icon label
!
nvela.iconLabelColor       : black
!
! label for nvela windows
!
nvela.staticIconLabel      : History
```

```
!
! icon used in the nvela application shells
!
nvela.staticIconBackground : /usr/OV/icons/gifs/event_history.gif
!
! label for nvela windows
!
nvela.dynamicIconLabel     : History
!
! icon used in the nvela application shells
!
nvela.dynamicIconBackground : /usr/OV/icons/gifs/event_history.gif
!
! font list used in the nvela application
!
nvela*FontList             : Rom10
!
! font to be used by the Card widget to write normal texts
!
nvela*card*cardFontList    : Rom10
!
! font to be selected by the Card widget when writing small texts
!
nvela*card*smallFontList   : tnrR10
!
! defines if application starts up outside of the control desk
! valid when running integrated to OVw
!
nvela.outside              : False
!
! defines if an application should be managed by OVw.  If this is True
! then nvela will have a handle for dragging in and out of the control
! desk.  If this is False then nvela will come up with no knowledge of
! the control desk
!nvela.euiManaged           : True
!
! defines color to be used in the text written in the cards
!
nvela*card*cardTextColor   : black
!
! defines foreground color to be used in the events application
!
nvela*foreground           : black
!
! defines if new workspaces are opened outside the control desk
!
nvela.wsOutside            : False
!
! defines color to be used in the severity field in the cards (cleared)
!
nvela.clearedColor         : white
!
! defines color to be used in the severity field in the cards (warning)
!
nvela.warningColor         : yellow
!
! defines color to be used in the severity field in the cards (minor)
!
nvela.minorColor           : green
!
! defines color to be used in the severity field in the cards (critical)
!
nvela.criticalColor        : red
!
! defines color to be used in the severity field in the cards (major)
!
nvela.majorColor           : orange
!
! defines color to be used in the severity field in the cards (indeterminate)
!
nvela.indeterminateColor   : #ffdcedf3d2d2
!
```

```
! defines mode to view event in card mode when double click the card
!
nvela.workCardDetailMode : 0
!
! defines mode to view event in list mode when double click the event in list
!
nvela.workListDetailMode : 0
!
! defines the main workspace title
!
nvela.mainWorkspaceTitle : Events History Application Main Workspace
!
! defines the static workspaces title
!
nvela.staticWorkspaceTitle : Static History Workspace
```

Archivo para el descubrimiento de las redes del Campus Gustavo Galindo (/usr/OV/seedfile)

```
192.168.253.*
192.168.254.*
192.168.1.*
192.168.2.*
192.168.3.*
192.168.4.*
192.168.5.*
192.168.6.*
192.168.7.*
192.168.8.*
192.168.9.*
192.168.10.*
192.168.11.*
192.168.12.*
192.168.13.*
192.168.14.*
192.168.15.*
192.168.16.*
192.168.17.*
192.168.18.*
192.168.19.*
192.168.20.*
192.168.21.*
192.168.22.*
192.168.23.*
192.168.24.*
192.168.25.*
192.168.26.*
192.168.27.*
192.168.28.*
192.168.29.*
192.168.30.*
192.168.31.*
172.16.*.*
192.188.59.*
200.10.148.*
200.10.149.*
200.10.150.*
200.10.151.*
200.9.176.*
```

Archivo de configuración de todos los programas que componen Tivoli Netview (/usr/OV/ovsuf)

```
0:ovwdb:/usr/OV/bin/ovwdb:OVs_YES_START:nvsecd:-O,-n5000,-t:OVs_WELL_BEHAVED:15:::
0:nvsecd:/usr/OV/bin/nvsecd:OVs_NO_STOP::-O:OVs_WELL_BEHAVED:15:::
0:pmd:/usr/OV/bin/pmd:OVs_YES_START:nvsecd:-Au,-At,-Mu,-Mt,-m:OVs_WELL_BEHAVED::::
0:OVORS_M:/usr/OV/bin/orsd:OVs_YES_START:nvsecd,pmd::OVs_WELL_BEHAVED:120:::
0:trapd:/usr/OV/bin/trapd:OVs_YES_START:nvsecd,pmd::OVs_WELL_BEHAVED::::
0:ovtopmd:/usr/OV/bin/ovtopmd:OVs_YES_START:nvsecd,trapd,ovwdb:-O,-
t:OVs_WELL_BEHAVED:15:::
0:ems_sieve_agent:/usr/OV/bin/ovesmd:OVs_YES_START:nvsecd,pmd,ovtopmd::OVs_WELL_BEHAVE
D::::
0:ems_log_agent:/usr/OV/bin/ovelmd:OVs_YES_START:nvsecd,ems_sieve_agent::OVs_WELL_BEHA
VED::::
0:trapgend:/usr/OV/bin/trapgend:OVs_YES_START:nvsecd:-f:OVs_NON_WELL_BEHAVED:5:::
0:mgragentd:/usr/OV/bin/mgragentd:OVs_YES_START:nvsecd:-f:OVs_NON_WELL_BEHAVED:5:::
0:ovactiond:/usr/OV/bin/ovactiond:OVs_YES_START:nvsecd,trapd::OVs_WELL_BEHAVED::::
0:nvcorrd:/usr/OV/bin/nvcorrd:OVs_YES_START:nvsecd,trapd::OVs_WELL_BEHAVED::::
0:nvpagerd:/usr/OV/bin/nvpagerd:OVs_YES_START:nvsecd::OVs_WELL_BEHAVED:60:::
0:actionsvr:/usr/OV/bin/actionsvr:OVs_YES_START:nvsecd,nvcorrd::OVs_WELL_BEHAVED:60:::
0:nvserverd:/usr/OV/bin/nvserverd:OVs_YES_START:nvsecd,nvcorrd::OVs_WELL_BEHAVED:30:::
0:nvcold:/usr/OV/bin/nvcold:OVs_YES_START:nvsecd,ovwdb:-O:OVs_WELL_BEHAVED:15:::
0:snmpCollect:/usr/OV/bin/snmpCollect:OVs_YES_START:nvsecd,trapd,ovwdb,ovtopmd::OVs_WE
LL_BEHAVED:120:::
0:nvlockd:/usr/OV/bin/nvlockd:OVs_YES_START:nvsecd::OVs_WELL_BEHAVED:30:::
P:OVs_WELL_BEHAVED:15:::
0:netmon:/usr/OV/bin/netmon:OVs_YES_START:nvsecd,ovtopmd,trapd,ovwdb:-P,-
s/usr/OV/conf/seedfile:OVs_WELL_BEHAVED:15:::
```

APÉNDICE B – CONFIGURACIÓN DE CISCOWORKS

Archivo de configuración principal de CiscoWorks (/opt/CSCOpx/campus/etc/cwsi/ANIServer.properties)

```
#
# ANI Device Properties
#
#
# NonSnmpAccessible: List of the cdpCachePlatform which are not SNMP accessible,
# so ignore those devices, don't create any SMFContainer for them.
#
NonSnmpAccessible=Cisco IP Phone 7960:Cisco IP Phone 7940:Cisco IP Phone 7910:Cisco IP
Conference Station 7935:Cisco IP Phone 7905:C
isco IP Phone 7912:Cisco IP Phone 7902
root:/opt/CSCOpx/campus/etc/cwsi >more ANIServer.properties
# ANI Properties

#

Server.version=4.2
Server.copyright=Copyright (C) 1997-2002 Cisco Systems, Inc., All rights reserved
Server.build=0000
#

# Thread pool definitions.

#

# Thread pools contain the following parameters:

#    ThreadPool.<name>.priority  = {LOW | NORMAL | HIGH}; (required)

#    ThreadPool.<name>.count.min = minimum # of threads (default == count)

#    ThreadPool.<name>.count.max = maximum # of threads (default == count)

#    ThreadPool.<name>.count     = # of threads (default == 1; use this

#                                  property if count.min==count.max)

#    ThreadPool.<name>.timeout   = timeout in seconds.  If the current

#                                  thread count is greater than count.min,

#                                  idle threads will die after this number

#                                  of seconds.  (default == 300; not

#                                  applicable if count.min==count.max)

#

# There are three standard thread pools that must be defined:

#   background  - Used by default for scheduled processes

#   interactive - Used often for interactive commands

#   polling     - Used for low-level background polling
```

```
#
# Note that developers are free to define other thread pools.
#
# Note that a TimeBase can define itself to run in a certain thread
# pool by specifying "TimeBase.<name>.ThreadPool=<poolName>".  If not
# specified, the default pool "interactive" is used for Demand
# time bases, and "background" is used for all other time bases.
#
ThreadPool.interactive.priority=HIGH
ThreadPool.interactive.count=12
ThreadPool.background.priority=NORMAL
ThreadPool.background.count=48
ThreadPool.polling.priority=LOW
ThreadPool.polling.count=12
ThreadPool.vmpsadmin.priority=NORMAL
ThreadPool.vmpsadmin.count=12
# Default evaluation queue weight.
#
# Thread pool users are assigned this weight by default.  If more
# than one process is using the pool, then threads are assigned based
# on the weight of the processes.
#
# All processes get this weight by default.
#
# However, the weight of each timebase can be specified using the
# property "TimeBase.<name>.weight=<weight_number>".  Modify these
# values relative to the default weight below.
#
# Weights are only meaningful relative to other weights.  The ratio
# of processes using threads in a pool will equal the ratio of their
# respective weights.  If only one process is using the pool, then
# it gets all of the threads.
#
EvalQueue.defaultWeight=100
# Time Base definitions.  A time base entry consists of several
# parts:
#       "TimeBase"          as the prefix to the property key
#       <CategoryPrefix>    The time base name and the prefix
#                           used in category strings
#       time base           the actual time base (one of Periodic,
```

```
#                           Fixed, or Demand).
#
# A 'Fixed' schedule is a list of entries separated by ':'.
#
# Each entry contains the following five fields, separated
# by ';':
#   - Month specifier (0-11)
#   - Day of month specifier (1-31)
#   - Day of week specifier (0-6, 0 being Sunday)
#   - Hour specifier (0-23)
#   - Minute specifier(0-59)
#
# Each specifier can be a number, a range, a comma separated list
# of numbers and ranges, or an asterisk (meaning all legal values).
#
# Day and day-of-week are joined using "and".  So, to specify
# only a day-of-week, use "*" for the "day", and vice-versa.
#
# Each entry can be preceded by a '^' which causes the values
# denoted by the entry to be excluded from the schedule rather
# than included.
#
# The ':' separated entries are evaluated in the order specified,
# so that entries that come later are applied to the schedule
# composed from entries that come earlier.  The initial state of
# the schedule, before entries are processed, is empty.

TimeBase.Discovery.Type=Fixed
TimeBase.Discovery.Schedule=*;*;*;0;0:*;*;*;4;0:*;*;*;8;0:*;*;*;12;0:*;*;*;16;0:*;*;*;
20;0
TimeBase.VMPSMinor.Type=Periodic
TimeBase.VMPSMinor.Schedule=3600
TimeBase.VMPSMinor.ThreadPool=vmpsadmin
TimeBase.VMPSMajor.Type=Fixed
TimeBase.VMPSMajor.Schedule=*;*;*;10,14;0
TimeBase.VMPSMajor.ThreadPool=vmpsadmin
TimeBase.Modify.Type=Demand
TimeBase.VmpsModifyTftp.Type=Demand
TimeBase.VmpsModifyVmps.Type=Demand
TimeBase.VmpsReadVmps.Type=Demand
TimeBase.VmpsWriteVmps.Type=Demand
TimeBase.VmpsModifyPortStatus.Type=Demand
TimeBase.AcquireOnDemand.Type=Demand
TimeBase.PollDevicesAndLinks.Type=Periodic
TimeBase.PollDevicesAndLinks.Schedule=7200
TimeBase.PollDevicesAndLinks.ThreadPool=polling
TimeBase.VerifyDevice.Type=Demand
```

```
TimeBase.ImportFromUT.Type=Fixed
TimeBase.ImportFromUT.Schedule=^*;*;*;*;*
TimeBase.ExportToRME.Type=Fixed
TimeBase.ExportToRME.Schedule=^*;*;*;*;*
TimeBase.ImportFromRME.Type=Fixed
TimeBase.ImportFromRME.Schedule=^*;*;*;*;*
TimeBase.RTPoller.Type=Demand
TimeBase.RTPoller.ThreadPool=polling
# Name of class which performs the timebase function

TimeBaseImpl.Discovery=com.cisco.nm.ani.server.core.DiscoveryTimeBaseImpl
TimeBaseImpl.PollDevicesAndLinks=com.cisco.nm.ani.server.status.StatusPollTimeBaseImpl
TimeBaseImpl.VMPSMajor=com.cisco.nm.ani.server.vmpsadmin.VmpsMajorTimeBaseImpl
TimeBaseImpl.VMPSMinor=com.cisco.nm.ani.server.vmpsadmin.VmpsMinorTimeBaseImpl
#For the various time base stat collectors

TimeBaseStat.Discovery.StatClass=com.cisco.nm.ani.server.core.DiscoveryTimeBaseStat
TimeBaseStat.Discovery.MaxNumberOfStats=10
#Validate Null Column in DB

AniDBNullColumnValidation=false
#

# Service Module Token list

#

ServiceModuleList=Base:Snmp:Core:Corex:Topo:Vlad:Ccm:VmpsAdmin:Lane:LaneConfiguration:
Dcrp:Status:Path:Sdiv:Atm:Apps
ServiceModulePackage=com.cisco.nm.ani.server
#

# Service Module token to class map

#

ServiceModule.Base=com.cisco.nm.ani.server.framework.BaseServiceModule
ServiceModule.Snmp=com.cisco.nm.ani.server.snmp.SnmpServiceModule
ServiceModule.Core=com.cisco.nm.ani.server.core.CoreServiceModule
ServiceModule.Corex=com.cisco.nm.ani.server.corex.CorexServiceModule
ServiceModule.Topo=com.cisco.nm.ani.server.topo.TopoServiceModule
ServiceModule.Vlad=com.cisco.nm.ani.server.vlad.VladServiceModule
ServiceModule.Ccm=com.cisco.nm.ani.server.ccm.CcmServiceModule
ServiceModule.Apps=com.cisco.nm.ani.server.apps.AppsServiceModule
ServiceModule.Lane=com.cisco.nm.ani.server.lane.LaneServiceModule
ServiceModule.LaneConfiguration=com.cisco.nm.ani.server.laneconfiguration.LaneConfigur
ationServiceModule
ServiceModule.VmpsAdmin=com.cisco.nm.ani.server.vmpsadmin.VmpsAdminServiceModule
ServiceModule.Dcrp=com.cisco.nm.ani.server.dcrp.DcrpServiceModule
ServiceModule.Status=com.cisco.nm.ani.server.status.StatusServiceModule
ServiceModule.Path=com.cisco.nm.ani.server.path.PathServiceModule
ServiceModule.Sdiv=com.cisco.nm.ani.server.sdiv.SdivServiceModule
ServiceModule.Query=com.cisco.nm.ani.server.query.QueryServiceModule
ServiceModule.Atm=com.cisco.nm.ani.server.atm.AtmServiceModule
ServiceModule.Wbem=com.cisco.nm.cim.wbem.WbemServiceModule
ServiceModule.Servlet=com.cisco.nm.moxie.httpserver.servlet.ServletServiceModule
ServiceModule.Asset=com.cisco.nm.ani.server.asset.AssetServiceModule
#

# Configure nameserver properties.

#

# When usedns is set to false, ANI will not resolve ip address to name

# using DNS.

#

nameserver.usedns=true
#
```

```
# When resolveByName is true and if a host name was found, then DNS

# will be used to lookup a primary IP address for the host name.

#

nameserver.resolveByName=true
#

# When resolveBySysName is true and if a host name was not found, then

# DNS will be used to lookup an IP address and host name from sysName.

#

nameserver.resolveBySysName=true
#

# Default name by which this ANI registers itself in OsAgent

#

AniName=ANIServer4.2-srv06csi
#

# Lock port used by ANI to ensure it is the only ANI running

#

AniPort=14004
#

# Connect to the daemon manager

#

dmgtd.aniConnect=yes
#

# File name where messages are stored. The name must begin with a /

# and must be available in classpath.

messageFileName=/com/cisco/nm/ani/share/messages/ani.messages
#

# Environmental variable replacement

#

DEVICESROOT=/opt/CSCOpx/campus/lib/classpath
DEVICESPACKAGE=com.cisco.nm.ani.server.devices
DeviceConfigFile=devices.xml
DeviceAppConfigFile=anidevices.xml
#

# CIM_CX settings

#

CIM_CXDeclarationFile=CIM_CX.xml
CIM_CXMappingPackagePrefix=com.cisco.nm.cim
CIM_CXPackageFolders=cisco.asset:cisco.core:cisco.services:cisco.topo:dmtf.application
:dmtf.core:dmtf.device:dmtf.network:dmtf.physi
cal:dmtf.system:dmtf.user:dmtf.vlan
#

# Servlet Engine settings

#

ServletEngine.ConfigFile=AniServletConfig.xml
```

```
ServletEngine.Threads=10
#

# Discovery parameters

#

#Do not edit this property.It is used for application internal logic

DeviceDiscovery=enable
Discovery.router=on
Discovery.seed=192.188.59.130
#

# Location of the Mojo style mib info file

#

MibInfo=mibinfo.dat
#

# Location of the SNMP Community config file

#

CommunityFile=anisnmp.conf
#

# SNMP configuration

# The following are default values for snmp operations.

# Values specified in communities configuration file will override

# these default values for that selected community entry.

#

snmp.maxRows=50000
snmp.threads.min=15
snmp.threads.max=48
snmp.maxRetry=1
snmp.timeoutSecs=6
snmp.retryPolicy=com.cisco.nm.lib.snmp.lib.ExpRetryPolicy
snmp.defaultReadCmty=public
snmp.defaultWriteCmty=private
snmp.getBulkSize=10
snmp.encryptCommunity=true
snmp.EnableMultiple=true
# Flag to turn on/off SNMP version2c. Set it to "off" to disable v2c.

snmp.version2c=on
#

# LogMsg configuration

#

#LogMsg.logfile=ani.log

#LogMsg.trace=frontend:framework:core:corex:topo:snmp:vlad:lane:vmpsadmin:dcrp:status:
path:sdiv:query:atm:apps:ccm:devices:laneconfi
guration:asset:devices.C5K:devices.LS1010:devices.Router:devices.C2900XL:devices.C2800
:devices.C3900:devices.AppHost

LogMsg.state=disable
LogMsg.threadStamp=true
LogMsg.timeStamp=true
LogMsg.logFileSize=1000000
#

# Database connection Definitions
```

```
#

# This uses JConnect v4 through a shared instance of Open Server Gateway

#

DB.driver=com.sybase.jdbc.SybDriver
DB.url=jdbc:sybase:Tds:localhost:?SERVICENAME=aniDb
DB.dsn=ani
DB.Connection.Timeout=60000
DB.Connection.MinIdleTime=60000
DB.ReaperSleepTime=30000
#

# VTP operations

#

VTP.onTransparent=off
#

# Event Channel Definitions

#

#AniEvents.factory=

#AniEvents.channel=AniEventsChannel

AniEvents=on
AniEvents.MaxInitWaitTime=30
#

# Discrepancy Options

#

#

# Status polling

#

PollDevicesAndLinks.onEnabled=on
#

# Security support

#

# To enable security you need to set UserAuthentication=on

# and define <CRM web server>. AuthenticationServer defaults to localhost if not
defined

UserAuthentication=on
#AuthenticationServer=<CRM web server>

# Timeout for valid cached sessions; if zero then sessions are not cached.

AuthenticationSessionTimeout=300000
# Sync-Up with RME

SyncUp.AddFromRME=off
SyncUp.AddToRME=off
SyncUp.RMEHost=
SyncUp.RMEPort=
SyncUp.RMEUser=
SyncUp.RMEPassword=
#
```

```
# Path HSRP support

#

# To enable path hsrp feature you need to set Path.Hsrp=on

Path.Hsrp=off
#

# UT DHCP Support

# To enable UserTracking to adapt to DHCP envionment set UTIPChange=1

UTIPChange=0
# Set this option to "ON" , if UT Major acquisition is to be restricted to specific
subnets

UT.SubnetDiscovery=OFF
# User Tracking to query only for Vlans with User ports .

UTGetVlansWithUserPorts=1
UTLeaveCPU=false
# New properties added after ANI 3.x to ANI 4.2 upgrade

# User Tracking for Enable IP Phone discovery  for IOS switches

UTGetVlansWithUserPortsIOS=1
#When nameserver.useloopbackaddress is set to true, ANI will use LoopBack IP Address
as Prefered IP

nameserver.useloopbackaddress=false
Cisco.ApplicationName=ani
Discovery.domains.include=ESPOL
```

Archivo de Configuración de los parámetros SNMP de CiscoWorks (/opt/CSCOpx/campus/etc/cwsi/anisnmp.conf)

```
# Enter the community string using this syntax:
#       target:read_community::timeout:retries:::write_community
#
# You must be sure to keep the correct number of colons between entries.
# If you add or delete colons,  the ANI Server will NOT be able to properly read the
community strings.
#
# To verify that you are entering valid community strings, use the online help for
guidance.

*.*.*.*:XXXXX::::::XXXXX
```

APÉNDICE C – CONFIGURACIÓN DE FLOWSCAN

Archivo de configuración del programa CFLOWD que recolecta los flows (/etc/cflowd/cflowd.conf)

```
##############################################################################
#  cflowd.conf - cflowd configuration file
#  $Name: cflowd-2-1-a9 $
##############################################################################
#  THIS IS JUST AN EXAMPLE!!!  IT MUST BE MODIFIED TO WORK WITH
#  YOUR CFLOWD INSTALLATION!!!
##############################################################################

#-----------------------------------------------------------------------------
#  OPTIONS stanza
#  --------------
#    The OPTIONS stanza contains global cflowd options.  It must be the
#    first stanza in the configuration.
#
#    Option fields:
#
#      LOGFACILITY      (Optional, default local6)
#         The syslog facility to use when logging.
#
#      TCPCOLLECTPORT   (Optional, default 2056)
#         The port on which to listen for connections from cfdcollect.
#
#      PKTBUFSIZE       (Optional, default 1048576)
#         The length (in bytes) to use for packet buffering in
#         shared memory.
#
#      TABLESOCKFILE    (Required)
#         The full path to be used for the named socket on which cflowd
#         will listen for connections from local clients (cfdases, et. al.)
#
#      FLOWDIR          (Required if storing raw flows, no default)
#         The directory in which to store memory-mapped raw flow files.
#         These files tend to have high I/O requirements.
#
#      FLOWFILELEN      (Optional, default 1048576)
#         The maximum length of an individual flow file.  You should
#         be careful with this value; the file is memory mapped and
#         hence should not be too large (1-2M is reasonable in most
#         cases).
#
#      NUMFLOWFILES     (Optional, default 10)
#         The number of raw flow files to retain per router.
#
#      MINLOGMISSED     (Optional, default 300)
#         The minimum number of perceived dropped flows to cause a
#         syslog() message from cflowd.
#
#-----------------------------------------------------------------------------

OPTIONS {
  # syslog to local6 facility.
  LOGFACILITY:          local6

  # Listen for connections from cfdcollect on port 2056.
  TCPCOLLECTPORT:       2056

  # Use a 2 megabyte packet buffer in shared memory.
  PKTBUFSIZE:           2097152

  # Use /usr/local/arts/etc/cflowdtable.socket as named stream socket
  # for connections from local clients (cfdases et. al.)
```

```
  TABLESOCKFILE:        /usr/local/arts/etc/cflowdtable.socket

  # Keep raw flow files in /usr/local/arts/data/cflowd/flows directory.
  FLOWDIR:              /var/www/html/flows

  #  Each raw flow file should be 1000000 bytes in length.
  FLOWFILELEN:          1000000

  #  Keep 10 raw flow files per router.
  NUMFLOWFILES:         10

  #  Log total missed flows from a router if it exceeds 1000 between
  #  connections from cfdcollect.
  MINLOGMISSED:         300
}

#-------------------------------------------------------------------------
#  COLLECTOR stanza
#  ----------------
#    The collector stanza is used to control access from collector
#    clients (e.g. cfdcollect).  Typically you have only one instance
#    of cfdcollect and hence only one COLLECTOR, but you can have as
#    many as you want (for example, if you have a backup host to run
#    cfdcollect when the primary cfdcollect host is down).
#-------------------------------------------------------------------------

COLLECTOR {
  HOST:         XXX.XXX.XXX.XXX   # IP address of central collector
  ADDRESSES:    { XXX.XXX.XXX.XXX }
  AUTH:         none
}


#-------------------------------------------------------------------------
#  CISCOEXPORTER stanza
#  --------------------
#    The CISCOEXPORTER stanza contains information about a Cisco that
#    is expected to export flow data to cflowd.
#
#    CISCOEXPORTER fields
#    --------------------
#      HOST - The IP address of the exporting Cisco.  This is essentially
#        used as an indexing mechanism, to differentiate one Cisco
#        from another.
#
#      ADDRESSES - addresses of individual interfaces on this Cisco.  This
#        allows cflowd to accept packets with a source address of
#        one of the interfaces, but still map the data to this
#        Cisco.
#
#      CFDATAPORT - the port to listen on for packets arriving from the
#         Cisco via flow-export.  This should match the port
#         argument of the 'ip flow-export ...' config line on
#         the Cisco.
#
#      LOCALAS - This is used to substitute an AS number when cflowd gets
#         data with an AS number of 0.  This is a kludge workaround
#         due to prefix cache misses on the Cisco and should be used
#         carefully (set it to 0 to not do substitution).
#
#      SNMPCOMM - SNMP community for the router.  This is used by
#         cflowd to get interface names and IP addresses via SNMP.
#         The community should be enclosed in single quotes.
#
#      COLLECT - What to save from the flow-export data received by the
#         Cisco.  The possible collect options:
#
#           protocol - IP protocol table (pkts/bytes per protocol...
#             ICMP, UDP, TCP, IGMP, etc.)
#
#           portmatrix - port matrix.  Pkts/bytes from port A to port B.
#
```

```
#           ifmatrix - interface matrix.  Pkts/bytes from interface A
#                      to interface B.
#
#           nexthop  - nexthop table.  Pkts/bytes to each IP next hop.
#
#           netmatrix - network matrix.  pkts/bytes from
#                       network A to network B.
#
#           asmatrix - AS matrix.  pkts/bytes from AS A to AS B.
#
#           tos - TOS (Type Of Service) table.  pkts/bytes vs. IP TOS.
#
#           flows - raw flow data.
#
#-------------------------------------------------------------------------

CISCOEXPORTER {
  HOST:          XXX.XXX.XXX.XXX          #  IP address of Cisco sending data.
  ADDRESSES:     { XXX.XXX.XXX.XXX        #  Addresses of interfaces on Cisco
                   }                      #   sending data.
  CFDATAPORT:    2055                     #  Port on which to listen for data.
  SNMPCOMM:      'XXXXX'                  #  SNMP community name.
  LOCALAS:       11377                    #  Local AS of Cisco sending data.
#  COLLECT:      { protocol, portmatrix, ifmatrix, nexthop, netmatrix,
#                  asmatrix, tos, flows }
}
```

Archivo de configuración FlowScan que procesa los flows (/var/www/html/flows/bin/flowscan.cf)

```
# flowscan Configuration Directives ##########################################

# FlowFileGlob (REQUIRED)
# use this glob (file pattern match) when looking for raw flow files to be
# processed, e.g.:
FlowFileGlob /var/www/html/flows/flows.*:*[0-9]
#FlowFileGlob flows.*:*[0-9]

# ReportClasses (REQUIRED)
# a comma-seperated list of FlowScan report classes, e.g.:
ReportClasses CampusIO
# ReportClasses SubNetIO
# ReportClasses CUFlow

# WaitSeconds (OPTIONAL)
# This should be <= the "-s" value passed on the command-line to cflowd, e.g.:
# WaitSeconds 300
WaitSeconds 30

# Verbose (OPTIONAL, non-zero = true)
Verbose 0
```

Archivo que complementa la configuración de FlowScan (/var/www/html/flows/bin/CampusIO.cf)

```
# { General Directives ########################################################

# NextHops (OPTIONAL, BUT SUGGESTED IF OutputIfIndexes IS NOT DEFINED)
# a comma-seperated list of IP addresses (or resolvable hostnames), e.g.:
# NextHops gateway.provider.net, gateway.other.net

# OutputIfIndexes (OPTIONAL, BUT SUGGESTED IF NextHops IS NOT DEFINED)
# a comma-seperated list of ifIndexes as determined using SNMP, e.g.:
#    $ snmpwalk router.our.domain public interfaces.ifTable.ifEntry.ifDescr
# or by looking at the raw flows from Cflowd to determine the $output_if.
# e.g.:
# OutputIfIndexes 1, 2, 3
OutputIfIndexes 3

# LocalSubnetFiles (REQUIRED)
# a comma-seperated list of one (or more) files containing the definitions
# of "local" subnets, e.g.:
# LocalSubnetFiles local_nets.boulder
LocalSubnetFiles /var/www/html/flows/bin/local_nets.espol

# OutputDir (REQUIRED)
# This is the directory in which RRD files will be written, e.g.:
OutputDir /var/www/html/flows/graphs
# OutputDir graphs

# LocalNextHops (OPTIONAL)
# a comma-seperated list of IP address (or resolvable hostnames).
#
# This is an "advanced" option which is only necessary if you are exporting
# and collecting flows from multiple Ciscos to the same FlowScan.
#
# Specify all the local Cisco router(s) from you are exporting and
# collecting flows on this FlowScan host.  This will ensure that the
# same flow isn't counted twice by ignoring flows destined for these
# next-hops, which otherwise would look as if they're inbound flows.
# (The flow will be counted by the last exporter that forwards it.)
# E.g.:
# LocalNextHops other-router.our.domain

# Verbose (OPTIONAL, non-zero = true)
# Verbose 1

# }{ Web Proxy ################################################################

# WebProxyIfIndex (OPTIONAL)
# The single ifIndex number of the router interface to which HTTP requests are
# being transparently redirected.
# E.g.:
# WebProxyIfIndex 5
WebProxyIfIndex 1

# }{ IP Protocols #############################################################

# Protocols (OPTIONAL)
# a comma-seperated list of IP protocols by name, e.g.:
# Protocols icmp, tcp, udp
Protocols icmp, tcp, udp

# }{ IP Services ##############################################################

# TCPServices (OPTIONAL)
# a comma-seperated list of TCP services by name or number, e.g.:
# TCPServices ftp-data, ftp, smtp, nntp, http, 7070, 554
TCPServices ftp-data, ftp, smtp, domain, nntp, http, https, 7070, 554, 8080, 110,
50000, 8311, 6346, 6347, 9618, 5190, 1214, 1863

# UDPServices (OPTIONAL)
# a comma-seperated list of UDP services by name or number, e.g.:
UDPServices domain, snmp, snmp-trap

# }{ Napster ##################################################################
```

```
# NapsterSubnetFiles (OPTIONAL)
# a comma-seperated list of one (or more) files containing the definitions
# of "Napster" subnets, e.g.:
# NapsterSubnetFiles Napster_subnets.boulder
#NapsterSubnetFiles bin/Napster_subnets.boulder

# NapsterSeconds (OPTIONAL)
# the number of seconds after which a given campus host has communicated
# with a host within the "Napster" subnet(s) will no longer be considered
# to be using the Napster application.  E.g. 1/2 an hour:
#NapsterSeconds 1800

# NapsterPorts (OPTIONAL)
# a comma-seperated list of default TCP ports used by Napster.
# These will be used to determine the confidence level of whether or not
# it's really Napster traffic.
# (If confidence is low, it will be reported as "NapsterMaybe".)
#NapsterPorts 8875, 4444, 5555, 6666, 6697, 6688, 6699, 7777, 8888

# }{ AS & BGP ########################################################################

# ASPairs (OPTIONAL)
# source_AS:destination_AS, e.g.:
# ASPairs 0:0
# (Note that the effect of setting ASPairs will be different based on whether
# you specified "peer-as" or "origin-as" when you configured your Cisco.)
ASPairs 0:0

# BGPDumpFile (OPTIONAL)
# the name of a file containing the output of "show ip bgp" on your Cisco
# exporter.  If this option is used, and the specified file exists, it will
# cause the "originAS" and "pathAS" reports to be generated.  Furthermore,
# if the BGPDumpFile's modification time is updated, it will be reloaded.
# BGPDumpFile /tmp/router.our.domain.bgp

# ASNFile (OPTIONAL)
# the path of a file containing ASN info in the format of the file at this URL:
#    ftp://ftp.arin.net/netinfo/asn.txt
# ASNFile etc/asn.txt

# }{ Top Talkers and AS Reports ######################################################

# TopN (OPTIONAL)
# Note that this requires the HTML::Table perl module.
# This is the number of top talkers and listeners to show in the tables
# that will be generated in the "top.html" HTML fragment output file
TopN 10

# ReportPrefixFormat (OPTIONAL)
# This option is used to specify the file name prefix for any HTML or text
# reports such as the "originAS" and "pathAS" reports.
# You may use strftime(3) format specifiers in the value, and it may also
# specify sub-directories.
# If not set, the prefix defaults to the null string, which means that
# each report to overwrite the previous of that type.
# Create reports with this sort of name "YYYYMMDD/HH:MI_report.html":
# ReportPrefixFormat %Y%m%d/%H:%M_
# Preserve one month by using the day of month in the dir name (like sar(1)):
ReportPrefixFormat %d/%H:%M_
# Preserve one day by using only the hour and minute in the dir name:
# ReportPrefixFormat %H:%M/

# } ##################################################################################
```

APÉNDICE D – CONFIGURACIÓN DE LOS DIFERENTES DISPOSITIVOS

Archivo de configuración SNMP en el Servidor Tivoli Netview (/etc/snmpd.conf)

```
# @(#)93        1.12  src/tcpip/etc/snmpd.conf, snmp, tcpip42G, g9650A 11/19/96
16:10:20
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# tcpip42G src/tcpip/etc/snmpd.conf
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1991,1994
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
#
# COMPONENT_NAME: (SNMP) Simple Network Management Protocol Daemon
#
# FUNCTIONS: none
#
# ORIGINS: 27 60
#
# (C) COPYRIGHT International Business Machines Corp. 1991, 1994
# All Rights Reserved
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# Licensed Material - Property of IBM
#
# Contributed by NYSERNet Inc.  This work was partially supported by the
# U.S. Defense Advanced Research Projects Agency and the Rome Air Development
# Center of the U.S. Air Force Systems Command under contract number
# F30602-88-C-0016.
#
# FILE: /etc/snmpd.conf
#
##############################################################################
logging         file=/usr/tmp/snmpd.log         enabled
logging         size=0                          level=0

community       public          127.0.0.1       255.255.255.255 readOnly
community       XXXXX           XXX.XXX.XXX.XXX 255.255.255.255 readWrite
community       private         127.0.0.1       255.255.255.255 readWrite
community       system          127.0.0.1       255.255.255.255 readWrite 1.17.2

view            1.17.2          system enterprises view

trap            public          XXX.XXX.XXX.XXX 1.2.3   fe      # loopback
trap            XXXXX           127.0.0.1       1.2.3   fe      # loopback

#snmpd          maxpacket=1024 querytimeout=120 smuxtimeout=60

smux    1.3.6.1.4.1.2.3.1.2.1.2         gated_password  # gated
smux    1.3.6.1.4.1.2.3.1.2.2.1.1.2     dpid_password   # dpid
```

```
snmpd   smuxtimeout=200 #muxatmd
smux    1.3.6.1.4.1.2.3.1.2.3.1.1       muxatmd_password #muxatmd
smux    1.3.6.1.4.1.2.6.4.1     nv6000  # TME 10 NetView: trapgend
smux    1.3.6.1.4.1.2.6.4.6     nv6000  # TME 10 NetView: mgragentd

# Administrator and Location
syscontact "Neil Nunez <nnunez@espol.edu.ec>"
syslocation "CSI, Campus Gustavo Galindo, ESPOL"
```

Archivo de Configuración SNMP del servidor CiscoWorks (/etc/snmp/conf/snmpd.conf)

```
# Copyright 1988 - 09/23/99 Sun Microsystems, Inc. All Rights Reserved.
#pragma ident  "@(#)snmpd.conf  2.23 99/09/23 Sun Microsystems"

# See below for file format and supported keywords

sysdescr        Sun SNMP Agent, Sun-Blade-1500
syscontact      Neil Nunez <nnunez@espol.edu.ec>
sysLocation     CSI, Campus Gustavo Galindo, ESPOL
#
#system-group-read-community     public
system-group-read-community     XXXXX
#system-group-write-community    private
#
#read-community  public
read-community  XXXXX
#write-community private
#
trap            XXX.XXX.XXX.XXX
trap-community  public
#
#kernel-file     /vmunix
#
#managers        lvs golden
managers        localhost XXX.XXX.XXX.XXX
```

Archivo de configuración de los servidores Linux (/etc/snmp/snmpd.conf)

```
###############################################################################
#
# snmpd.conf:
#   An example configuration file for configuring the ucd-snmp snmpd agent.
#
###############################################################################
#
# This file is intended to only be as a starting point.  Many more
# configuration directives exist than are mentioned in this file.  For
# full details, see the snmpd.conf(5) manual page.
#
# All lines beginning with a '#' are comments and are intended for you
# to read.  All other lines are configuration commands for the agent.

###############################################################################
# Access Control
###############################################################################

# As shipped, the snmpd demon will only respond to queries on the
# system mib group until this file is replaced or modified for
# security purposes.  Examples are shown below about how to increase the
# level of access.

# By far, the most common question I get about the agent is "why won't
# it work?", when really it should be "how do I configure the agent to
# allow me to access it?"
#
# By default, the agent responds to the "public" community for read
# only access, if run out of the box without any configuration file in
# place.  The following examples show you other ways of configuring
# the agent so that you can change the community names, and give
# yourself write access to the mib tree as well.
#
# For more information, read the FAQ as well as the snmpd.conf(5)
# manual page.

####
# First, map the community name "public" into a "security name"

#       sec.name  source          community
com2sec local   localhost              private
com2sec espol   XXX.XXX.XXX.XXX          XXXXX
com2sec espol   XXX.XXX.XXX.XXX          XXXXX
com2sec espol   XXX.XXX.XXX.XXX          XXXXX

####
# Second, map the security name into a group name:

#       groupName      securityModel securityName
group   local   v1      local
group   local   v2c     local
group   local   usm     local
group   grpespol        v1      espol
group   grpespol        v2c     espol

####
# Third, create a view for us to let the group have rights to:

# Make at least  snmpwalk -v 1 localhost -c public system fast again.
#       name           incl/excl     subtree         mask(optional)
view    all             included        .1              80
view    system          included        system          fe
view    mib2            included        .iso.org.dod.internet.mgmt.mib-2 fc

####
# Finally, grant the group read-only access to the systemview view.
```

```
#       group          context sec.model sec.level prefix read   write  notif
access  grpespol        ""      any     noauth  exact   all     none    none
access  local           ""      any     noauth  exact   all     none    none

# ----------------------------------------------------------------------------

# Here is a commented out example configuration that allows less
# restrictive access.

# YOU SHOULD CHANGE THE "COMMUNITY" TOKEN BELOW TO A NEW KEYWORD ONLY
# KNOWN AT YOUR SITE.  YOU *MUST* CHANGE THE NETWORK TOKEN BELOW TO
# SOMETHING REFLECTING YOUR LOCAL NETWORK ADDRESS SPACE.

##       sec.name  source          community
#com2sec local     localhost       COMMUNITY
#com2sec mynetwork NETWORK/24      COMMUNITY

##     group.name sec.model  sec.name
#group MyRWGroup  any        local
#group MyROGroup  any        mynetwork

#
#group MyRWGroup  any        otherv3user
#...

##           incl/excl subtree                          mask

## -or just the mib2 tree-

#view mib2   included  .iso.org.dod.internet.mgmt.mib-2 fc


##               context sec.model sec.level prefix read   write  notif
#access MyROGroup ""      any       noauth    0      all    none   none
#access MyRWGroup ""      any       noauth    0      all    all    all

###############################################################################
# System contact information
#

# It is also possible to set the sysContact and sysLocation system
# variables through the snmpd.conf file:


syslocation CSI, Campus Gustavo Galindo, ESPOL
syscontact Neil Nunez <nnunez@espol.edu.ec>
sysservices 72


###############################################################################
# SECTION: Trap Destinations
#
#   Here we define who the agent will send traps to.

# trapsink: A SNMPv1 trap receiver
#   arguments: host [community] [portnum]

trapsink        XXX.XXX.XXX.XXX   XXXXX

# trapcommunity: Default trap sink community to use
#   arguments: community-string

trapcommunity public

# authtrapenable: Should we send traps when authentication failures occur
#   arguments: 1 | 2   (1 = yes, 2 = no)

authtrapenable  1
```

Para la configuración de los servidores con Windows 2000, en la consola de administración de servicios, se selecciona propiedades de SNMP Service.

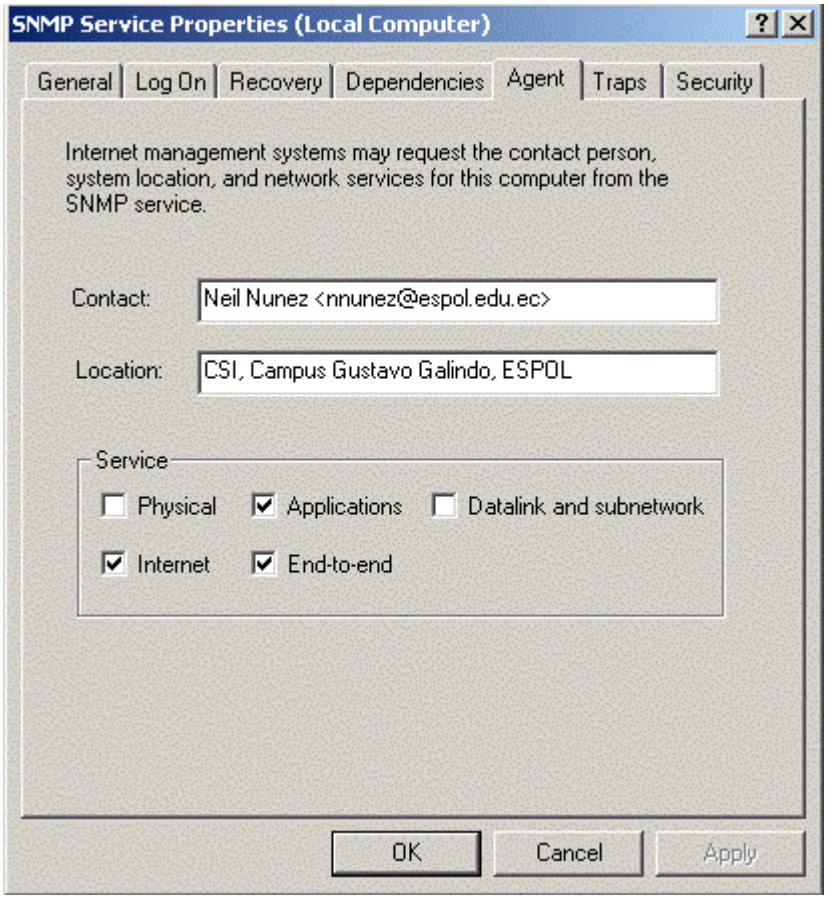


Figura 60 Configuración del servicio SNMP(contacto) para Windows 2000

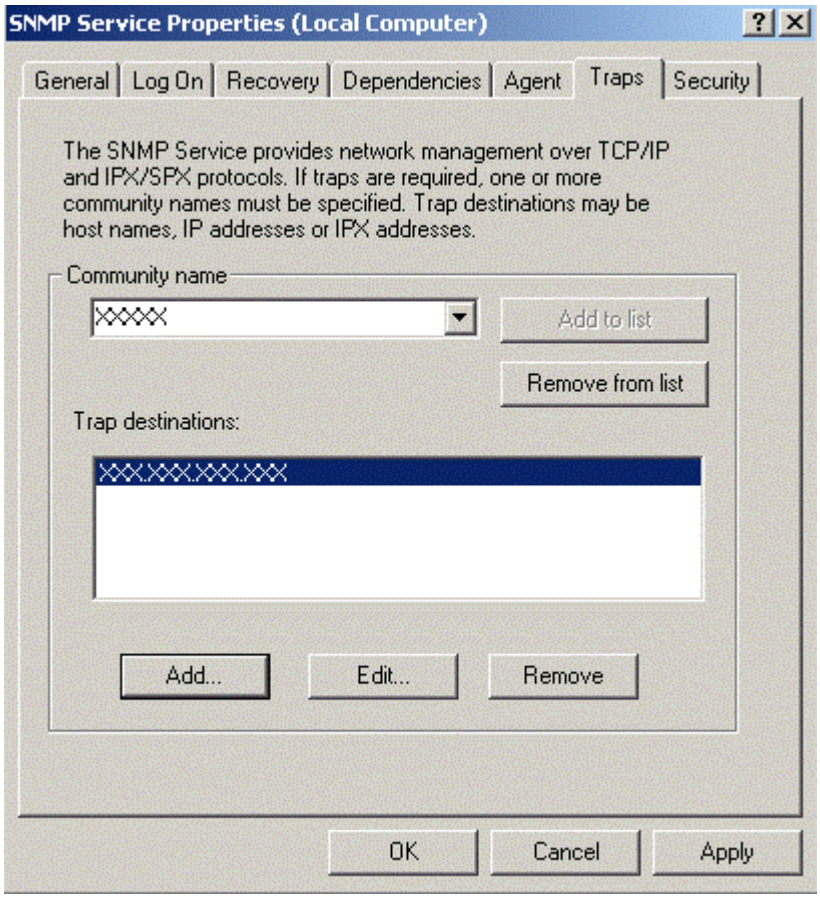


Figura 61 Configuración del servicio SNMP(trap) para Windows 2000

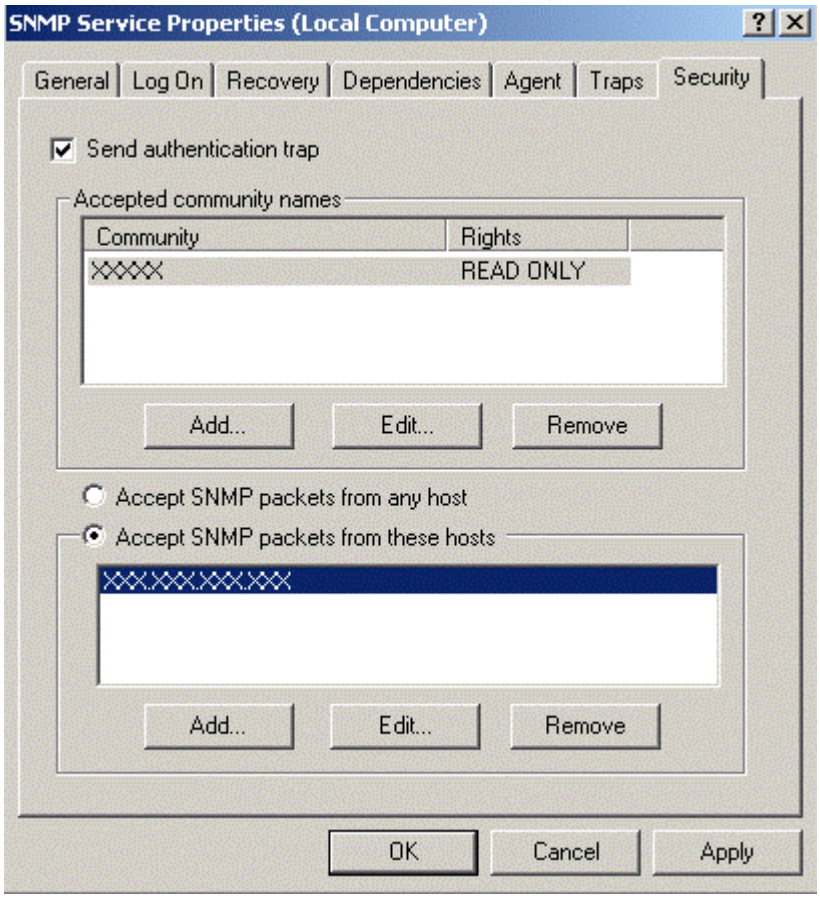


Figura 62 Configuración del servicio SNMP(seguridad) para Windows 2000

Para los equipos cisco, deben agregarse las siguientes líneas a la configuración:

```
access-list 10 remark Lista de Acceso para SNMP
access-list 10 permit XXX.XXX.XXX.XXX 0.0.0.255
access-list 10 deny any any log
snmp-server community XXXXX RO 10
snmp-server location CSI, Campus Gustavo Galindo, ESPOL
snmp-server contact Neil Nunez <nnunez@espol.edu.ec>
snmp-server enable traps
snmp-server host XXX.XXX.XXX.XXX public
```

APÉNDICE E – CÓDIGO FUENTE DE LOS PROGRAMAS USADOS

Archivo Makefile usado para generar los gráficos de flor a partir de los datos proporcionados por Flowscan (/var/www/html/flows/graphs/Makefile). Este archivo debe ejecutarse cada 5 minutos para generar los gráficos.

```
# FlowScan Makefile for graphs
# $Id: graphs.mf.in,v 1.24 2001/02/14 21:52:39 dplonka Exp $
#
# usage:
#    make -f graphs.mf [filetype=<png|gif>] [width=x] [height=y] [ioheight=y+n]
[hours=h] [tag=_tagval] [events=public_events.txt] [
organization='Foobar U - Springfield Campus']
#
# e.g.:
#
#    $ make -f graphs.mf hours=24 tag=_day
#
#    $ make -f graphs.mf filetype=gif hours=168 tag=_week
#
#    $ make -f graphs.mf width=320 height=100 ioheight=120 tag=_small
#
#
# Dave Plonka <plonka@doit.wisc.edu>

SHELL = /usr/bin/ksh

perl = /usr/bin/perl
rrdtool = /usr/local/rrdtool-1.0.45/bin/rrdtool
rrddir = .

event2vrule = /var/www/html/flows/bin/event2vrule

# { you might want to specify these on the make(1) command line:
width = 640
height = 150
hours = 48

# this specifies the heigth for the graphs named "io_*":
ioheight = 320

#filetype = png
filetype = gif

# this is a suffix that you can add to the end of the graph file names:
# e.g. make hours=24 tag=_1d
tag =

# this is a file containing events that you'd like to be displayed in the graph:
events = /dev/null

# this is the name of your organization for graph titles
organization = 'Espol'

past_hours = $$($(perl) -e 'print time - $(hours)*60*60')
# }

# Turn the filetype into uppercase - as rrdtool likes it with "--imgformat":
```

```
IMGFORMAT = "$$(typeset -u IMGFORMAT=$(filetype); print $${IMGFORMAT?})"

# This is the time before which you do not want to graph the data values,
# because they are unreliable:
# Fri Apr  9 10:30:00 1999
totals_last_error = 923671800

totals_past_hours = $$($(perl) -e '$$when = time - $(hours)*60*60; if (0 ==
$(totals_last_error) || $(totals_last_error) > $$when) {
 print "$(totals_last_error)" } else { print $$when } ')

all: services protocols io totals
protocols: protocols_Mbps$(tag).$(filetype) protocols_pkts$(tag).$(filetype)
protocols_flows$(tag).$(filetype)
services: services_Mbps$(tag).$(filetype) services_flows$(tag).$(filetype)
services_pkts$(tag).$(filetype)
totals: totals_Mbps$(tag).$(filetype) totals$(tag).$(filetype)

# These are the 'i'nbound/'o'utbound graphs inspired by example graphs
# by Alexander Kunz <Alexander.Kunz@nextra.de>:
io: io_services_bits$(tag).$(filetype) io_services_pkts$(tag).$(filetype)
io_services_flows$(tag).$(filetype) io_protocols_bits$(tag
).$(filetype) io_protocols_pkts$(tag).$(filetype) io_protocols_flows$(tag).$(filetype)

DEF_total_out_bytes = DEF:total_out_bytes=$(rrddir)/total.rrd:out_bytes:AVERAGE
CDEF_total_out_bits = CDEF:total_out_bits=total_out_bytes,8,*
DEF_total_in_bytes = DEF:total_in_bytes=$(rrddir)/total.rrd:in_bytes:AVERAGE
CDEF_total_in_bits = CDEF:total_in_bits=total_in_bytes,8,*
CDEF_total_bytes = CDEF:total_bytes=total_out_bytes,total_in_bytes,+

DEF_total_out_flows = DEF:total_out_flows=$(rrddir)/total.rrd:out_flows:AVERAGE
DEF_total_in_flows = DEF:total_in_flows=$(rrddir)/total.rrd:in_flows:AVERAGE
CDEF_total_flows = CDEF:total_flows=total_out_flows,total_in_flows,+

DEF_total_out_pkts = DEF:total_out_pkts=$(rrddir)/total.rrd:out_pkts:AVERAGE
DEF_total_in_pkts = DEF:total_in_pkts=$(rrddir)/total.rrd:in_pkts:AVERAGE
CDEF_total_pkts = CDEF:total_pkts=total_out_pkts,total_in_pkts,+

DEF_MCAST_out_bytes = DEF:MCAST_out_bytes=$(rrddir)/MCAST.rrd:out_bytes:AVERAGE
CDEF_MCAST_out_bits = CDEF:MCAST_out_bits=MCAST_out_bytes,8,*
DEF_MCAST_in_bytes = DEF:MCAST_in_bytes=$(rrddir)/MCAST.rrd:in_bytes:AVERAGE
CDEF_MCAST_in_bits = CDEF:MCAST_in_bits=MCAST_in_bytes,8,*
CDEF_MCAST_out_Mbps = CDEF:MCAST_out_Mbps=MCAST_out_bytes,.000008,*
CDEF_MCAST_in_Mbps = CDEF:MCAST_in_Mbps=MCAST_in_bytes,.000008,*
CDEF_MCAST_Mbps = CDEF:MCAST_Mbps=MCAST_out_Mbps,MCAST_in_Mbps,+

CDEF_TOTAL_Mbps = CDEF:TOTAL_Mbps=total_Mbps
CDEF_TOTAL_in_bits = CDEF:TOTAL_in_bits=total_in_bits
CDEF_TOTAL_out_bits = CDEF:TOTAL_out_bits=total_out_bits
CDEF_TOTAL_pkts = CDEF:TOTAL_pkts=total_pkts
CDEF_TOTAL_in_pkts = CDEF:TOTAL_in_pkts=total_in_pkts
CDEF_TOTAL_out_pkts = CDEF:TOTAL_out_pkts=total_out_pkts
CDEF_TOTAL_flows = CDEF:TOTAL_flows=total_flows
CDEF_TOTAL_in_flows = CDEF:TOTAL_in_flows=total_in_flows
CDEF_TOTAL_out_flows = CDEF:TOTAL_out_flows=total_out_flows

DEF_MCAST_out_pkts = DEF:MCAST_out_pkts=$(rrddir)/MCAST.rrd:out_pkts:AVERAGE
DEF_MCAST_in_pkts = DEF:MCAST_in_pkts=$(rrddir)/MCAST.rrd:in_pkts:AVERAGE
CDEF_MCAST_pkts = CDEF:MCAST_pkts=MCAST_out_pkts,MCAST_in_pkts,+

DEF_MCAST_out_flows = DEF:MCAST_out_flows=$(rrddir)/MCAST.rrd:out_flows:AVERAGE
DEF_MCAST_in_flows = DEF:MCAST_in_flows=$(rrddir)/MCAST.rrd:in_flows:AVERAGE
CDEF_MCAST_flows = CDEF:MCAST_flows=MCAST_out_flows,MCAST_in_flows,+

DEF_192_188_59_out_bytes =
DEF:x192_188_59_out_bytes=$(rrddir)/192.188.59.0_24.rrd:out_bytes:AVERAGE
DEF_192_188_59_in_bytes =
DEF:x192_188_59_in_bytes=$(rrddir)/192.188.59.0_24.rrd:in_bytes:AVERAGE

DEF_200_9_176_out_bytes =
DEF:x200_9_176_out_bytes=$(rrddir)/200.9.176.0_24.rrd:out_bytes:AVERAGE
```

```
DEF_200_9_176_in_bytes =
DEF:x200_9_176_in_bytes=$(rrddir)/200.9.176.0_24.rrd:in_bytes:AVERAGE

DEF_200_10_148_out_bytes =
DEF:x200_10_148_out_bytes=$(rrddir)/200.10.148.0_24.rrd:out_bytes:AVERAGE
DEF_200_10_148_in_bytes =
DEF:x200_10_148_in_bytes=$(rrddir)/200.10.148.0_24.rrd:in_bytes:AVERAGE

DEF_200_10_149_out_bytes =
DEF:x200_10_149_out_bytes=$(rrddir)/200.10.149.0_24.rrd:out_bytes:AVERAGE
DEF_200_10_149_in_bytes =
DEF:x200_10_149_in_bytes=$(rrddir)/200.10.149.0_24.rrd:in_bytes:AVERAGE

DEF_200_10_150_out_bytes =
DEF:x200_10_150_out_bytes=$(rrddir)/200.10.150.0_24.rrd:out_bytes:AVERAGE
DEF_200_10_150_in_bytes =
DEF:x200_10_150_in_bytes=$(rrddir)/200.10.150.0_24.rrd:in_bytes:AVERAGE

DEF_200_10_151_out_bytes =
DEF:x200_10_151_out_bytes=$(rrddir)/200.10.151.0_24.rrd:out_bytes:AVERAGE
DEF_200_10_151_in_bytes =
DEF:x200_10_151_in_bytes=$(rrddir)/200.10.151.0_24.rrd:in_bytes:AVERAGE

CDEF_192_188_59_bytes =
CDEF:x192_188_59_bytes=x192_188_59_out_bytes,x192_188_59_in_bytes,+
CDEF_200_9_176_bytes =
CDEF:x200_9_176_bytes=x200_9_176_out_bytes,x200_9_176_in_bytes,+
CDEF_200_10_148_bytes =
CDEF:x200_10_148_bytes=x200_10_148_out_bytes,x200_10_148_in_bytes,+
CDEF_200_10_149_bytes =
CDEF:x200_10_149_bytes=x200_10_149_out_bytes,x200_10_149_in_bytes,+
CDEF_200_10_150_bytes =
CDEF:x200_10_150_bytes=x200_10_150_out_bytes,x200_10_150_in_bytes,+
CDEF_200_10_151_bytes =
CDEF:x200_10_151_bytes=x200_10_151_out_bytes,x200_10_151_in_bytes,+

CDEF_subnet_bytes =
CDEF:subnet_bytes=x192_188_59_bytes,x200_9_176_bytes,+,x200_10_148_bytes,+,x200_10_149
_bytes,+,x200_10_150_bytes
,+,x200_10_151_bytes,+

CDEF_backbone_in_bits = CDEF:backbone_in_bits=x192_188_59_in_bytes,8,*
CDEF_backbone_out_bits = CDEF:backbone_out_bits=x192_188_59_out_bytes,8,*
CDEF_fiec_in_bits = CDEF:fiec_in_bits=x200_9_176_in_bytes,8,*
CDEF_fiec_out_bits = CDEF:fiec_out_bits=x200_9_176_out_bytes,8,*
CDEF_penas_in_bits = CDEF:penas_in_bits=x200_10_148_in_bytes,8,*
CDEF_penas_out_bits = CDEF:penas_out_bits=x200_10_148_out_bytes,8,*
CDEF_biblioteca_in_bits = CDEF:biblioteca_in_bits=x200_10_149_in_bytes,8,*
CDEF_biblioteca_out_bits = CDEF:biblioteca_out_bits=x200_10_149_out_bytes,8,*
CDEF_cti_in_bits = CDEF:cti_in_bits=x200_10_150_in_bytes,8,*
CDEF_cti_out_bits = CDEF:cti_out_bits=x200_10_150_out_bytes,8,*
CDEF_basico_in_bits = CDEF:basico_in_bits=x200_10_151_in_bytes,8,*
CDEF_basico_out_bits = CDEF:basico_out_bits=x200_10_151_out_bytes,8,*

CDEF_total_Mbps = CDEF:total_Mbps=total_bytes,.000008,*

CDEF_backbone_Mbps = CDEF:backbone_Mbps=x192_188_59_bytes,.000008,*
CDEF_fiec_Mbps = CDEF:fiec_Mbps=x200_9_176_bytes,.000008,*
CDEF_penas_Mbps = CDEF:penas_Mbps=x200_10_148_bytes,.000008,*
CDEF_biblioteca_Mbps = CDEF:biblioteca_Mbps=x200_10_149_bytes,.000008,*
CDEF_cti_Mbps = CDEF:cti_Mbps=x200_10_150_bytes,.000008,*
CDEF_basico_Mbps = CDEF:basico_Mbps=x200_10_151_bytes,.000008,*

red = ff0000
green = 00ff00
blue = 0000ff

totals_Mbps$(tag).$(filetype): 192.188.59.0_24.rrd 200.9.176.0_24.rrd
200.10.149.0_24.rrd 200.10.150.0_24.rrd 200.10.151.0_24.rrd to
tal.rrd unknown.rrd
        $(rrdtool) graph \
        $@ \
```

```
        --interlaced \
        --imgformat $(IMGFORMAT) \
        --width $(width) \
        --height $(height) \
        -v 'megabits/sec' \
        -t '$(organization) I/O by Network (Mb/s)' \
        -s $(totals_past_hours) \
        $(DEF_total_out_bytes) \
        $(DEF_total_in_bytes) \
        $(CDEF_total_bytes) \
        $(CDEF_total_Mbps) \
        $(DEF_192_188_59_out_bytes) \
        $(DEF_192_188_59_in_bytes) \
        $(CDEF_192_188_59_bytes) \
        $(CDEF_backbone_Mbps) \
        $(DEF_200_9_176_out_bytes) \
        $(DEF_200_9_176_in_bytes) \
        $(CDEF_200_9_176_bytes) \
        $(CDEF_fiec_Mbps) \
        $(DEF_200_10_149_out_bytes) \
        $(DEF_200_10_149_in_bytes) \
        $(CDEF_200_10_149_bytes) \
        $(CDEF_biblioteca_Mbps) \
        $(DEF_200_10_150_out_bytes) \
        $(DEF_200_10_150_in_bytes) \
        $(CDEF_200_10_150_bytes) \
        $(CDEF_cti_Mbps) \
        $(DEF_200_10_151_out_bytes) \
        $(DEF_200_10_151_in_bytes) \
        $(CDEF_200_10_151_bytes) \
        $(CDEF_basico_Mbps) \
        $(DEF_MCAST_in_bytes) \
        $(DEF_MCAST_out_bytes) \
        $(CDEF_MCAST_in_Mbps) \
        $(CDEF_MCAST_out_Mbps) \
        $(CDEF_MCAST_Mbps) \
        $(CDEF_TOTAL_Mbps) \
        'CDEF:backbone_pct=backbone_Mbps,total_Mbps,/,100,*' \
        'CDEF:fiec_pct=fiec_Mbps,total_Mbps,/,100,*' \
        'CDEF:biblioteca_pct=biblioteca_Mbps,total_Mbps,/,100,*' \
        'CDEF:cti_pct=cti_Mbps,total_Mbps,/,100,*' \
        'CDEF:basico_pct=basico_Mbps,total_Mbps,/,100,*' \
        AREA:backbone_Mbps#aaaa00:'Backbone I/O (192.188.59.0/24)' \
        STACK:fiec_Mbps#ff0000:'Fiec I/O (200.9.176.0/24)' \
        STACK:biblioteca_Mbps#00ff00:'Biblioteca I/O (200.10.149.0/24)' \
        STACK:cti_Mbps#0000ff:'CTI I/O (200.10.150.0/24)' \
        STACK:basico_Mbps#ffff00:'Basico I/O (200.10.151.0/24)' \
        LINE1:TOTAL_Mbps#880088:'TOTAL I/O' \
        COMMENT:'\n' \
        COMMENT:'\n' \
        GPRINT:backbone_pct:AVERAGE:'Backbone %.1lf%%' \
        GPRINT:fiec_pct:AVERAGE:'FIEC %.1lf%%' \
        GPRINT:biblioteca_pct:AVERAGE:'Biblioteca %.1lf%%' \
        GPRINT:cti_pct:AVERAGE:'CTI %.1lf%%' \
        GPRINT:basico_pct:AVERAGE:'Basico %.1lf%%'


totals$(tag).$(filetype): 192.188.59.0_24.rrd 200.9.176.0_24.rrd 200.10.149.0_24.rrd
200.10.150.0_24.rrd 200.10.151.0_24.rrd total.r
rd unknown.rrd
        $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        --width $(width) \
        --height $(ioheight) \
        -v 'bits/sec' \
        -t '$(organization) I/O by Network, Bytes, +out/-in' \
        -s $(totals_past_hours) \
        $(DEF_total_out_bytes) \
        $(DEF_total_in_bytes) \
        $(CDEF_total_out_bits) \
```

```
        $(CDEF_total_in_bits),-1,* \
        $(CDEF_total_bytes) \
        $(CDEF_total_Mbps) \
        $(DEF_192_188_59_out_bytes) \
        $(DEF_192_188_59_in_bytes) \
        $(CDEF_backbone_out_bits) \
        $(CDEF_backbone_in_bits),-1,* \
        $(DEF_200_9_176_out_bytes) \
        $(DEF_200_9_176_in_bytes) \
        $(CDEF_fiec_out_bits) \
        $(CDEF_fiec_in_bits),-1,* \
        $(DEF_200_10_149_out_bytes) \
        $(DEF_200_10_149_in_bytes) \
        $(CDEF_biblioteca_out_bits) \
        $(CDEF_biblioteca_in_bits),-1,* \
        $(DEF_200_10_150_out_bytes) \
        $(DEF_200_10_150_in_bytes) \
        $(CDEF_cti_out_bits) \
        $(CDEF_cti_in_bits),-1,* \
        $(DEF_200_10_151_out_bytes) \
        $(DEF_200_10_151_in_bytes) \
        $(CDEF_basico_out_bits) \
        $(CDEF_basico_in_bits),-1,* \
        $(DEF_MCAST_in_bytes) \
        $(DEF_MCAST_out_bytes) \
        $(CDEF_MCAST_in_Mbps) \
        $(CDEF_MCAST_out_Mbps) \
        $(CDEF_MCAST_Mbps) \
        $(CDEF_TOTAL_Mbps) \
        'CDEF:backbone_out_pct=backbone_out_bits,total_out_bits,/,100,*' \
        'CDEF:backbone_in_pct=backbone_in_bits,total_in_bits,/,100,*' \
        'CDEF:fiec_out_pct=fiec_out_bits,total_out_bits,/,100,*' \
        'CDEF:fiec_in_pct=fiec_in_bits,total_in_bits,/,100,*' \
        'CDEF:biblioteca_out_pct=biblioteca_out_bits,total_out_bits,/,100,*' \
        'CDEF:biblioteca_in_pct=biblioteca_in_bits,total_in_bits,/,100,*' \
        'CDEF:cti_out_pct=cti_out_bits,total_out_bits,/,100,*' \
        'CDEF:cti_in_pct=cti_in_bits,total_in_bits,/,100,*' \
        'CDEF:basico_out_pct=basico_out_bits,total_out_bits,/,100,*' \
        'CDEF:basico_in_pct=basico_in_bits,total_in_bits,/,100,*' \
        AREA:backbone_out_bits#aaaa00:'Backbone      ' \
        GPRINT:backbone_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:backbone_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:fiec_out_bits#ff0000:'Fiec          ' \
        GPRINT:fiec_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:fiec_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:biblioteca_out_bits#00ff00:'Biblioteca    ' \
        GPRINT:biblioteca_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:biblioteca_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:cti_out_bits#0000ff:'CTI           ' \
        GPRINT:cti_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:cti_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:basico_out_bits#ffff00:'Basico        ' \
        GPRINT:basico_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:basico_in_pct:AVERAGE:'%.1lf%% In\n' \
        LINE1:total_out_bits#880088:'TOTAL I/O' \
        AREA:backbone_in_bits#aaaa00 \
        STACK:fiec_in_bits#ff0000 \
        STACK:biblioteca_in_bits#00ff00 \
        STACK:cti_in_bits#0000ff \
        STACK:basico_in_bits#ffff00 \
        LINE1:total_in_bits#880088 \
        HRULE:0#f5f5f5

DEF_tcp_out_bytes = DEF:tcp_out_bytes=$(rrddir)/tcp.rrd:out_bytes:AVERAGE
DEF_tcp_in_bytes = DEF:tcp_in_bytes=$(rrddir)/tcp.rrd:in_bytes:AVERAGE
CDEF_tcp_out_bits = CDEF:tcp_out_bits=tcp_out_bytes,8,*
CDEF_tcp_in_bits = CDEF:tcp_in_bits=tcp_in_bytes,8,*
CDEF_tcp_out_Mbps = CDEF:tcp_out_Mbps=tcp_out_bytes,.000008,*
CDEF_tcp_in_Mbps = CDEF:tcp_in_Mbps=tcp_in_bytes,.000008,*
CDEF_tcp_Mbps = CDEF:tcp_Mbps=tcp_out_Mbps,tcp_in_Mbps,+

DEF_udp_out_bytes = DEF:udp_out_bytes=$(rrddir)/udp.rrd:out_bytes:AVERAGE
```

```
DEF_udp_in_bytes = DEF:udp_in_bytes=$(rrddir)/udp.rrd:in_bytes:AVERAGE
CDEF_udp_out_bits = CDEF:udp_out_bits=udp_out_bytes,8,*
CDEF_udp_in_bits = CDEF:udp_in_bits=udp_in_bytes,8,*
CDEF_udp_out_Mbps = CDEF:udp_out_Mbps=udp_out_bytes,.000008,*
CDEF_udp_in_Mbps = CDEF:udp_in_Mbps=udp_in_bytes,.000008,*
CDEF_udp_Mbps = CDEF:udp_Mbps=udp_out_Mbps,udp_in_Mbps,+

DEF_icmp_out_bytes = DEF:icmp_out_bytes=$(rrddir)/icmp.rrd:out_bytes:AVERAGE
DEF_icmp_in_bytes = DEF:icmp_in_bytes=$(rrddir)/icmp.rrd:in_bytes:AVERAGE
CDEF_icmp_out_bits = CDEF:icmp_out_bits=icmp_out_bytes,8,*
CDEF_icmp_in_bits = CDEF:icmp_in_bits=icmp_in_bytes,8,*
CDEF_icmp_out_Mbps = CDEF:icmp_out_Mbps=icmp_out_bytes,.000008,*
CDEF_icmp_in_Mbps = CDEF:icmp_in_Mbps=icmp_in_bytes,.000008,*
CDEF_icmp_Mbps = CDEF:icmp_Mbps=icmp_out_Mbps,icmp_in_Mbps,+

protocols_Mbps$(tag).$(filetype): icmp.rrd tcp.rrd udp.rrd MCAST.rrd
        $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        --width $(width) \
        --height $(height) \
        -v 'megabits per second' \
        -t '$(organization) I/O by IP Protocol, Bytes' \
        -s $(past_hours) \
        $(DEF_total_out_bytes) \
        $(DEF_total_in_bytes) \
        $(CDEF_total_bytes) \
        $(CDEF_total_Mbps) \
        $(DEF_tcp_out_bytes) \
        $(DEF_tcp_in_bytes) \
        $(CDEF_tcp_out_Mbps) \
        $(CDEF_tcp_in_Mbps) \
        $(CDEF_tcp_Mbps) \
        $(DEF_udp_out_bytes) \
        $(DEF_udp_in_bytes) \
        $(CDEF_udp_out_Mbps) \
        $(CDEF_udp_in_Mbps) \
        $(CDEF_udp_Mbps) \
        $(DEF_icmp_out_bytes) \
        $(DEF_icmp_in_bytes) \
        $(CDEF_icmp_out_Mbps) \
        $(CDEF_icmp_in_Mbps) \
        $(CDEF_icmp_Mbps) \
        $(DEF_MCAST_in_bytes) \
        $(DEF_MCAST_out_bytes) \
        $(CDEF_MCAST_in_Mbps) \
        $(CDEF_MCAST_out_Mbps) \
        $(CDEF_MCAST_Mbps) \
        $(CDEF_TOTAL_Mbps) \
        AREA:tcp_in_Mbps#ff0000:'TCP in' \
        STACK:tcp_out_Mbps#880000:'TCP out' \
        STACK:MCAST_in_Mbps#aaaa00:'MCAST in' \
        STACK:MCAST_out_Mbps#555500:'MCAST out' \
        STACK:udp_in_Mbps#00ff00:'UDP in' \
        STACK:udp_out_Mbps#008800:'UDP out' \
        STACK:icmp_in_Mbps#0000ff:'ICMP in' \
        STACK:icmp_out_Mbps#000088:'ICMP out' \
        LINE1:TOTAL_Mbps#880088:'TOTAL I/O'

io_protocols_bits$(tag).$(filetype): icmp.rrd tcp.rrd udp.rrd MCAST.rrd $(events)
        $(event2vrule) -h $(hours) $(events) $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        -v 'bits per second' \
        -t '$(organization) I/O by IP Protocol, Bytes, +out/-in' \
        -s $(past_hours) \
        --width $(width) \
        --height $(ioheight) \
        --alt-autoscale \
        $(DEF_total_out_bytes) \
```

```
        $(DEF_total_in_bytes) \
        $(CDEF_total_out_bits) \
        $(CDEF_total_in_bits),-1,* \
        $(DEF_tcp_out_bytes) \
        $(DEF_tcp_in_bytes) \
        $(CDEF_tcp_out_bits) \
        $(CDEF_tcp_in_bits),-1,* \
        $(DEF_udp_out_bytes) \
        $(DEF_udp_in_bytes) \
        $(CDEF_udp_out_bits) \
        $(CDEF_udp_in_bits),-1,* \
        $(DEF_icmp_out_bytes) \
        $(DEF_icmp_in_bytes) \
        $(CDEF_icmp_out_bits) \
        $(CDEF_icmp_in_bits),-1,* \
        $(DEF_MCAST_in_bytes) \
        $(DEF_MCAST_out_bytes) \
        $(CDEF_MCAST_in_bits),-1,* \
        $(CDEF_MCAST_out_bits) \
        $(CDEF_TOTAL_out_bits) \
        $(CDEF_TOTAL_in_bits) \
        AREA:tcp_out_bits#ff0000:'TCP out' \
        STACK:MCAST_out_bits#aaaa00:'MCAST out' \
        STACK:udp_out_bits#00ff00:'UDP out' \
        STACK:icmp_out_bits#0000ff:'ICMP out' \
        LINE1:TOTAL_out_bits#880088:'TOTAL out' \
        COMMENT:'\n' \
        AREA:tcp_in_bits#880000:'TCP in ' \
        STACK:MCAST_in_bits#555500:'MCAST in ' \
        STACK:udp_in_bits#008800:'UDP in ' \
        STACK:icmp_in_bits#000088:'ICMP in ' \
        LINE1:TOTAL_in_bits#880088:'TOTAL in ' \
        HRULE:0#f5f5f5

DEF_tcp_out_pkts = DEF:tcp_out_pkts=$(rrddir)/tcp.rrd:out_pkts:AVERAGE
DEF_tcp_in_pkts = DEF:tcp_in_pkts=$(rrddir)/tcp.rrd:in_pkts:AVERAGE
CDEF_tcp_pkts = CDEF:tcp_pkts=tcp_out_pkts,tcp_in_pkts,+

DEF_udp_out_pkts = DEF:udp_out_pkts=$(rrddir)/udp.rrd:out_pkts:AVERAGE
DEF_udp_in_pkts = DEF:udp_in_pkts=$(rrddir)/udp.rrd:in_pkts:AVERAGE
CDEF_udp_pkts = CDEF:udp_pkts=udp_out_pkts,udp_in_pkts,+

DEF_icmp_out_pkts = DEF:icmp_out_pkts=$(rrddir)/icmp.rrd:out_pkts:AVERAGE
DEF_icmp_in_pkts = DEF:icmp_in_pkts=$(rrddir)/icmp.rrd:in_pkts:AVERAGE
CDEF_icmp_pkts = CDEF:icmp_pkts=icmp_out_pkts,icmp_in_pkts,+

protocols_pkts$(tag).$(filetype): icmp.rrd tcp.rrd udp.rrd MCAST.rrd
        $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        --width $(width) \
        --height $(height) \
        -v 'packets per second' \
        -t '$(organization) I/O by IP Protocol, Packets' \
        -s $(past_hours) \
        $(DEF_total_out_pkts) \
        $(DEF_total_in_pkts) \
        $(CDEF_total_pkts) \
        $(DEF_tcp_out_pkts) \
        $(DEF_tcp_in_pkts) \
        $(CDEF_tcp_pkts) \
        $(DEF_udp_out_pkts) \
        $(DEF_udp_in_pkts) \
        $(CDEF_udp_pkts) \
        $(DEF_icmp_out_pkts) \
        $(DEF_icmp_in_pkts) \
        $(CDEF_icmp_pkts) \
        $(DEF_MCAST_in_pkts) \
        $(DEF_MCAST_out_pkts) \
        $(CDEF_MCAST_pkts) \
        $(CDEF_TOTAL_pkts) \
```

```
        AREA:tcp_in_pkts#ff0000:'TCP in' \
        STACK:tcp_out_pkts#880000:'TCP out' \
        STACK:MCAST_in_pkts#aaaa00:'MCAST in' \
        STACK:MCAST_out_pkts#555500:'MCAST out' \
        STACK:udp_in_pkts#00ff00:'UDP in' \
        STACK:udp_out_pkts#008800:'UDP out' \
        STACK:icmp_in_pkts#0000ff:'ICMP in' \
        STACK:icmp_out_pkts#000088:'ICMP out' \
        LINE1:TOTAL_pkts#880088:'TOTAL I/O'

io_protocols_pkts$(tag).$(filetype): icmp.rrd tcp.rrd udp.rrd MCAST.rrd $(events)
        $(event2vrule) -h $(hours) $(events) $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        --width $(width) \
        --height $(ioheight) \
        --alt-autoscale \
        -v 'packets per second' \
        -t '$(organization) I/O by IP Protocol, Packets, +out/-in' \
        -s $(past_hours) \
        $(DEF_total_out_pkts) \
        $(DEF_total_in_pkts) \
        CDEF:total_in_pkts_neg=total_in_pkts,-1,* \
        $(DEF_tcp_out_pkts) \
        $(DEF_tcp_in_pkts) \
        CDEF:tcp_in_pkts_neg=tcp_in_pkts,-1,* \
        $(DEF_udp_out_pkts) \
        $(DEF_udp_in_pkts) \
        CDEF:udp_in_pkts_neg=udp_in_pkts,-1,* \
        $(DEF_icmp_out_pkts) \
        $(DEF_icmp_in_pkts) \
        CDEF:icmp_in_pkts_neg=icmp_in_pkts,-1,* \
        $(DEF_MCAST_in_pkts) \
        CDEF:MCAST_in_pkts_neg=MCAST_in_pkts,-1,* \
        $(DEF_MCAST_out_pkts) \
        $(CDEF_TOTAL_in_pkts),-1,* \
        $(CDEF_TOTAL_out_pkts) \
        AREA:tcp_out_pkts#ff0000:'TCP out' \
        STACK:MCAST_out_pkts#aaaa00:'MCAST out' \
        STACK:udp_out_pkts#00ff00:'UDP out' \
        STACK:icmp_out_pkts#0000ff:'ICMP out' \
        LINE1:TOTAL_out_pkts#880088:'TOTAL out' \
        COMMENT:'\n' \
        AREA:tcp_in_pkts_neg#880000:'TCP in ' \
        STACK:MCAST_in_pkts_neg#555500:'MCAST in ' \
        STACK:udp_in_pkts_neg#008800:'UDP in ' \
        STACK:icmp_in_pkts_neg#000088:'ICMP in ' \
        LINE1:TOTAL_in_pkts#880088:'TOTAL in ' \
        HRULE:0#f5f5f5

DEF_tcp_out_flows = DEF:tcp_out_flows=$(rrddir)/tcp.rrd:out_flows:AVERAGE
DEF_tcp_in_flows = DEF:tcp_in_flows=$(rrddir)/tcp.rrd:in_flows:AVERAGE
CDEF_tcp_flows = CDEF:tcp_flows=tcp_out_flows,tcp_in_flows,+

DEF_udp_out_flows = DEF:udp_out_flows=$(rrddir)/udp.rrd:out_flows:AVERAGE
DEF_udp_in_flows = DEF:udp_in_flows=$(rrddir)/udp.rrd:in_flows:AVERAGE
CDEF_udp_flows = CDEF:udp_flows=udp_out_flows,udp_in_flows,+

DEF_icmp_out_flows = DEF:icmp_out_flows=$(rrddir)/icmp.rrd:out_flows:AVERAGE
DEF_icmp_in_flows = DEF:icmp_in_flows=$(rrddir)/icmp.rrd:in_flows:AVERAGE
CDEF_icmp_flows = CDEF:icmp_flows=icmp_out_flows,icmp_in_flows,+

protocols_flows$(tag).$(filetype): icmp.rrd tcp.rrd udp.rrd MCAST.rrd
        $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        --width $(width) \
        --height $(height) \
        -v 'flows per second' \
        -t '$(organization) I/O by IP Protocol, Flows' \
```

```
	-s $(past_hours) \
	$(DEF_total_out_flows) \
	$(DEF_total_in_flows) \
	$(CDEF_total_flows) \
	$(DEF_tcp_out_flows) \
	$(DEF_tcp_in_flows) \
	$(CDEF_tcp_flows) \
	$(DEF_udp_out_flows) \
	$(DEF_udp_in_flows) \
	$(CDEF_udp_flows) \
	$(DEF_icmp_out_flows) \
	$(DEF_icmp_in_flows) \
	$(CDEF_icmp_flows) \
	$(DEF_MCAST_in_flows) \
	$(DEF_MCAST_out_flows) \
	$(CDEF_MCAST_flows) \
	$(CDEF_TOTAL_flows) \
	AREA:tcp_in_flows#ff0000:'TCP in' \
	STACK:tcp_out_flows#880000:'TCP out' \
	STACK:MCAST_in_flows#aaaa00:'MCAST in' \
	STACK:MCAST_out_flows#555500:'MCAST out' \
	STACK:udp_in_flows#00ff00:'UDP in' \
	STACK:udp_out_flows#008800:'UDP out' \
	STACK:icmp_in_flows#0000ff:'ICMP in' \
	STACK:icmp_out_flows#000088:'ICMP out' \
	LINE1:TOTAL_flows#880088:'TOTAL I/O'

io_protocols_flows$(tag).$(filetype): icmp.rrd tcp.rrd udp.rrd MCAST.rrd $(events)
	$(event2vrule) -h $(hours) $(events) $(rrdtool) graph \
	$@ \
	--interlaced \
	--imgformat $(IMGFORMAT) \
	--width $(width) \
	--height $(ioheight) \
	--alt-autoscale \
	-v 'flows per second' \
	-t '$(organization) I/O by IP Protocol, Flows, +out/-in' \
	-s $(past_hours) \
	$(DEF_total_out_flows) \
	$(DEF_total_in_flows) \
	CDEF:total_in_flows_neg=total_in_flows,-1,* \
	$(CDEF_total_flows) \
	$(DEF_tcp_out_flows) \
	$(DEF_tcp_in_flows) \
	CDEF:tcp_in_flows_neg=tcp_in_flows,-1,* \
	$(CDEF_tcp_flows) \
	$(DEF_udp_out_flows) \
	$(DEF_udp_in_flows) \
	CDEF:udp_in_flows_neg=udp_in_flows,-1,* \
	$(CDEF_udp_flows) \
	$(DEF_icmp_out_flows) \
	$(DEF_icmp_in_flows) \
	CDEF:icmp_in_flows_neg=icmp_in_flows,-1,* \
	$(CDEF_icmp_flows) \
	$(DEF_MCAST_in_flows) \
	CDEF:MCAST_in_flows_neg=MCAST_in_flows,-1,* \
	$(DEF_MCAST_out_flows) \
	$(CDEF_TOTAL_in_flows),-1,* \
	$(CDEF_TOTAL_out_flows) \
	AREA:tcp_out_flows#ff0000:'TCP out' \
	STACK:MCAST_out_flows#aaaa00:'MCAST out' \
	STACK:udp_out_flows#00ff00:'UDP out' \
	STACK:icmp_out_flows#0000ff:'ICMP out' \
	LINE1:TOTAL_out_flows#880088:'TOTAL out' \
	COMMENT:'\n' \
	AREA:tcp_in_flows_neg#880000:'TCP in ' \
	STACK:MCAST_in_flows_neg#555500:'MCAST in ' \
	STACK:udp_in_flows_neg#008800:'UDP in ' \
	STACK:icmp_in_flows_neg#000088:'ICMP in ' \
	LINE1:TOTAL_in_flows#880088:'TOTAL in ' \
	HRULE:0#f5f5f5
```

```
DEF_http_src_out_bytes =
DEF:http_src_out_bytes=$(rrddir)/http_src.rrd:out_bytes:AVERAGE
CDEF_http_src_out_bits = CDEF:http_src_out_bits=http_src_out_bytes,8,*
DEF_http_src_in_bytes = DEF:http_src_in_bytes=$(rrddir)/http_src.rrd:in_bytes:AVERAGE
CDEF_http_src_in_bits = CDEF:http_src_in_bits=http_src_in_bytes,8,*
CDEF_http_src_Mbps =
CDEF:http_src_Mbps=http_src_out_bytes,http_src_in_bytes,+,.000008,*

DEF_http_dst_out_bytes =
DEF:http_dst_out_bytes=$(rrddir)/http_dst.rrd:out_bytes:AVERAGE
CDEF_http_dst_out_bits = CDEF:http_dst_out_bits=http_dst_out_bytes,8,*
DEF_http_dst_in_bytes = DEF:http_dst_in_bytes=$(rrddir)/http_dst.rrd:in_bytes:AVERAGE
CDEF_http_dst_in_bits = CDEF:http_dst_in_bits=http_dst_in_bytes,8,*
CDEF_http_dst_Mbps =
CDEF:http_dst_Mbps=http_dst_out_bytes,http_dst_in_bytes,+,.000008,*

DEF_ftp_data_src_out_bytes = DEF:ftp_data_src_out_bytes=$(rrddir)/ftp-
data_src.rrd:out_bytes:AVERAGE
CDEF_ftp_data_src_out_bits = CDEF:ftp_data_src_out_bits=ftp_data_src_out_bytes,8,*
DEF_ftp_data_src_in_bytes = DEF:ftp_data_src_in_bytes=$(rrddir)/ftp-
data_src.rrd:in_bytes:AVERAGE
CDEF_ftp_data_src_in_bits = CDEF:ftp_data_src_in_bits=ftp_data_src_in_bytes,8,*
CDEF_ftp_data_src_Mbps =
CDEF:ftp_data_src_Mbps=ftp_data_src_out_bytes,ftp_data_src_in_bytes,+,.000008,*

DEF_ftp_data_dst_out_bytes = DEF:ftp_data_dst_out_bytes=$(rrddir)/ftp-
data_dst.rrd:out_bytes:AVERAGE
CDEF_ftp_data_dst_out_bits = CDEF:ftp_data_dst_out_bits=ftp_data_dst_out_bytes,8,*
DEF_ftp_data_dst_in_bytes = DEF:ftp_data_dst_in_bytes=$(rrddir)/ftp-
data_dst.rrd:in_bytes:AVERAGE
CDEF_ftp_data_dst_in_bits = CDEF:ftp_data_dst_in_bits=ftp_data_dst_in_bytes,8,*
CDEF_ftp_data_dst_Mbps =
CDEF:ftp_data_dst_Mbps=ftp_data_dst_out_bytes,ftp_data_dst_in_bytes,+,.000008,*

DEF_ftpPASV_src_out_bytes =
DEF:ftpPASV_src_out_bytes=$(rrddir)/ftpPASV_src.rrd:out_bytes:AVERAGE
CDEF_ftpPASV_src_out_bits = CDEF:ftpPASV_src_out_bits=ftpPASV_src_out_bytes,8,*
DEF_ftpPASV_src_in_bytes =
DEF:ftpPASV_src_in_bytes=$(rrddir)/ftpPASV_src.rrd:in_bytes:AVERAGE
CDEF_ftpPASV_src_in_bits = CDEF:ftpPASV_src_in_bits=ftpPASV_src_in_bytes,8,*
CDEF_ftpPASV_src_Mbps =
CDEF:ftpPASV_src_Mbps=ftpPASV_src_out_bytes,ftpPASV_src_in_bytes,+,.000008,*

DEF_ftpPASV_dst_out_bytes =
DEF:ftpPASV_dst_out_bytes=$(rrddir)/ftpPASV_dst.rrd:out_bytes:AVERAGE
CDEF_ftpPASV_dst_out_bits = CDEF:ftpPASV_dst_out_bits=ftpPASV_dst_out_bytes,8,*
DEF_ftpPASV_dst_in_bytes =
DEF:ftpPASV_dst_in_bytes=$(rrddir)/ftpPASV_dst.rrd:in_bytes:AVERAGE
CDEF_ftpPASV_dst_in_bits = CDEF:ftpPASV_dst_in_bits=ftpPASV_dst_in_bytes,8,*
CDEF_ftpPASV_dst_Mbps =
CDEF:ftpPASV_dst_Mbps=ftpPASV_dst_out_bytes,ftpPASV_dst_in_bytes,+,.000008,*

CDEF_ftpDATA_src_Mbps = CDEF:ftpDATA_src_Mbps=ftp_data_src_Mbps,ftpPASV_src_Mbps,+
CDEF_ftpDATA_dst_Mbps = CDEF:ftpDATA_dst_Mbps=ftp_data_dst_Mbps,ftpPASV_dst_Mbps,+
CDEF_ftpDATA_src_in_bits =
CDEF:ftpDATA_src_in_bits=ftp_data_src_in_bits,ftpPASV_src_in_bits,+
CDEF_ftpDATA_dst_in_bits =
CDEF:ftpDATA_dst_in_bits=ftp_data_dst_in_bits,ftpPASV_dst_in_bits,+
CDEF_ftpDATA_src_out_bits =
CDEF:ftpDATA_src_out_bits=ftp_data_src_out_bits,ftpPASV_src_out_bits,+
CDEF_ftpDATA_dst_out_bits =
CDEF:ftpDATA_dst_out_bits=ftp_data_dst_out_bits,ftpPASV_dst_out_bits,+

DEF_pop_src_out_bytes = DEF:pop_src_out_bytes=$(rrddir)/pop-
3_src.rrd:out_bytes:AVERAGE
CDEF_pop_src_out_bits = CDEF:pop_src_out_bits=pop_src_out_bytes,8,*
DEF_pop_src_in_bytes = DEF:pop_src_in_bytes=$(rrddir)/pop-3_src.rrd:in_bytes:AVERAGE
CDEF_pop_src_in_bits = CDEF:pop_src_in_bits=pop_src_in_bytes,8,*
CDEF_pop_src_Mbps = CDEF:pop_src_Mbps=pop_src_out_bytes,pop_src_in_bytes,+,.000008,*

DEF_pop_dst_out_bytes = DEF:pop_dst_out_bytes=$(rrddir)/pop-
3_dst.rrd:out_bytes:AVERAGE
```

```
CDEF_pop_dst_out_bits = CDEF:pop_dst_out_bits=pop_dst_out_bytes,8,*
DEF_pop_dst_in_bytes = DEF:pop_dst_in_bytes=$(rrddir)/pop-3_dst.rrd:in_bytes:AVERAGE
CDEF_pop_dst_in_bits = CDEF:pop_dst_in_bits=pop_dst_in_bytes,8,*
CDEF_pop_dst_Mbps = CDEF:pop_dst_Mbps=pop_dst_out_bytes,pop_dst_in_bytes,+,.000008,*

#DEF_nntp_src_out_bytes =
DEF:nntp_src_out_bytes=$(rrddir)/nntp_src.rrd:out_bytes:AVERAGE
#CDEF_nntp_src_out_bits = CDEF:nntp_src_out_bits=nntp_src_out_bytes,8,*
#DEF_nntp_src_in_bytes = DEF:nntp_src_in_bytes=$(rrddir)/nntp_src.rrd:in_bytes:AVERAGE
#CDEF_nntp_src_in_bits = CDEF:nntp_src_in_bits=nntp_src_in_bytes,8,*
#CDEF_nntp_src_Mbps =
CDEF:nntp_src_Mbps=nntp_src_out_bytes,nntp_src_in_bytes,+,.000008,*

#DEF_nntp_dst_out_bytes =
DEF:nntp_dst_out_bytes=$(rrddir)/nntp_dst.rrd:out_bytes:AVERAGE
#CDEF_nntp_dst_out_bits = CDEF:nntp_dst_out_bits=nntp_dst_out_bytes,8,*
#DEF_nntp_dst_in_bytes = DEF:nntp_dst_in_bytes=$(rrddir)/nntp_dst.rrd:in_bytes:AVERAGE
#CDEF_nntp_dst_in_bits = CDEF:nntp_dst_in_bits=nntp_dst_in_bytes,8,*
#CDEF_nntp_dst_Mbps =
CDEF:nntp_dst_Mbps=nntp_dst_out_bytes,nntp_dst_in_bytes,+,.000008,*

DEF_kazaa_src_out_bytes =
DEF:kazaa_src_out_bytes=$(rrddir)/kazaa_src.rrd:out_bytes:AVERAGE
CDEF_kazaa_src_out_bits = CDEF:kazaa_src_out_bits=kazaa_src_out_bytes,8,*
DEF_kazaa_src_in_bytes =
DEF:kazaa_src_in_bytes=$(rrddir)/kazaa_src.rrd:in_bytes:AVERAGE
CDEF_kazaa_src_in_bits = CDEF:kazaa_src_in_bits=kazaa_src_in_bytes,8,*
CDEF_kazaa_src_Mbps =
CDEF:kazaa_src_Mbps=kazaa_src_out_bytes,kazaa_src_in_bytes,+,.000008,*

DEF_kazaa_dst_out_bytes =
DEF:kazaa_dst_out_bytes=$(rrddir)/kazaa_dst.rrd:out_bytes:AVERAGE
CDEF_kazaa_dst_out_bits = CDEF:kazaa_dst_out_bits=kazaa_dst_out_bytes,8,*
DEF_kazaa_dst_in_bytes =
DEF:kazaa_dst_in_bytes=$(rrddir)/kazaa_dst.rrd:in_bytes:AVERAGE
CDEF_kazaa_dst_in_bits = CDEF:kazaa_dst_in_bits=kazaa_dst_in_bytes,8,*
CDEF_kazaa_dst_Mbps =
CDEF:kazaa_dst_Mbps=kazaa_dst_out_bytes,kazaa_dst_in_bytes,+,.000008,*

DEF_smtp_src_out_bytes =
DEF:smtp_src_out_bytes=$(rrddir)/smtp_src.rrd:out_bytes:AVERAGE
CDEF_smtp_src_out_bits = CDEF:smtp_src_out_bits=smtp_src_out_bytes,8,*
DEF_smtp_src_in_bytes = DEF:smtp_src_in_bytes=$(rrddir)/smtp_src.rrd:in_bytes:AVERAGE
CDEF_smtp_src_in_bits = CDEF:smtp_src_in_bits=smtp_src_in_bytes,8,*
CDEF_smtp_src_Mbps =
CDEF:smtp_src_Mbps=smtp_src_out_bytes,smtp_src_in_bytes,+,.000008,*

DEF_smtp_dst_out_bytes =
DEF:smtp_dst_out_bytes=$(rrddir)/smtp_dst.rrd:out_bytes:AVERAGE
CDEF_smtp_dst_out_bits = CDEF:smtp_dst_out_bits=smtp_dst_out_bytes,8,*
DEF_smtp_dst_in_bytes = DEF:smtp_dst_in_bytes=$(rrddir)/smtp_dst.rrd:in_bytes:AVERAGE
CDEF_smtp_dst_in_bits = CDEF:smtp_dst_in_bits=smtp_dst_in_bytes,8,*
CDEF_smtp_dst_Mbps =
CDEF:smtp_dst_Mbps=smtp_dst_out_bytes,smtp_dst_in_bytes,+,.000008,*

DEF_7070_src_out_bytes =
DEF:x7070_src_out_bytes=$(rrddir)/7070_src.rrd:out_bytes:AVERAGE
CDEF_7070_src_out_bits = CDEF:x7070_src_out_bits=x7070_src_out_bytes,8,*
DEF_7070_src_in_bytes = DEF:x7070_src_in_bytes=$(rrddir)/7070_src.rrd:in_bytes:AVERAGE
CDEF_7070_src_in_bits = CDEF:x7070_src_in_bits=x7070_src_in_bytes,8,*
CDEF_7070_src_Mbps =
CDEF:x7070_src_Mbps=x7070_src_out_bytes,x7070_src_in_bytes,+,.000008,*

DEF_7070_dst_out_bytes =
DEF:x7070_dst_out_bytes=$(rrddir)/7070_dst.rrd:out_bytes:AVERAGE
CDEF_7070_dst_out_bits = CDEF:x7070_dst_out_bits=x7070_dst_out_bytes,8,*
DEF_7070_dst_in_bytes = DEF:x7070_dst_in_bytes=$(rrddir)/7070_dst.rrd:in_bytes:AVERAGE
CDEF_7070_dst_in_bits = CDEF:x7070_dst_in_bits=x7070_dst_in_bytes,8,*
CDEF_7070_dst_Mbps =
CDEF:x7070_dst_Mbps=x7070_dst_out_bytes,x7070_dst_in_bytes,+,.000008,*

DEF_554_src_out_bytes = DEF:x554_src_out_bytes=$(rrddir)/554_src.rrd:out_bytes:AVERAGE
```

```
CDEF_554_src_out_bits = CDEF:x554_src_out_bits=x554_src_out_bytes,8,*
DEF_554_src_in_bytes = DEF:x554_src_in_bytes=$(rrddir)/554_src.rrd:in_bytes:AVERAGE
CDEF_554_src_in_bits = CDEF:x554_src_in_bits=x554_src_in_bytes,8,*
CDEF_554_src_Mbps =
CDEF:x554_src_Mbps=x554_src_out_bytes,x554_src_in_bytes,+,.000008,*

DEF_554_dst_out_bytes = DEF:x554_dst_out_bytes=$(rrddir)/554_dst.rrd:out_bytes:AVERAGE
CDEF_554_dst_out_bits = CDEF:x554_dst_out_bits=x554_dst_out_bytes,8,*
DEF_554_dst_in_bytes = DEF:x554_dst_in_bytes=$(rrddir)/554_dst.rrd:in_bytes:AVERAGE
CDEF_554_dst_in_bits = CDEF:x554_dst_in_bits=x554_dst_in_bytes,8,*
CDEF_554_dst_Mbps =
CDEF:x554_dst_Mbps=x554_dst_out_bytes,x554_dst_in_bytes,+,.000008,*

#DEF_real_out_bytes = DEF:real_out_bytes=$(rrddir)/RealAudio.rrd:out_bytes:AVERAGE
#CDEF_real_out_bits = CDEF:real_out_bits=real_out_bytes,8,*
#DEF_real_in_bytes = DEF:real_in_bytes=$(rrddir)/RealAudio.rrd:in_bytes:AVERAGE
#CDEF_real_in_bits = CDEF:real_in_bits=real_in_bytes,8,*
#CDEF_real_Mbps =
CDEF:real_Mbps=real_out_bytes,real_in_bytes,+,.000008,*,x7070_dst_Mbps,+,x7070_src_Mbp
s,+,x554_dst_Mbps,+,x554_src
_Mbps,+

#DEF_napster_out_bytes = DEF:napster_out_bytes=$(rrddir)/NapUser.rrd:out_bytes:AVERAGE
#CDEF_napster_out_bits = CDEF:napster_out_bits=napster_out_bytes,8,*
#DEF_napster_in_bytes = DEF:napster_in_bytes=$(rrddir)/NapUser.rrd:in_bytes:AVERAGE
#CDEF_napster_in_bits = CDEF:napster_in_bits=napster_in_bytes,8,*
#CDEF_napster_Mbps = CDEF:napster_Mbps=napster_out_bytes,napster_in_bytes,+,.000008,*

services_Mbps$(tag).$(filetype): ftp-data_dst.rrd ftp-data_src.rrd ftpPASV_dst.rrd
ftpPASV_src.rrd ftp_dst.rrd ftp_src.rrd http_dst.
rrd http_src.rrd smtp_dst.rrd smtp_src.rrd total.rrd 554_src.rrd 554_dst.rrd
7070_src.rrd 7070_dst.rrd icmp.rrd
        $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        -v 'megabits/sec' \
        -t '$(organization) Well Known Services Mb/s' \
        -s $(past_hours) \
        --width $(width) \
        --height $(height) \
        $(DEF_total_out_bytes) \
        $(DEF_total_in_bytes) \
        $(CDEF_total_bytes) \
        $(CDEF_total_Mbps) \
        $(CDEF_TOTAL_Mbps) \
        $(DEF_http_src_out_bytes) \
        $(DEF_http_src_in_bytes) \
        $(CDEF_http_src_Mbps) \
        $(DEF_http_dst_out_bytes) \
        $(DEF_http_dst_in_bytes) \
        $(CDEF_http_dst_Mbps) \
        $(DEF_pop_src_out_bytes) \
        $(DEF_pop_src_in_bytes) \
        $(CDEF_pop_src_Mbps) \
        $(DEF_pop_dst_out_bytes) \
        $(DEF_pop_dst_in_bytes) \
        $(CDEF_pop_dst_Mbps) \
        $(DEF_ftp_data_src_out_bytes) \
        $(DEF_ftp_data_src_in_bytes) \
        $(CDEF_ftp_data_src_Mbps) \
        $(DEF_ftp_data_dst_out_bytes) \
        $(DEF_ftp_data_dst_in_bytes) \
        $(CDEF_ftp_data_dst_Mbps) \
        $(DEF_ftpPASV_src_out_bytes) \
        $(DEF_ftpPASV_src_in_bytes) \
        $(CDEF_ftpPASV_src_Mbps) \
        $(DEF_ftpPASV_dst_out_bytes) \
        $(DEF_ftpPASV_dst_in_bytes) \
        $(CDEF_ftpPASV_dst_Mbps) \
        $(CDEF_ftpDATA_src_Mbps) \
        $(CDEF_ftpDATA_dst_Mbps) \
```

```
        $(DEF_smtp_src_out_bytes) \
        $(DEF_smtp_src_in_bytes) \
        $(CDEF_smtp_src_Mbps) \
        $(DEF_smtp_dst_out_bytes) \
        $(DEF_smtp_dst_in_bytes) \
        $(CDEF_smtp_dst_Mbps) \
        $(DEF_7070_src_out_bytes) \
        $(DEF_7070_src_in_bytes) \
        $(CDEF_7070_src_Mbps) \
        $(DEF_7070_dst_out_bytes) \
        $(DEF_7070_dst_in_bytes) \
        $(CDEF_7070_dst_Mbps) \
        $(DEF_554_src_out_bytes) \
        $(DEF_554_src_in_bytes) \
        $(CDEF_554_src_Mbps) \
        $(DEF_554_dst_out_bytes) \
        $(DEF_554_dst_in_bytes) \
        $(CDEF_554_dst_Mbps) \
        $(DEF_icmp_out_bytes) \
        $(DEF_icmp_in_bytes) \
        $(CDEF_icmp_out_Mbps) \
        $(CDEF_icmp_in_Mbps) \
        $(CDEF_icmp_Mbps) \
        'CDEF:http_pct=http_src_Mbps,http_dst_Mbps,+,TOTAL_Mbps,/,100,*' \
        'CDEF:ftp_pct=ftpDATA_src_Mbps,ftpDATA_dst_Mbps,+,TOTAL_Mbps,/,100,*' \
        'CDEF:smtp_pct=smtp_src_Mbps,smtp_dst_Mbps,+,TOTAL_Mbps,/,100,*' \
        'CDEF:icmp_pct=icmp_Mbps,TOTAL_Mbps,/,100,*' \
        'CDEF:pop_pct=pop_src_Mbps,pop_dst_Mbps,+,TOTAL_Mbps,/,100,*' \
        'CDEF:other_pct=100,http_pct,-,ftp_pct,-,smtp_pct,-,icmp_pct,-,pop_pct,-' \
        AREA:http_src_Mbps#ff0000:'HTTP src I/O' \
        STACK:http_dst_Mbps#880000:'HTTP dst I/O' \
        STACK:ftpDATA_src_Mbps#00ff00:'FTP DATA src I/O' \
        STACK:ftpDATA_dst_Mbps#008800:'FTP DATA dst I/O' \
        STACK:pop_src_Mbps#aaaa00:'POP src I/O' \
        STACK:pop_dst_Mbps#0000ff:'POP dst I/O' \
        STACK:smtp_src_Mbps#888888:'SMTP src I/O' \
        STACK:smtp_dst_Mbps#000000:'SMTP dst I/O' \
        STACK:icmp_Mbps#ff8888:'ICMP' \
        LINE1:TOTAL_Mbps#880088:'TOTAL I/O' \
        COMMENT:'\n' \
        COMMENT:'\n' \
        GPRINT:http_pct:AVERAGE:'HTTP %.1lf%%' \
        GPRINT:ftp_pct:AVERAGE:'FTP DATA %.1lf%%' \
        GPRINT:pop_pct:AVERAGE:'POP %.1lf%%' \
        GPRINT:smtp_pct:AVERAGE:'SMTP %.1lf%%' \
        GPRINT:icmp_pct:AVERAGE:'ICMP %.1lf%%' \
        GPRINT:other_pct:AVERAGE:'other %.1lf%%'

io_services_bits$(tag).$(filetype): ftp-data_dst.rrd ftp-data_src.rrd ftpPASV_dst.rrd
ftpPASV_src.rrd ftp_dst.rrd ftp_src.rrd http_d
st.rrd http_src.rrd smtp_dst.rrd smtp_src.rrd total.rrd 554_src.rrd 554_dst.rrd
7070_src.rrd 7070_dst.rrd icmp.rrd kazaa_src.rrd kaz
aa_dst.rrd $(events)
        $(event2vrule) -h $(hours) $(events) $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        -v 'bits per second' \
        -t '$(organization) Well Known Services, +out/-in' \
        -s $(past_hours) \
        --width $(width) \
        --height $(ioheight) \
        --alt-autoscale \
        $(DEF_total_out_bytes) \
        $(DEF_total_in_bytes) \
        $(CDEF_total_out_bits) \
        $(CDEF_total_in_bits),-1,* \
        $(CDEF_total_bytes) \
        $(CDEF_total_Mbps) \
        $(CDEF_TOTAL_Mbps) \
        $(CDEF_TOTAL_out_bits) \
        $(CDEF_TOTAL_in_bits) \
```

```
$(DEF_http_src_out_bytes) \
$(DEF_http_src_in_bytes) \
$(CDEF_http_src_out_bits) \
$(CDEF_http_src_in_bits),-1,* \
$(CDEF_http_src_Mbps) \
$(DEF_http_dst_out_bytes) \
$(DEF_http_dst_in_bytes) \
$(CDEF_http_dst_out_bits) \
$(CDEF_http_dst_in_bits),-1,* \
$(CDEF_http_dst_Mbps) \
$(DEF_pop_src_out_bytes) \
$(DEF_pop_src_in_bytes) \
$(CDEF_pop_src_out_bits) \
$(CDEF_pop_src_in_bits),-1,* \
$(CDEF_pop_src_Mbps) \
$(DEF_pop_dst_out_bytes) \
$(DEF_pop_dst_in_bytes) \
$(CDEF_pop_dst_out_bits) \
$(CDEF_pop_dst_in_bits),-1,* \
$(CDEF_pop_dst_Mbps) \
$(DEF_ftp_data_src_out_bytes) \
$(DEF_ftp_data_src_in_bytes) \
$(CDEF_ftp_data_src_out_bits) \
$(CDEF_ftp_data_src_in_bits),-1,* \
$(CDEF_ftp_data_src_Mbps) \
$(DEF_ftp_data_dst_out_bytes) \
$(DEF_ftp_data_dst_in_bytes) \
$(CDEF_ftp_data_dst_out_bits) \
$(CDEF_ftp_data_dst_in_bits),-1,* \
$(CDEF_ftp_data_dst_Mbps) \
$(DEF_ftpPASV_src_out_bytes) \
$(DEF_ftpPASV_src_in_bytes) \
$(CDEF_ftpPASV_src_out_bits) \
$(CDEF_ftpPASV_src_in_bits),-1,* \
$(CDEF_ftpPASV_src_Mbps) \
$(DEF_ftpPASV_dst_out_bytes) \
$(DEF_ftpPASV_dst_in_bytes) \
$(CDEF_ftpPASV_dst_out_bits) \
$(CDEF_ftpPASV_dst_in_bits),-1,* \
$(CDEF_ftpPASV_dst_Mbps) \
$(CDEF_ftpDATA_src_Mbps) \
$(CDEF_ftpDATA_dst_Mbps) \
$(CDEF_ftpDATA_src_in_bits) \
$(CDEF_ftpDATA_src_out_bits) \
$(CDEF_ftpDATA_dst_in_bits) \
$(CDEF_ftpDATA_dst_out_bits) \
$(DEF_smtp_src_out_bytes) \
$(DEF_smtp_src_in_bytes) \
$(CDEF_smtp_src_out_bits) \
$(CDEF_smtp_src_in_bits),-1,* \
$(CDEF_smtp_src_Mbps) \
$(DEF_smtp_dst_out_bytes) \
$(DEF_smtp_dst_in_bytes) \
$(CDEF_smtp_dst_out_bits) \
$(CDEF_smtp_dst_in_bits),-1,* \
$(CDEF_smtp_dst_Mbps) \
$(DEF_7070_src_out_bytes) \
$(DEF_7070_src_in_bytes) \
$(CDEF_7070_src_out_bits) \
$(CDEF_7070_src_in_bits),-1,* \
$(CDEF_7070_src_Mbps) \
$(DEF_7070_dst_out_bytes) \
$(DEF_7070_dst_in_bytes) \
$(CDEF_7070_dst_out_bits) \
$(CDEF_7070_dst_in_bits),-1,* \
$(CDEF_7070_dst_Mbps) \
$(DEF_554_src_out_bytes) \
$(DEF_554_src_in_bytes) \
$(CDEF_554_src_out_bits) \
$(CDEF_554_src_in_bits),-1,* \
$(CDEF_554_src_Mbps) \
$(DEF_554_dst_out_bytes) \
```

```
        $(DEF_554_dst_in_bytes) \
        $(CDEF_554_dst_out_bits) \
        $(CDEF_554_dst_in_bits),-1,* \
        $(CDEF_554_dst_Mbps) \
        $(DEF_kazaa_src_out_bytes) \
        $(DEF_kazaa_src_in_bytes) \
        $(CDEF_kazaa_src_out_bits) \
        $(CDEF_kazaa_src_in_bits),-1,* \
        $(CDEF_kazaa_src_Mbps) \
        $(DEF_kazaa_dst_out_bytes) \
        $(DEF_kazaa_dst_in_bytes) \
        $(CDEF_kazaa_dst_out_bits) \
        $(CDEF_kazaa_dst_in_bits),-1,* \
        $(CDEF_kazaa_dst_Mbps) \
        $(DEF_icmp_out_bytes) \
        $(DEF_icmp_in_bytes) \
        $(CDEF_icmp_out_bits) \
        $(CDEF_icmp_in_bits),-1,* \
        $(CDEF_icmp_out_Mbps) \
        $(CDEF_icmp_in_Mbps) \
        $(CDEF_icmp_Mbps) \
        'CDEF:http_in_pct=http_src_in_bits,http_dst_in_bits,+,TOTAL_in_bits,/,100,*' \

'CDEF:ftp_in_pct=ftpDATA_src_in_bits,ftpDATA_dst_in_bits,+,TOTAL_in_bits,/,100,*' \
        'CDEF:pop_in_pct=pop_src_in_bits,pop_dst_in_bits,+,TOTAL_in_bits,/,100,*' \
        'CDEF:smtp_in_pct=smtp_src_in_bits,smtp_dst_in_bits,+,TOTAL_in_bits,/,100,*' \
        'CDEF:icmp_in_pct=icmp_in_bits,TOTAL_in_bits,/,100,*' \

'CDEF:kazaa_in_pct=kazaa_src_in_bits,kazaa_dst_in_bits,+,TOTAL_in_bits,/,100,*' \
        'CDEF:other_in_pct=100,http_in_pct,-,kazaa_in_pct,-,ftp_in_pct,-,pop_in_pct,-
,smtp_in_pct,-,icmp_in_pct,-' \

'CDEF:http_out_pct=http_src_out_bits,http_dst_out_bits,+,TOTAL_out_bits,/,100,*' \

'CDEF:ftp_out_pct=ftpDATA_src_out_bits,ftpDATA_dst_out_bits,+,TOTAL_out_bits,/,100,*'
\
        'CDEF:pop_out_pct=pop_src_out_bits,pop_dst_out_bits,+,TOTAL_out_bits,/,100,*'
\

'CDEF:smtp_out_pct=smtp_src_out_bits,smtp_dst_out_bits,+,TOTAL_out_bits,/,100,*' \
        'CDEF:icmp_out_pct=icmp_out_bits,TOTAL_out_bits,/,100,*' \

'CDEF:kazaa_out_pct=kazaa_src_out_bits,kazaa_dst_out_bits,+,TOTAL_out_bits,/,100,*' \
        'CDEF:other_out_pct=100,http_out_pct,-,kazaa_out_pct,-,ftp_out_pct,-
,pop_out_pct,-,smtp_out_pct,-,icmp_out_pct,-' \
        AREA:http_src_out_bits#ff0000:'HTTP src     +' \
        STACK:http_dst_out_bits#880000:'HTTP dst    ' \
        GPRINT:http_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:http_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:kazaa_src_out_bits#FFFF00:'KaZaA* src   +' \
        STACK:kazaa_dst_out_bits#C8C800:'KaZaA* dst  ' \
        GPRINT:kazaa_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:kazaa_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:ftpDATA_src_out_bits#00ff00:'FTP DATA src +' \
        STACK:ftpDATA_dst_out_bits#008800:'FTP DATA dst ' \
        GPRINT:ftp_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:ftp_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:pop_src_out_bits#0000ff:'POP src      +' \
        STACK:pop_dst_out_bits#000088:'POP dst      ' \
        GPRINT:pop_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:pop_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:smtp_src_out_bits#888888:'SMTP src     +' \
        STACK:smtp_dst_out_bits#000000:'SMTP dst     ' \
        GPRINT:smtp_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:smtp_in_pct:AVERAGE:'%.1lf%% In\n' \
        STACK:icmp_out_bits#ff8888:'ICMP                          ' \
        GPRINT:icmp_out_pct:AVERAGE:'%.1lf%% Out' \
        GPRINT:icmp_in_pct:AVERAGE:'%.1lf%% In\n' \
        GPRINT:other_out_pct:AVERAGE:'   Other                          %.1lf%% Out' \
        GPRINT:other_in_pct:AVERAGE:'%.1lf%% In\n' \
        LINE1:TOTAL_out_bits#880088:'TOTAL' \
        AREA:http_src_in_bits#ff0000 \
```

```
        STACK:http_dst_in_bits#880000 \
        STACK:kazaa_src_in_bits#FFFF00 \
        STACK:kazaa_dst_in_bits#C8C800 \
        STACK:ftpDATA_src_in_bits#00ff00 \
        STACK:ftpDATA_dst_in_bits#008800 \
        STACK:pop_src_in_bits#0000ff \
        STACK:pop_dst_in_bits#000088 \
        STACK:smtp_src_in_bits#888888 \
        STACK:smtp_dst_in_bits#000000 \
        STACK:icmp_in_bits#ff8888 \
        LINE1:TOTAL_in_bits#880088 \
        HRULE:0#f5f5f5

DEF_http_src_out_flows =
DEF:http_src_out_flows=$(rrddir)/http_src.rrd:out_flows:AVERAGE
DEF_http_src_in_flows = DEF:http_src_in_flows=$(rrddir)/http_src.rrd:in_flows:AVERAGE
CDEF_http_src_flows = CDEF:http_src_flows=http_src_out_flows,http_src_in_flows,+
DEF_http_dst_out_flows =
DEF:http_dst_out_flows=$(rrddir)/http_dst.rrd:out_flows:AVERAGE
DEF_http_dst_in_flows = DEF:http_dst_in_flows=$(rrddir)/http_dst.rrd:in_flows:AVERAGE
CDEF_http_dst_flows = CDEF:http_dst_flows=http_dst_out_flows,http_dst_in_flows,+

DEF_ftp_data_src_out_flows = DEF:ftp_data_src_out_flows=$(rrddir)/ftp-
data_src.rrd:out_flows:AVERAGE
DEF_ftp_data_src_in_flows = DEF:ftp_data_src_in_flows=$(rrddir)/ftp-
data_src.rrd:in_flows:AVERAGE
CDEF_ftp_data_src_flows =
CDEF:ftp_data_src_flows=ftp_data_src_out_flows,ftp_data_src_in_flows,+
DEF_ftp_data_dst_out_flows = DEF:ftp_data_dst_out_flows=$(rrddir)/ftp-
data_dst.rrd:out_flows:AVERAGE
DEF_ftp_data_dst_in_flows = DEF:ftp_data_dst_in_flows=$(rrddir)/ftp-
data_dst.rrd:in_flows:AVERAGE
CDEF_ftp_data_dst_flows =
CDEF:ftp_data_dst_flows=ftp_data_dst_out_flows,ftp_data_dst_in_flows,+
DEF_ftpPASV_src_out_flows =
DEF:ftpPASV_src_out_flows=$(rrddir)/ftpPASV_src.rrd:out_flows:AVERAGE
DEF_ftpPASV_src_in_flows =
DEF:ftpPASV_src_in_flows=$(rrddir)/ftpPASV_src.rrd:in_flows:AVERAGE
CDEF_ftpPASV_src_flows =
CDEF:ftpPASV_src_flows=ftpPASV_src_out_flows,ftpPASV_src_in_flows,+
DEF_ftpPASV_dst_out_flows =
DEF:ftpPASV_dst_out_flows=$(rrddir)/ftpPASV_dst.rrd:out_flows:AVERAGE
DEF_ftpPASV_dst_in_flows =
DEF:ftpPASV_dst_in_flows=$(rrddir)/ftpPASV_dst.rrd:in_flows:AVERAGE
CDEF_ftpPASV_dst_flows =
CDEF:ftpPASV_dst_flows=ftpPASV_dst_out_flows,ftpPASV_dst_in_flows,+
CDEF_ftpDATA_src_flows = CDEF:ftpDATA_src_flows=ftp_data_src_flows,ftpPASV_src_flows,+
CDEF_ftpDATA_dst_flows = CDEF:ftpDATA_dst_flows=ftp_data_dst_flows,ftpPASV_dst_flows,+

DEF_nntp_src_out_flows =
DEF:nntp_src_out_flows=$(rrddir)/nntp_src.rrd:out_flows:AVERAGE
DEF_nntp_src_in_flows = DEF:nntp_src_in_flows=$(rrddir)/nntp_src.rrd:in_flows:AVERAGE
CDEF_nntp_src_flows = CDEF:nntp_src_flows=nntp_src_out_flows,nntp_src_in_flows,+
DEF_nntp_dst_out_flows =
DEF:nntp_dst_out_flows=$(rrddir)/nntp_dst.rrd:out_flows:AVERAGE
DEF_nntp_dst_in_flows = DEF:nntp_dst_in_flows=$(rrddir)/nntp_dst.rrd:in_flows:AVERAGE
CDEF_nntp_dst_flows = CDEF:nntp_dst_flows=nntp_dst_out_flows,nntp_dst_in_flows,+

DEF_smtp_src_out_flows =
DEF:smtp_src_out_flows=$(rrddir)/smtp_src.rrd:out_flows:AVERAGE
DEF_smtp_src_in_flows = DEF:smtp_src_in_flows=$(rrddir)/smtp_src.rrd:in_flows:AVERAGE
CDEF_smtp_src_flows = CDEF:smtp_src_flows=smtp_src_out_flows,smtp_src_in_flows,+
DEF_smtp_dst_out_flows =
DEF:smtp_dst_out_flows=$(rrddir)/smtp_dst.rrd:out_flows:AVERAGE
DEF_smtp_dst_in_flows = DEF:smtp_dst_in_flows=$(rrddir)/smtp_dst.rrd:in_flows:AVERAGE
CDEF_smtp_dst_flows = CDEF:smtp_dst_flows=smtp_dst_out_flows,smtp_dst_in_flows,+

DEF_7070_src_out_flows =
DEF:x7070_src_out_flows=$(rrddir)/7070_src.rrd:out_flows:AVERAGE
DEF_7070_src_in_flows = DEF:x7070_src_in_flows=$(rrddir)/7070_src.rrd:in_flows:AVERAGE
CDEF_7070_src_flows = CDEF:x7070_src_flows=x7070_src_out_flows,x7070_src_in_flows,+
```

```
DEF_7070_dst_out_flows =
DEF:x7070_dst_out_flows=$(rrddir)/7070_dst.rrd:out_flows:AVERAGE
DEF_7070_dst_in_flows = DEF:x7070_dst_in_flows=$(rrddir)/7070_dst.rrd:in_flows:AVERAGE
CDEF_7070_dst_flows = CDEF:x7070_dst_flows=x7070_dst_out_flows,x7070_dst_in_flows,+

DEF_554_src_out_flows = DEF:x554_src_out_flows=$(rrddir)/554_src.rrd:out_flows:AVERAGE
DEF_554_src_in_flows = DEF:x554_src_in_flows=$(rrddir)/554_src.rrd:in_flows:AVERAGE
CDEF_554_src_flows = CDEF:x554_src_flows=x554_src_out_flows,x554_src_in_flows,+
DEF_554_dst_out_flows = DEF:x554_dst_out_flows=$(rrddir)/554_dst.rrd:out_flows:AVERAGE
DEF_554_dst_in_flows = DEF:x554_dst_in_flows=$(rrddir)/554_dst.rrd:in_flows:AVERAGE
CDEF_554_dst_flows = CDEF:x554_dst_flows=x554_dst_out_flows,x554_dst_in_flows,+

DEF_real_out_flows = DEF:real_out_flows=$(rrddir)/RealAudio.rrd:out_flows:AVERAGE
DEF_real_in_flows = DEF:real_in_flows=$(rrddir)/RealAudio.rrd:in_flows:AVERAGE
CDEF_REAL_in_flows =
CDEF:REAL_in_flows=real_in_flows,x7070_src_in_flows,+,x7070_dst_in_flows,+,x554_src_in
_flows,+,x554_dst_in_flow
s,+
CDEF_REAL_out_flows =
CDEF:REAL_out_flows=real_out_flows,x7070_src_out_flows,+,x7070_dst_out_flows,+,x554_sr
c_out_flows,+,x554_dst_o
ut_flows,+
CDEF_REAL_flows =
CDEF:REAL_flows=real_out_flows,real_in_flows,+,x7070_src_flows,+,x7070_dst_flows,+,x55
4_src_flows,+,x554_dst_flows
,+

services_flows$(tag).$(filetype): ftp-data_dst.rrd ftp-data_src.rrd ftp_dst.rrd
ftp_src.rrd http_dst.rrd http_src.rrd nntp_dst.rrd n
ntp_src.rrd smtp_dst.rrd smtp_src.rrd total.rrd 554_src.rrd 554_dst.rrd 7070_src.rrd
7070_dst.rrd RealAudio.rrd icmp.rrd MCAST.rrd
        $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        -v 'flows/sec' \
        -t '$(organization) Well Known Services Flows' \
        -s $(past_hours) \
        --width $(width) \
        --height $(height)  \
        $(DEF_total_out_flows) \
        $(DEF_total_in_flows) \
        $(CDEF_total_flows) \
        $(DEF_http_src_out_flows) \
        $(DEF_http_src_in_flows) \
        $(CDEF_http_src_flows) \
        $(DEF_http_dst_out_flows) \
        $(DEF_http_dst_in_flows) \
        $(CDEF_http_dst_flows) \
        $(DEF_ftp_data_src_out_flows) \
        $(DEF_ftp_data_src_in_flows) \
        $(CDEF_ftp_data_src_flows) \
        $(DEF_ftp_data_dst_out_flows) \
        $(DEF_ftp_data_dst_in_flows) \
        $(CDEF_ftp_data_dst_flows) \
        $(DEF_ftpPASV_src_out_flows) \
        $(DEF_ftpPASV_src_in_flows) \
        $(CDEF_ftpPASV_src_flows) \
        $(DEF_ftpPASV_dst_out_flows) \
        $(DEF_ftpPASV_dst_in_flows) \
        $(CDEF_ftpPASV_dst_flows) \
        $(CDEF_ftpDATA_src_flows) \
        $(CDEF_ftpDATA_dst_flows) \
        $(DEF_nntp_src_out_flows) \
        $(DEF_nntp_src_in_flows) \
        $(CDEF_nntp_src_flows) \
        $(DEF_nntp_dst_out_flows) \
        $(DEF_nntp_dst_in_flows) \
        $(CDEF_nntp_dst_flows) \
        $(DEF_smtp_src_out_flows) \
        $(DEF_smtp_src_in_flows) \
        $(CDEF_smtp_src_flows) \
```

```
        $(DEF_smtp_dst_out_flows) \
        $(DEF_smtp_dst_in_flows) \
        $(CDEF_smtp_dst_flows) \
        $(DEF_7070_src_out_flows) \
        $(DEF_7070_src_in_flows) \
        $(CDEF_7070_src_flows) \
        $(DEF_7070_dst_out_flows) \
        $(DEF_7070_dst_in_flows) \
        $(CDEF_7070_dst_flows) \
        $(DEF_554_src_out_flows) \
        $(DEF_554_src_in_flows) \
        $(CDEF_554_src_flows) \
        $(DEF_554_dst_out_flows) \
        $(DEF_554_dst_in_flows) \
        $(CDEF_554_dst_flows) \
        $(DEF_real_out_flows) \
        $(DEF_real_in_flows) \
        $(CDEF_REAL_flows) \
        $(DEF_icmp_out_flows) \
        $(DEF_icmp_in_flows) \
        $(CDEF_icmp_flows) \
        $(DEF_MCAST_in_flows) \
        $(DEF_MCAST_out_flows) \
        $(CDEF_MCAST_flows) \
        $(CDEF_TOTAL_flows) \
        AREA:http_src_flows#ff0000:'HTTP src I/O' \
        STACK:http_dst_flows#880000:'HTTP dst I/O' \
        STACK:ftpDATA_src_flows#00ff00:'FTP DATA src I/O' \
        STACK:ftpDATA_dst_flows#008800:'FTP DATA dst I/O' \
        STACK:nntp_src_flows#0000ff:'NNTP src I/O' \
        STACK:nntp_dst_flows#000088:'NNTP dst I/O' \
        STACK:REAL_flows#00ffff:'RealServer I/O' \
        STACK:smtp_src_flows#888888:'SMTP src I/O' \
        STACK:smtp_dst_flows#000000:'SMTP dst I/O' \
        STACK:icmp_flows#ff8888:'ICMP' \
        STACK:MCAST_in_flows#aaaa00:'MCAST in' \
        STACK:MCAST_out_flows#555500:'MCAST out' \
        LINE1:TOTAL_flows#880088:'TOTAL I/O'

io_services_flows$(tag).$(filetype): ftp-data_dst.rrd ftp-data_src.rrd ftp_dst.rrd
ftp_src.rrd http_dst.rrd http_src.rrd nntp_dst.rr
d nntp_src.rrd smtp_dst.rrd smtp_src.rrd total.rrd 554_src.rrd 554_dst.rrd
7070_src.rrd 7070_dst.rrd RealAudio.rrd icmp.rrd MCAST.rr
d $(events)
        $(event2vrule) -h $(hours) $(events) $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        -v 'flows per second' \
        -t '$(organization) Well Known Services Flows, +out/-in' \
        -s $(past_hours) \
        --width $(width) \
        --height $(ioheight) \
        --alt-autoscale \
        $(DEF_total_out_flows) \
        $(DEF_total_in_flows) \
        CDEF:total_in_flows_neg=total_in_flows,-1,* \
        $(CDEF_total_flows) \
        $(DEF_http_src_out_flows) \
        $(DEF_http_src_in_flows) \
        CDEF:http_src_in_flows_neg=http_src_in_flows,-1,* \
        $(CDEF_http_src_flows) \
        $(DEF_http_dst_out_flows) \
        $(DEF_http_dst_in_flows) \
        CDEF:http_dst_in_flows_neg=http_dst_in_flows,-1,* \
        $(CDEF_http_dst_flows) \
        $(DEF_ftp_data_src_out_flows) \
        $(DEF_ftp_data_src_in_flows) \
        CDEF:ftp_data_src_in_flows_neg=ftp_data_src_in_flows,-1,* \
        $(CDEF_ftp_data_src_flows) \
        $(DEF_ftp_data_dst_out_flows) \
        $(DEF_ftp_data_dst_in_flows) \
```

```
CDEF:ftp_data_dst_in_flows_neg=ftp_data_dst_in_flows,-1,* \
$(CDEF_ftp_data_dst_flows) \
$(DEF_ftpPASV_src_out_flows) \
$(DEF_ftpPASV_src_in_flows) \
CDEF:ftpPASV_src_in_flows_neg=ftpPASV_src_in_flows,-1,* \
$(CDEF_ftpPASV_src_flows) \
$(DEF_ftpPASV_dst_out_flows) \
$(DEF_ftpPASV_dst_in_flows) \
CDEF:ftpPASV_dst_in_flows_neg=ftpPASV_dst_in_flows,-1,* \
$(CDEF_ftpPASV_dst_flows) \
$(CDEF_ftpDATA_src_flows) \
$(CDEF_ftpDATA_dst_flows) \
CDEF:ftpDATA_dst_out_flows=ftp_data_dst_out_flows,ftpPASV_dst_out_flows,+ \
CDEF:ftpDATA_src_out_flows=ftp_data_src_out_flows,ftpPASV_src_out_flows,+ \
CDEF:ftpDATA_dst_in_flows=ftp_data_dst_in_flows,ftpPASV_dst_in_flows,+ \
CDEF:ftpDATA_dst_in_flows_neg=ftpDATA_dst_in_flows,-1,* \
CDEF:ftpDATA_src_in_flows=ftp_data_src_in_flows,ftpPASV_src_in_flows,+ \
CDEF:ftpDATA_src_in_flows_neg=ftpDATA_src_in_flows,-1,* \
$(DEF_nntp_src_out_flows) \
$(DEF_nntp_src_in_flows) \
CDEF:nntp_src_in_flows_neg=nntp_src_in_flows,-1,* \
$(CDEF_nntp_src_flows) \
$(DEF_nntp_dst_out_flows) \
$(DEF_nntp_dst_in_flows) \
CDEF:nntp_dst_in_flows_neg=nntp_dst_in_flows,-1,* \
$(CDEF_nntp_dst_flows) \
$(DEF_smtp_src_out_flows) \
$(DEF_smtp_src_in_flows) \
CDEF:smtp_src_in_flows_neg=smtp_src_in_flows,-1,* \
$(CDEF_smtp_src_flows) \
$(DEF_smtp_dst_out_flows) \
$(DEF_smtp_dst_in_flows) \
CDEF:smtp_dst_in_flows_neg=smtp_dst_in_flows,-1,* \
$(CDEF_smtp_dst_flows) \
$(DEF_7070_src_out_flows) \
$(DEF_7070_src_in_flows) \
$(CDEF_7070_src_flows) \
$(DEF_7070_dst_out_flows) \
$(DEF_7070_dst_in_flows) \
$(CDEF_7070_dst_flows) \
$(DEF_554_src_out_flows) \
$(DEF_554_src_in_flows) \
$(CDEF_554_src_flows) \
$(DEF_554_dst_out_flows) \
$(DEF_554_dst_in_flows) \
$(CDEF_554_dst_flows) \
$(DEF_real_out_flows) \
$(DEF_real_in_flows) \
$(CDEF_REAL_in_flows) \
CDEF:REAL_in_flows_neg=REAL_in_flows,-1,* \
$(CDEF_REAL_out_flows) \
$(CDEF_REAL_flows) \
$(DEF_icmp_out_flows) \
$(DEF_icmp_in_flows) \
CDEF:icmp_in_flows_neg=icmp_in_flows,-1,* \
$(CDEF_icmp_flows) \
$(DEF_MCAST_in_flows) \
CDEF:MCAST_in_flows_neg=MCAST_in_flows,-1,* \
$(DEF_MCAST_out_flows) \
$(CDEF_MCAST_flows) \
$(CDEF_TOTAL_in_flows) \
CDEF:TOTAL_in_flows_neg=TOTAL_in_flows,-1,* \
$(CDEF_TOTAL_out_flows) \
$(CDEF_TOTAL_flows) \
AREA:http_src_out_flows#ff0000:'HTTP src' \
STACK:http_dst_out_flows#880000:'HTTP dst' \
STACK:ftpDATA_src_out_flows#00ff00:'FTP DATA src' \
STACK:ftpDATA_dst_out_flows#008800:'FTP DATA dst' \
STACK:nntp_src_out_flows#0000ff:'NNTP src' \
STACK:nntp_dst_out_flows#000088:'NNTP dst' \
STACK:REAL_out_flows#00ffff:'RealServer' \
STACK:smtp_src_out_flows#888888:'SMTP src' \
```

```
        STACK:smtp_dst_out_flows#000000:'SMTP dst' \
        STACK:icmp_out_flows#ff8888:'ICMP' \
        STACK:MCAST_out_flows#aaaa00:'MCAST' \
        LINE1:TOTAL_out_flows#880088:'TOTAL' \
        AREA:http_src_in_flows_neg#ff0000 \
        STACK:http_dst_in_flows_neg#880000 \
        STACK:ftpDATA_src_in_flows_neg#00ff00 \
        STACK:ftpDATA_dst_in_flows_neg#008800 \
        STACK:nntp_src_in_flows_neg#0000ff \
        STACK:nntp_dst_in_flows_neg#000088 \
        STACK:REAL_in_flows_neg#00ffff \
        STACK:smtp_src_in_flows_neg#888888 \
        STACK:smtp_dst_in_flows_neg#000000 \
        STACK:icmp_in_flows_neg#ff8888 \
        STACK:MCAST_in_flows_neg#aaaa00 \
        LINE1:TOTAL_in_flows_neg#880088 \
        HRULE:0#f5f5f5

DEF_http_src_out_pkts = DEF:http_src_out_pkts=$(rrddir)/http_src.rrd:out_pkts:AVERAGE
DEF_http_src_in_pkts = DEF:http_src_in_pkts=$(rrddir)/http_src.rrd:in_pkts:AVERAGE
CDEF_http_src_pkts = CDEF:http_src_pkts=http_src_out_pkts,http_src_in_pkts,+
DEF_http_dst_out_pkts = DEF:http_dst_out_pkts=$(rrddir)/http_dst.rrd:out_pkts:AVERAGE
DEF_http_dst_in_pkts = DEF:http_dst_in_pkts=$(rrddir)/http_dst.rrd:in_pkts:AVERAGE
CDEF_http_dst_pkts = CDEF:http_dst_pkts=http_dst_out_pkts,http_dst_in_pkts,+

DEF_ftp_data_src_out_pkts = DEF:ftp_data_src_out_pkts=$(rrddir)/ftp-
data_src.rrd:out_pkts:AVERAGE
DEF_ftp_data_src_in_pkts = DEF:ftp_data_src_in_pkts=$(rrddir)/ftp-
data_src.rrd:in_pkts:AVERAGE
CDEF_ftp_data_src_pkts =
CDEF:ftp_data_src_pkts=ftp_data_src_out_pkts,ftp_data_src_in_pkts,+
DEF_ftp_data_dst_out_pkts = DEF:ftp_data_dst_out_pkts=$(rrddir)/ftp-
data_dst.rrd:out_pkts:AVERAGE
DEF_ftp_data_dst_in_pkts = DEF:ftp_data_dst_in_pkts=$(rrddir)/ftp-
data_dst.rrd:in_pkts:AVERAGE
CDEF_ftp_data_dst_pkts =
CDEF:ftp_data_dst_pkts=ftp_data_dst_out_pkts,ftp_data_dst_in_pkts,+
DEF_ftpPASV_src_out_pkts =
DEF:ftpPASV_src_out_pkts=$(rrddir)/ftpPASV_src.rrd:out_pkts:AVERAGE
DEF_ftpPASV_src_in_pkts =
DEF:ftpPASV_src_in_pkts=$(rrddir)/ftpPASV_src.rrd:in_pkts:AVERAGE
CDEF_ftpPASV_src_pkts =
CDEF:ftpPASV_src_pkts=ftpPASV_src_out_pkts,ftp_data_src_in_pkts,+
DEF_ftpPASV_dst_out_pkts =
DEF:ftpPASV_dst_out_pkts=$(rrddir)/ftpPASV_dst.rrd:out_pkts:AVERAGE
DEF_ftpPASV_dst_in_pkts =
DEF:ftpPASV_dst_in_pkts=$(rrddir)/ftpPASV_dst.rrd:in_pkts:AVERAGE
CDEF_ftpPASV_dst_pkts =
CDEF:ftpPASV_dst_pkts=ftpPASV_dst_out_pkts,ftpPASV_dst_in_pkts,+
CDEF_ftpDATA_src_pkts = CDEF:ftpDATA_src_pkts=ftp_data_src_pkts,ftp_data_src_pkts,+
CDEF_ftpDATA_dst_pkts = CDEF:ftpDATA_dst_pkts=ftp_data_dst_pkts,ftp_data_dst_pkts,+


DEF_nntp_src_out_pkts = DEF:nntp_src_out_pkts=$(rrddir)/nntp_src.rrd:out_pkts:AVERAGE
DEF_nntp_src_in_pkts = DEF:nntp_src_in_pkts=$(rrddir)/nntp_src.rrd:in_pkts:AVERAGE
CDEF_nntp_src_pkts = CDEF:nntp_src_pkts=nntp_src_out_pkts,nntp_src_in_pkts,+
DEF_nntp_dst_out_pkts = DEF:nntp_dst_out_pkts=$(rrddir)/nntp_dst.rrd:out_pkts:AVERAGE
DEF_nntp_dst_in_pkts = DEF:nntp_dst_in_pkts=$(rrddir)/nntp_dst.rrd:in_pkts:AVERAGE
CDEF_nntp_dst_pkts = CDEF:nntp_dst_pkts=nntp_dst_out_pkts,nntp_dst_in_pkts,+

DEF_smtp_src_out_pkts = DEF:smtp_src_out_pkts=$(rrddir)/smtp_src.rrd:out_pkts:AVERAGE
DEF_smtp_src_in_pkts = DEF:smtp_src_in_pkts=$(rrddir)/smtp_src.rrd:in_pkts:AVERAGE
CDEF_smtp_src_pkts = CDEF:smtp_src_pkts=smtp_src_out_pkts,smtp_src_in_pkts,+
DEF_smtp_dst_out_pkts = DEF:smtp_dst_out_pkts=$(rrddir)/smtp_dst.rrd:out_pkts:AVERAGE
DEF_smtp_dst_in_pkts = DEF:smtp_dst_in_pkts=$(rrddir)/smtp_dst.rrd:in_pkts:AVERAGE
CDEF_smtp_dst_pkts = CDEF:smtp_dst_pkts=smtp_dst_out_pkts,smtp_dst_in_pkts,+

DEF_7070_src_out_pkts = DEF:x7070_src_out_pkts=$(rrddir)/7070_src.rrd:out_pkts:AVERAGE
DEF_7070_src_in_pkts = DEF:x7070_src_in_pkts=$(rrddir)/7070_src.rrd:in_pkts:AVERAGE
CDEF_7070_src_pkts = CDEF:x7070_src_pkts=x7070_src_out_pkts,x7070_src_in_pkts,+
DEF_7070_dst_out_pkts = DEF:x7070_dst_out_pkts=$(rrddir)/7070_dst.rrd:out_pkts:AVERAGE
DEF_7070_dst_in_pkts = DEF:x7070_dst_in_pkts=$(rrddir)/7070_dst.rrd:in_pkts:AVERAGE
```

```
CDEF_7070_dst_pkts = CDEF:x7070_dst_pkts=x7070_dst_out_pkts,x7070_dst_in_pkts,+

DEF_554_src_out_pkts = DEF:x554_src_out_pkts=$(rrddir)/554_src.rrd:out_pkts:AVERAGE
DEF_554_src_in_pkts = DEF:x554_src_in_pkts=$(rrddir)/554_src.rrd:in_pkts:AVERAGE
CDEF_554_src_pkts = CDEF:x554_src_pkts=x554_src_out_pkts,x554_src_in_pkts,+
DEF_554_dst_out_pkts = DEF:x554_dst_out_pkts=$(rrddir)/554_dst.rrd:out_pkts:AVERAGE
DEF_554_dst_in_pkts = DEF:x554_dst_in_pkts=$(rrddir)/554_dst.rrd:in_pkts:AVERAGE
CDEF_554_dst_pkts = CDEF:x554_dst_pkts=x554_dst_out_pkts,x554_dst_in_pkts,+

DEF_real_out_pkts = DEF:real_out_pkts=$(rrddir)/RealAudio.rrd:out_pkts:AVERAGE
DEF_real_in_pkts = DEF:real_in_pkts=$(rrddir)/RealAudio.rrd:in_pkts:AVERAGE
CDEF_REAL_in_pkts =
CDEF:REAL_in_pkts=real_in_pkts,x7070_src_in_pkts,+,x7070_dst_in_pkts,+,x554_src_in_pkt
s,+,x554_dst_in_pkts,+
CDEF_REAL_out_pkts =
CDEF:REAL_out_pkts=real_out_pkts,x7070_src_out_pkts,+,x7070_dst_out_pkts,+,x554_src_ou
t_pkts,+,x554_dst_out_pkt
s,+
CDEF_REAL_pkts =
CDEF:REAL_pkts=real_out_pkts,real_in_pkts,+,x7070_src_pkts,+,x7070_dst_pkts,+,x554_src
_pkts,+,x554_dst_pkts,+

services_pkts$(tag).$(filetype): ftp-data_dst.rrd ftp-data_src.rrd ftp_dst.rrd
ftp_src.rrd http_dst.rrd http_src.rrd nntp_dst.rrd nn
tp_src.rrd smtp_dst.rrd smtp_src.rrd total.rrd 554_src.rrd 554_dst.rrd 7070_src.rrd
7070_dst.rrd RealAudio.rrd icmp.rrd MCAST.rrd
        $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        -v 'packets/sec' \
        -t '$(organization) Well Known Services Packets' \
        -s $(past_hours) \
        --width $(width) \
        --height $(height) \
        $(DEF_total_out_pkts) \
        $(DEF_total_in_pkts) \
        $(CDEF_total_pkts) \
        $(DEF_http_src_out_pkts) \
        $(DEF_http_src_in_pkts) \
        $(CDEF_http_src_pkts) \
        $(DEF_http_dst_out_pkts) \
        $(DEF_http_dst_in_pkts) \
        $(CDEF_http_dst_pkts) \
        $(DEF_ftp_data_src_out_pkts) \
        $(DEF_ftp_data_src_in_pkts) \
        $(CDEF_ftp_data_src_pkts) \
        $(DEF_ftp_data_dst_out_pkts) \
        $(DEF_ftp_data_dst_in_pkts) \
        $(CDEF_ftp_data_dst_pkts) \
        $(DEF_ftpPASV_src_out_pkts) \
        $(DEF_ftpPASV_src_in_pkts) \
        $(CDEF_ftpPASV_src_pkts) \
        $(DEF_ftpPASV_dst_out_pkts) \
        $(DEF_ftpPASV_dst_in_pkts) \
        $(CDEF_ftpPASV_dst_pkts) \
        $(CDEF_ftpDATA_src_pkts) \
        $(CDEF_ftpDATA_dst_pkts) \
        $(DEF_nntp_src_out_pkts) \
        $(DEF_nntp_src_in_pkts) \
        $(CDEF_nntp_src_pkts) \
        $(DEF_nntp_dst_out_pkts) \
        $(DEF_nntp_dst_in_pkts) \
        $(CDEF_nntp_dst_pkts) \
        $(DEF_smtp_src_out_pkts) \
        $(DEF_smtp_src_in_pkts) \
        $(CDEF_smtp_src_pkts) \
        $(DEF_smtp_dst_out_pkts) \
        $(DEF_smtp_dst_in_pkts) \
        $(CDEF_smtp_dst_pkts) \
        $(DEF_7070_src_out_pkts) \
        $(DEF_7070_src_in_pkts) \
```

```
        $(CDEF_7070_src_pkts) \
        $(DEF_7070_dst_out_pkts) \
        $(DEF_7070_dst_in_pkts) \
        $(CDEF_7070_dst_pkts) \
        $(DEF_554_src_out_pkts) \
        $(DEF_554_src_in_pkts) \
        $(CDEF_554_src_pkts) \
        $(DEF_554_dst_out_pkts) \
        $(DEF_554_dst_in_pkts) \
        $(CDEF_554_dst_pkts) \
        $(DEF_real_out_pkts) \
        $(DEF_real_in_pkts) \
        $(CDEF_REAL_pkts) \
        $(DEF_icmp_out_pkts) \
        $(DEF_icmp_in_pkts) \
        $(CDEF_icmp_pkts) \
        $(DEF_MCAST_in_pkts) \
        $(DEF_MCAST_out_pkts) \
        $(CDEF_MCAST_pkts) \
        $(CDEF_TOTAL_pkts) \
        AREA:http_src_pkts#ff0000:'HTTP src I/O' \
        STACK:http_dst_pkts#880000:'HTTP dst I/O' \
        STACK:ftpDATA_src_pkts#00ff00:'FTP DATA src I/O' \
        STACK:ftpDATA_dst_pkts#008800:'FTP DATA dst I/O' \
        STACK:nntp_src_pkts#0000ff:'NNTP src I/O' \
        STACK:nntp_dst_pkts#000088:'NNTP dst I/O' \
        STACK:REAL_pkts#00ffff:'RealServer I/O' \
        STACK:smtp_src_pkts#888888:'SMTP src I/O' \
        STACK:smtp_dst_pkts#000000:'SMTP dst I/O' \
        STACK:icmp_pkts#ff8888:'ICMP' \
        STACK:MCAST_in_pkts#aaaa00:'MCAST in' \
        STACK:MCAST_out_pkts#555500:'MCAST out' \
        LINE1:TOTAL_pkts#880088:'TOTAL I/O'

io_services_pkts$(tag).$(filetype): ftp-data_dst.rrd ftp-data_src.rrd ftp_dst.rrd
ftp_src.rrd http_dst.rrd http_src.rrd nntp_dst.rrd
 nntp_src.rrd smtp_dst.rrd smtp_src.rrd total.rrd 554_src.rrd 554_dst.rrd 7070_src.rrd
7070_dst.rrd RealAudio.rrd icmp.rrd MCAST.rrd
 $(events)
        $(event2vrule) -h $(hours) $(events) $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        -v 'packets per second' \
        -t '$(organization) Well Known Services Packets, +out/-in' \
        -s $(past_hours) \
        --width $(width) \
        --height $(ioheight) \
        --alt-autoscale \
        $(DEF_total_out_pkts) \
        $(DEF_total_in_pkts) \
        CDEF:total_in_pkts_neg=total_in_pkts,-1,* \
        $(CDEF_total_pkts) \
        $(DEF_http_src_out_pkts) \
        $(DEF_http_src_in_pkts) \
        CDEF:http_src_in_pkts_neg=http_src_in_pkts,-1,* \
        $(CDEF_http_src_pkts) \
        $(DEF_http_dst_out_pkts) \
        $(DEF_http_dst_in_pkts) \
        CDEF:http_dst_in_pkts_neg=http_dst_in_pkts,-1,* \
        $(CDEF_http_dst_pkts) \
        $(DEF_ftp_data_src_out_pkts) \
        $(DEF_ftp_data_src_in_pkts) \
        CDEF:ftp_data_src_in_pkts_neg=ftp_data_src_in_pkts,-1,* \
        $(CDEF_ftp_data_src_pkts) \
        $(DEF_ftp_data_dst_out_pkts) \
        $(DEF_ftp_data_dst_in_pkts) \
        CDEF:ftp_data_dst_in_pkts_neg=ftp_data_dst_in_pkts,-1,* \
        $(CDEF_ftp_data_dst_pkts) \
        $(DEF_ftpPASV_src_out_pkts) \
        $(DEF_ftpPASV_src_in_pkts) \
        CDEF:ftpPASV_src_in_pkts_neg=ftpPASV_src_in_pkts,-1,* \
```

```
$(CDEF_ftpPASV_src_pkts) \
$(DEF_ftpPASV_dst_out_pkts) \
$(DEF_ftpPASV_dst_in_pkts) \
CDEF:ftpPASV_dst_in_pkts_neg=ftpPASV_dst_in_pkts,-1,* \
$(CDEF_ftpPASV_dst_pkts) \
$(CDEF_ftpDATA_src_pkts) \
$(CDEF_ftpDATA_dst_pkts) \
CDEF:ftpDATA_dst_out_pkts=ftp_data_dst_out_pkts,ftpPASV_dst_out_pkts,+ \
CDEF:ftpDATA_src_out_pkts=ftp_data_src_out_pkts,ftpPASV_src_out_pkts,+ \
CDEF:ftpDATA_dst_in_pkts=ftp_data_dst_in_pkts,ftpPASV_dst_in_pkts,+ \
CDEF:ftpDATA_dst_in_pkts_neg=ftpDATA_dst_in_pkts,-1,* \
CDEF:ftpDATA_src_in_pkts=ftp_data_src_in_pkts,ftpPASV_src_in_pkts,+ \
CDEF:ftpDATA_src_in_pkts_neg=ftpDATA_src_in_pkts,-1,* \
$(DEF_nntp_src_out_pkts) \
$(DEF_nntp_src_in_pkts) \
CDEF:nntp_src_in_pkts_neg=nntp_src_in_pkts,-1,* \
$(CDEF_nntp_src_pkts) \
$(DEF_nntp_dst_out_pkts) \
$(DEF_nntp_dst_in_pkts) \
CDEF:nntp_dst_in_pkts_neg=nntp_dst_in_pkts,-1,* \
$(CDEF_nntp_dst_pkts) \
$(DEF_smtp_src_out_pkts) \
$(DEF_smtp_src_in_pkts) \
CDEF:smtp_src_in_pkts_neg=smtp_src_in_pkts,-1,* \
$(CDEF_smtp_src_pkts) \
$(DEF_smtp_dst_out_pkts) \
$(DEF_smtp_dst_in_pkts) \
CDEF:smtp_dst_in_pkts_neg=smtp_dst_in_pkts,-1,* \
$(CDEF_smtp_dst_pkts) \
$(DEF_7070_src_out_pkts) \
$(DEF_7070_src_in_pkts) \
$(CDEF_7070_src_pkts) \
$(DEF_7070_dst_out_pkts) \
$(DEF_7070_dst_in_pkts) \
$(CDEF_7070_dst_pkts) \
$(DEF_554_src_out_pkts) \
$(DEF_554_src_in_pkts) \
$(CDEF_554_src_pkts) \
$(DEF_554_dst_out_pkts) \
$(DEF_554_dst_in_pkts) \
$(CDEF_554_dst_pkts) \
$(DEF_real_out_pkts) \
$(DEF_real_in_pkts) \
$(CDEF_REAL_in_pkts) \
CDEF:REAL_in_pkts_neg=REAL_in_pkts,-1,* \
$(CDEF_REAL_out_pkts) \
$(CDEF_REAL_pkts) \
$(DEF_icmp_out_pkts) \
$(DEF_icmp_in_pkts) \
CDEF:icmp_in_pkts_neg=icmp_in_pkts,-1,* \
$(CDEF_icmp_pkts) \
$(DEF_MCAST_in_pkts) \
CDEF:MCAST_in_pkts_neg=MCAST_in_pkts,-1,* \
$(DEF_MCAST_out_pkts) \
$(CDEF_MCAST_pkts) \
$(CDEF_TOTAL_in_pkts) \
CDEF:TOTAL_in_pkts_neg=TOTAL_in_pkts,-1,* \
$(CDEF_TOTAL_out_pkts) \
$(CDEF_TOTAL_pkts) \
AREA:http_src_out_pkts#ff0000:'HTTP src' \
STACK:http_dst_out_pkts#880000:'HTTP dst' \
STACK:ftpDATA_src_out_pkts#00ff00:'FTP DATA src' \
STACK:ftpDATA_dst_out_pkts#008800:'FTP DATA dst' \
STACK:nntp_src_out_pkts#0000ff:'NNTP src' \
STACK:nntp_dst_out_pkts#000088:'NNTP dst' \
STACK:REAL_out_pkts#00ffff:'RealServer' \
STACK:smtp_src_out_pkts#888888:'SMTP src' \
STACK:smtp_dst_out_pkts#000000:'SMTP dst' \
STACK:icmp_out_pkts#ff8888:'ICMP' \
STACK:MCAST_out_pkts#aaaa00:'MCAST' \
LINE1:TOTAL_out_pkts#880088:'TOTAL' \
AREA:http_src_in_pkts_neg#ff0000 \
```

```
        STACK:http_dst_in_pkts_neg#880000 \
        STACK:ftpDATA_src_in_pkts_neg#00ff00 \
        STACK:ftpDATA_dst_in_pkts_neg#008800 \
        STACK:nntp_src_in_pkts_neg#0000ff \
        STACK:nntp_dst_in_pkts_neg#000088 \
        STACK:REAL_in_pkts_neg#00ffff \
        STACK:smtp_src_in_pkts_neg#888888 \
        STACK:smtp_dst_in_pkts_neg#000000 \
        STACK:icmp_in_pkts_neg#ff8888 \
        STACK:MCAST_in_pkts_neg#aaaa00 \
        LINE1:TOTAL_in_pkts_neg#880088 \
        HRULE:0#f5f5f5

# AS to AS stuff:
DEF_vBNS2WiscNet_bytes =
DEF:vBNS2WiscNet_bytes=$(rrddir)/vBNS2WiscNet.rrd:bytes:AVERAGE
CDEF_vBNS2WiscNet_Mbps = CDEF:vBNS2WiscNet_Mbps=vBNS2WiscNet_bytes,.000008,*

DEF_WiscNet2vBNS_bytes =
DEF:WiscNet2vBNS_bytes=$(rrddir)/WiscNet2vBNS.rrd:bytes:AVERAGE
CDEF_WiscNet2vBNS_Mbps = CDEF:WiscNet2vBNS_Mbps=WiscNet2vBNS_bytes,.000008,*

DEF_vBNS2Campus_bytes = DEF:vBNS2Campus_bytes=$(rrddir)/vBNS2Campus.rrd:bytes:AVERAGE
CDEF_vBNS2Campus_Mbps = CDEF:vBNS2Campus_Mbps=vBNS2Campus_bytes,.000008,*

DEF_Campus2vBNS_bytes = DEF:Campus2vBNS_bytes=$(rrddir)/Campus2vBNS.rrd:bytes:AVERAGE
CDEF_Campus2vBNS_Mbps = CDEF:Campus2vBNS_Mbps=Campus2vBNS_bytes,.000008,*

DEF_WiscNet2Campus_bytes =
DEF:WiscNet2Campus_bytes=$(rrddir)/WiscNet2Campus.rrd:bytes:AVERAGE
CDEF_WiscNet2Campus_Mbps = CDEF:WiscNet2Campus_Mbps=WiscNet2Campus_bytes,.000008,*

DEF_Campus2WiscNet_bytes =
DEF:Campus2WiscNet_bytes=$(rrddir)/Campus2WiscNet.rrd:bytes:AVERAGE
CDEF_Campus2WiscNet_Mbps = CDEF:Campus2WiscNet_Mbps=Campus2WiscNet_bytes,.000008,*

DEF_Campus2Campus_bytes =
DEF:Campus2Campus_bytes=$(rrddir)/Campus2Campus.rrd:bytes:AVERAGE
CDEF_Campus2Campus_Mbps = CDEF:Campus2Campus_Mbps=Campus2Campus_bytes,.000008,*

DEF_Campus2Berbee_bytes =
DEF:Campus2Berbee_bytes=$(rrddir)/Campus2Berbee.rrd:bytes:AVERAGE
CDEF_Campus2Berbee_Mbps = CDEF:Campus2Berbee_Mbps=Campus2Berbee_bytes,.000008,*

DEF_Berbee2Campus_bytes =
DEF:Berbee2Campus_bytes=$(rrddir)/Berbee2Campus.rrd:bytes:AVERAGE
CDEF_Berbee2Campus_Mbps = CDEF:Berbee2Campus_Mbps=Berbee2Campus_bytes,.000008,*

DEF_Campus2Chorus_bytes =
DEF:Campus2Chorus_bytes=$(rrddir)/Campus2Chorus.rrd:bytes:AVERAGE
CDEF_Campus2Chorus_Mbps = CDEF:Campus2Chorus_Mbps=Campus2Chorus_bytes,.000008,*

DEF_Chorus2Campus_bytes =
DEF:Chorus2Campus_bytes=$(rrddir)/Chorus2Campus.rrd:bytes:AVERAGE
CDEF_Chorus2Campus_Mbps = CDEF:Chorus2Campus_Mbps=Chorus2Campus_bytes,.000008,*

DEF_Campus2TDS_bytes = DEF:Campus2TDS_bytes=$(rrddir)/Campus2TDS.rrd:bytes:AVERAGE
CDEF_Campus2TDS_Mbps = CDEF:Campus2TDS_Mbps=Campus2TDS_bytes,.000008,*

DEF_TDS2Campus_bytes = DEF:TDS2Campus_bytes=$(rrddir)/TDS2Campus.rrd:bytes:AVERAGE
CDEF_TDS2Campus_Mbps = CDEF:TDS2Campus_Mbps=TDS2Campus_bytes,.000008,*

DEF_Campus2ESnet_bytes =
DEF:Campus2ESnet_bytes=$(rrddir)/Campus2ESnet.rrd:bytes:AVERAGE
CDEF_Campus2ESnet_Mbps = CDEF:Campus2ESnet_Mbps=Campus2ESnet_bytes,.000008,*

DEF_ESnet2Campus_bytes =
DEF:ESnet2Campus_bytes=$(rrddir)/ESnet2Campus.rrd:bytes:AVERAGE
CDEF_ESnet2Campus_Mbps = CDEF:ESnet2Campus_Mbps=ESnet2Campus_bytes,.000008,*

as2as_Mbps$(tag).$(filetype): total.rrd Berbee2Campus.rrd Campus2Berbee.rrd
Campus2Campus.rrd Campus2Chorus.rrd Campus2ESnet.rrd Cam
```

```
pus2TDS.rrd Campus2WiscNet.rrd Campus2vBNS.rrd Chorus2Campus.rrd ESnet2Campus.rrd
TDS2Campus.rrd WiscNet2Campus.rrd vBNS2Campus.rrd
MCAST.rrd
        $(rrdtool) graph \
        $@ \
        --interlaced \
        --imgformat $(IMGFORMAT) \
        -v 'megabits per second' \
        -t '$(organization) AS to AS, Mb/s' \
        -s $(past_hours) \
        --width $(width) \
        --height $(height) \
        $(DEF_total_out_bytes) \
        $(DEF_total_in_bytes) \
        $(CDEF_total_bytes) \
        $(CDEF_total_Mbps) \
        $(DEF_vBNS2WiscNet_bytes) \
        $(CDEF_vBNS2WiscNet_Mbps) \
        $(DEF_WiscNet2vBNS_bytes) \
        $(CDEF_WiscNet2vBNS_Mbps) \
        $(DEF_WiscNet2Campus_bytes) \
        $(CDEF_WiscNet2Campus_Mbps) \
        $(DEF_Campus2WiscNet_bytes) \
        $(CDEF_Campus2WiscNet_Mbps) \
        $(DEF_vBNS2Campus_bytes) \
        $(CDEF_vBNS2Campus_Mbps) \
        $(DEF_Campus2vBNS_bytes) \
        $(CDEF_Campus2vBNS_Mbps) \
        $(DEF_Campus2Campus_bytes) \
        $(CDEF_Campus2Campus_Mbps) \
        $(DEF_Campus2Berbee_bytes) \
        $(CDEF_Campus2Berbee_Mbps) \
        $(DEF_Berbee2Campus_bytes) \
        $(CDEF_Berbee2Campus_Mbps) \
        $(DEF_Campus2Chorus_bytes) \
        $(CDEF_Campus2Chorus_Mbps) \
        $(DEF_Chorus2Campus_bytes) \
        $(CDEF_Chorus2Campus_Mbps) \
        $(DEF_Campus2TDS_bytes) \
        $(CDEF_Campus2TDS_Mbps) \
        $(DEF_TDS2Campus_bytes) \
        $(CDEF_TDS2Campus_Mbps) \
        $(DEF_Campus2ESnet_bytes) \
        $(CDEF_Campus2ESnet_Mbps) \
        $(DEF_ESnet2Campus_bytes) \
        $(CDEF_ESnet2Campus_Mbps) \
        $(DEF_MCAST_in_bytes) \
        $(DEF_MCAST_out_bytes) \
        $(CDEF_MCAST_in_Mbps) \
        $(CDEF_MCAST_out_Mbps) \
        $(CDEF_MCAST_Mbps) \
        $(CDEF_TOTAL_Mbps) \
        'CDEF:mcast_pct=MCAST_Mbps,TOTAL_Mbps,/,100,*' \

'CDEF:WiscNet_pct=WiscNet2Campus_Mbps,Campus2WiscNet_Mbps,+,TOTAL_Mbps,/,100,*' \

'CDEF:vBNS_pct=vBNS2Campus_Mbps,Campus2vBNS_Mbps,+,MCAST_Mbps,+,TOTAL_Mbps,/,100,*' \
        'CDEF:Berbee_pct=Berbee2Campus_Mbps,Campus2Berbee_Mbps,+,TOTAL_Mbps,/,100,*' \
        'CDEF:Chorus_pct=Chorus2Campus_Mbps,Campus2Chorus_Mbps,+,TOTAL_Mbps,/,100,*' \
        'CDEF:TDS_pct=TDS2Campus_Mbps,Campus2TDS_Mbps,+,TOTAL_Mbps,/,100,*' \
        'CDEF:ESnet_pct=ESnet2Campus_Mbps,Campus2ESnet_Mbps,+,TOTAL_Mbps,/,100,*' \
        AREA:MCAST_Mbps#aaaa00:'MCAST I/O' \
        STACK:Campus2WiscNet_Mbps#00ff00:'Campus to WiscNet' \
        STACK:WiscNet2Campus_Mbps#008800:'WiscNet to Campus' \
        STACK:Campus2vBNS_Mbps#0000ff:'Campus to vBNS' \
        STACK:vBNS2Campus_Mbps#000088:'vBNS to Campus' \
        STACK:Campus2ESnet_Mbps#00ffff:'Campus to ESnet' \
        STACK:ESnet2Campus_Mbps#008888:'ESnet to Campus' \
        STACK:Campus2Chorus_Mbps#ff00ff:'Campus to Chorus' \
        STACK:Chorus2Campus_Mbps#880088:'Chorus to Campus' \
        STACK:Campus2TDS_Mbps#ffa500:'Campus to TDS' \
        STACK:TDS2Campus_Mbps#885200:'TDS to Campus' \
```

```
        STACK:Campus2Berbee_Mbps#ff7f50:'Campus to Berbee' \
        STACK:Berbee2Campus_Mbps#884225:'Berbee to Campus' \
        LINE1:TOTAL_Mbps#ff0000:'TOTAL Inter-AS & MCAST' \
        STACK:Campus2Campus_Mbps#880000:'Intra-Campus (at the peering point, in
addition to Inter-AS)' \
        COMMENT:'\n' \
        COMMENT:'\n' \
        GPRINT:WiscNet_pct:AVERAGE:'WiscNet %.1lf%%' \
        GPRINT:vBNS_pct:AVERAGE:'vBNS+MCAST %.1lf%%' \
        GPRINT:ESnet_pct:AVERAGE:'ESnet %.1lf%%' \
        GPRINT:Chorus_pct:AVERAGE:'Chorus %.1lf%%' \
        GPRINT:TDS_pct:AVERAGE:'TDS %.1lf%%' \
        GPRINT:Berbee_pct:AVERAGE:'Berbee %.1lf%%'

.SUFFIXES: .rrd .xml

.rrd.xml:
        $(rrdtool) dump $< > $@ || rm -f $@
```

Programa que muestra el gráfico de uso del enlace de Internet (/usr/OV/httpd/cgi-bin/wanGraph-2.pl.cgi). Este programa usa los datos proporcionados por Tivoli Netview.

```
#!/usr/local/bin/perl
# File name: wanGraph-2.pl.cgi
# Last Modified: 7/16/96
# Written by Otto J. Helweg - ohelweg@ahc.e-mail.com or otto@ottomatic.com
#   As Uncle Larry says, "Abundance in all things - material, emotional,
#     intellectual, spiritual - is the goal of any first-rate soul."

# This program combines the following functions into one cgi-bin:
#    - displays a html form
#    - processes the form and extracts the variables
#    - executes 'snmpColDump' to get snmpCollected data in text format
#    - sorts and parses the data based on the particular router and interface
#    - feeds the data to 'gnuplot' to get a bitmap in 'pbm' format
#    - uses 'ppmtogif' to translate the 'pbm' to 'gif' and 'PostScript'
#    - feeds the 'gif' back to the browser

# This program assumes that the standard snmpCollect configuration has been
#   modified to collect on Interface Utilization (ignore this if you're using
#   HP OpenView).

# Note: This PERL program is intended to be run from a 'table aware' WWW browser
#   (ie: Netscape v.1.1 or later).

# ** You will need to fix the form pointer below to your http location.
$webLocation = "http://192.168.253.1/TME10/NetView/images";

# Gets the present working directory (I don't trust my httpd).
#$pwd = `pwd`;
$pwd = "/usr/OV/web/httpd/htdocs/TME10/NetView/images";
#chop($pwd);

# Load the desired Networks into an array. Note that the important fields; 'Node Name'
and
#   'Interface Number' (from MIB2) must match the configuration of the NetView
snmpCollect daemon. If snmpCollect
#   is not collecting data on the node/interface that you are requesting, this program
will explode!
```

```
&netLoad;

if ($ENV{"QUERY_STRING"} ne "") {
  # Converts the QUERY_STRING into a more readable array
  &queryParse;
  # Displays the entry form in html to the browser
  &formDisplay;
} else {
  &formDisplay;
  exit();
}

# Creates a temporary directory if it doesn't already exist and checks for needed
programs.
umask(0000);
if (!opendir(TEMP,"$pwd/wanGraph-tmp")) {
  if(!mkdir("$pwd/wanGraph-tmp",0777)) {
    @errors = "<CENTER>Error creating the temp directory, make sure this program has
'rwx' permissions in the directory</CENTER>\n";
  }
}
@reqProg = ("xdate","gnuplot","ppmtogif");
foreach $reqProg (@reqProg) {
  $output = `which $reqProg`;
  @outSplit = split(/ /,$output);
  if ($outSplit[0] eq "no") {
    @errors = (@errors,"<CENTER>Couldn't find $reqProg, a required program, please add
to path.</CENTER>\n");
  }
}
if($errors[0] ne "") {
  print("<HR>\n");
  print("Present Working Directory: $pwd<BR>\n");
  foreach $errors (@errors) {
    print("$errors\n");
  }
  exit();
}

# Create a Temporary file name for the image based
#   on the date/time. This forces the browser to
#   load the image rather then using it's cache.
$TimeStamp=`date +%H%M%S`;
chop($TimeStamp);

# Delete previously created temp files if an 'in-use' token file is not present.
if (! -e "$pwd/wanGraph-tmp/in-use.token") {
  $token = "$pwd/wanGraph-tmp/in-use.token";
  `rm -f $pwd/wanGraph-tmp/*`;
  # Leave a token incase someone else tries to generate a graph at the same time.
  #   There is always the possibility that they will try to look at the same router
  #   interface on the same router, in which case data will get stepped on.
  `touch $token`;
}

# I'm jumping through a bunch of hoops to try and figure out the actual start and end
dates
#   to graph.
#   - First I get the current time in seconds from 1970.
$currentTime = time();
$secDay = 86400;
# Then by using the 'localtime' function, I can determine the hour/minute/second of
the
#   current time, and I use that to calculate an end time (in seconds) which falls on
midnight.
@endDate = localtime($currentTime);
$endTime = ($secDay - (($endDate[2] * 60 * 60) + ($endDate[1] * 60) + $endDate[0])) +
$currentTime;
# The dateCalc function simply returns a date string that corresponds to my time in
seconds from 1970.
$EndDate = &dateCalc ($endTime);
```

```
if ($lookup{timeframe} eq "today") {
  $startTime=$endTime - $secDay;
  $StartDate = &dateCalc ($startTime);
} elsif ($lookup{timeframe} eq "yesterday") {
  $startTime=$endTime - ($secDay * 2);
  $StartDate = &dateCalc ($startTime);
} elsif ($lookup{timeframe} eq "week") {
  $startTime=$endTime - ($secDay * 8);
  $StartDate = &dateCalc ($startTime);
} elsif ($lookup{timeframe} eq "month") {
  $startTime=$endTime - ($secDay * 31);
  $StartDate = &dateCalc ($startTime);
} elsif ($lookup{timeframe} eq "other") {
  $EndDate = $lookup{edate};
  $StartDate = $lookup{sdate};
}

# Gather the desired segments and remove the empties
$count = 0;
@segments = ("$lookup{Segment1}","$lookup{Segment2}");
foreach $segments (@segments) {
  if ($segments ne "none") {
    @netGrep = grep(/$segments/,@netInfo);
    @netSplit = split(/,/,$netGrep[0]);
    $netRouter[$count] = $netSplit[1];
    $netIntf[$count] = $netSplit[2];
    $netDesc[$count] = $segments;
    $netSpeed[$count] = $netSplit[3];

    # The following commands create the datasets to be graphed for gnuplot
    if ($lookup{timeframe} ne "other") {
      `/usr/OV/bin/snmpColDump -t
/usr/OV/databases/snmpCollect/ifInOctets.$netIntf[$count] | grep $netRouter[$count] |
awk -F\"\\t\
" '{ if (\$4 > $startTime) { printf(\"%s\\t%s\\t%s\\n\", \$1, \$2, \$3*8) }}' >
$pwd/wanGraph-tmp/seg_in_$netIntf[$count]_$netRouter
[$count]`;
      `/usr/OV/bin/snmpColDump -t
/usr/OV/databases/snmpCollect/ifOutOctets.$netIntf[$count] | grep $netRouter[$count] |
awk -F\"\\t
\" '{ if (\$4 > $startTime) { printf(\"%s\\t%s\\t%s\\n\", \$1, \$2, \$3*8) }}' >
$pwd/wanGraph-tmp/seg_out_$netIntf[$count]_$netRout
er[$count]`;
    } else {
      `/usr/OV/bin/snmpColDump
/usr/OV/databases/snmpCollect/ifInOctets.$netIntf[$count] | grep $netRouter[$count] |
awk -F\"\\t\" '
{ printf(\"%s\\t%s\\t%s\\n\", \$1, \$2, \$3*8) }' > $pwd/wanGraph-
tmp/seg_in_$netIntf[$count]_$netRouter[$count]`;
      `/usr/OV/bin/snmpColDump
/usr/OV/databases/snmpCollect/ifOutOctets.$netIntf[$count] | grep $netRouter[$count] |
awk -F\"\\t\"
'{ printf(\"%s\\t%s\\t%s\\n\", \$1, \$2, \$3*8) }' > $pwd/wanGraph-
tmp/seg_out_$netIntf[$count]_$netRouter[$count]`;
    }

    $count++;
  }
}
$numseg = $count;

# Creates a command file for gnuplot
open (GPFILE, ">$pwd/wanGraph-tmp/$TimeStamp.gp");
print GPFILE ("set data style lines\n");
if ($lookup{color} eq "on") {
  print GPFILE ("set term pbm small color\n");
} else {
  print GPFILE ("set term pbm small monochrome\n");
}
print GPFILE ("set xdata time\n");
print GPFILE ("set timefmt \"%m/%d/%y %H:%M:%S\"\n");
print GPFILE ("set format x \"%m/%d\\n%H:%M\"\n");
```

```
if ($lookup{grid} eq "on") {
  print GPFILE ("set grid\n");
}
print GPFILE ("set title \"Wan Utilization\"\n");
print GPFILE ("set ylabel \"Utilization (Bps)\"\n");
print GPFILE ("set xrange [ \"$StartDate\":\"$EndDate\" ]\n");
print GPFILE ("set output '$pwd/wanGraph-tmp/$TimeStamp.pbm'\n");
$circuitSpeed = $netSpeed[0] * 1000;
if($lookup{speed} eq "on") {
  $circuitSpeed = $netSpeed[0] * 1000;
  print GPFILE ("plot $circuitSpeed t '$netSpeed[0] Kb Link', '$pwd/wanGraph-
tmp/seg_in_$netIntf[0]_$netRouter[0]' using 1:4 t '$net
Desc[0]-in'");
  print GPFILE (", '$pwd/wanGraph-tmp/seg_out_$netIntf[0]_$netRouter[0]' using 1:4 t
'$netDesc[0]-out'");
  for ($count=1; $count < $numseg; $count++) {
    $circuitSpeed = $netSpeed[$count] * 1000;
    print GPFILE (", $circuitSpeed t '$netSpeed[$count] Kb Link', '$pwd/wanGraph-
tmp/seg_in_$netIntf[$count]_$netRouter[$count]' usi
ng 1:4 t '$netDesc[$count]-in'");
    print GPFILE (", '$pwd/wanGraph-tmp/seg_out_$netIntf[$count]_$netRouter[$count]'
using 1:4 t '$netDesc[$count]-out'");
  }
  print GPFILE ("\n");
} else {
  print GPFILE ("plot '$pwd/wanGraph-tmp/seg_in_$netIntf[0]_$netRouter[0]' using 1:4 t
'$netDesc[0]-in'");
  print GPFILE (", '$pwd/wanGraph-tmp/seg_out_$netIntf[0]_$netRouter[0]' using 1:4 t
'$netDesc[0]-out'");
  for ($count=1; $count < $numseg; $count++) {
    print GPFILE (", '$pwd/wanGraph-tmp/seg_in_$netIntf[$count]_$netRouter[$count]'
using 1:4 t '$netDesc[$count]-in'");
    print GPFILE (", '$pwd/wanGraph-tmp/seg_out_$netIntf[$count]_$netRouter[$count]'
using 1:4 t '$netDesc[$count]-out'");
  }
  print GPFILE ("\n");
}
if ($lookup{color} eq "on") {
  print GPFILE ("set term postscript color solid\n");
} else {
  print GPFILE ("set term postscript monochrome dashed\n");
}
print GPFILE ("set output '$pwd/wanGraph-tmp/$TimeStamp.ps'\n");
if($lookup{speed} eq "on") {
  $circuitSpeed = $netSpeed[0] * 1000;
  print GPFILE ("plot $circuitSpeed t '$netSpeed[0] Kb Link', '$pwd/wanGraph-
tmp/seg_in_$netIntf[0]_$netRouter[0]' using 1:4 t '$net
Desc[0]-in'");
  print GPFILE (", '$pwd/wanGraph-tmp/seg_out_$netIntf[0]_$netRouter[0]' using 1:4 t
'$netDesc[0]-out'");
  for ($count=1; $count < $numseg; $count++) {
    $circuitSpeed = $netSpeed[$count] * 1000;
    print GPFILE (", $circuitSpeed t '$netSpeed[$count] Kb Link', '$pwd/wanGraph-
tmp/seg_in_$netIntf[$count]_$netRouter[$count]' usi
ng 1:4 t '$netDesc[$count]-in'");
    print GPFILE (", '$pwd/wanGraph-tmp/seg_out_$netIntf[$count]_$netRouter[$count]'
using 1:4 t '$netDesc[$count]-out'");
  }
  print GPFILE ("\n");
} else {
  print GPFILE ("plot '$pwd/wanGraph-tmp/seg_in_$netIntf[0]_$netRouter[0]' using 1:4 t
'$netDesc[0]-in'");
  print GPFILE (", '$pwd/wanGraph-tmp/seg_out_$netIntf[0]_$netRouter[0]' using 1:4 t
'$netDesc[0]-out'");
  for ($count=1; $count < $numseg; $count++) {
    print GPFILE (", '$pwd/wanGraph-tmp/seg_in_$netIntf[$count]_$netRouter[$count]'
using 1:4 t '$netDesc[$count]-in'");
    print GPFILE (", '$pwd/wanGraph-tmp/seg_out_$netIntf[$count]_$netRouter[$count]'
using 1:4 t '$netDesc[$count]-out'");
  }
  print GPFILE ("\n");
}
```

```
close (GPFILE);

# Execute gnuplot
`gnuplot $pwd/wanGraph-tmp/$TimeStamp.gp`;

# Excute ppmtogif in order to convert the 'pbm' output from gnuplot to 'gif'
`ppmtogif -interlace $pwd/wanGraph-tmp/$TimeStamp.pbm > $pwd/wanGraph-
tmp/$TimeStamp.gif 2> /dev/null`;

# Remove the PBM file (just in case)
`rm $pwd/wanGraph-tmp/$TimeStamp.pbm`;

# Display the graph image to the browser
print("<HR>\n");
print("<center>Bajar:\n");
print("<a href=\"$webLocation/wanGraph-tmp/$TimeStamp.ps\">GrM-afico
(Postscript)</A>,\n");
print("<a href=\"$webLocation/wanGraph-tmp/seg_in_$netIntf[0]_$netRouter[0]\">Datos
Enlace Uno (entrada)</A>,\n");
print("<a href=\"$webLocation/wanGraph-tmp/seg_out_$netIntf[0]_$netRouter[0]\">Datos
Enlace Uno (salida)</A><BR>\n");
print("<HR>\n");
print("<center><img src=\"$webLocation/wanGraph-tmp/$TimeStamp.gif\"
></center><BR>\n");

$elapsed=(time() - $^T);
print("<HR>Tiempo de cM-alculo transcurrido: $elapsed segundos\n");

# Print the Footer
print("</BODY>\n");
print("</HTML>\n");

# Remove the token because we're finished.
if ($token ne "") {
  unlink ($token);
}

# ----------------------------------------------------
# We're all DONE!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# ----------------------------------------------------

sub netLoad {
  # The format of this array is as follows:
  #     Network Description,Node Name,Interface Number,Link Speed(Kb)
  # ie: Williston,hac46b,1,128
  @netInfo=("Internet,diamante.espol.edu.ec,3,10000","Penas,192.188.59.16,2,512");

  @netInfo = sort (@netInfo);
}

sub queryParse {
  # Begin parsing the QUERY_STRING from the form input...
  #   Expects something like;
  #     foo=www%21&bar=hello&baz=blah

  # Split the string into each of the key-value pairs
  (@fields) = split('&', $ENV{'QUERY_STRING'});

  # For each of these key-value pairs, decode the value
  for $field (@fields) {
    # Split the key-value pair on the equal sign.
    ($name, $value) = split('=', $field);

    # Change all plus signs to spaces. This is an
    # remnant of ISINDEX
    $value =~ y/\+/ /;

    # Change all carrage-return/line feeds to spaces. This is an
    # remnant of ISINDEX?
    # $value =~ y/\%0D\%0A/ /;

    # Decode the value & removes % escapes.
```

```
    $value =~ s/%([\da-f]{1,2})/pack(C,hex($1))/eig;

    # Create the appropriate entry in the
    # associative array lookup
    if(defined $lookup{$name}) {
      # If there are multiple values, seperate
      # them by newlines
      $lookup{$name} .= "\n".$value;
    } else {
      $lookup{$name} = $value;
    }
  }
}

sub formDisplay {
  # Note that there are a bunch of if/then statements spattered throughout this sub-
routine. They
  #   attempt to rebuild the form in the previous state (leaving check-boxes
unchecked, etc.).
  print("Content-type: text/html\n");
  print("\n");
  print("<HTML>\n");
  print("<HEAD>\n");
  print("<TITLE>GrM-afica Wan</TITLE>\n");
  print("</HEAD>\n");
  print("<BODY BGCOLOR=\"#FFFFFF\">\n");
  print("<FORM METHOD=\"GET\" ACTION=\"/cgi-bin/wanGraph-2.pl.cgi\">\n");
  print("<CENTER><TABLE><TR><TD COLSPAN=\"2\">\n");
  print("<CENTER><H2>GeneraciM-sn de GrM-aficos de UtilizaciM-sn de
Enlaces</H2></CENTER>\n");
  print("<DIV ALIGN=\"RIGHT\"><I><A HREF=\"/cgi-bin/lanGraph.pl.cgi\">GrM-aficos
LAN</A> | GrM-aficos WAN</I></DIV></TD></TR>\n");
  print("<TR><TD COLSPAN=2><TABLE BORDER WIDTH=\"100%\"><TR><TD WIDTH=\"80%\">\n");
  if ($lookup{timeframe} eq "today" || $ENV{"QUERY_STRING"} eq "") {
    print("<input type=radio name=\"timeframe\" value=\"today\"
checked><B>Hoy</B>\n");
  } else {
    print("<input type=radio name=\"timeframe\" value=\"today\"><B>Hoy</B>\n");
  }
  print("<br>\n");
  if ($lookup{timeframe} eq "yesterday") {
    print("<input type=radio name=\"timeframe\" value=\"yesterday\" checked><B>Ayer y
Hoy</B>\n");
  } else {
    print("<input type=radio name=\"timeframe\" value=\"yesterday\"><B>Ayer y
Hoy</B>\n");
  }
  print("<br>\n");
  if ($lookup{timeframe} eq "week") {
    print("<input type=radio name=\"timeframe\" value=\"week\" checked><B>La semana
pasada y Hoy</B>\n");
  } else {
    print("<input type=radio name=\"timeframe\" value=\"week\"><B>La semana pasada y
Hoy</B>\n");
  }
  print("<br>\n");
  if ($lookup{timeframe} eq "month") {
    print("<input type=radio name=\"timeframe\" value=\"month\" checked><B>El mes
pasado y Hoy</B>\n");
  } else {
    print("<input type=radio name=\"timeframe\" value=\"month\"><B>El mes pasado y
Hoy</B>\n");
  }
  print("<br>\n");
  if ($lookup{timeframe} eq "other") {
    print("<input type=radio name=\"timeframe\" value=\"other\"
checked><B>Otro</B>\n");
  } else {
    print("<input type=radio name=\"timeframe\" value=\"other\"><B>Otro</B>\n");
  }
  if ($lookup{sdate} ne "") {
```

```
    print("<input size=10 name=\"sdate\" value=\"$lookup{sdate}\"><B><i> -Fecha
Inicial</i></B>\n");
  } else {
    print("<input size=10 name=\"sdate\"><B><i> -Fecha Inicial</i></B>\n");
  }
  if ($lookup{edate} ne "") {
    print("<input size=10 name=\"edate\" value=\"$lookup{edate}\"><B><i> -Fecha
Final<BR><CENTER>Formato: MM/DD/YY</B></i><BR>\n");
  } else {
    print("<input size=10 name=\"edate\"><B><i> -Fecha Final<BR><CENTER>Formato:
MM/DD/YY</B></i><BR>\n");
  }
  print("</CENTER>");
  print("</TD><TD VALIGN=TOP ALIGN=RIGHT WDTH=\"20%\">\n");
  if ($lookup{grid} eq "on" || $ENV{"QUERY_STRING"} eq "") {
    print("<B>Malla XY</B><input type=checkbox name=\"grid\" checked><BR>\n");
  } else {
    print("<B>Malla XY</B><input type=checkbox name=\"grid\"><BR>\n");
  }
  if ($lookup{color} eq "on" || $ENV{"QUERY_STRING"} eq "") {
    print("<B>Color</B><input type=checkbox name=\"color\" checked><BR>\n");
  } else {
    print("<B>Color</B><input type=checkbox name=\"color\"><BR>\n");
  }
  if ($lookup{speed} eq "on") {
    print("<B>Velocidad</B><input type=checkbox name=\"speed\"
checked><BR></TD></TR>\n");
  } else {
    print("<B>Velocidad</B><input type=checkbox name=\"speed\"><BR></TD></TR>\n");
  }
  print("</TABLE><CENTER><TABLE BORDER WIDTH=\"100%\"><TR><TD
ALIGN=\"CENTER\"><B>Enlace Uno</B><BR><select name=\"Segment1\">\n");
  if ($lookup{Segment1} ne "") {
    print("<option selected>$lookup{Segment1}\n");
  }
  foreach $netInfo (@netInfo) {
    @netDesc = split(/,/,$netInfo);
    print("<option>$netDesc[0]\n");
  }
  print("</select></CENTER></TD>\n");
  print("<TD COLSPAN=2><CENTER><B>Enlace Dos</B><BR><select name=\"Segment2\">\n");
  if ($lookup{Segment2} ne "") {
    print("<option selected>$lookup{Segment2}\n");
  } else {
    print("<option selected>none\n");
  }
  print("<option>none\n");
  foreach $netInfo (@netInfo) {
    @netDesc = split(/,/,$netInfo);
    print("<option>$netDesc[0]\n");
  }
  print("</select></CENTER></TD></TR></TABLE></TD></TR>\n");
  print("<TR><TD><CENTER><INPUT TYPE=\"submit\" VALUE=\"Enviar\"></CENTER></TD>\n");
  print("<TD><CENTER><INPUT TYPE=\"reset\" VALUE=\"Borrar\"></CENTER></TD></TR>\n");
  print("</TABLE></CENTER>\n");
  print("</FORM>\n");
}


# The dateCalc function simply returns a date string that corresponds to a time in
seconds from 1970.
sub dateCalc {
  local ($calcTime) = @_;
  local (@calcDate);
  @calcDate = localtime($calcTime);
  $calcDate[4]++;
  if ($calcDate[4] < "10") {
    $calcDate[4]="0$calcDate[4]";
  }
  if ($calcDate[3] < "10") {
    $calcDate[3]="0$calcDate[3]";
  }
```

```
  if ($calcDate[5] < "10") {
    $calcDate[5]="0$calcDate[5]";
  }
  if ($calcDate[5] >= "100") {
    $calcDate[5]=$calcDate[5]-100;
  }
  $calcDate="$calcDate[4]/$calcDate[3]/$calcDate[5]";
}
```

BIBLIOGRAFÍA

1. Allan Leinwand, Karen Fang, Network Management A Practical Perspective, Addison-Wesley, Segunda Edición, 1996

2. Kornel Terplan, Benchmarking for Effective Network Management, McGraw-Hill, 1996

3. Richard H. Baker, Network Security, McGraw-Hill, 1995

4. Craig Hunt, TCP/IP Network Administration, O'Reilly & Associates, Segunda Edición, 1998

5. Mark A. Miller, Managing Internetworks with SNMP, M&T Books, 1993

6. Tivoli, TME10 Framework Planning and Installation Guide Versión 3.6, Tivoli, 1998

7. Tivoli, TME10 Framework User's Guide Versión 3.6, Tivoli, Septiembre 1998

8. Tivoli, TME10 NetView Installation and Configuration Versión 5, Tivoli, Junio 1997

9. Tivoli, TME10 NetView Administrator's Guide Versión 5, Tivoli, Junio 1997

10. ITSO, Examples of using TME10 NetView for AIX and TME10 NetView for Windows NT, IBM, Segunda Edición, Mayo 1997

11. Douglas W. Stevenson, Network Management What it is and what it isn't, http://suresh_kr.tripod.com/snmp/dstevenson.html

12. FlowScan, Network Traffic Flow Visualization and Reporting Tool, http://www.caida.org/tools/utilities/flowscan/

13. Cisco, CiscoWorks LAN Management Solution, http://www.cisco.com/en/US/products/sw/cscowork/ps2425/index.html