UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

GESTIÓN DE REDES CON SNMP

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR :

MILTON JESÚS DÍAZ OLIVET

PROMOCIÓN
1999 – II

LIMA – PERÚ
2005

DEDICO ESTE TRABAJO A:
MIS PADRES, POR SU APOYO
INCONDICIONAL EN MI CARRERA.

GESTIÓN DE REDES CON SNMP

SUMARIO

El presente informe de suficiencia nos ofrece una visión amplia y comprensible de las tecnologías y los estándares de la gestión de redes, así como permite un conocimiento actualizado de las principales plataformas y arquitecturas de los protocolos de Administración de redes basados en los modelos de referencia OSI y TCP/IP.

El informe se estructura en dos partes. La primera nos muestra un enfoque teórica en la que se exponen en detalle los diferentes estándares de gestión de red, los conceptos y arquitectura SNMP, bases de información (MIBs), el modelo de comunicación SNMP, gestión de grandes redes y supervisión remota.

La segunda parte nos muestra una aplicación práctica en la que se detallan las demostraciones del funcionamiento de algunas aplicaciones para la gestión y una aplicación para la monitorización de algunos equipos de Voz sobre IP de Telefonica del Peru, desarrollada por el graduando con el software MRTG.

ÍNDICE

CAPÍTULO IV

ASN.1 (Notación de Sintaxis Abstracta.1)

PRÓLOGO

La proliferación de redes de datos a lo largo de la década de los 90, tanto LANs como WANs, y el interfuncionamiento entre ellas hace que los aspectos relativos a su control y gestión cada vez sean más tenidos en cuenta, convirtiéndoseen algo a lo que todos los responsables de redes han de prestar una gran atención.

Este punto, el de gestión de red, es uno de los más controvertidos en teleinformática, ya que, prácticamente, no existe una solución única, aceptada por todos y que sea fácilmente implantable. Las soluciones existentes suelen ser propietarias -Netview de IBM, OpenView de HP, etc.- lo que hace que en una red compleja, formada por equipos multifabricante, no exista un único sistema capaz de realizar la gestión completa de la misma, necesitándose varias plataformas -una por cada fabricante-, lo que dificulta y complica enormemente la labor del gestor de red.

Con la idea de presentar una solución única, válida para cualquier tipo de red, varios grupos de normalización están trabajando en ello y, aunque hay dos tendencias claras (SNMP para redes de empresa y CMIS/CMIP para redes públicas), sólo SNMP es la que está consiguiendo una aceptación e implantación amplia, a lo que ha contribuido su sencillez y rapidez de desarrollo.

CAPÍTULO I

LA IMPORTANCIA DE LA GESTIÓN DE REDES

1.1 INTRODUCCIÓN

Las redes de comunicaciones han evolucionado con el paso del tiempo ante la necesidad de satisfacer las demandas de los diferentes servicios de telecomunicaciones, que día a día necesitan un mayor ancho de banda y una mejor calidad de servicio para las nuevas aplicaciones que se han venido desarrollando hasta la actualidad. La tecnología de redes ha incrementado su complejidad generándose la necesidad de contar con una mejor administración de los recursos de estos sistemas, lo cual ha favorecido la evolución conjunta de la gestión de redes.

La gestión de redes tiene como propósito la utilización y coordinación de los recursos para:

Planificar, organizar, mantener, supervisar, evaluar, y controlar los elementos de las redes de comunicaciones para adaptarse a la calidad de servicio necesaria, a un

determinado costo. Su campo de aplicación es amplio y de gran importancia dadas las características tecnológicas que poseen los sistemas de telecomunicaciones y los servicios que ofrecen, mantiene un cierto grado de complejidad al interactuar con sistemas heterogéneos que involucran diversos fabricantes con productos eminentemente propietarios, así como productos apegados a estándares en forma total o parcial.

1.2 ELEMENTOS DE UN SISTEMA DE GESTIÓN

La gestión de un entorno de telecomunicaciones es una aplicación de procesamiento de información, en la cual intervienen elementos fundamentales como son el gestor, el agente, el protocolo de gestión, y la base de información de gestión (MIB, Management Information Base), los cuales interactúan entre sí empleando el modelo gestor-agente como se muestra en la figura 1.1

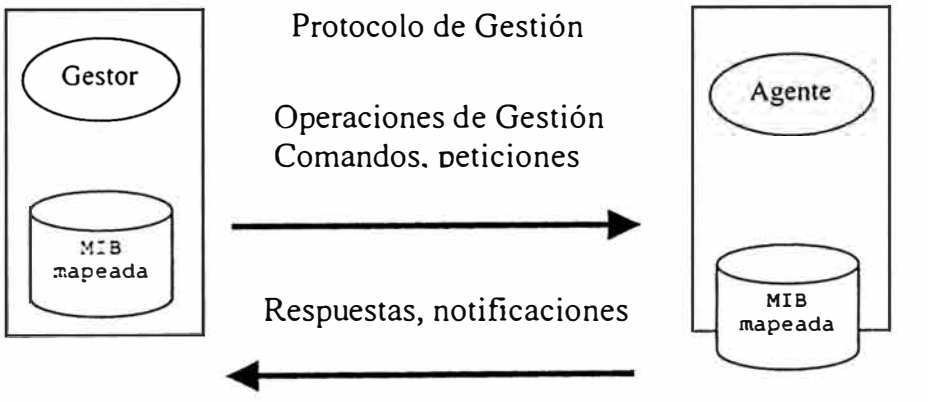


Figura 1.1 Modelo Gestor - Agente

El gestor es la parte de la aplicación que emite las directivas de operaciones de gestión y recibe notificaciones y respuestas. Este se implementa en una estación de gestión en la cual se debe disponer de la MIB del dispositivo en gestión y una interfaz de usuario.

El agente tiene la función de responder a las directivas enviadas por el gestor y lo realiza accesando a la MIB para manipular los objetos involucrados en la operación. El agente se encuentra ubicado en el dispositivo de telecomunicaciones gestionado.

La MIB es el conjunto de objetos gestionados (el concepto de objeto es diferente al empleado en la programación orientada a objetos) que representan a los recursos de la red que permiten algún tipo de gestión en una forma abstracta.

La MIB se encuentra ubicada en el dispositivo de telecomunicaciones, y una referencia de ésta es necesaria en el gestor.

El protocolo es el conjunto de especificaciones y convenciones que gobiernan la interacción de procesos y elementos dentro de un sistema de gestión. En la actualidad SNMP (Simple Network Management Protocol), forma parte del modelo de gestión de Internet y CMIP (Common Management Information Protocol), es parte del modelo de gestión OSI son los protocolos predominantes.

1.3 IMPORTANCIA DE LA GESTIÓN DE REDES

La gestión de red juega un papel importante en el buen funcionamiento de las redes y se hace imprescindible su aplicación por las siguientes razones:

- Los sistemas de información son vitales y están soportados sobre redes.
- La información manejada tiende a ser cada día mayor y a estar mas dispersa

- Las nuevas tecnologías de red requieren de una gestión cada vez mas especializada, que le permita el empleo eficiente de sus recursos de telecomunicaciones.
- El adecuado empleo de las tecnologías de gestión de red permite mejorar la eficiencia, disponibilidad y el rendimiento de las redes, aumentar la relación calidad / costo en el diseño de las redes, así como aumentar la satisfacción de los usuarios por el servicio de red proporcionado.

Para lograr una gestión de red eficiente es necesario contar con un sistema integrado de gestión que conlleve a mejorar la eficiencia en la operación de la red. Un sistema integrado de gestión de red debe contar con los siguientes elementos: Recursos humanos, métodos de trabajo y desarrollo tecnológico.

1.4 RESUMEN DEL CAPITULO

La gestión de redes es una necesidad primordial en las organizaciones.

De la gestión de red se espera que:

- Asegure un servicio casi continuo a los usuarios finales descrito por la disponibilidad y velocidad de respuesta, sin que se vean afectados por las actualizaciònes tecnológicas en la red.
- Incremente el desempeño de una red con el empleo de la mejor tecnología de redes, recursos humanos adecuados, métodos de trabajo probados y herramientas integradas que automaticen las operaciones de gestión.
- Controle los costos dedicados a las comunicaciones y a la seguridad de la información.

CAPÍTULO II

ARQUITECTURA DE LA GESTIÓN DE RED

En este apartado se describen las tres principales arquitecturas de gestión de red:

- Modelo OSI.
- Modelo TMN.
- Modelo Internet (SNMP).

2.1 MODELO OSI

OSI ha definido una arquitectura de gestión OSI (Open Systems Interconnection) cuya función es permitir supervisar, controlar y mantener una red de datos. Está dividida en cinco categorías de servicios de gestión denominadas Areas Funcionales Específicas de Gestión (Specific Management Functional Areas: SMFA). Estas categorías son las siguientes:

2.1.1 Gestión De Configuración

La gestión de configuración comprende una serie de facilidades mediante las cuales se realizan las siguientes funciones:

- Iniciación y desactivación.
- Definición o cambio de parámetros de configuración.
- Recogida de información de estado.
- Denominación de los elementos de la red.

2.1.2 Gestión De Fallos

Detección, diagnóstico y corrección de los fallos de la red y de las condiciones de error. Incluye:

- Notificación de fallos.
- Sondeo periódico en busca de mensajes de error.
- Establecimiento de alarmas.

2.1.3 Gestión De Prestaciones

Se define como la evaluación del comportamiento de los elementos de la red. Para poder efectuar este análisis es preciso mantener un histórico con datos estadísticos y de configuración.

2.1.4 Gestión De Contabilidad

Determinación de los costes asociados a la utilización de los recursos y la asignación de sus correspondientes cargas.

2.1.5 Gestión De Seguridad

Comprende el conjunto de facilidades mediante las cuales el administrador de la red modifica la funcionalidad que proporciona seguridad frente a intentos de acceso no autorizados. Incluye aspectos como la gestión de claves, cortafuegos e históricos de seguridad.

La arquitectura de gestión OSI define un objeto gestionable como la interfaz conceptual que han de presentar los dispositivos que ofrecen funciones de gestión. El proceso de supervisión y control de un objeto gestionable se realiza mediante una serie de interacciones. Estas interacciones son de dos tipos:

- **De operación:** el gestor solicita algún dato al objeto gestionable o desea realizar alguna acción sobre él.
- **De notificación:** cuando el objeto gestionable intenta enviar algún dato al gestor como consecuencia de algún evento ocurrido en el dispositivo.

Un objeto gestionable se caracteriza además por un conjunto de **atributos** que son las propiedades o características del objeto, y un **comportamiento** en respuesta a las operaciones solicitadas. En la figura 2.1 se presenta un ejemplo de estas interacciones.

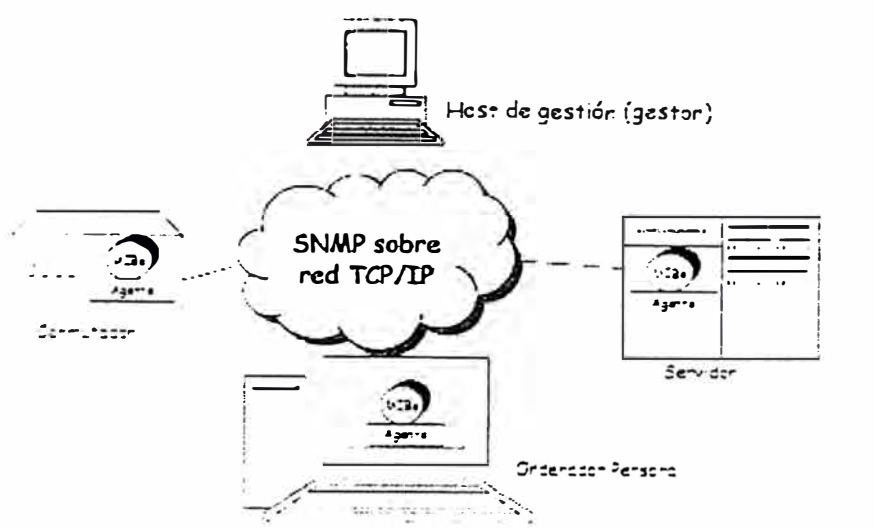


Figura 2.1 Operación y notificación de Gestión.

La comunicación entre el gestor y el objeto gestionable no es directa, se realiza mediante un intermediario: el agente de gestión (esto se corresponde con un modelo centralizado Gestor-Agente). La función del agente es controlar el flujo de información de gestión entre el gestor y el objeto. Este control lo realiza comprobando una serie de reglas de gestión (por ejemplo que el gestor tenga la capacidad para solicitar una determinada operación), que han de cumplirse para poder realizar la operación. Estas reglas se incluyen en los datos como parte de la solicitud de una operación.

El flujo normal de información de gestión y control entre el gestor y el agente se realiza mediante el protocolo CMIP, perteneciente al nivel de aplicación OSI.

El protocolo permite que un sistema se pueda configurar para que opere como gestor o como agente. La mayoría de las realizaciones prácticas de sistemas gestionados se configuran con unos pocos sistemas operando en modo gestor, controlando las actividades de un gran número de sistemas operando en modo agente.

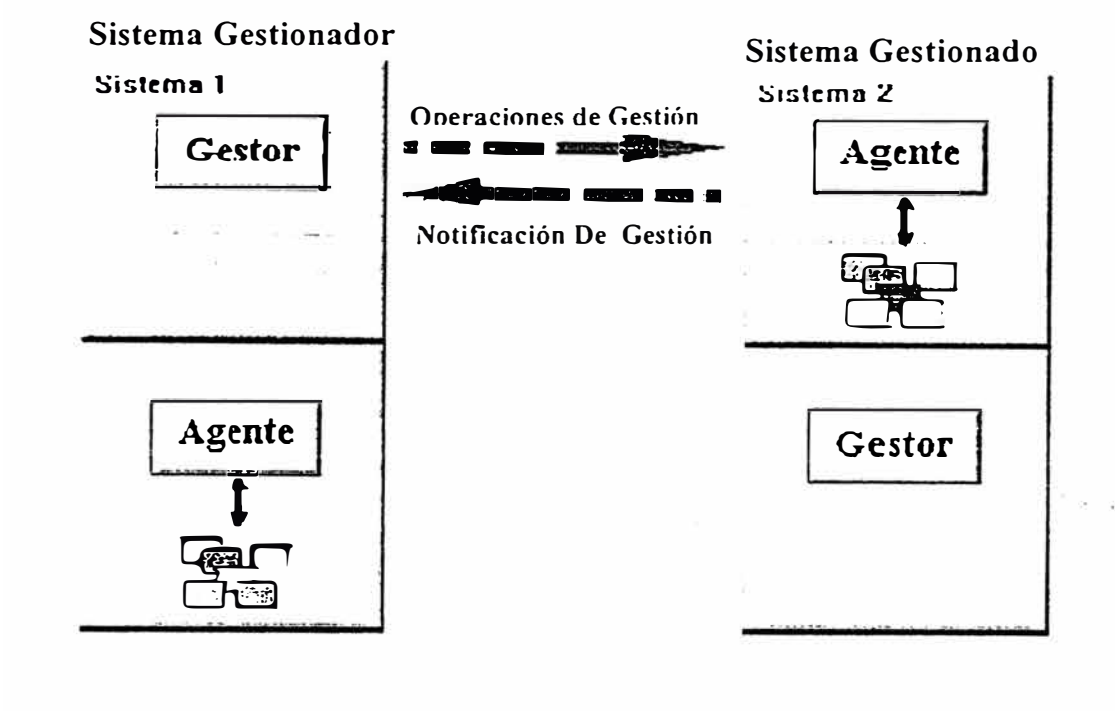


Figura 2.2 La configuración puede ser como gestor o como agente.

Cuando dos procesos se asocian para realizar una gestión de sistemas, deben establecer en qué modo va a operar cada uno de ellos (en modo Agente o en modo Gestor). Los procesos indican, mediante las denominadas unidades funcionales, qué funcionalidades de gestión y estándares utilizarán durante la asociación.

Otros componentes de la arquitectura de gestión OSI son:

- **Estructura de la Información de Gestión (Structure of Management Information, SMI):** Define la estructura lógica de la información de gestión OS. Establece las reglas para nombrar a los objetos gestionables y a sus atributos.

Define un conjunto de subclases y tipos de atributos que son en principio aplicables todos los tipos de clases de objetos gestionables.

- **Base de Información de Gestión (Management Information Base, MIB):** Representa la información que se está utilizando, modificando o transfiriendo en la arquitectura de los protocolos de gestión OSI. La MIB conoce todos los objetos gestionables y sus atributos. No es necesario que este centralizada físicamente en un lugar concreto, puede estar distribuida a través del sistema y en cada uno de sus niveles.
- **CMIS (Common Management Information Services):** Es un conjunto de reglas que identifican las funciones de una interfaz OSI entre aplicaciones, utilizado por cada aplicación para intercambiar información y parámetros. CMIS define la estructura de la información que es necesaria para describir el entorno.

  Prácticamente todas las actividades de la gestión de red OSI están basadas en diez primitivas de servicio CMIS que son utilizadas por las SMFAs.

2.2 MODELO TNM

El término TMN (Telecommunications Management Network) fue introducido por la ITU-T, y está definido en la recomendación M.3010. Aunque en un principio no hubo mucha colaboración entre los grupos de gestión de red de la ISO y el CCITT (germen de la ITU-T), posteriormente fueron incorporados varios conceptos del modelo OSI al estándar TMN. En concreto:

- Se adoptó el modelo gestor-agente del modelo OSI.

- Se siguió el paradigma de la orientación a objetos de la arquitectura OSI.
- Se trabajó conjuntamente en el desarrollo del concepto de dominios de gestión.

Un aspecto diferenciador de ambos modelos consiste en la introducción, en el modelo TMN, de una red separada de aquella que se gestiona, con el fin de transportar la información de gestión.

A diferencia del modelo OSI, en el cual se definen cinco áreas funcionales, el estándar TMN no entra en consideraciones sobre las aplicaciones de la información gestionada, por el contrario, se define la siguiente funcionalidad:

- El intercambio de información entre la red gestionada y la red TMN.
- El intercambio de información entre redes TMN.
- La conversión de formatos de información para un intercambio consistente de información.
- La transferencia de información entre puntos de una TMN.
- El análisis de la información de gestión y la capacidad de actuar en función de ella.
- La manipulación y presentación de la información de gestión en un formato útil para el usuario de la misma.
- El control del acceso a la información de gestión por los usuarios autorizados.

2.2.1 Arquitectura TMN

El modelo TMN define tres arquitecturas diferenciadas:

- **Arquitectura Funcional:** Que describe la distribución de la funcionalidad dentro de la TMN, con el objeto de definir los bloques funcionales a partir de los cuales se construye la TMN.

- **Arquitectura Física:** Que describe las interfaces y el modo en que los bloques funcionales se implementan en equipos físicos.
- **Arquitectura de la Información:** Que sigue los principios de los modelos OSI de gestión (CMIS y CMIP) y directorio (X.500).

2.2.2 Arquitectura Funcional

Se definen cinco tipos de bloques funcionales. Estos bloques proporcionan la funcionalidad que permite a la TMN realizar sus funciones de gestión. Dos bloques funcionales que intercambian información están separados mediante puntos de referencia. A continuación se describen los distintos tipos de bloques funcionales:

- **Función de Operación de Sistemas (OSF):** Los OSF procesan la información relativa a la gestión de la red con el objeto de monitorizar y controlar las funciones de gestión. Cabe definir múltiples OSF dentro de una única TMN.
- **Función de Estación de Trabajo (WSF):** Este bloque funcional proporciona los mecanismos para que un usuario pueda interactuar con la información gestionada por la TMN.
- **Función de Elemento de Red (NEF):** Es el bloque que actúa como agente, susceptible de ser monitorizado y controlado. Estos bloques proporcionan las funciones de intercambio de datos entre los usuarios de la red de telecomunicaciones gestionada.
- **Adaptadores Q (QAF):** Este tipo de bloque funcional se utiliza para conectar a la TMN aquellas entidades que no soportan los puntos de referencia estandarizados por TMN.

- **Función de Mediación (MF):** La función de mediación se encarga de garantizar que la información intercambiada entre los bloques del tipo OSF o NEF cumpla con los requisitos demandados por cada uno de ellos.

  Puede realizar funciones de almacenamiento, adaptación, filtrado y condensación de la información.

Cada bloque funcional se compone a su vez de un conjunto de componentes funcionales, considerados como los bloques elementales para su construcción. Estos componentes se identifican en la norma pero no están sujetos a estandarización.

2.2.3 Arquitectura física

La arquitectura física se encarga de definir como se implementan los bloques funcionales mediante equipamiento físico y los puntos de referencia en interfaces. En la arquitectura física se definen los siguientes bloques constructivos:

- Elemento de red (NE).
- Dispositivo de mediación (MD).
- Adaptador Q (QA).
- Sistema de operaciones (OS).
- Red de comunicación de datos (DCN).

Cada uno de estos bloques puede implementar uno o más bloques funcionales (excepto el DCN que se encarga de realizar el intercambio de información entre bloques), pero siempre hay uno que ha de contener obligatoriamente y que determina su denominación.

2.2.4 Interfaces

Las interfaces son implementaciones de los puntos de referencia, y son comparables a las pilas de protocolos. Existe una correspondencia uno a uno entre los puntos de referencia y las interfaces, excepto para aquellos que están fuera de la TMN, es decir, los puntos de referencia G y M.

2.2.5 Arquitectura Lógica De Niveles

En el estándar TMN define una serie de capas o niveles de gestión mediante las cuales se pretende abordar la gran complejidad de la gestión de redes de telecomunicación.

Cada uno de estos niveles agrupa un conjunto de funciones de gestión. El estándar LLA define cuáles son esos niveles y las relaciones entre ellos. Se definen los siguientes niveles:

- **Nivel de Elementos de Red:** Incluye las funciones que proporcionan información en formato TMN del equipamiento de red así como las funciones de adaptación para proporcionar interfaces TMN a elementos de red no-TMN.
- **Nivel de Gestión de Elementos:** Incluye la gestión remota e individual de cualquier elemento de red que se precise para el establecimiento de conexiones entre dos puntos finales para proporcionar un servicio dado. Este nivel proporcionará funciones de gestión para monitorizar y controlar elementos de gestión individuales en la capa de elemento de red.
- **Nivel de Gestión de Red:** Incluye el control, supervisión, coordinación y configuración de grupos de elementos de red constituyendo redes y subredes para la realización de una conexión.

- **Nivel de Gestión de Servicios:** Incluye las funciones que proporcionan un manejo eficiente de las conexiones entre los puntos finales de la red, asegurando un óptimo aprovisionamiento y configuración de los servicios prestados a los usuarios.
- **Nivel de Gestión de Negocio:** Incluye la completa gestión de la explotación de la red. incluyendo contabilidad, gestión y administración, basándose en las entradas procedentes de los niveles de Gestión de Servicios y de Gestión de Red.

Figura 2.3 Arquitectura lógica de niveles en el estandar TNM

2.3 MODELO DE INTERNET - SNMP

En 1988, el IAB (Internet Activities Board, Comité de Actividades Inter-Red) determinó la estrategia de gestión para TCP/IP (Transfer Control Protocol/Internet Protocol, Protocolo de Control de Transmisión/protocolo de Inter-Red). Esto significó el

nacimiento de dos esfuerzos paralelos: la solución a corto plazo, SNMP, y la solución eventual a largo plazo. CMOT (CMIP Over TCP/IP, CMIP sobre TCP/IP).

CMOT pretendía implantar los estándares del modelo de gestión OSI en el entorno Internet (TCP/IP). CMOT tuvo que afrontar los problemas derivados de la demora en la aparición de especificaciones y la ausencia de implementaciones prácticas. Como consecuencia de ello, la iniciativa CMOT fue paralalizada en 1992.

SNMP es una extensión del protocolo de gestión de red para gateways SGMP (Simple Gateway Monitoring Protocol, Protocolo Sencillo de Supervisión de Pasarelas), que se convirtió en 1989 en el estándar recomendado por Internet. Está dirigido a proporcionar una gestión de red centralizada que permita la observación, el control y la gestión de las instalaciones. Utilizando SNMP, un administrador de red puede direccionar preguntas y comandos a los dispositivos de la red.

SNMP se ha convertido, debido al enorme éxito que ha tenido desde su publicación, en el estándar de facto de gestión de redes. Prácticamente todo el equipamiento de redes puede ser gestionado vía SNMP.

Algunas de las funciones que proporciona SNMP son:

- Supervisión del rendimiento de la red y su estado.
- Control de los parámetros de operación.
- Obtención de informes de fallos.
- Análisis de fallos.

2.3.1 SNMP

El protocolo SNMP incorpora varios elementos presentes en otros estándares como el modelo gestor-agente, la existencia de una base de datos de información de gestión (MIB) o el uso de primitivas de tipo PUT y GET para manipular dicha información. A continuación se describen dichos elementos:

- **Agente:** Equipamiento lógico alojado en un dispositivo gestionable de la red. Almacena datos de gestión y responde a las peticiones sobre dichos datos.
- **Gestor:** Equipamiento lógico alojado en la estación de gestión de red. Tiene la capacidad de preguntar a los agentes utilizando diferentes comandos SNMP.
- **MIB (Management Information Base, Base de Información de Gestión):** base de datos virtual de los objetos gestionables, accesible por un agente, que puede ser manipulada vía SNMP para realizar la gestión de red.

El protocolo SNMP realiza las funciones descritas anteriormente llevando información de gestión entre los gestores y los agentes. En la figura 2.4 siguiente se presenta un ejemplo de sistema de gestión SNMP.

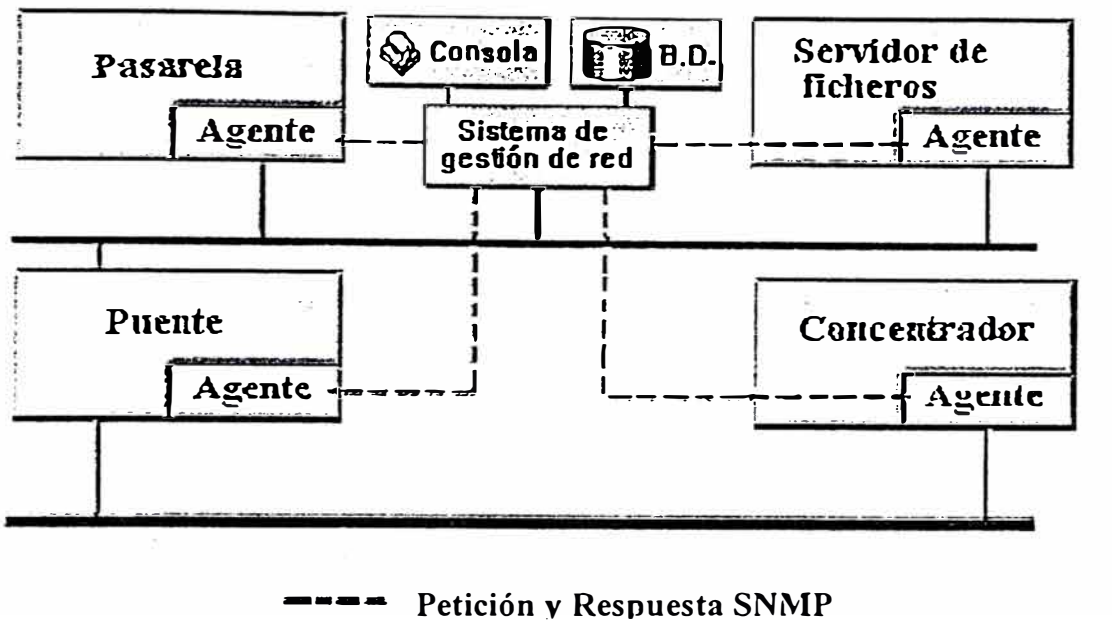


Figura 2.4 Esquema de un sistema de gestión SNMP

El protocolo SNMP es sólo un aspecto dentro de toda la estructura de gestión, la cual está compuesta de los siguientes elementos:

- **Estación de Gestión de Red (Network Management Station, NMS):** Es el elemento central que proporciona al administrador una visión del estado de la red y unas funciones de modificación de este estado (puede ser una estación de trabajo o un ordenador personal).
- **Estructura de la Información de Gestión (SMI, Structure of Management Information):** Es un conjunto de reglas que define las características de los objetos de la red y cómo obtienen los protocolos de gestión información de ellos. Aunque ha sido diseñado después del SMI de OSI, no es compatible con este.

2.3.2 Base De Información De Gestión (MIB).

Es una colección de objetos, que representan de forma abstracta los dispositivos de la red y sus componentes internos. La MIB es conforme a la SMI para TCP/IP. Cada agente SNMP contiene instrumentación que, como mínimo, debe ser capaz de reunir objetos MIB estándar. Estos objetos incluyen direcciones de red, tipos de interfaz, contadores y datos similares.

El estándar MIB de Internet define 126 objetos relacionados con los protocolos TCP/IP. Los fabricantes que deseen pueden desarrollar extensiones del estándar MIB. Estas MIBs privadas incorporan un amplio rango de objetos gestionables, y algunas veces contienen objetos que son funcionalmente similares a los MIBs ya definidos, en otros casos el cambio de una variable en un objeto inicia una batería de funciones en el dispositivo gestionado (como por ejemplo un autodiagnóstico).

La carga de la gestión de todas las MIBs y de las extensiones privadas recae en el sistema de gestión. Las MIBs están escritas en una variante simple del lenguaje de definición OSI ASN.1.

En 1990 se introdujo una nueva versión de MIB, MIB II, donde la mayor aportación es la utilización de 185 nuevos objetos de extensiones privadas.

Aparte de la MIB, existe la **Base de Datos de Estadísticas de Red** (Network Statistics Datbase, NSD) que está en la estación de trabajo de gestión. En esta base de datos se recoge información de los agentes para realizar funciones de correlación y planificación.

Las limitaciones de SNMP se deben a no haber sido diseñado para realizar funciones de gestión de alto nivel. Sus capacidades lo restringen a la supervisión de redes y a la

detección de errores. Como todos los elementos TCP/IP, han sido creados pensando más en su funcionalidad y dejando a un lado la seguridad.

2.4 SNMPv2 Y SNMPv3

En 1996 se publicó un nuevo estándar, el protocolo SNMPv2, resultado de una serie de propuestas para mejorar las características de SNMP. Los cambios se traducen fundamentalmente en una mejora de las prestaciones, un aumento de la seguridad y en la introducción de una jerarquía de gestión.

2.4.1 Prestaciones

SNMPv2 mejora el mecanismo de transferencia de información hacia los gestores, de forma que se necesitan realizar menos peticiones para obtener paquetes de información grandes.

2.4.2 Seguridad

A diferencia de SNMP, que no incorpora ningún mecanismo de seguridad, SNMPv2 define métodos para controlar las operaciones que están permitidas.

Desafortunadamente surgieron dos planteamientos diferentes en cuanto al modelo de seguridad, que han dado lugar a dos especificaciones conocidas como SNMPv2* y SNMPv2u. Se están realizando esfuerzos para unificar ambos enfoques en un único estándar: SNMPv3.

2.4.3 Gestión jerárquica

Cuando el número de agentes a gestionar es elevado, la gestión mediante el protocolo SNMP se vuelve ineficaz debido a que el gestor debe sondear periódicamente todos los agentes que gestiona.

SNMPv2 soluciona este inconveniente introduciendo los gestores de nivel intermedio. Son estos últimos los que se encargan de sondear a los agentes bajo su control. Los gestores intermedios son configurados desde un gestor principal de forma que solo se realiza un sondeo de aquellas variables demandadas por este último, y solo son notificados los eventos programados. SNMPv2 también introduce un vocabulario más

2.5 RMON

La especificación RMON (Remote MONitor, Monitorización Remota) es una base de información de gestión (MIB) desarrollada por el organismo IETF (Internet Engineering Task Force) para proporcionar capacidades de monitorización y análisis de protocolos en redes de área local (segmentos de red). Esta información proporciona a los gestores una mayor capacidad para poder planificar y ejecutar una política preventiva de mantenimiento de la red.

Las implementaciones de RMON consisten en soluciones cliente/servidor. El cliente es la aplicación que se ejecuta en la estación de trabajo de gestión, presentando la información de gestión al usuario. El servidor es el agente que se encarga de analizar el tráfico de red y generar la información estadística. La comunicación entre aplicación y agente se realiza mediante el protocolo SNMP.

RMON es una herramienta muy útil para el gestor de red pues le permite conocer el estado de un segmento de red sin necesidad de desplazarse físicamente hasta el mismo y realizar medidas con analizadores de redes y protocolos.

Las iniciativas se dirigen en estos momentos hacia la obtención de una mayor y más precisa información. En concreto, se trabaja en la línea de analizar los protocolos de nivel superior, monitorizando aplicaciones concretas y comunicaciones extremo a extremo (niveles de red y superiores). Estas facilidades se incorporarán en versiones sucesivas de la especificación (RMON II).

2.6 COMPARACIÓN SNMP/CMIP

A continuación se hace una comparación entre los protocolos SNMIP y CMIP:

- SNMP está basado en técnicas de sondeo, mientras que CMIP utiliza una técnica basada en eventos. Esto permite que CMIP sea más eficiente que SNMP en el control de grandes redes.
- CMIP es un protocolo orientado a conexión mientras que SNMP es un protocolo sin conexión. Esto significa que la carga de proceso de SNMP es reducida, pero cuando se envía un mensaje nunca se puede asegurar que el mensaje llega a su destino. La seguridad de los datos no es prioritaria para SNMP.
- CMIP permite la implementación de comandos condicionales sofisticados, mientras que SNMP necesita el nombre de cada objeto.

- CMIP permite, mediante una única petición, la recogida de gran cantidad de datos de los objetos gestionables, enviando información de retorno en múltiples respuestas. Esto no está permitido en SNMP.
- CMIP está especialmente preparado para gestionar grandes redes distribuidas, mientras que SNMP está recomendado para la gestión inter-red.
- CMIP realiza una distinción clara entre los objetos y sus atributos. SNMP no permite esto. lo cual hace imposible la reutilización de atributos y definiciones.

CAPÍTULO III

EL PROTOCOLO SNMP

3.1 JUSTIFICACIÓN

Existen diversos fabricantes de equipo para construir las redes digitales, conscientes de la necesidad de administrar el funcionamiento de sus equipos crearon soluciones propias para sus equipos que no necesariamente eran compatibles con otras marcas.

Como esfuerzo para lograr que equipos de diversos fabricantes pudieran ser moni toreados, estándares deben ser creados para que todos trabajen igual, pero, ¿porqué estarían dispuestas las compañías a hacerlo?.

Los fabricantes de equipo hacen mucho dinero vendiendo paquetes de soluciones propietarias en lugar de vender productos individuales que sean intercambiables.¿Porqué debería Cisco vender switches si puede dominar toda la empresa?, ¿Porqué debe desarrollar productos que puedan ínter operar con 3Com o Bay proveyendo una entrada fácil en sus cuentas de negocio?.

Los fabricantes participan y forman grupos que definen estándares porque temen que si no hacen, los clientes tendrán la percepción que no están bien abiertas. De hecho se preocupan más de las repercusiones negativas que de los estándares mismos.

3.1.1 Definición Administración de Redes

Black (1992) define el término administración de redes tomando prestada la definición de administración de negocios, ya que involucra la planeación, organización, monitoreo y control de actividades y recursos.

La idea básica de un sistema de administración de red es que haya dos tipos de sistemas en cualquier configuración en red : Agentes y Administradores. Cualquier nodo en la red que tiene que ser administrado incluye un modulo con un agente, como PCs, estaciones de trabajo, servidores, ruteadores, etc.

Black (1992) nos da una idea acerca de los esfuerzos que se hicieron con el fin de definir un estándar para administración de redes

1. Los modelos de protocolo del Open Systems Interconnection (OSI).
    - Common Managment Information Protocol (CMIP).
    - Common Managment Information Service Element (CMISE).

Estos estándares proveen cinco áreas funcionales para la administración de redes como es: alarmas, eficiencia, configuración, contabilidad y seguridad.

1. Los protocolos de Internet .
    - Simple Network Management Protocol (SNMP).
    - CMIP sobre TCP (CMOT).

2. Los protocolos propuestos por la IEEE.
   - CMIP sobre LLC (CMOL).
   - Una aproximación de IEEE sin CMIP.

3.2 HISTORIA

Desde finales de los 70 las redes computacionales han crecido y han dejado de ser redes separadas para convertirse en redes más grandes fueron llamadas Interred (Internet) y crecieron a tasas exponenciales, a medida que crecieron, fueron más difíciles de mantener y fue evidente la necesidad de que se desarrollara un protocolo de administración.

El primer protocolo usado fue SNMP, que fue considerado temporal y más bien parecía un curita mientras otros protocolos mejores y más grandes fueran desarrollados.

Dentro de estos protocolos, surgieron SNMPv2 que incorporaba la mayoría de las características del SNMP original, el segundo fue el CMIP, el cual estaba mejor organizado y contenía muchas características adicionales que SNMP o SNMPv2.

(Wilson 98) especifica que los primeros estándares para estándares para interoperabilidad de sistemas de administración de redes fueron definidos bajo el modelo OSI, que incluyeron los cuatro elementos esenciales: un protocolo, repository, un conjunto de API's y un marco para definir objetos. Pero el proceso estuvo lleno de burocracia y política, por lo que tomo más de cuatro años para completar las especificaciones OSI, muchas de las cuales nunca fueron implementadas.

Cuando se enfocaban en OSI, cuatro personas desarrollaron SNMP en menos de un año. Su eficiencia fue debido a que evitaron los comités de estándares que requiere que el desarrollador demuestre una implementación trabajando antes que pueda ser revisada para proponerse como estándar.

SNMP resolvió los problemas de una definición de protocolo y definición de objetos, pero no resolvió la definición del API's para ligar las aplicaciones de administración. Algunos grupos de estándares, incluyendo la Fundación de Software Abierto (Open software Foundation) y el consorcio de administración trataron de llenar esas necesidades, pero esos esfuerzos terminaron en las manos del proceso de estandarización.

Como un esfuerzo para definir un estándar que pudiera ser implementado en el equipo que fabrican las diversas compañías, el protocolo preferido fue SNMP, veamos porqué:

Como un ejemplo de simplicidad contra poder se puede comparar SNMP contra CIMP (Common Management Information Protocol), este último tiene un conjunto muy amplio de primitivas y elementos de datos, para CIMP, un subconjunto del protocolo tenía que ser elegido para lograr la interoperabilidad con otros equipos de diferentes marcas y en contraparte, SNMP fue especificado sin opciones, la cual aseguraba la interoperabilidad.

(Stallings) menciona que la elección entre SNMPv2 y CMIP se hizo por el público en general y fue extremadamente costoso, en promedio, las compañías gastaban cerca del 15% de su presupuesto de sistemas de información en administración de redes, entre las

100 compañías americanas más importantes, se gastaron un promedio de 1.3 millones de dólares.

Stalling (1998) comenta que SNMP es un protocolo de administración de redes que fue diseñado para ser fácil de implementar y consumir recursos mínimos de procesador y de red, pero Hein y Griffiths (1995) dicen que el termino simple es fuente de malos entendidos.

SNMP no es simple en especificación ni en su implementación real.

No fue diseñado para ser simple.

Pero los conceptos integrados en el protocolo SNMP son de una simplicidad que varia desde bella hasta trivial. Esto hizo SNMP versión 1 decepcionantemente fácil de implementar.

Muchos fabricantes de productos de comunicaciones desarrollaron su propia versión de SNMP pero desestimaron el trabajo requerido para hacer de una especificación simple un producto vendible, lo que fue un error muy caro.

La versión 2 de SNMP es por mucho más compleja que la versión 1.

Como se mencionó, SNMP fue desarrollado para proveer un protocolo de administración de redes interconectadas, pero su primera meta fue ser simple para dar robustez en caso de una falla de red, por lo que tiene una carga de procesador muy baja en los productos que está corriendo y la facilidad de depurar el código en el protocolo mismo, por lo que se limitó al UDP (Users Datagram Protocol), esto daba la capacidad de manejar paquetes perdidos y hacer las retransmisiones necesarias. Como parte de la depuración, se podían cambiar la ruta de los paquetes para que no se pierdan, por lo que

manteniendo el control del servicio de transporte (capa 4), era considerado esencial. Por lo que un protocolo de administración será obligatoriamente simple, para que el consumo de recursos de la red sea mínimo. tanto como sea posible, UDP permite el control necesario sobre las transmisiones de paquetes, tamaños de los paquetes y contenido.

Stallings (1998) comenta que SNMP tuvo su primera publicación en 1988 y ha llegado a ser la herramienta de administración de redes más comúnmente usada para TCP/IP.

El crecimiento que tuvo desde finales de los 80 no ha sido lo suficientemente rápido y lo ha llevado a tener algunas deficiencias principalmente de seguridad.

Algunas, de estas deficiencias se trataron de resolver con una nueva versión del protocolo llamado SNMPv2. que fue publicado como un conjunto RFC (Request For Comments) en los cuales se definen los cambios propuestos e intervienen gente de diversas compañías para lograr un consenso.

Este esfuerzo no fue ampliamente aceptado por ser muy caro y difícil de implementar, por lo que se desarrolló la versión 3, llamada SNMPv3, que cubrió las expectativas y es el que se usa en estos días.

Existen monitores que son compatibles con los tres estándares por fines de compatibilidad.

IBM, en aquellos días orientados a mainframe de los años 80, fue la primera compañía en desarrollar un sistema de administración de empresas capaz de monitorear y controlar redes consistentes de miles (incluso decenas de miles) de anfitriones y terminales. Este paquete de software. llamado Net View, abrió un nuevo rango de aplicaciones que hoy

en día está calificado como uno de los mercados más actuales de la industria de las comunicaciones de datos, el cual incluye competidores como Computer Associates, Digital Equipement, HP y Sun Microsystems.

El problema, es que la mayoría de las llamadas herramientas de administración de red (tal como Cisco Works de Cisco y Mange Wise de Novell) están diseñadas para administrar dispositivos o sistemas particulares, tales como ruteadores y LAN's en las empresas, no en toda le red.

Estas aplicaciones (conocidas colectivamente como sistemas de administración de elementos), los cuales incluyen sistemas de administración de módem, núcleo y multiplexor, reúnen datos sobre el estado de tipos específicos de dispositivos de red(por ejemplo, servidores LAN, ruteadores y núcleos) y mostrarla en una estación de trabajo UNÍX o PC monitoreada por el administrador de red.

3.3 ADMINISTRADOR DE REDES CON SNMP

Este protocolo, forma el ambiente de comunicación más comúnmente usado, el soporte es construido por el fabricante en la mayoría de los equipos.

Este protocolo, se encuentra en constante evolución y se viene adaptando a las necesidades crecientes que se tienen de la red de Internet, sobre todo en seguridad.

Se desea investigar sobre este protocolo porque es una herramienta que tiene mucho potencial para controlar inteligentemente los dispositivos que conforman una red como estaciones de trabajo, servidores, concentradores, ruteadores, etc. Y hasta la fecha, el protocolo se usa casi para monitoreo de alarmas, dejando a un lado el área de control.

El estudio de este protocolo tiene mucho trabajo por ser desarrollado y se requiere de muchos profesionales que puedan diseñar e implementar un agente SNMP, se está orientando a manejar el administrador de redes con tecnología de la Web.

3.4 GENERALIDADES

3.4.1 ¿Cómo se Trabaja la Tecnología?

SNMP recolecta información administrativa de los dispositivos que conforman una red, por medio de un agente que se encuentra en dicho dispositivo, el cual graba la información en una base de información de administración (MIB). El MIB incluye varias características como estadísticas de tráfico de la red, sobrecargas y errores.

Un MIB tiene un formato que es común a los otros dispositivos, por lo que las utilerías de administración de SNMP pueden recolectar la información, no importando la marca o fabricante, y de esta manera la presentan al administrador o la pasan a otros agentes.

SNMP opera sobre el Use Datagram Protocol (UDP), el cual es parte del protocolo TCP/IP. UDP provee las reglas para agregar la dirección del puerto a un proceso de aplicación para formar el encabezado de los paquetes de información.

El encabezado identifica la fuente y el destino del paquete y otra información referente a lo que hay en el paquete, como la longitud del mismo.

Una dirección de puerto, es la dirección específica de una aplicación o un proceso corriendo en la computadora, en donde la dirección IP es la dirección de la estación de trabajo en la red.

SNMP tiene tres primitivas de control que inician el flujo de datos (Get, Get-Next y Set). Hay dos primitivas de control que se usan para responder una petición, una es usada en respuesta a la petición directa (Get-Reponse) ya la otra es respuesta asíncrona para obtener la atención de quien hace la petición (Trap).

Estas cinco primitivas son transportadas por UDP, por lo que están limitadas en tamaño por la cantidad de datos que caben en un paquete sencillo UDP. Este mensaje relativamente pequeño fue la meta para un conjunto de funciones de administración.

3.4.2 Recibo de Información de Administración

SNMP tiene la primitiva Get-Next la cual permite la visualización de datos sin requerir conocimiento previo, si se sabe que es lo que se está buscando. Cuando se requiere una tabla entera de información, la primitiva Get-Next la obtendrá. Pero a menos que sea usado con cuidado, la primitiva Get-Next puede consumir recursos intensivamente, como ancho de banda y tiempo de CPU. La manera más simple de usar Get-Next es empezar en el principio de una tabla, esperara la respuesta y hasta entonces enviar otro Get-Next.

Por ejemplo, si se requiere la información del Next-Hop, la Interface de Next-Hop y Route-Type, de una tabla que contiene 100 elementos, usando la forma simple del Get-Next, requiere 2x3x1000=6000 paquetes (Get-Next, Get-Response, columnas, renglones).

Una optimización podría ser el pedir las tres columnas en un paquete sencillo, esto forma un número de paquetes de 2x1000=2000 paquetes. En tiempo real, es el producto

de viaje de ida y de regreso por el número de consultas. En tiempo de CPU, siguen siendo 6000 bloqueos en la tabla de rutas para ambos casos.

3.4.3 Consultar Más de un Dato

El problema de SNMP ha sido consultar solo un objeto, esto es particularmente importante cuando se usa redes inalámbricas donde el flujo de datos es menor y la carga de administración se considera sobrecarga, se ve una aproximación donde se tiene múltiples Get-Next corroendo simultáneamente, un segundo algoritmo reduce el conteo de paquetes combinando los múltiples Get-Next en un solo paquete.

Ninguna de estas dos aproximaciones fue implementada en el ambiente inalámbrico, pero eso fue resuelto en SNMPv2, donde se defino una primitiva llamada Get-Bulk.

Pero aunque SNMPv2 parecía ser la solución, realmente solo vive en la teoría, falló por que sus creadores (Case, Mc Cloghrie, Rose, Waldbusser) no se pudieron poner de acuerdo en muchas facetas clave. Mejor se optó por seguir con SNMPv3.

3.5 DEFINICIÓN DE SNMP

Hein y Griffiths (1995) definen SNMP como un mecanismo de transporte de información entre componentes de la red que permiten la administración mediante la supervisión y verificación de ciertas condiciones de red.

Stallings (1998) define la función de este protocolo como el intercambio de información entre los agentes y el sistema de administración.

3.6 ESTRUCTURA DE SNMP

El agente es responsable de:

- Recolectar y mantener información acerca de su medio ambiente local.
- Proveer la información a un administrador, ya sea en respuesta a una petición de información o a una que no ha sido solicitada cuando algo sobresaliente pasa, como una alarma.
- Responder a los comandos del administrador de alterar la configuración local o os parámetros de operación.

SNMP define un método para reunir y formatear información sobre los dispositivos que serán administrados a través de una pequeña pieza software conocido como un agente. Un agente SNMP puede ser instalado en cualquier dispositivo inteligente.

Los dispositivos que no son SNMP pueden ser administrados a través de agentes sustitutos que traducen cualquier tipo de administración de estado enviada a un agente inteligente.

Dentro de este protocolo, se usan bases de datos llamadas bases de información de administración (Management Information Bases MIBS).

Dependiendo de los datos que sean de nuestro interés, se pueden programar las variables que se quieran consultar, que contiene la siguiente información

- Tipo de variable.
- Datos de la variable.
- Si es solo lectura o lectura-escritura.
- El valor de la variable.

En esencia el protocolo provee cuatro funciones

- Set : Usado por el administrador para definir un valor en el agente.
- Trap : Usado por el agente para mandar una alerta al administrador.
- Inform : Usado por el administrador para mandar una alerta a otro administrador.

Estos son unos de los que son usados por USM (User-Based Secury Model) el cual es especificado en el RFC 2574:

- SnmpEnginelD: Identificador único para un dispositivo SNMP dentro de un dominio administrado.
- SnmpEngineBoots: un conteo de numero de veces en que un dispositivo SNMP ha reiniciado desde que el SNMP EngineID fue indicado.
- SbnmpEngineTime: El número de segundos desde la ultima vez que el contador SnmpEngineBoots fue incrementado.
- SnmpSecurityLevel: Hay tres posibles niveles de seguridad: no AuthNoPriv, AuthNoPriv y AuthPriv.

CAPÍTULO IV

ASN.1 (Notación de Sintaxis Abstracta.1)

Aunque no es el motivo de este documento, parece conveniente realizar esta referencia acerca del lenguaje de definición de notaciones como es el ASN.1. Se trata de un conjunto de producciones que determinaran a las estructuras de datos, tipos y variables de una especificación. Las producciones son del tipo de cualquier definición sintáctica de la forma:

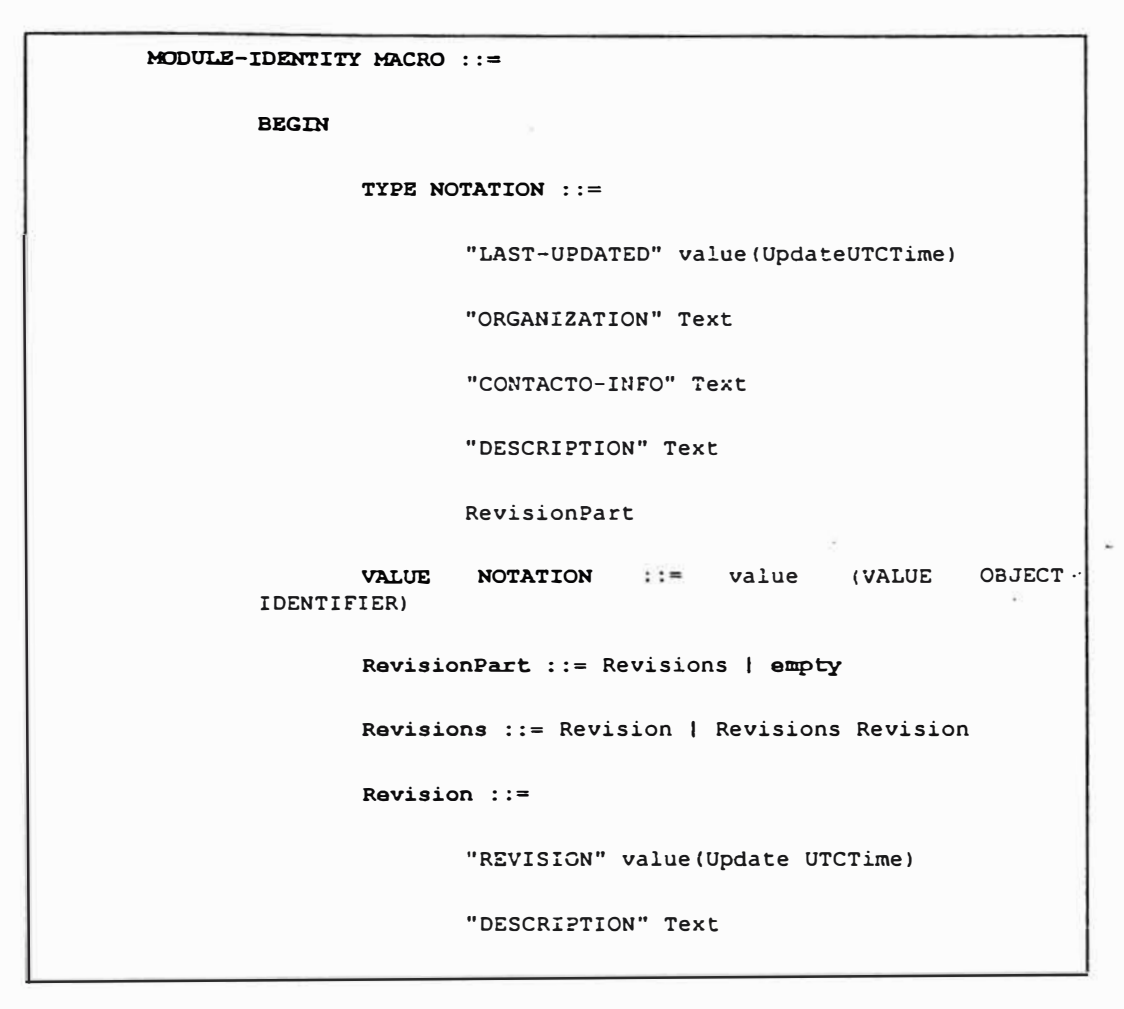
nombre ::= regla

Donde **nombre** indica el título de la producción y la **regla**, parte derecha de la producción, contendrá uno o varios tipos simples o símbolos distinguidos separados por "|" indicando cada uno de ellos una posible derivación del símbolo distinguido determinado por **nombre**.

En el caso que alguna de las partes de **regla** sea un símbolo distinguido, las producciones de este deberán ser especificadas inmediatamente después de las producciones de **nombre**.

La organización ISO proporciona al ASN.1 un mecanismo para hacer mas cómoda esta notación, que son las macros, estructuras que en si no añaden nada pero que establecen la definición de ciertas estructuras complejas, como se puede ver en las distintas macros que se utilizan en el resto de la documentación.

En el siguiente ejemplo, se define un módulo mediante una macro. **TYPE NOTATION** es una regla que define el type, **VALUE NOTATION** es otra regla que define Value, **RevisionPart** y **Revisions** son producciones con símbolo distinguido, a continuación viene la definición de **Revisión**, con varios tipos simples y por último **Text** es una producción simple. La manera de introducir comentarios es poner el texto que se quiera tras dos guiones.



```
MODULE-IDENTITY MACRO ::=

        BEGIN

                TYPE NOTATION ::=

                        "LAST-UPDATED" value(UpdateUTCTime)

                        "ORGANIZATION" Text

                        "CONTACTO-INFO" Text

                        "DESCRIPTION" Text

                        RevisionPart

                VALUE NOTATION ::= value (VALUE OBJECT
        IDENTIFIER)

                RevisionPart ::= Revisions | empty

                Revisions ::= Revision | Revisions Revision

                Revision ::=

                        "REVISION" value(Update UTCTime)

                        "DESCRIPTION" Text
```

Ejemplo 4.1 Definición de un Módulo Mediante una Macro.

4.1 MODELO DE INFORMACIÓN

Para examinar el papel de la información de gestión en el entorno de administración, consideraremos las siguientes cinco partes:

- Reglas para definir la información de administración.
- Ejemplos de colecciones de definiciones existentes.
- Reglas para definir los convenios textuales (definición de tipos de uso frecuente).

- Cómo se accede a éstas al definir información de administración.
- Coexistencia entre el entorno original y el entorno SNMPv2.

Antes de comenzar, hay que aclarar la relación entre variables, objetos y tipos de objetos. Un **objeto administrable** tiene asociado una sintaxis y una semántica de tipo abstracto, mientras que una variable es una instancia de un objeto particular; en este caso también se denomina instancia de un objeto.

4.2 ESTRUCTURA DE LA INFORMACIÓN DE ADMINISTRACIÓN

La **Estructura de la Información de Administración** (SMI) define las reglas para definir la información de administración independientemente de los detalles de implementación. La SMI se define usando **ASN.1 (Abstract Syntax Notation)**.

Si se piensa que una colección de objetos administrados están almacenados, por ejemplo, en una base de datos, la **SMI** define el esquema de esa base de datos. En realidad, esa base de datos se denomina **Base de Información de Administración (MIB)**.

Índice para este punto:

1.- Módulos de información.

2.- Definiciones de objetos.

3.- Convenciones textuales.

4.- Grupos de objetos.

5.- Identificando una instancia de un objeto.

6.- Definiciones de notificaciones.

7.- Ejemplos de uso.

4.3 MÓDULOS DE INFORMACIÓN

Existen tres clases de módulos ASN.1 , también llamados Módulos de Información, definidos por él.

- Módulos MIB, que define una colección de objetos de administración afines.
- Sentencias de Conformidad, que definen un conjunto de requisitos de los nodos con respecto a uno o más módulos MIB.
- Sentencias de Capacidad, que describe la capacidad de un nodo para implementar los objetos definidos en uno o más módulos MIB.

Por supuesto, estas funciones deberían estar combinadas en un sólo módulo.

4.3.1 Identificando un Módulo de Información

Cada módulo de información debe comenzar con la indicación de su identidad y la historia de sus revisiones. Para ello, la SMI define una macro especial (**MODULE-IDENTITY**):

```
MODULE-IDENTITY MACRO ::=
        BEGIN
                TYPE NOTATION ::=
                        "LAST-UPDATED" value(Update UTCTime)
                        "ORGANIZATION" Text
                        "CONTACTO-INFO" Text
                        "DESCRIPTION" Text
                        RevisionPart
                VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)
                        RevisionPart ::= Revisions | empty
                        Revisions ::= Revision | Revisions Revision
                        Revision ::=
                                "REVISION" value(Update UTCTime)
                                "DESCRIPTION" Text
                                -- se usa el juego de caracteres ASCII NVT
                                Texto ::= """string"""
```

Ejemplo 4.2 Definición Macro MODULE IDENTITY.

La macro MODULE-IDENTITY se usa para mantener los históricos de las revisiones de cada módulo de información. Debe existir una y sólo una en cada módulo de información.

4.3.2 Cláusula LAST-UPDATED

La cláusula LAST-UPDATED, que debe existir, contiene la fecha y hora de la última revisión realizada sobre este módulo de información.

4.3.3 Cláusula ORGANIZATION

La cláusula ORGANIZATION, que debe existir, contiene una descripción textual de la organización bajo cuyo auspicios se desarrollo este módulo de información.

4.3.4 Cláusula CONTACT-INFO

La cláusula CONTACT-INFO, que debe estar presente, contiene el nombre, dirección postal. número del teléfono, y la dirección de correo electrónico de la persona a quien deben ser enviadas las preguntas de carácter técnico relacionadas con este módulo de información.

4.3.5 Cláusula DESCRIPTION

La cláusula DESCRIPTION, indispensable, contiene una descripción textual de alto nivel de los contenidos de este módulo de información.

4.3.6 Cláusula REVISIÓN

La cláusula REVISION, que no es necesaria, es usada normalmente para describir las revisiones hechas al módulo de información, en orden inverso a su sucesión cronológica. Cada instancia de esta cláusula contiene la fecha y hora de la revisión.

4.3.7 Cláusula DESCRIPTION

La cláusula DESCRIPTION, que debe estar presente por cada cláusula REVISION, contiene una descripción textual de alto nivel de la revisión identificada en esa cláusula REVISION.

4.3.8 Valor de MODULE-IDENTITY

El valor devuelto de una llamada a la macro MODULE-IDENTITY es un identificador de objetos (OBJECT IDENTIFIER). Como tal, este valor puede usarse al referirse al módulo de información que contiene la llamada.

4.3.9 Ejemplo de Uso

Ejemplo de cómo podría construirse el esqueleto de un módulo MIB:

```
FIZBIN-MIB DEFINITIONS ::= BEGIN
        IMPORTS
                MODULE-IDENTITY, OBJECT-TYPE, experimental
                FROM SNMPv2-SMI;
        fizbin MODULE-IDENTITY
                LAST-UPDATED "9210070433Z"
                ORGANIZATION "IETF SNMPv2 Grupo de trabajo"
                CONTACT-INFO
                " Marshall T. Rose
                Postal: Dover Beach Consulting, Inc.
                420 Whisman Court
                Mountain View, CA 94043-2186,
                US
                Tel: +1 415 968 1052
                Fax: +1 415 968 2510
                E-mail: mrose@dbc.mtview.ca.us "
        DESCRIPTION
                "Módulo MIB para entidades SNMPv2."
                REVISION "9210070433Z"
        DESCRIPTION
                "Versión inicial de este módulo MIB."
                ::= { snmpModules 1 }
        END
```

Ejemplo 4.3 Esqueleto de un Módulo MIB.

4.4 USO DE OBJECT IDENTIFIER

Un identificador de objeto es un nombre asignado arbitrariamente. Los identificadores de objetos definidos en el SMI para el protocolo de gestión son:

```
-- camino al root
internet OBJECT IDENTIFIER ::= {iso 3 6 1}
directory OBJECT IDENTIFIER ::= {internet 1}
mgmt OBJECT IDENTIFIER ::= {internet 2}
experimental OBJECT IDENTIFIER ::= {internet 3}
private OBJECT IDENTIFIER ::= {internet 4}
enterprises OBJECT IDENTIFIER ::= {private 1}
security OBJECT IDENTIFIER ::= {internet 5}
snmpV2 OBJECT IDENTIFIER ::= {internet 6}
```

Ejemplo 4.4 Identificadores de Objetos Definidos en el SMI.

La raíz del subárbol administrada por la Autoridad de Números Asignados para Internet (IANA) es:

Internet **OBJECT IDENTIFIER** ::= { iso 3 6 1 }

Es decir. el subárbol de Internet de identificadores de objetos comienza con el prefijo: **1.3.6.1.**

Varias ramas debajo de este subárbol se usan para la gestión de la red:

mgmt **OBJECT IDENTIFIER** ::= { internet 2 }

experimental **OBJECT IDENTIFIER** ::= { internet 3 }

private **OBJECT IDENTIFIER** ::= { internet 4 }

enterprises **OBJECT IDENTIFIER** ::= { private 1 }

Sin embargo, el SMI no prohíbe la definición de objetos en otras porciones del árbol de objetos.

El subárbol Mgmt(2) es usado para identificar objetos "estándar".

El subárbol experimental(3) es usado para identificar objetos diseñados por grupos de trabajo del IETF. Si un módulo de información producido por un grupo de trabajo se convierte en un módulo de información "estándar", entonces en el momento de su entrada en los cauces estándar de Internet, los objetos se mueven al subárbol Mgmt(2).

El subárbol Private(4) se usa para identificar objetos definidos de forma unilateral. El subárbol Enterprises(1) bajo el Private se usa, entre otras cosas, para permitir a los proveedores de subsistemas de red registrar modelos de sus productos.

El subárbol SnmpV2 se usa con propósitos de mantenimiento.

4.5 DEFINICIÓN DE OBJETOS

La macro OBJECT-TYPE se usa para definir un objeto gestionado. La expansión de la macro OBJECT-TYPE conceptualmente sucede durante la implementación y no en tiempo de ejecución.

La definición de la macro es:

```
OBJECT-TYPE MACRO::=
        BEGIN
                TYPE NOTATION ::=
                "SINTAX" type(Syntax)
                UnitsPart
                "MAX-ACCESS" Access
                "STATUS" Status
                "DESCRIPTION" Text
                ReferPart
                IndexPart
                DefValPart
                VALUE NOTATION ::= value (VALUE ObjectName)
                UnitsPart ::= "UNITS" Text | empty
                Access ::= "no-accesible"
                | "solo-lectura "
                | "lectura y escritura"
                | "lectura y creación "
                Status ::= "actual"
                | "desaprobado"
                | "obsoleto"
                ReferPart ::= "REFERENCIA" Text | empty
                IndexPart ::= "ÍNDICE" "{" IndexTypes "}"
                | "INCREMENTOS" "{" Entry "}"
                | empty
                IndexTypes ::= IndexType | IndexTypes "," IndexType
                IndexType ::= "IMPLIED" Index | Index
                Index ::=
                -- utiliza el valor de SYNTAX
                -- de la correspondiente
                -- llamada OBJECT-TYPE
                value ( Indexobject ObjectName)
                Entry ::=
                --Utiliza el valor de INDEX de la
                --correspondiente llamada OBJECT-TYPE
                value (Entryobject ObjectName)
```

Ejemplo 4.5 Definición de la Macro OBJECT-TYPE.

4.5.1 Cláusula SINTAX

La cláusula SINTAX, que debe estar presente, define las estructuras de datos abstractas correspondientes a ese objeto. La estructura de datos debe ser una de las alternativas definidas en ObjectSyntax.

Se permiten la creación de subtipos de **ASN.1**, así como adaptar los tipos de ASN.1, principalmente como una ayuda a los implementadores para mejorar la comprensibilidad del objeto. Cualquier restricción en tamaño, rango, enumeraciones o repertorio especificados en esta cláusula representan el nivel máximo de apoyo que tiene "sentido en el protocolo". Por supuesto, no está permitida la creación de subtipos para los tipos **Counter32** o **Counter64**, pero se permite para el tipo **Gauge32** (estos tipos se explican posteriormente).

La semántica de ObjectSyntax es:

```
ObjectSyntax ::=
        CHOICE {
                Simple SimpleSyntax,
                --Las sucesiones (SEQUENCES) para las tablas
                --y filas conceptuales no se mencionan aquí...
                application-wide ApplicationSyntax
                }
                -- tipos integrados en ASN.1
                SimpleSyntax ::=
        CHOICE {
                --Los enteros con un rango más restrictivo
                --también pueden ser usados
                integer-value INTEGER (-2147483648..2147483647),
                string-value OCTECT STRING,
                objectID-value OBJECT IDENTIFIER,
                --sólo se permiten formas enumeradas
                bit-value BIT STRING
                }
```

Ejemplo 4.6 Semàntica de ObjetSynttax.

A continuación pasamos a comentar brevemente algunos tipos de dato usado:

4.5.2 Integer32 e INTEGER

El tipo Integer32 representa información de valor entero entre -2^{31} y $2^{31}-1$ inclusive (-2147483648 a 2147483647 decimal). Este tipo es indistinguible del tipo INTEGER.

El tipo INTEGER también puede usarse para representar información de valor entero, si contiene enumeraciones numéricas, o si se crea un subtipo para restringir más que el tipo Integer32. En el caso anterior, sólo esas enumeraciones de números pueden estar presentes como un valor. Aunque se recomienda que los valores enumerados empiecen en 1 y estén numerados de forma consecutiva, cualquier valor válido para Integer32 es permitido como valor enumerado y, además, los valores enumerados no necesitan ser asignados de forma consecutiva.

Finalmente, el carácter guión no se permite como una parte del nombre de ninguna enumeración de números.

4.5.3 Octec String

El tipo OCTET STRING representa datos binarios o textuales arbitrarios. Aunque SMI no especifica ninguna limitación del tamaño para este tipo, los diseñadores de MIB deben comprender que puede haber limitaciones en la implementación e interoperabilidad para tamaños superiores a 255 octetos.

4.5.4 Object Identifier

El tipo OBJECT IDENTIFIER representa administrativamente a los nombres asignados. Cualquier instancia de este tipo puede tener, como mucho, 128 subidentificadores. Además, cada subidentificador no debe exceder el valor $2^{32}-1$ (4294967295 decimal).

4.5.5 Bit String

El tipo BIT STRING representa una enumeración de bits. Esta colección contendrá valores no negativos, consecutivos y comenzando por el cero. Sólo esos bits enumerados pueden estar presentes en un valor.

Un requisito en los módulos MIB "estándar" es que el carácter guión no está permitido como parte del nombre de una enumeración de bits.

4.5.6 IpAddress

El tipo IpAddress representa una dirección internet de 32-bits. Esta es representada como un OCTET STRING de longitud 4, en formato de red.

El tipo IpAddress es un tipo etiquetado por razones históricas. Deben representarse las direcciones de red usando una llamada a la macro TEXTUAL-CONVENTION.

4.5.7 Counter32

El tipo Counter32 representa un entero no negativo que se incrementa de forma monótona hasta el valor máximo de 2^{32}-1 (4294967295 decimal), cuando se alcanza dicha cifra, se vuelve al cero y se incrementa de nuevo.

Los contadores no tienen definido un valor "inicial", y así, un valor de un Contador sólo no representa (en general) ningún volumen de información. Suelen ocurrir discontinuidades en el incremento monótono del valor en la reinicialización del sistema de gestión, y en otros ocasiones como se especificó en la descripción de un tipo de objeto que usa este tipo ASN.1. Si los otros casos pueden ocurrir, por ejemplo, la creación de una instancia de objeto en un momento distinto al de la reinicialización,

entonces el objeto correspondiente debe definirse con un valor de cláusula SINTAX de TimeStamp indicando el tiempo de la última discontinuidad.

El valor de la cláusula MAX-ACCESS para los objetos con un valor de Counter32 de su cláusula SINTAX es siempre de sólo lectura.

La cláusula DEFVAL no está permitida para los objetos con un valor de su cláusula SINTAX de Counter32.

4.5.8 Gauge32

El tipo Gauge32 representa un entero no negativo que puede aumentar o disminuir, pero nunca excederá un valor máximo. El valor máximo no puede superar 2^{32}-1 (4294967295 decimal). El máximo valor de un Gauge será siempre el valor máximo que la información que contiene pueda alcanzar; si la información contenida disminuye por debajo del máximo valor, el Gauge también disminuye.

4.6 CONVENIOS TEXTUALES

Aunque los tipos de datos específicos de la aplicación contenidos en el SMI son usados, la experiencia en la creación de módulos MIB muestra que, a veces, es conveniente definir tipos de datos con sintaxis similar a la estándar, pero con una semántica mucho más precisa. Estos tipos, se denominan **convenciones textuales**. Su codificación es idéntica a la de los otros tipos, pero dentro del MIB, poseen una semántica especial, la cual es capturada por la macro TEXTUAL-CONVENTION:

Esta macro no se invoca de la misma forma que el resto:

```
TEXTUAL-CONVENTION MACRO ::=
        BEGIN
                TYPE NOTATION ::=
                        DisplayPart
                        "STATUS" Status
                        "DESCRIPTION" Text
                        ReferPart
                        "SYNTAX" type(Syntax)
                VALUE NOTATION ::= value(VALUE Syntax)
                        DisplayPart ::= "DISPLAY-HINT" Text | empty
                        Status ::= "current" | "deprecated" | "obsolete"
                        ReferPart ::= "REFERENCE" Text | empty
                        --Usa el conjunto de caracteres ASCII NVT
                        Text ::= """"string""""
        END
```

Ejemplo 4.7 Semantica de la Macro TEXTUAL-CONVENTION.

4.7 GRUPOS DE OBJETOS

Se definen objetos afines dentro de grupos de objetos (Object Group). Desde el punto de vista de la implementación, un grupo de objetos se ve como una unidad de implementación, y un desarrollador puede codificar cero o más objetos de los contenidos en el grupo.

La asignación de identificadores de objeto a los tipos en un módulo MIB, sigue la siguiente pauta:

- Los tipos de objeto se ponen dentro de grupos de objetos.

- Un identificador de objeto es asignado a cada grupo. Por convenio, se suele anteponer al nombre, como prefijo, el nombre del módulo MIB.
- Los objetos poseen un identificador secuencial subordinado al del grupo.

4.8 IDENTIFICANDO UNA INSTANCIA DE UN OBJETO

Es necesario conocer el nombre de un objeto, para poder gestionarlo, pero los objetos en sí no son más que plantillas, y son las instancias de los mismos, las que maneja el protocolo, con lo que se debe especificar el identificador de instancia.

El protocolo de gestión utiliza, para identificar instancias, un identificador de objeto, formado por la concatenación del nombre del tipo de objeto y un sufijo. Si el objeto no es parte de una tabla, entonces, representa una instancia de un tipo de objeto en un dispositivo particular. Para identificarlo sólo hay que añadir al nombre del objeto, ".0". Ejemplo:

"nombre de objeto".0

En caso de ser un objeto parte de una tabla, su identificación posee tres posibles alternativas dependiendo de:

- El objeto es una tabla.
- El objeto es una fila de una tabla.
- El objeto es una columna de una fila de la tabla.

El protocolo no permite manejar objetos agregados, con lo que sólo se pueden gestionar objetos columna que forman las celdas de una tabla.

4.9 DEFINICIONES DE NOTIFICACIONES

La macro NOTIFICATION-TYPE se usa para definir la información contenida dentro de una transmisión no solicitada de información de gestión (es decir, dentro de una SNMPv2-Trap-PDU o InformRequest-PDU).

La definición de la macro es la siguiente:

```
NOTIFICATION-TYPE MACRO ::=
BEGIN
TYPE NOTATION ::=
ObjectsPart
"STATUS" Status
"DESCRIPTION" Text
ReferPart
VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
ObjectsPart ::= "OBJETOS" "{" Object "}"| empty
Objetos ::= Objeto | Objects "," Object
Objeto ::= value(Name ObjectName)
Status ::= "actual" | "desaprobado" | "obsoleto"
ReferPart ::= "REFERENCE" Text | empty
--usa el juego de caracteres NVT ASCII
Text ::= """ string """
END
```

Ejemplo 4.8 Definición de la Macro NOTIFICATION-TYPE.

4.9.1 Cláusula OBJECTS

La cláusula OBJECTS, no obligatoria, define la secuencia ordenada de objetos del MIB que están contenidos dentro de cada instancia de la notificación.

4.9.2 Cláusula STATUS

La cláusula STATUS, que debe existir, indica si esta definición es actual o antigua ("Current" u "Obselete" respectivamente).

El valor "Deprecated" (desaprobada) indica que la notificación está obsoleta, pero que un implementador puede desear dotar al objeto de interoperabilidad con aplicaciones más viejas.

4.9.3 Cláusula DESCRIPTION

La cláusula DESCRIPTION, que debe estar presente, contiene una definición textual de la notificación que mantiene todas las definiciones semánticas necesarias para la implementación, y debe incluir cualquier información que pueda ser comunicada en cualquier notificación asociada con el objeto. En particular, la cláusula DESCRIPTION debe documentar qué instancias de los objetos mencionados en la cláusula OBJECTS deben ser contenidas dentro de las notificaciones de este tipo.

4.9.4 Cláusula REFERENCE

La cláusula REFERENCE, no obligatoria, contiene una referencia cruzada de tipo textual a una notificación definida en algún otro módulo de información. Se usa si se tiene un módulo MIB no OSI, producido por alguna otra organización.

4.9.5 Valor de NOTIFICATION-TYPE

El valor devuelto en una llamada a la macro NOTIFICATION-TYPE es el nombre de la notificación, que es un identificador de objeto (OBJECT IDENTIFIER), un nombre asignado administrativamente.

4.10 RESUMEN

En este apartado se ha hablado de una parte fundamental de cualquier sistema, nos referimos a las reglas que rigen la formación de los distintos módulos que forman el sistema de gestión.

El **SMI** (Estructura de la información de administración) nos ofrece las reglas para definir la información de administración independientemente de los detalles de implementación. la sintaxis usará ASN.1.

Existen tres tipos de módulos de información definidos con las reglas ofrecidas por el **SMI:**

- **Módulo de Información Base (MIB):** Conjunto fundamental de objetos de administración.
- **Sentencias de Conformidad** Conjunto de requisitos de los nodos respecto a uno o más módulos MIB.
- **Sentencias de Capacidad:** Capacidad de un nodo para implementar objetos definidos en uno o más módulos MIB.

Cada módulo de información debe comenzar con la indicación de su identidad y la historia de sus revisiones (macro MODULE-IDENTITY).

Dentro de cada modulo existirán objetos, los cuales se definen con la macro **OBJECT-TYPE,** la expansión de estos se produce durante la implementación. También se usaran convenciones textuales (macro TEXTUAL-CONVENTION), que son redefiniciones mas precisas de algún tipo de datos, dentro de un MIB.

Existen tres tipos de MIB:

- Estándar : son módulos que se han convertido en un estándar
- Experimental: Esperan su oportunidad de convertirse en estándar
- Específico : son propios de alguna empresa

El módulo MIB de la V2 contiene 5 grupos de objetos: **System, Snmp, SnmpComunity, SnmpSet y SnmpBasicNotification**.

CAPÍTULO V

BASE DE INFORMACIÓN DE GESTIÓN (MIB)

Es una colección de objetos, que representan de forma abstracta los dispositivos de la red y sus componentes internos. La MIB es conforme a la SMI para TCP/IP. Cada agente SNMP contiene instrumentación que, como mínimo, debe ser capaz de reunir objetos MIB estándar. Estos objetos incluyen direcciones de red, tipos de interfaz, contadores y datos similares.

El estándar MIB de Internet define 126 objetos relacionados con los protocolos TCP/IP.

Los fabricantes que deseen pueden desarrollar extensiones del estándar MIB. Estas MIBs privadas incorporan un amplio rango de objetos gestionables, y algunas veces contienen objetos que son fundamentalmente similares a los MIBs ya definidos, en otros casos el cambio de una variable en un objeto inicia una batería de funciones en el dispositivo gestionado (como por ejemplo un autodiagnóstico).

INTERNET. es un nodo que cuelga de DOD y se le puede identificar mediante {ISO(1)ORG(3)DOD(6)INTERNET(1)} o simplemente 1.3.6.1, este valor sirve como prefijo de los nodos en niveles inferiores.

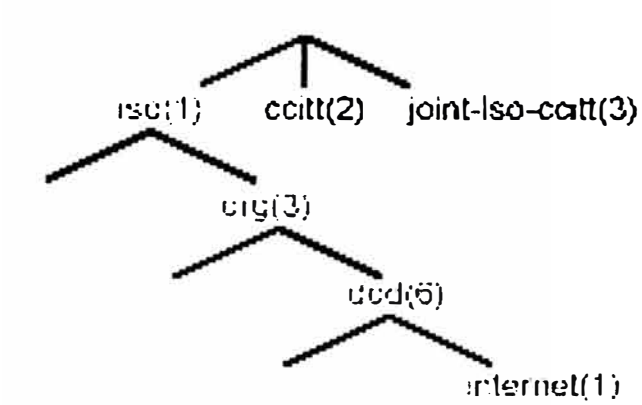


Figura 5.1 Estructura en forma de árbol.

SMI define cuatro nodos bajo el nodo INTERNET que son:

- DIRECTORY : Que está reservado para uso futuro en el directorio OSI.
- MGMT : Se usa para objetos definidos en documentos aprobados por IAB.
- EXPERIMENTAL: Se usa para identificar objetos en experimentos sobre internet.
- PRIVATE : Se usa para identificar objetos definidos unilateralmente (empresas).

5.2 DESCRIPCIÓN

El MIB define los objetos que pueden ser gestionados para cada capa en el protocolo TCP/IP. Hay dos versiones, MIB-I and MIB-II. MIB-I fue definida en el RFC 1156, y está clasificado ahora como protocolo histórico con status no recomendado.

GRUPO	NOMBRE	DESCRIPCION
System	System	Descripción del sistema
Interfaces	Interfaces	Descripción de los interfaces del sistema
Address traslation	At	Mapeo entre direcciones físicas e IP
Internet Protocol	Ip	Estadísticas del protocolo IP
Internet Control Messages Protocol	Icmp	Estadísticas del protocolo ICMP
Transmisión Control Protocol	Tcp	Estadísticas del protocolo TCP
User Datagram rotocol	Udp	Estadística del protocolo UDP
Exterior Gateway Protocol	Egp	Estadísticas del protocolo EGP
Transmisión	Transmission	MIB de los medios de transmisión
SNMP	Snmp	Estadísticas del protocolo SNMP

Tabla 5.1: MIB - Definición de grupo.

Cada nodo gestionado soporta sólo los grupos apropiados. Por ejemplo, si no hay pasarela, el grupo EGP no tiene por qué estar incluido. Pero si un grupo es apropiado, todos los objetos en ese grupo deben estar soportados.

La lista de objetos gestionados definidos deriva de aquellos elementos considerados esenciales. Este enfoque, consistente en tomar sólo los objetos esenciales no es restrictivo, ya que el SMI proporciona mecanismos de extensibilidad tales como la definición de una nueva versión de MIB o de objetos privados o no estandarizados.

Debajo hay algunos ejemplos de objetos de cada grupo. La lista completa está definida en el RFC 1213.

5.2.1 Grupo de sistema

- SysDescr : Descripción completa del sistema(versión, HW, OS).
- SysObjectID : Identificación que da el distribuidor al objeto .
- SysUpTime : Tiempo desde la última reinicialización .
- SysContact : Nombre de la persona que hace de contacto .
- SysServices : Servicios que ofrece el dispositivo .

5.2.2 Grupo de interfaces

- IfIndex : Número de interfaz.
- IfDescr : Descripción de la interfaz.
- IfTyp :Tipo de la interfaz.
- IfMtu : Tamaño máximo del datagrama IP .
- IfAdminisStatus : Status de la interfaz .
- IfLastChange : Tiempo que lleva la interfaz en el estado actual .
- IfINErrors : Número de paquetes recibidos que contenían errores.
- IfOutDiscards : Número de paquetes enviados y desechados.

5.2.3 Grupo de traducción de direcciones

- AtTable : Tabla de traducción de direcciones.
- AtEntry : Cada entrada que contiene una correspondencia de dirección de red. a dirección física .
- AtPhysAddress : La dirección física dependiente del medio .

- AtNetAddress : La dirección de red correspondiente a la dirección física .

5.2.4 Grupo IP

- IpForwarding : Indica si la entidad es una pasarela IP.
- IpInHdrErrors : Número de datagramas de entrada desechados debido a errores en sus cabeceras IP
- IpInAddrErrors : Número de datagramas de entrada desechados debido a errores en sus direcciones IP .
- IpInUnknownProtos : Número de datagramas de entrada desechados debido a protocolos desconocidos o no soportados .
- IpReasmOKs : Número de datagramas IP reensamblados con éxito .
- IpRouteMask : Máscara de subred para el encaminamiento .

5.2.5 Grupo ICMP

- IcmpInMsgs : Número de mensajes ICMP recibidos .
- IcmpInDestUnreachs : Número de mensajes ICMP "destino inalcanzable"(destination unreachable) recibidos.
- IcmpInTimeExcds Número de mensajes ICMP "Time Exceeded"(tiempo excedido) recibidos.
- IcmpInSrcQuenchs : Número de mensajes ICMP "Source Quench(desbordamiento del emisor) recibidos.
- IcmpOutErrors : Número de mensajes ICMP no enviados debido a problemas en ICMP.

5.2.6 Grupo TCP

- TcpRtoAlgorithm : Algoritmo que determina el timeout para retransmitir octetos para los que no se ha recibido reconocimiento.
- TcpMaxConn : Límite en el número de conexiones TCP que puede soportar la entidad.
- TcpActiveOpens : Número de veces que las conexiones TCP han efectuado. Una transición directa del estado SYN-SENT al estado CLOSED.
- TcpInSegs : Número de segmentos recibidos, incluyendo aquellos con error.
- TcpConnRemAddress : La dirección IP remota para esta conexión TCP.
- TcpInErrs : Número de segmentos desechados debido a errores de formato.
- TcpOutRsts : Número de resets generados.

5.2.7 Grupo UDP

- UdpInDatagrams : Número de datagramas UDP entregados a usuarios UDP.
- UdpNoPorts : Número de datagramas UDP recibidos para los que no existía aplicación en el puerto de destino.
- UdpInErrors : Número de datagramas UDP recibidos que no se pudieron. entregar por razones otras que la ausencia de la aplicación en el puerto de destino.
- UdpOutDatagrams : Número de datagramas UDP enviados por la entidad.

5.2.8 Grupo EGP

- EgpInMsgs : Número de mensajes EGP recibidos sin error.
- EgpnErrors : Número de mensajes EGP con error.
- EgpOutMsgs : Número de mensajes EGP generados localmente.
- EgpNeighAddr : La dirección IP del vecino de esta entrada EGP.
- EgpNeighState : El estado EGP del sistema local con respecto a la entrada EGP vecino.

El grupo de interfaces contiene dos objetos de nivel superior: el número de interfaces del nodo(IfNumber) y una tabla con información de estas(IfTable). Cada entrada de la tabla(IfEntry) contiene los objetos de esa interfaz. Entre ellos, el tipo de interfaz(IfType) se identifica en el árbol MIB con notación ASN.1 como 1.3.6.1.2.1.2.2.1.3. Para un adaptador de red en anillo, su valor sería 9("Iso88025-TokenRing").

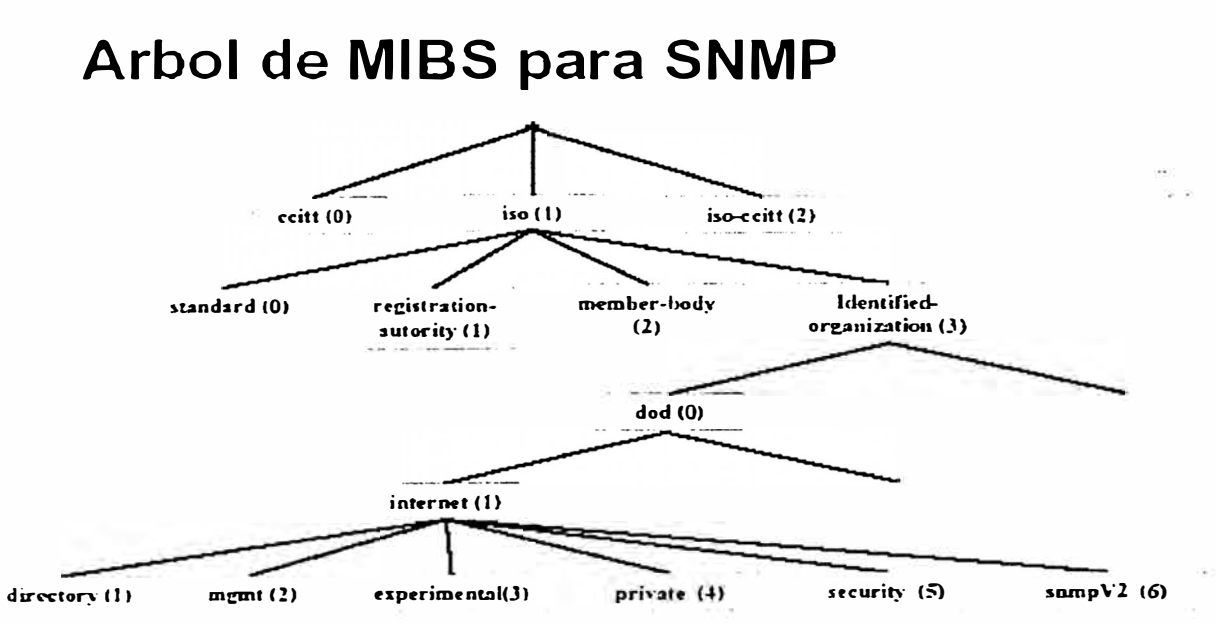


Figura 5.2: Identificador de Objeto - Asignación para redes TCP/IP.

5.3 El MIB PARA SNMPv2

Este MIB define los objetos gestionados que determinan el comportamiento de la entidad SNMPv2.

Nota: No es una sustitución del MIB-II.

Las siguientes son algunas definiciones de objetos para hacerse una idea de sus contenidos:

```
snmpORLastChange OBJECT-TYPE
  SYNTAX    TimeStamp
  MAX-ACCESS read-only
  STATUS    current
  DESCRIPTION
      "El valor de sysUpTime en el momento del cambio más reciente en el valor
    o estado de cualquier instancia de
    snmpORID."
warmStart NOTIFICATION-TYPE
  STATUS  current
  DESCRIPTION
```

Ejemplo 5.1 Definición de Objetos.

5.4 MIB MANAGER - MANAGER

La finalidad de este MIB es proporcionar los medios para la coordinación entre múltiples estaciones de gestión. Es decir, los medios por los que las funciones de control y monitorización de la gestión de red se pueden distribuir entre múltiples NMS en una gran red. Específicamente, este MIB suministra mecanismos para que una NMS solicite servicios de gestión de otra. Por tanto, una entidad SNMPv2 puede tener un doble papel; cuando proporciona información de gestión a otro manager, actúa como agente, y

cuando pide información, actúa como manager. El MIB manager-manager consta de las tres tablas siguientes:

- Alarmas
- Eventos
- Notificaciones

Cada alarma es una condición específica detectada mediante la monitorización periódica, en un intervalo de muestreo configurable, de los valores de una determinada variable con información de gestión. Un ejemplo de condición de alarma es cuando la variable monitorizada toma un valor fuera de rango. Cada condición de alarma dispara un evento, que puede a su vez desencadenar una o más notificaciones para otras NMS usando el InformRequest.

5.5 MIB PRIVADAS

Los fabricantes pueden crear objetos específicos para gestionar sus productos. El uso de SMI y un esquema de identificadores de objetos estándar permite que se pueda acceder fácilmente a esos objetos.

Los Mibs privados se cargan dentro de un sub-árbol PRIVATE (1.3.6.4.1) y más concretamente dentro de la rama ENTERPRISES (1.3.1.4.1).

Una estación gestora solo puede acceder a la información cuya existencia conoce y sobre la que puede preguntar. Para que la estación gestora pueda gestionar objetos MIB privados, su estructura debe ser cargada en la estación gestora, para mayor facilidad de

implementación algunos fabricantes proporcionan una descripción formal de su MIB en texto.

5.6 FORMATO DE MENSAJE SNMP

SNMP es un protocolo de sesión que está encapsulado en UDP. el formato de mensaje es de la siguiente manera:

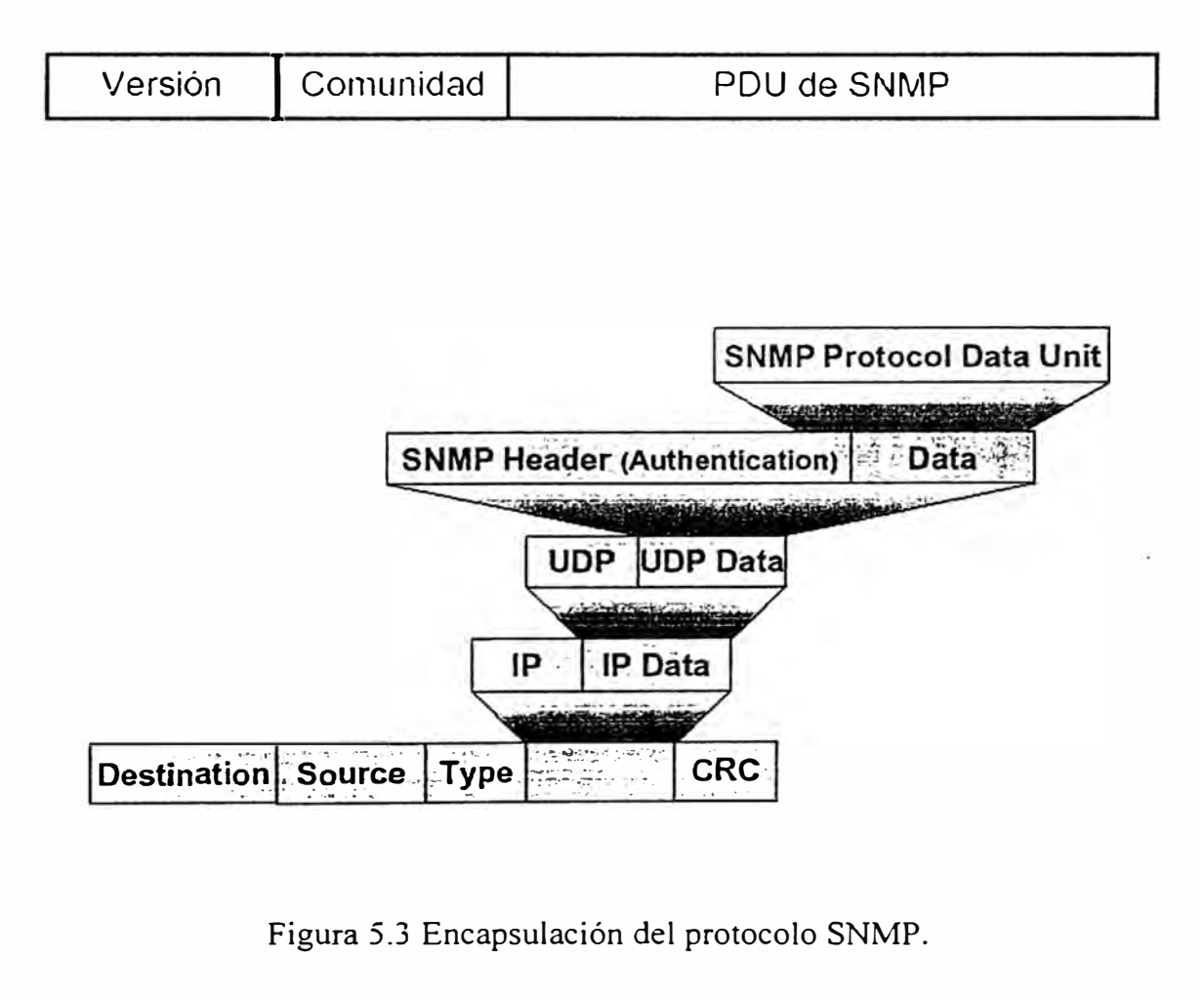


Figura 5.3 Encapsulación del protocolo SNMP.

- **Versión Number :** El Administrador y el agente deben usar la misma versión de SNMP. Mensajes que contienen diferentes números de versión son descartados sin procesarlos.
- **Community** : El nombre de comunidad es usado para autenticar al administrador antes de permitir el acceso al Agente.
- **PDU** : Existen cinco tipos de PDU: GetRequest, GetNextRequest, GetResponse. SetRequest, Trap.

5.7 FORMATO PDU

PDU type	Request ID	Error status	Error index	Object 1 value 1	Object 2, value 2	others

Tabla 5.2 Formato PDU.

5.7.1 PDU Type

Especifica el tipo de PDU.

0 = GetRequest.

1 = GetNextRequest.

2 = GetResponse.

3 = SetRequest.

5.7.2 Request ID

Campo de enteros que correlaciona la petición del administrador con la respuesta del agente.

5.7.3 Error Status

Entero enumerado que indica si fue una operación normal o una de cinco condiciones de error. los valores posibles son:

- 0 noError : Operación Administrador/Agente exitosa.
- 1 TooBig : Tamaño del PDU de GetResponse excede una limitación local.
- 2 NoSuchName : El objeto pedido no corresponde con los nombres disponibles en la se MIB.
- 3 BadValue : Un SetRequest contiene un tipo inconsistente, longitud y valor de la variable.
- 4 ReadOnly : No está definido en RFC 1157.
- 5 GenErr : Otros errores, que no están definidos explícitamente.

5.7.4 Error index

Identifica la entrada en la cual lista la variable cubierta que causo el error.

5.7.5 Object/Value

Par de variables cubiertas de nombre variable con su valor.

5.8 FORMATO DEL PDU TRAP

PDU type	Enterp	Agent addr	Gen trap	Spec trap	Time stamp	Obj 1 Val 1		

Tabla 5.3 Formato del PDU TRAP.

5.8.1 PDU Type

Especifica el tipo de PDU (4=trap).

5.8.2 Enterprise

Identifica la empresa de administración cuya autorización de registro fue definida.

5.8.3 Agent Address

Campo que describe el evento siendo reportado, los siguientes siete valores están definidos:

- 0 ColdStart : La entidad que lo manda ha sido reinicializada, indicando que la configuración del agente o la implementación de esa entidad puede ser alterado.
- 1 WarmStart : El protocolo que envía esto ha sido reinicializado, pero ni los agentes ni la implementación del protocolo han sido alterados.
- 2 LinkDown : Un enlace de comunicación ha fallado.
- 3 LinkUp : El enlace se ha restablecido.
- 4 AuthenticationFailure : El agente ha recibido un mensaje SNMP que no ha sido autentificado propiamente por el administrador, ejemplo, nombre de comunidad fue incorrecto.
- 5 EgpNeigborLos : Un punto EGP vecino esta fuera.
- 6 EnterpriseSpecific : Un trap no genérico ha ocurrido el cual es posteriormente identificado por el campo de empresa y tipo de trap específico.

5.8.4 Specific Trap Type

Usado para identificar un trap no genérico cuando el tipo genérico de trap es EnterpriseSpecific.

5.8.5 Timestamp

Valor del objeto SysUpTime, que representa el tiempo acumulado entre la ultima reinicializaciòn y la generación de ese trap.

Object/value

- **Operación del protocolo SNMP**

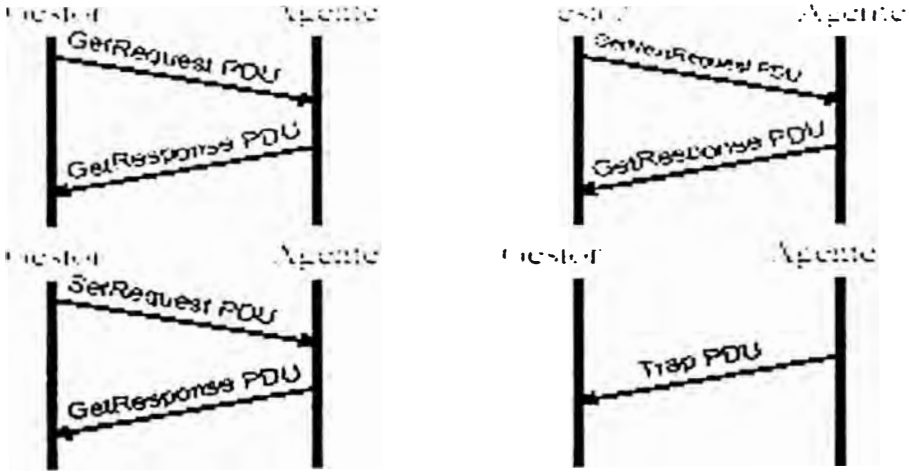


Figura 5.4 Operación del protocolo SNMP

El corazón del marco de SNMP es el protocolo en si mismo, este provee un mecanismo básico y poderoso para el intercambio de información de gestión entre el software administrador y los agentes.

La unidad básica de intercambio es el mensaje, el cual consiste de un envoltorio exterior de mensaje y un PDU (Protocol Data Unit, Protocolo de Unidad de Dato) en su interior.

Ocho tipos de PDUs pueden ser transportados en un mensaje SNMP.

Mensaje	Descripción
Get-Request	Solicita el valor de una o mas variables.
Get-Next-Request	Solicita la siguiente variable.
Get-Bulk-Request	Busca una gran tabla.
Set-Request	Actualiza una o mas variables.
Inform-Request	Administrador a administrador describiendo el MIB local.
SnmpV2-Trap	Reporte de trap de agente a administrador.
Response	Respuesta del agente a una solicitud Get-Request

Tabla 5.4 Tipos de PDU'S.

5.9 DESCRIPCIÓN DEL FORMATO DE LAS TRAMAS PDU

A continuación se explican en detalle cada una de las tramas PDU:

5.9.1 Get-Request-PDU

Es emitido por la estación de gestión, incluye una lista de uno o varios nombres de objetos cuyos valores han sido solicitados. Si la operación Get es exitosa, entonces el agente responde enviando una trama Response-PDU. La lista de variables-vinculadas contendrá el identificador y el valor de todos los objetos recuperados.

5.9.2 GetNextRequest-PDU

Es emitido por la estación de gestión, e incluye una lista de uno o mas objetos. En este caso, para cada objeto definido en el campo de variables-vinculadas, se retorna un valor correspondiente al siguiente en el orden lexicografico, es decir, el siguiente en el MIB en términos de su posición en la estructura del árbol de identificación de objetos. Una de las fortalezas del GetNextRequest-PDU, es que el habilita una entidad de gestión que descubre la estructura de una vista MIB de forma dinámica, lo cual es muy útil si la estación de gestión no conoce a priori el conjunto de objetos que son soportados por un agente o que están en un MIB particular.

5.9.3 GetBulkRequest

Es de las mejoras más resaltantes que provee el SNMPv2. El propósito de este PDU es minimizar el número de intercambios requeridos para recuperar una larga cantidad de información gestión. El PDU GetBulkRequest permite a la estación de gestión solicitar una respuesta tan grande como el tamaño del mensaje lo permita.

EL PDU GetBulkRequest, utiliza los mismos principios que el PDU GetNextRequest, esto es, seleccionar siempre la instancia del siguiente objeto en orden lexicográfico. La diferencia es que, con GetBulkRequest se hace posible especificar que varios sucesores lexicográficos han sido seleccionados.

5.9.4 SetRequest-PDU

Es transmitido por la estación de gestión para solicitar que uno o más valores de uno o más objetos sean alterados. La entidad SNMPv2 que reciba la trama PDU SetRequest.

responde con un Responce-PDU conteniendo el mismo Request-Id. La operación SetRequest es atómica, es decir, la variable es actualizada o no.

5.10 SNMPv2-Trap-PDU

Es generado y transmitido por una entidad actuando en el rol de agente hacia la estación de gestión cuando un evento inusual ocurre. Es utilizado para proveer a la estación de gestión una notificación asíncrona acerca de algún evento significativo. La lista de variables-vinculadas contiene la información asociada con el mensaje Trap. A diferencia de los otros PDU, el SNMPv2-Trap-PDU no genera una respuesta de la entidad receptora, es un mensaje sin confirmación.

CAPÍTULO VI

SNMP EN LINUX

Ahora bien, SNMP es un mundo muy complejo y amplio sin embargo son muchas las herramientas disponibles con SNMP en su servidor Linux. Pues bien, con Linux y con herramientas de software libre se pueden hacer, entre otras cosas, las siguientes:

- Instalar un agente SNMP para monitorizar variables en un servidor con Linux.
- Utilizar en una estación con Linux una herramienta de gestión para observar variables de agentes SNMP.
- Programar un interfaz para tomar medidas en base a la consulta (monitorización de variables de un elemento SNMP).
- Programar un interfaz para recibir alertas SNMP y tratarlas como sea necesario.

Existen, herramientas propietarias para llevar a cabo estas funciones. Algunas de las más conocidas, como HP OpenView, SunNet Manager e IBM Netview, soportan muchas de las funciones que se van a tratar aquí. Sin embargo se va a entrar en detalle en herramientas de software libre, por considerarse que serán las herramientas más útiles

para un desarrollador que quiera conocer "por dentro" el funcionamiento de los protocolos.

El estar en posesión del código fuente ayuda en gran medida a la persona que tiene que entrar en contacto con la tecnología ya que, rápidamente, puede familiarizarse con ésta a través de una implementación, o la puede poner a prueba compilándola e instalándola.

6.1 AGENTES SNMP EN LINUX

En la mayoría de los sistemas Linux, se incluye un agente de SNMP que se trata de uno de los más desarrollados en la actualidad. Se trata de la actualización de la librería SNMP de la Universidad de California en Davis (que a su vez se basa en la librería de la Universidad de Carnegie Mellon). La librería se llamaba, en versiones previas Ucd-Snmp, ahora se denomina Net-Snmp. La versión actual ha sido portada a Linux de la librería original por Juergen Schoenwaelder y Erik Schoenfelder, el desarrollador principal es Wes Hardaker.

Esta librería ha sido muy actualizada y desarrollada e incluye las herramientas de SNMP "tradicionales". Las últimas versiones parten de la base de código de la versión 2.1 y han sido tremendamente mejoradas.

La versión actual, la 4.1, incluye soporte para todas las versiones de SNMP (desde la uno, a la tres). Los agentes de SNMP que instala son perfectamente extensibles, tanto a través del propio código (con la API proporcionada) como a través de comandos definidos en la configuración.

Al tratarse de un software de agentes tan extendido, es conveniente detenerse un poco en su instalación y configuración, así como en las herramientas que proporciona.

6.2 INSTALACIÓN DE NET-SNMP EN UN SERVIDOR LINUX

El primer paso será, sin duda, obtener el código fuente de la distribución. Anteriormente estaba disponible en http://ucd-snmp.ucdavis.edu, pero ahora se ha movido a Sourceforge (para aprovechar los recursos que éste ofrece para proyectos libres) y está disponible en **http://www.sourceforge.net/proyects/net-snmp**.

Una vez descargado y descomprimido en un directorio, se puede proceder a compilar el código fuente para ello se hace desde la raíz:

```
Linux$ configure
Linux$ make all
```

Ejemplo 6.1 Raíz Para Copilar el Código Fuente.

Con suerte, quedará compilado y preparado para instalar. Esta librería no depende de otras, es autocontenida, lo que facilita su compilación Lo cual se podrá hacer con Make Install. Ahora queda poner una configuración adecuada en el fichero /etc/snmp/snmpd.conf.

Las distribuciones actuales, por ejemplo Debian ó RedHat, incorporan ya el paquete de Ucd-Snmp de forma que su instalación es mucho más sencilla (son binarios ya compilados) y su configuración rápida. Por ejemplo, para el paquete Debian de Ucd-Snmp, basta con instalarlo para tener ya un agente ejecutándose de forma transparente al usuario.

De hecho en la distribución se incluyen dos agentes. El primero Snmpd es un agente que permanece escuchando en el puerto 161 (Udp) esperando recibir peticiones, cuando le llega una solicitud la procesa y devuelve la información. El segundo, Snmptrapd se trata de un agente que procesa las alertas de otros agentes. Para ello permanece escuchando en el puerto 162 (Udp), cuando recibe una alerta por este puerto procede a guardarla en el registro (Syslog). Sin embargo también puede ser configurado para utilizar programas externos en el tratamiento de las alertas.

Los agentes de Net-Snmp incluyen una serie de extensiones para poder obtener información específica del sistema como son:

- Información general del sistema .
- Conexiones tcp/udp/ip/snmp abiertas y estado .
- Discos duros.
- Procesos y carga del procesador .

6.3 CONFIGURACIÓN DE LOS AGENTES

Una vez instalados los agentes sólo será necesario adaptarlo a las necesidades del equipo en el que va a estar instalado. La librería incluye una buena documentación que describe el formato de los ficheros de configuración.

En la página de manual Snmpd_Config se describe el funcionamiento general de los ficheros de configuración. En la instalación en sistemas Debian, el agente queda instalado con un fichero de ejemplo de configuración, en otros caso será necesario copiar (o crear) uno en **/etc/snmp/snmpd.conf**. Pero muchos de los problemas pueden venir por no entender correctamente el modo de funcionamiento de la autenticación en SNMP.

Las primeras definiciones en el fichero de configuración definen las limitaciones para el acceso al agente desde cualquier servidor. Uno de los problemas más comunes es no ser capaz de acceder al agente por que estas restricciones son muy fuertes o no se han definido correctamente. El funcionamiento es, quizás, un tanto complejo, pero esto se debe a que el agente tiene soporte para la autenticación en SNMPv1, en SNMPv2c (con comunidades) y en SNMPv3 (a través de usuarios y grupos). Net-Snmp implementa el Modelo de Control de Accesos Basados en Vistas (VACM, View-Access Control Model) definido como RFC.

Lo primero que se debe definir es una relación entre comunidades y modelos de seguridad en el agente SNMP, tras esto se define una relación entre modelos de seguridad y grupos, se definen vistas (que son zonas del árbol de la MIB) y, finalmente, se indica el acceso permitido de los grupos a las vistas.

Esto puede parecer complejo, pero quedará más claro con un ejemplo. Si se tiene definida la siguiente relación:

```
#     sec.name source community
com2sec readonly default public
com2sec readwrite 127.0.0.1 private

# sec.model sec.name group
MyROSystem v1 paranoid group
MyROSystem v2c paranoid group |
MyROSystem usm paranoid group
MyROGroup v1 readonly group
MyROGroup v2c readonly group
MyROGroup usm readonly group
MyRWGroup v1 readwrite group
MyRWGroup v2c readwrite group
```

Ejemplo 6.2 Definición de Relación entre Modelos de Seguridad y Grupos.

Se está incluyendo todos los accesos como comunidad "Public" desde cualquier lugar al grupo MyROGroup, mientras que los accesos como comunidad "Private" desde el servidor local se vinculan al grupo MyRWGroup. Con las siguientes vistas definidas se termina la definición de los accesos a los agentes:

```
# incl/excl subtree mask
view all included .1 80
view system included .iso.org.dod.internet.mgmt.mib-
2.system

# context sec.model sec.level match read write notif
access MyROSystem "" any noauth exact all none none
access MyROGroup "" any noauth exact all none none
access MyRWGroup "" any noauth exact all all none
```

Ejemplo 6.3 Definición de los Accesos a los Agentes.

Con esta configuración garantizamos el acceso de escritura al grupo definido anteriormente (MyRWGroup) a cualquier parte de la MIB, mientras que sólo se permite leer dentro de la vista System (que está definida como una parte limitada de la MIB disponible) al grupo de sólo lectura.

Sin embargo a través de la configuración permite adaptar mucho más que sólo el acceso al agente. Entre otras cosas se puede:

Hacer que el agente monitorice la existencia de procesos: Proc. De esta forma se puede controlar que, por ejemplo, el proceso apache tenga más de 100 procesos.

También es posible tomar acciones en caso de que las limitaciones impuestas a los procesos no se cumplan, definiéndolas con Procfix.

Hacer que el agente ejecute comandos con la función Exec. El agente ejecutará estos comandos cuando se consulte la variable de la MIB que se defina. De esta forma se puede utilizar el agente como una herramienta de gestión que tome acciones dentro del sistema. ampliando su comportamiento a algo más allá que un mero elemento que monitoriza variables en el sistema.

- Hacer que el agente controle la carga de la máquina para que se mantenga en unos límites determinados con el parámetro load.
- Definir algunos de los parámetros internos del agente en la MIB, como la ubicación del sistema (Syslocation) o la persona de contacto (Syscontact).
- Configurar el agente para enviar alertas a otros agentes cuando se den las condiciones necesarias. Para ello se tiene que definir la comunidad a utilizar con Trapcommunity y el servidor concreto a utilizar con Trapsink, Trap2sink, ó Informsink.

El formato en detalle de la configuración de los agentes se puede consultar en la página de manual Snmpd.conf

6.4 FAMILIARIZÁNDOSE CON EL AGENTE

Ya se debería tener el agente configurado y funcionando, hecho que se puede comprobar mirando e listado de procesos (con, por ejemplo, Ps Aux |Grep Snmp) y de conexiones para ver que hay un proceso escuchando en el puerto 161 (con, Netstat -Anp-u). Si no se ha lanzado aún habrá que lanzarlo ejecutando /usr/sbin/snmpd, la mayoría de las distribuciones instalarán un programa para poder parar y lanzar el demonio de forma

sencilla. En el caso de Debían esto se consigue llamando el Script /etc/init.d/snmp con la orden Start.

Tras esto. llega el momento de familiarizarse con las herramientas de gestión SNMP incluidas dentro de Net-Snmp. Estas son:

- Snmpstatus que permite acceder a la situación del agente.
- Snmpwalk que permite recorrer la MIB del agente y sus variables.
- Snmpget y Snmpset que permiten, respectivamente, consultar y fijar atributos de SNMP.
- Snmptranslate permite traducir de un identificador de objeto (OID) de la MIB a una cadena de caracteres representativa de éste.
- Snmpdelta, establece un proceso de monitorización sobre una o más variables del agente. de forma que recuperar el valor de estas variables en periodos de tiempo definidos.
- Snmptest es una herramienta de prueba del agente, al conectarse permite, a través de una interfaz de línea de comandos, recuperar cualquier variable que este contenga. Indica los métodos de comunicación usados contra el agente, por si fuera necesaria su depuración.
- Snmpnetstat, es un comando atípico en las distribuciones de SNMP ya que es particular de la distribución Set-Snmp. Nos permite obtener un listado de los canales de comunicación abiertos en una máquina, al igual que Nestat, pero utilizando un agente SNMP para recuperar la información.

Muchas de estas funciones son comunes de cualquier implementación de SNMP y el desarrollador las encontrará en cualquier distribución. Así, si se desea saber si el agente está activo se haría:

```
S snmpstatus -v 1 localhost public
[127.0.0.1]=>[Linux templar2.2.16-storm #1 Thu Aug 24 18:29:48 PDT 2000
i686] Up: 0:17:56.24
Interfaces: 0, Recv/Trans packets: 1908/1908 | IP: 1906/1906
```

Ejemplo 6.4 Definición para verificar Agente Activo.

Para consultar toda una rama se puede utilizar el comando Snmpwalk un ejemplo de su uso se muestra en el listado 1. Para obtener un valor concreto del árbol (por ejemplo, la fecha del sistema) se ejecutaría:

```
LinuxS snmpget localhost public host.hrSystem.hrSystemDate.0
host.hrSystem.hrSystemDate.0 = 2001-2-12,18:51:20.0,+1:0
```

Ejemplo 6.5 Definición Para Obtener la Fecha del Sistema.

6.5 OTROS AGENTES SNMP PARA LINUX

El agente de Net-Snmp no es el único agente disponible para los sistemas Linux aunque sí el que posiblemente se incluya en más distribuciones y esté más probado y extendido. Otros agentes a considerar dentro de Linux son:

- **Snmpd-Tcl (disponible en http://geekcorp.com/snmpd) :** Se trata de una extensión mas que un agente en sí, para proveer la MIB de Recursos del Sistema (RFC 1514) dentro de un agente SNMP. Está pensado para poder multiplexar varios agentes a través del mismo puerto, de forma que cada uno ofrezca una MIB determinada.
- **Opennms :** El proyecto de Gestión Abierta de Redes (Open Network Management, www.opennms.org) ofrece una librería Java con licencia LGPL denominada JoeSNMP. Esta librería incluye una arquitectura completa de agentes SNMP desde el agente en sí a agentes para procesar las alarmas y un gestor genérico.
- **Agent++ :** Que es una implementación de agentes SNMP en C++ que soporta desde la versión 1 hasta la 3. Su licencia de distribución no es, sin embargo, libre (disponible en http://www.agentpp.com) .
- **SNMP++ :** Al igual que el anterior está programado en C++, pero no soporta las mismas versiones de SNMP (de hecho es la base sobre la que se construyó Agent++). Su licencia de distribución tampoco es libre, disponible en: **http://rosegarden.external.hp.com/snmp++/ .**

6.6 HERRAMIENTAS PARA MONITORIZAR AGENTES

Sin embargo un agente no sirve para mucho sin herramientas que lo monitoricen, esto lo dirá cualquier administrador. Dejando de un lado, temporalmente, el punto de vista del desarrollador y programador podemos pasar a responder la siguiente pregunta:

6.6.1 ¿Qué Necesita un Administrador?

Pues ni más ni menos que una herramienta visual, desde las que poder consultar las variables de los agentes, poner monitores para comprobar su evolución, y ver "gráficamente" las alertas.

Nuestro administrador de sistemas no tiene que pensar que no va a encontrar ésto aquí, y que ésto de Linux es sólo para los desarrolladores. Se le puede informar de que existen un buen número de herramientas disponibles para los sistemas Linux con soporte de SNMP (o que van a tenerlo pronto, según sus autores). Importante a destacar, y esto hará las delicias del personal de contabilidad, es que estas herramientas no tienen **licencias** de miles de euros (como sus equivalentes propietarias) y que no les van tan a la zaga en cuanto a características y funcionalidad disponible.

Entre otras herramientas podemos hablar de:

- **Scotty (también conocido como Tkined) :** Es una herramienta completa de monitorización incluye capacidades de gestión/monitorización de dispositivos SNMP. Está implementada en Tcl/Tk, con extensiones propias, e incluye hasta un navegador de MIBs. Disponible en http://wwwsnmp.cs.utwente.nl/ schoenw/scotty/
- **Softguard :** Se trata de un navegador para agentes SNMP y sus MIBs, incluye funciones de auto descubrimiento y está también implementado en Tcl/Tk aunque está mucho más orientado hacia SNMP que Scotty. Se puede obtener de http://www.osn.de/user/finzel/html/sgSpies.html
- **NetraMet. :** Se trata de una herramienta diseñada para gestionar el accounting de servidores (RFC 1272, 2063, 2064 y 2123). Incluye una implementación para tratar

las extensiones de NetFlow de CISCO. Descargable desde http://www.auckland.ac.nz/net/NeTraMet/

- **Gxsnmp :** Es un gestor de elementos SNMP aún en desarrollo pero con un gran potencial. Las versiones actuales son betas muy recientes que carecen de un gran número de funcionalidades. Sin embargo sus bases son sólidas. Utiliza una base de datos para almacenar la información de agentes y redes (tiene interfaces programados a varias, entre otras, Mysql), y la librería SMI para acceder a las MIBs. Es posible que, en un futuro y cuando sea más madura, sea la aplicación por excelencia para gestión de agentes SNMP en entornos Linux. Se puede obtener en http://www.gxsnmp.org .
- **Gkrellm :** Es un monitor que permite monitorizar múltiples características del sistema, desde la capacidad del disco al uso de la CPU con un bonito Widget en el escritorio. Existe una extensión a este monitor que permite incorporar variables SNMP para monitorizarlas junto con el resto de características del sistema.
- **Mrtg. :** Herramienta con interfaz WWW que permite una lectura en tiempo real de estadísticas de distintos elementos, entre otros, dispositivos SNMP. Es una de las herramientas más conocidas para monitorización de tráfico, y una de las más extendidas. Consultar www.mrtg.org
- **Cheops. :** Herramienta sustitutiva de Scotty para la gestión de elementos de red, aún no incluye soporte de SNMP pero es tremendamente gráfica e intuitiva.
- **Mon. :** Se trata de una herramienta integrada para la gestión de red, soportando múltiples sistemas en los que, a través de agentes, se pueden monitorizar las

aplicaciones de éstos y su rendimiento. Tiene soporte de SNMP y ofrece la posibilidad de definir muchos niveles de alertas, desde correo electrónico a notificaciones con voz en tiempo real. Disponible en http://www.kernel.org/software/mon.

- **Big Brother. :** Pretende ser una herramienta integrada, con una interfaz orientada a WWW para monitorizar estadísticas. Su licencia no permite la distribución comercial, pero aún así tiene características interesantes, aunque se ve superada por las anteriores.

6.7 DESARROLLO DE AGENTES

No se puede terminar este artículo sin estudiar la forma de integrar la gestión SNMP con aplicaciones a través de la programación de interfaces de acceso SNMP. De no hacerlo así es probable que algún programador experimentado se sienta defraudado por no haber llegado hasta el último detalle.

Pues bien, aunque existan herramientas, como ya se han visto, de gestión de dispositivos, un desarrollador puede querer generar su propia aplicación a medida para acceder a valores ofrecidos por elementos SNMP. De hecho esto puede ser incluso un requisito de un cliente y quizás la solución no este disponible entre los elementos que ya hemos destacado.

Actualmente se pueden desarrollar aplicaciones con acceso a SNMP utilizando librerías libres en tres lenguajes distintos: Perl, Tcl/Tk, PHP, C y Java.

En el caso de Perl existen tres librerías distintas de acceso a SNMP, la librería Net:SNMP (disponible en cpan.perl.org), la librería SNMP_Session (disponible en ftp://ftp.switch.ch/software/sources/network/snmp/perl/) y el módulo de extensión de Perl para la librería UCD SNMPv3 (disponible en ftp://ftp-east.baynetworks.com/netman/snmp/perl5). Salvo la última, estas librerías no necesitan tener ninguna librería de agentes instalada en el sistema.

Para Tcl existe la extensión proporcionada por el interfaz de gestión Scotty denominada Tnm:snmp (Tnm es la extensión para herramientas de gestión de red) que permite incorporar funciones de gestión de SNMP tanto dentro de Scotty como fuera de éste. También el lenguaje PHP (versión 3 y versión 4) incorpora extensiones con módulos SNMP para poder programa aplicaciones con interfaces WWW con acceso a dispositivos de red.

Para las versiones compiladas (C y Java) se puede utilizar cualquiera de las librerías de desarrollo mencionadas anteriormente (Ent.-Sep u Hopéenme) ya que cualquiera de ellas ofrece una API completa para el acceso a las funciones de SNMP.

CAPÍTULO VII

EL GRAFICADOR MRTG

7.1 ¿Qué es el Graficador de Tráfico Multi Router?

El Graficador de Tráfico Multi Router (Multi Router Traffic Grapher, MRTG) es una herramienta para monitorear la carga de tráfico en los enlaces de una red. El MRTG genera páginas HTML las cuales contienen gráficos GIF que proveen una representación visual EN VIVO de este tráfico. El MRTG está basado en Perl y C y trabaja en estaciones de trabajo UNIX, Windows, etc. El MRTG está siendo usado exitosamente en muchos sitios alrededor de la red.

7.2 SISTEMAS QUE CORREN EL MRTG

- Linux 1.2.x, 2.0.x, 2.2.x (Intel and Alpha and Sparc)
- SunOS 4.1.3
- Solaris 2.4, 2.5, 2.5.1, 2.6, 7
- AIX 4.1.4, 4.2.0.0

- HPUX 9,10,11
- WindowsNT 3.51, 4.0, 2k
- IRIX 5.3, 6.2
- BSDI BSD/OS 2.1, 4.x, 3.1
- NetBSD (sparc)
- FreeBSD 2.1.x, 2.2.x, 3.1, 3.4
- OpenBSD 2.5, 2.6
- Digital Unix 4.0
- SCO Open Server 5.0
- Reliant UNIX
- NeXTStep 3.3
- OpenStep 4.2

7.3 PRINCIPALES CARACTERÌSTICAS

Portable	El MRTG trabaja sobre la mayoría de las plataformas UNIX y sobre windows NT.
Perl	El MRTG está escrito en Perl y viene con la fuente completa.
Portable SNMP	El MRTG usa una implementación de SNMP altamente portable escrita completamente en Perl gracias a Simon Leinen. No es necesario instalar

	ningún paquete de SNMP externo.
Soporte para SNMPv2	El MRTG puede leer los nuevos contadores de 64bit de SNMPv2. No más enredos de contadores.
Identificación de Interfaces Confiables	Las interfaces de los enrutadores pueden ser identificadas por su dirección IP. Descripción y dirección Ethernet además del número de interfaz normal.
Bitácores (logs) de tamaño constante	Las bitácoras del MRTG NO crecen. Gracias al uso de un algoritmo único de consolidación de datos.
Configuración Automática	El MRTG viene con un conjunto de herramientas de configuración las cuales hacen la configuración muy simple.
Desempeño	Las rutinas de tiempo crítico están escritas en C gracias a la iniciativa de Dave Rand (Co-Autor).
Gráficos libres de GIF	Los gráficos son generados directamente en formato PNG, usando la biblioteca GD de Thomas Boutell.
Personabilidad	La apariencia de las páginas web producidas por el MRTG son altamente configurables.
RRDtool	El MRTG tiene ganchos intrínsecos para usar RRDtool. Si usted esta atado por un buen desempeño, esto puede ayudar.

Tabla 7.1 Caracteristicas del MRTG.

7.4 DETALLES

El MRTG consiste en un programa en Perl que usa SNMP para leer los contadores de tráfico de sus enrutadores y de un rápido programa en "C" el cual archiva los datos de tráfico y crea hermosas imágenes que representan el tráfico en la conexión de red monitoreada. Esos gráficos se insertan en páginas web que pueden ser vistas desde cualquier browser moderno.

Además de una vista diaria detallada, el MRTG crea también representaciones visuales para el tráfico de los últimos siete días, las cuatro últimas semanas y los últimos doce meses. Esto es posible pues el MRTG mantiene un archivo de todos los datos que ha obtenido del enrutador. Este archivo es consolidado automáticamente, así que no crece con el tiempo, pero contiene todos los datos relevantes del tráfico de los últimos dos años. Todo esto se realiza de una manera eficiente. Por lo tanto, usted puede monitorear 200 o más enlaces de red desde cualquier máquina Unix medio decente.

El MRTG no está limitado al monitoreo de tráfico, es posible monitorear cualquier variable SNMP que usted elija. Usted puede hasta usar un programa externo para recolectar datos que serán monitoreados por el MRTG. Muchas personas están usando el MRTG para monitorear cosas como Carga del Sistema, Logueo de Sesiones, disponibilidad de módems y más. El MRTG hasta le permite acumular dos a más fuentes de datos en un único gráfico.

7.4.1 Snapshots o Fotos Instantáneas

¿Qué significan los estados del icono **VERDE** y **ROJO**?

Básicamente, MRTG registra un cierto router cada 5 minutos o menos. Los resultados SNMP de las consultas dados por los routers designados son procesados por el programa. Las salidas de MRTG son los gráficos (formato.Png) y los textos basados en Logs.

Basándonos en este principio, podemos utilizar el tiempo de actualización o Timestamp de los Logs o archivos Mrtg, para determinar qué vinculo tiene un problema. Un gráfico de MRTG no actualizado o si presenta un icono rojo pestañeando, puede sugerir que un vinculo ha caído.

Se puede diagnosticar el vinculo mas adelante usando un Ping/Traceroute u otra herramienta..

7.4.2 ¿Cómo Interpretar los Gráficos?

Los gráficos llenos de colores en la página web son el resultado generado por el MRTG. Como en sistema grafico elemental, el diagrama tiene ejes verticales y horizontales. Básicamente, el gráfico muestra el tráfico entrante y saliente en cierta interfase con respecto al tiempo.

VERDE ###	Tráfico Entrante en Bits por Segundo
AZUL ###	Tráfico Saliente en Bits por Segundo
VERDE OSCURO ###	Tráfico Entrante máximo en 5 Minutos
MAGENTA ###	Tráfico Saliente máximo en 5 Minutos

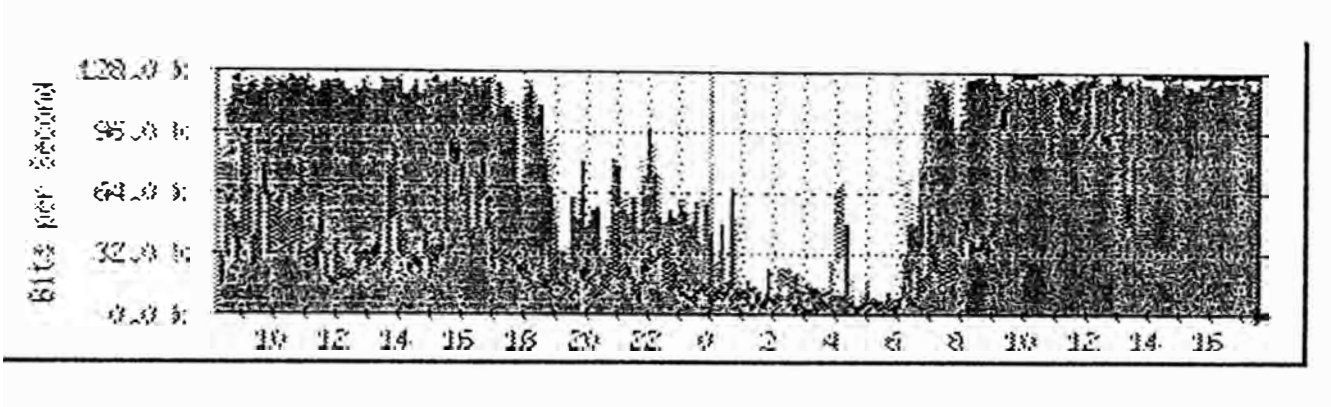


Figura 7.1 Foto instantanea de un gráfico con MRTG.

El gráfico, muestra un ejemplo del tráfico de un vinculo de un serial 1/3 del Asti-Router saliendo a Internet. El gráfico muestra las estadísticas de 24 horas, los valores horizontales son de 0 a 24. Se puede notar que aproximadamente desde las 7 de la mañana

Hasta las 10 de la noche, el tráfico entrante (verde) casi alcanza el máximo de 120 Kbps promedio. Este es el momento en el cual la mayoría de las personas están navegando en la red, bajando archivos, recibiendo e-mails, etc. Pero, desde las 10 el gráfico se cae.

7.4.3 Arquitectura y Componentes Relacionados de MRTG

Aplicación de administración de red que puede supervisar (monitorear) a cualquier Host remoto de la red que tenga habilitado el soporte de protocolo SNMP. MRTG, tal como

una aplicación SNMP, ejecuta peticiones de SNMP contra los Hosts designados sobre una base regular.

Originalmente MRTG fue diseñado para adquirir información de banda ancha relacionada a las interfaces de la red en un Host de la red. Actualmente MRTG puede interrogar a cualquier Host que soporte SNMP OID y construir el gráfico de la variación. Más que eso, las nuevas versiones de MRTG pueden extenderse más allá de las capacidades de SNMP y recopilar información numérica de cualquier Host que colecciona y guarda este tipo de información (ver RRDTOOL, herramienta de bases de datos en Round Robin).

Un diseño de red distribuido de MRTG puede representarse como sigue:

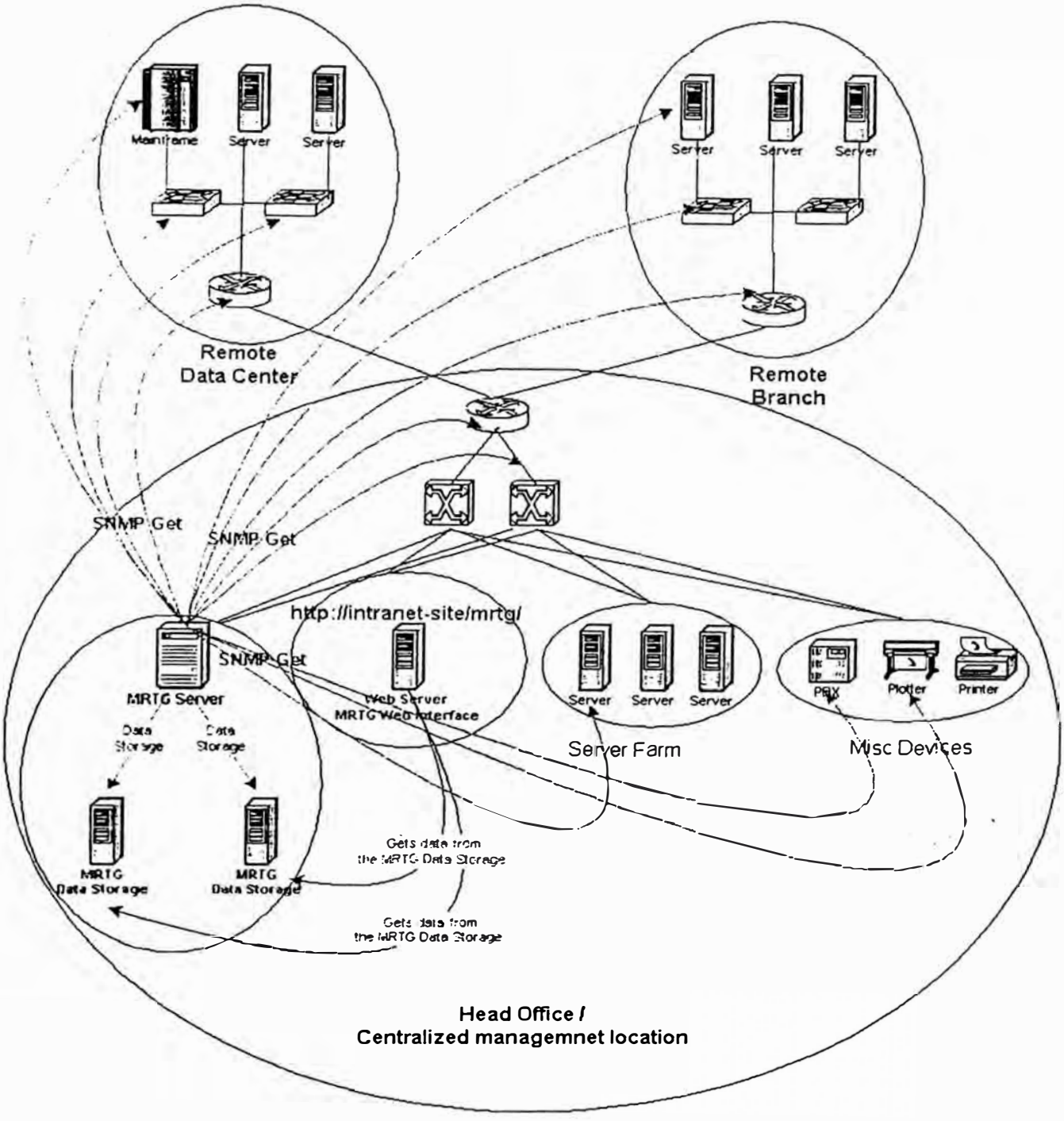


Figura 7.2 Diseño distribuido de una red con MRTG.

MRTG adquiere la información de SNMP que realiza las tareas siguientes:

- Interroga al host remoto y consigue el valor del OID SNMP específico.
- Actualiza el gráfico de la variación con los nuevos valores y borra el gráfico viejo.
- Los gráficos son imágenes en formato PNG. El nuevo gráfico de la variación se guarda en una dirección que puede ser local o remoto en un servidor de almacenamiento dedicado MRTG.
- Guarda el nuevo valor en el archivo log. El archivo log puede colocarse en el host local o remotamente en un servidor de almacenamiento MRTG.

La versión clásica de MRTG construye los gráficos inmediatamente después de que un nuevo valor SNMP es adquirido y no guarda ningún dato histórico para referencias futuras.

La nueva versión viene con un soporte de base de datos muy flexible donde los datos históricos pueden ser guardados según el tamaño de la base de datos que nosotros configuremos. MRTG no genera los gráficos de la variación cuando un nuevo valor SNMP es adquirido, solamente los guarda en la base de datos, haciendo todo el proceso más rápido. Las variaciones gráficas se generan "por-demanda" usando scripts dedicados.

MTRG necesita los siguientes componentes:

- Es necesario que el interprete de Perl esté instalado en la maquina con MRTG, debidos a que el MRTG es una aplicación escrita en Perl. (www.activeperl.com).
- El paquete MRTG. (http://people.ee.ethz.ch/~oetiker/webtools/mrtg/pub).

- Un Servidor Web en la máquina local o en una máquina remota (instalación distribuida) para visualizar los gráficos de MRTG en web/http.

7.5 PROCESO DE INSTALACIÒN

El proceso de instalación se presentará como una implementación no distribuida. Esta sigue las mismas instrucciones pero necesita modificaciones en la especificación de la ruta.

Bajo Sistemas Operativos Windows se deberán seguir los siguientes pasos:

- Según S.O., puede que sea necesario instalar la actualización InstMsiA.exe, 1.670Kb
- Instalar Perl. ActivePerl-5.6.1.633-MSWin32-x86.msi.exe 8Mb
- Instalar MRTG 2.9.25.zip. Debe descomprimirse en el directorio C:\mrtg-2.9.25
- Ejecutar el Servidor Web
- Habilitar soporte para SNMP en el host a monitorear. W95snmpzp.exe, 67Kb

7.6 CONFIGURANDO EL MRTG

7.6.1 Configuración de los Archivos ".cfg"

Es necesario un archivo ".cfg por cada Host a monitorear o también puede utilizarse un archivo ".cfg" global para monitorear todos los Hosts, pero la flexibilidad decaerá.

Los archivos ".cfg" definen los SNMP OIDs de cada entidad que nos proponemos monitorear. MRTG analiza estos archivos y colecciona los valores de todos los SMNP OIDs definidos en el mismo.

Para construir un archivo ".cfg" se debe ejecutar el script "CFGMAKER", este se encuentra en el directorio \mrtg\bin\. Este script escanea las interfaces de red del host y construye el archivo ".cfg". La sintaxis es: perl cfgmaker [options] [community@]router1 [[options] [community@]router2 ...]

7.6.2 Configurar el Archivo "Index.html"

El archivo Index.html se construye basándose en el archivo ".cfg" creado previamente. El rol de este archivo es el de colocar links Html a las interfaces individuales de red, aquellas que están siendo monitoreadas.

Para construirlo se utiliza el Script "INDEXMAKER". La sintaxis es: perl indexmaker [options] [.cfg file].

Finalizando con los pasos previos, el estado del MRTG deberá ser:

- Perl, MRTG y Servidor Web instalados.
- El archivo (.cfg) configurado para monitorear todos los hosts, utilizando el COFIMAKER
- Estos archivos deberán guiar el curso de acción del MRTG en la adquisición del SNMP
- El archivo Index para todos los Hosts, será generado y copiado en la carpeta mage/log/html. Es recomendable colocar como nombre "Index.html" y utilizar una carpeta para cada Host monitoreado con el fin de almacenar sus datos de manera organizada.
- Próximo paso : Ejecutar el proceso del MRTG y visualizar la variación en los gráficos.

7.6.3 Ejecutando las tareas de MRTG

- Ejecutar en la línea de comandos una instancia de MRTG.
- Las tareas que ejecuta el MRTG, se basan en la configuración del archivo ".cfg". Para el monitoreo de cada Host debe crearse una instancia separada del MRTG.
- Las opciones siguientes deben ser agregadas a la configuración del archivo.

```
Workdir: d:\inetpub\wwwroot\
Htmldir: d:\inetpub\wwwroot\ \
Imagedir: d:\inetpub\wwwroot\
Logdir: d:\inetpub\wwwroot\
Icondir: /images/
Refresh: 300  // Periodo de refresco del Browser en
segundos

RunAsDaemon: Yes // Ejecución en Segundo plano

Interval: 5 // Intervalo de ejecución de 5min. Este es el
valor por defecto
```

Ejemplo 7.1 Ejecución de Tareas del MRTG.

Para iniciar una instancia de MRTG, ejecutar el Script Mrtg (se encuentra en mrtg\bin\) con la sintaxis siguiente: Perl Mrtg [config file]

La instancia MRTG seguirá ejecutándose siempre y cuando la ventana de comandos permanezca abierta.

Ejecutar la instancia de MRTG como un servicio en Windows NT/2000.

Ejecutar una instancia MRTG en segundo plano, es posible solo si la instancia MRTG se ejecuta como un servicio. Para registrar y ejecutar una instancia MRTG como un servicio. es necesario una tercera aplicación o servicio de Microdoft "RunAsService".

En un entorno complejo. MRTG puede ser configurado para monitorear múltiples Hosts y cada uno de estos puede tener su propia configuración (carpetas, intervalos de tiempo). Por ello se recomienda ejecutar una instancia separada de MRTG para cada Host o grupo de Hosts. cada uno con su configuración.

Asimismo. un servicio separado deberá ser registrado por cada instancia de MRTG para obtener un alto nivel control.

7.7 SOPORTE DE BASE DE DATOS PARA MRTG

La versión clásica de MRTG no almacenaba ningún historial de los datos SNMP para futuras investigaciones, ni la generación de gráficos. Este solo colecciona información SNMP y construye la variación de los gráficos en el momento. Hay que tipos de gráficos: diarios, semanales, mensuales y anuales.

La base de datos soportada que suministra a MRTG by RRDTOOL (herramienta de bases de datos en Round Robin) viene con capacidades extendibles, las cuales hace de MRTG un análisis más complejo de la red.

La base de datos almacena valores SNMP por largos periodos de tiempo, los cuales se configuran a través de los parámetros de tamaño de la base de datos. Además puede configurarse para almacenar un número específico de registros, permite especificar el tamaño de estos para un determinado periodo de tiempo. Todos estos datos pueden ser exportados como texto plano o en formato XML.

7.8 MONITOREO DE SNMP OID GENÈRICOS CON MRTG

7.8.1 Configuración de los Archivos de MRTG

Las tareas en MRTG son ejecutadas basándose en opciones que presentan los archivos de configuración. Es necesario un archivo de configuración para cada monitoreo, ya sea de manera individual o por grupo. Como mencionamos anteriormente, todas las tareas de MRTG empiezan con lo siguiente: perl mrtg [.cfg file].

Como se puede observar, el elemento clave son los archivos de configuración, estos son un conjunto de Tags opcionales. Los Tags opcionales pueden ser generados en forma automática a través del Script CFGMAKER contra un Host, estos Scripts generan un archivo de configuración, pero solo para los objetos Interface de red.

MRTG tiene la capacidad de monitorear cualquier objeto SNMP de un Host; como por ejemplo espacio en disco, utilización de CPU, etc. La estructura del archivo de configuración consta de dos tipos de Tags opcionales:

- **Opciones Globales :** definen el camino que MRTG maneja para la colección de datos dentro del archivo.

- **Opciones Específicas** *:* definen configuraciones específicas utilizadas para la colección de los datos y la construcción de los gráficos.
- Los Tags opcionales hacen la diferencia entre la configuración Standart del MRTG (IN/OUT de las interfaces de red) y monitorear cualquier objeto SNMP. La sintáxis es la siguiente:

Target [target-generic-name]: OID1&OID2:comunity-name@IP-address , donde « target » es el objetivo a monitorear.

- OID1 y OID2, son las fuentes de datos y corresponden a los valores de originales de Entrada/Salida (el stándart de MRTG funciona con las IN/OUT de las Interfaces de red).

Estos permiten los siguientes valores:

- **OID1 y OID2 son diferentes OIDs** Si necesitamos graficar 2 valores diferentes en el mismo gráfico.
- **OID1 y OID2 son los mismos objetos** Si necesitamos graficar solo uno de los valores en cada gráfico.

La figura siguiente muestra la estructura del archivo de configuración:

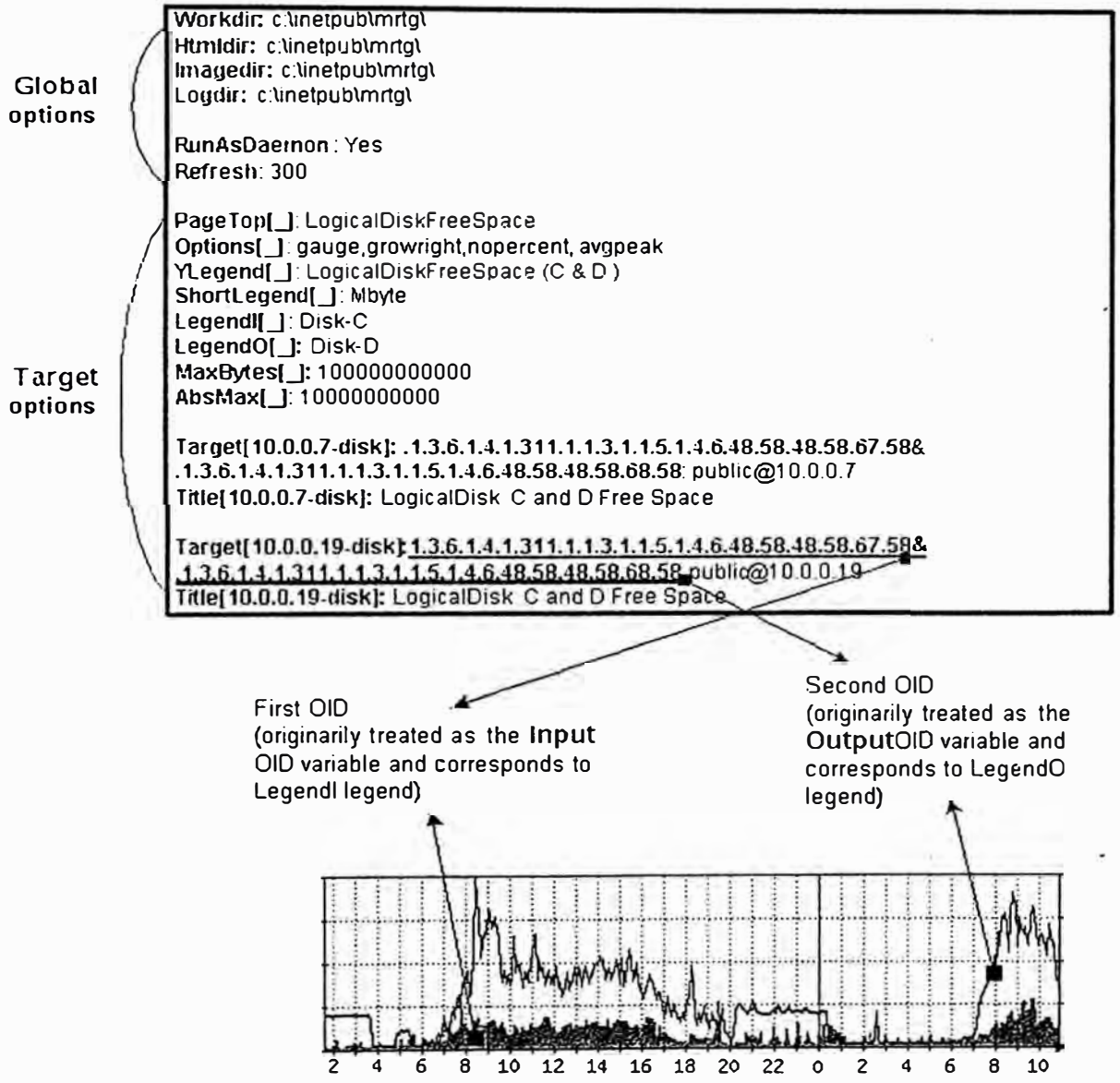


Figura 7.3 La figura siguiente muestra la estructura del archivo de configuración

7.8.2 Opciones de Configuración de archivos

MRTG ofrece un conjunto complejo de Tags opcionales, los cuales son utilizados en la configuración de los archivos. A continuación se listan algunos de ellos:

7.8.3 Opciones Globales

- **HtmlDir** : Especifica el directorio donde las páginas Html se almacenarán. Ej: /www/mrtg.
- **ImageDir** : Especifica el directrio donde se almacenarán las imágenes, este deberá estar debajo del directorio de Html. Ej: /www/mrtg/images.
- **LogDir** : Directorio donde se guardan los archivos Logs.
- **Refresh** : Determina el tiempo en segundos (por defecto es de 300segs.) de refresco o actualización de las páginas.
- **Interval** : Frecuencia en la cual se ejecuta el MRTG, por defecto es de 5 minutos.
- **LoadMIBs** : Busca en los archivos de la MIB para permitir referenciar los objetos a través de su nombre simbólico y no así, a través del OID. Para una mayor eficiencia se mantiene una cache de MIBs en el directorio WorkDir.
- **RunAsDaemon :** habilita la ejecución del MRTG en segundo plano.

7.8.4 Opciones Específicas

- **Title** : Título para la página Html que mostrará los gráficos generados.
- **RouterUptime :** Tiempo de actividad de un router.
- **PageFoot, PageTop, AddHead :** Parámetros para configurar la estructura u organización de las páginas que contendrán los gráficos.

- **MaxBytes1, MaxBytes2 :** Cantidad máxima de Bytes para cada una de las variables del gráfico.
- **Unscaled :** por defecto cada gráfico tiene una escala vertical en cual se muestra la cantidad de Bytes hasta la constante MaxBytes. El parámetro "Unscaled" determina la escala horizontal, d=day w=week m=month y=year
- **Suppress :** por defecto MRTG produce 4 gráficos. Con esta opción podemos suprimir la generación de un determinado gráfico.
- **Extension :** MRTG crea por defecto páginas Html. Esta opción le dice al MRTG optar por una extensión diferente. Ej: Extension[ezwf]: phtml
- **XSize and Ysize :** Por defecto los gráficos son de 100 x 400 pixels
- **XZoom and Yzoom :** Si es necesario agrandar los gráficos.
- **XScale and Yscale :** Si se necesita otra escala que no sea la actual.
- **Noinfo :** Suprime la información sobre el tiempo de actividad y nombres de los dispositivos generados en la página web.
- **Nopercent :** no imprime los porcentajes.
- **Noborder, noarrow, colours :** Configura el aspecto de los gráficos.

7.9 EJEMPLOS DE CONFIGURACIÓN COMUNES

Ejemplo para monitorear el espacio libre en disco (unidades C,D), memoria RAM y utilización de CPU para un determinado Host.

7.9.1 Configuración Para el Monitoreo de Utilización de CPU

```
### Global Config Options

Workdir: c:\inetpub\mrtg\
Htmldir: c:\inetpub\mrtg\
Imagedir: c:\inetpub\mrtg\
Logdir: c:\inetpub\mrtg\
Icondir: /mrtg/images

Refresh: 300
RunAsDaemon: Yes
Interval: 5

### CPU Utilization

Target[10.0.0.254-
cpu]:.1.3.6.1.4.1.311.1.1.3.1.1.33.9.0&.1.3.6.1.4.1.311.1.1.3.1.1.33.9.0:public@10.0.0.254
AbsMax[10.0.0.254-cpu]: 100
MaxBytes[10.0.0.254-cpu]: 100
Title[10.0.0.254-cpu]: CPU Utilization (average)
PageTop[10.0.0.254-cpu]: CPU Utilization
Options[10.0.0.254-cpu]: gauge,growright,nopercent, noo
YLegend[10.0.0.254-cpu]: CPU Utilization
ShortLegend[10.0.0.254-cpu]: %
LegendI[10.0.0.254-cpu]: CPU Utilization (percentage) 
```

Ejemplo 7.2 Configuración para el monitoreo de la carga del CPU.

7.9.2 Configuración Para el Monitoreo de Espacio en el Disco

```
### Free disk C space

Target[10.0.0.254-
diskc]:.1.3.6.1.4.1.311.1.1.3.1.1.5.1.4.6.48.58.48.58.67.58&1.3.6.1.4.1.311.1.1.3.1.1.5.1.4.6.48.58.48'
58.67.58:public@10.0.0.254
AbsMax[10.0.0.254-diskc]: 100000
MaxBytes[10.0.0.254-diskc]: 100000
Title[10.0.0.254-diskc]: DISK FREE SPACE (C)
PageTop[10.0.0.254-diskc]: DISK FREE SPACE (C)
Options[10.0.0.254-diskc]: gauge,growright,nopercent, noo
YLegend[10.0.0.254-diskc]: DISK FREE SPACE (C)
ShortLegend[10.0.0.254-diskc]: MB
LegendI[10.0.0.254-diskc]: DISK FREE SPACE (C) 

### Free disk D space

Target[10.0.0.254-
diskd]:1.3.6.1.4.1.311.1.1.3.1.1.5.1.4.6.48.58.48.58.68.58&1.3.6.1.4.1.311.1.1.3.1.1.5.1.4.6.48.58.48.
58.68.58:public@10.0.0.254
AbsMax[10.0.0.254-diskd]: 100000
MaxBytes[10.0.0.254-diskd]: 100000
Title[10.0.0.254-diskd]: DISK FREE SPACE (D)
PageTop[10.0.0.254-diskd]: DISK FREE SPACE (D)
Options[10.0.0.254-diskd]: gauge,growright,nopercent, noo
YLegend[10.0.0.254-diskd]: DISK FREE SPACE (D)
ShortLegend[10.0.0.254-diskd]: MB
LegendI[10.0.0.254-diskd]: DISK FREE SPACE (D) 
```

Ejemplo 7.3 Configuración Para el Monitoreo de Espacio en el Disco.

7.9.3 Configuración Para el Monitoreo de la Memoria RAM

```
### Free RAM

Target[10.0.0.254-
mem]:.1.3.6.1.4.1.311.1.1.3.1.1.1.1.0&.1.3.6.1.4.1.311.1.1.3.1.1.1.1.0:public@10.0.0.
254
AbsMax[10.0.0.254-mem]: 900000000
MaxBytes[10.0.0.254-mem]: 900000000
Title[10.0.0.254-mem]: FREE RAM
PageTop[10.0.0.254-mem]: FREE RAM
Options[10.0.0.254-mem]: gauge,growright,nopercent, noo
YLegend[10.0.0.254-mem]: FREE RAM
ShortLegend[10.0.0.254-mem]: B
LegendI[10.0.0.254-mem]: FREE RAM&nbs
```

Ejemplo 7.4 Configuración Para el Monitoreo de la Memoria RAM.

CAPÍTULO VIII

APLICACIÓN

MONITORIZACIÓN GRÁFICA DEL TRAFICO DE VOZ Y DATOS DE UNA RED DE VOZ SOBRE IP VIA EL PROTOCOLO SNMP

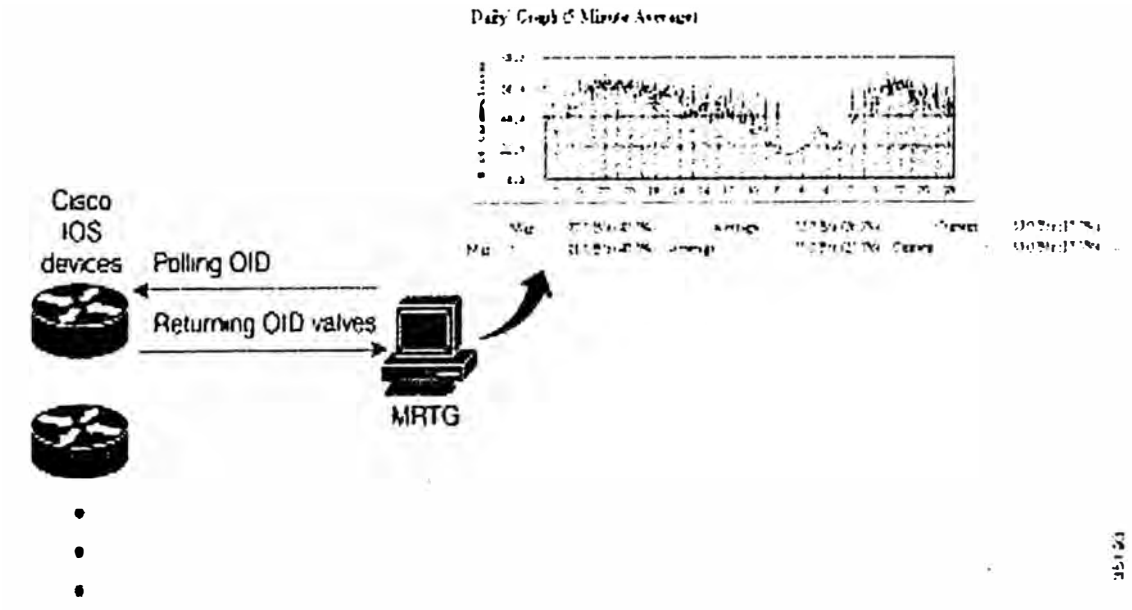


Figura 8.1 Gestion de una red de voz sobre ip, mediante MRTG.

8.1 INTRODUCCIÓN

La presente aplicación fue desarrollada por el graduando, y consiste de un sistema de monitoreo de voz y datos para una plataforma de VOZ sobre IP (VOIP) perteneciente a una de las principales Operadoras de tráfico internacional en el Perú.

Esta aplicación nos muestra en una pagina web la curva de tráfico de las llamadas activas y del trafico IP a través de las interfaces de los Gateways y Routers respectivamente. adicionalmente nos presenta un consolidado de los minutos y causa de liberación de las llamadas ocurrido durante el día, mostrándonos así una visión rápida del estado de la red y la plataforma.

Esta Aplicación se basa en un software de monitoreo MRTG, de distribución gratuita que fue instalado en un servidor con sistema operativo Linux, y de un servidor Web APACHE. también de libre distribución que viene incorporado en las versiones de Linux RedHat.

El servidor Linux hace consulta vía el protocolo SNMP a los equipos de VOIP, para lo cual fue necesario investigar las MIB de los equipos e interpretar los valores que estos arrojan y pasarlos al software MRTG, para que grafique y genere la página Web.

Los gráficos que se generan son similares al siguiente:

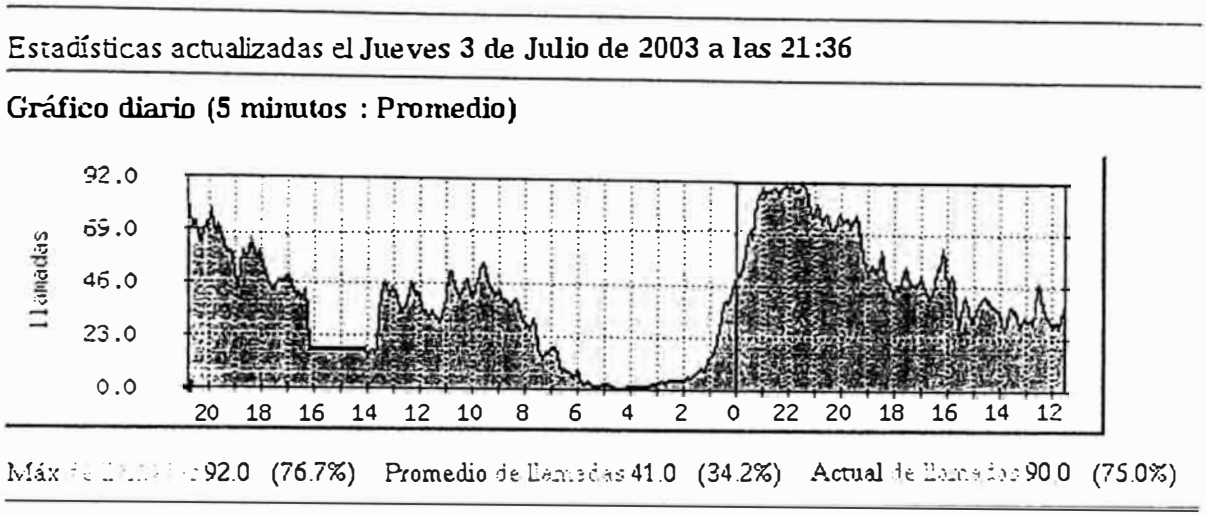


Figura 8.2 Foto instantánea del monitoreo de tráfico en una red VOIP.

Donde se puede observar que el grafico corre de izquierda a derecha. En el eje horizontal se observa la hora con un rango de 24 horas, donde la separación de cada día se muestra con una línea roja. En el eje vertical se observa el número de llamadas activas en el momento de la consulta.

Se puede observar una leyenda en la parte inferior que nos muestra el valor máximo, el promedio y el valor actual de llamadas. Como una referencia nos muestra la hora y fecha de la última consulta SNMP a los equipos.

8.2 REQUERIMIENTOS DE HARDWARE

- PC Pentium Compaq DESKPRO .
- Mínimo 64 MB de Ram.
- Mínimo 5 Giga Bytes de disco duro.
- Tarjeta de Red 10/100 MBPS.

8.3 REQUERIMIENTOS DE SOFTWARE

- Sistema Operativo Linux RedHat. 7.1 o superior.
- Net-Snmp. para consultas SNMP.
- MRTG. para monitoreo de tráfico.
- Servidor Web APACHE.
- Software de desarrollo PERL, de distribución gratuita.

8.4 DESCRIPCIÓN DE LA RED DE VOIP

La Red de VOIP consiste de dos nodos, ubicados en Lince y el Cercado, los cuales se interconectan ha dos centrales internacionales a través de los gateways. Los Gateways hacen la conversión de la voz proveniente de los canales E1 de las centrales Internacionales a datos mediante las tarjetas DSP's. Utilizando la codificación G.729 estandarizado en los protocolos H323.

Los nodos de VOIP tanto del Cercado como de Lince se interconectan a través de un enlace E3 dedicado y cuyo medio de transmisión es fibra óptica.

La Salida internacional es a través de un nodo ATM ubicado en Lince. Este Nodo forma parte del anillo de fibra óptica de la red internacional del grupo Telefonica que interconecta los países de Chile, Brasil, USA, España y Argentina.

La plataforma de VOIP, maneja en su mayoría trafico internacional entrante, proveniente principalmente de los países de EEUU y Argentina, este trafico es recolectado de todo el mundo por las grandes operadoras internacionales de voz sobre Ip, como IBASIS, CTI, ARBINET etc. , y es enviado desde sus respectivos Gateways a

los gateways destinos, en este caso Perú, que se interconecta a las centrales nacionales y termina la llamada en el usuario final, siendo para este completamente transparente ya que la calidad es muy optima.

La plataforma de VOIP, también maneja tráfico internacional saliente, pero en pocas proporciones comparado con el tráfico internacional entrante, esto debido a que por tener ya una capacidad instalada de centrales le es más rentable enviar tráfico de manera tradicional entre centrales internacionales, pero hay casos por ejemplo con las llamadas de móviles que se tienen un margen de ganancia mayor al enviar el tráfico vía IP que de manera tradicional.

8.5 SEÑALIZACIÓN DE LA LLAMADA.

La plataforma de VOIP maneja señalización SS7, mediante el controlador de señalización CiscoSC2200. El Cisco SC2200 es una aplicación software que se ejecuta en plataforma UNIX Sun estándar de manera redundante, este crea un nodo que desde una perspectiva de señalización funciona como un extremo de terminación u origen de señalización SS7/C7. El SC2200 lleva un control de todos los circuitos de la plataforma, ya que tiene registrados todos los gateways de la red, proporcionando funciones de portadora como alarmas, estadísticas y registro de tarificación denominados CDR's, necesarios para la facturación.

El SC2200 procesa los niveles MTP3 y superiores de la arquitectura SS7, este proporciona funcionalidades de ISUP (Integrated Services Digital Network User Part) ,

NUP (National User Part) y TUP (Telephony User Part) además maneja aproximadamente 50 variantes SS7 para 32 países.

Los niveles MTP1 y MTP2 de la arquitectura de señalización son manejados por el terminal de enlace de señalización SLT. El SLT utiliza el router Cisco serie 2600 que utiliza un potente procesador RISC, mediante el protocolo RUDP encapsula el MTP3 y superior y lo envía al SC2200 , El SLT recibe directamente el E1 proveniente de la central y extrae el canal de señalización dejando intacto los canales de voz volviéndolos a dirigir hacia el gateway para su tratamiento.

8.6 INSTALACIÓN DEL NET-SNMP EN UN SERVIDOR LINUX

El primer paso será descargar el software que está disponible en la dirección:

http://www.sourceforge.net/proyects/net-snmp.

Una vez descargado y descomprimido en un directorio, se puede proceder a compilar el código fuente para ello se hace desde la raíz:

```
Linux# configure
Linux# make all
```

Ejemplo 8.1 Definición Código fuente.

Y listo quedará compilado y preparado para instalar. Esta librería no depende de otras. es autocontenida, lo que facilita su compilación Lo cual se podrá hacer con **make install**. Ahora queda poner una configuración adecuada en el fichero **/etc/snmp/snmpd.conf.**

Las versiones actuales de RedHat, incorporan ya el paquete de **Net-Snmp** de forma que su instalación es mucho más sencilla (son binarios ya compilados) y su configuración rápida.

```
Linux # snmpwalk –c PUBLIC 200.81.123.45 .1.3.6.1.4.1.9.1.0.19.1.1.4.0
Enterprises.9.10.19.1.1.4.0 = Gauge32 : 24
```

Ejemplo 8.2 Instalaciòn del NET-SNMP.

8.7 INSTALANDO Y CONFIGURANDO EL MRTG

El proceso de instalación general se explica en detalle en el capítulo referente al MRTG. Explicaremos en forma resumida el proceso de instalación y configuración de manera particular para monitorear las llamadas activas de la plataforma de voz sobre Ip.

Una vez bajado el software se compila el Mrtg.

```
Linux# cd /usr/local
Linux# gunzip -c mrtg-2.9.0pre31.tar.gz | tar xvf -
Linux# cd mrtg-2.9.0pre31

figure
```

Ejemplo 8.3 Instalación y Configuración del MRTG.

8.8 CONFIGURACIÓN

El siguiente paso es ahora configurar el MRTG, para monitorear los dispositivos de la red como los Gateways y Routers. Esto se hace al crear un archivo Mrtg.cfg el cual define los parámetros de configuración propios del dispositivo, para crear el archivo Mrtg.cfg, nos valemos de un script denominado **Cfgmaker**. Este script se apunta al enrutador escogido y él creará un archivo de configuración, este script se encuentra en el directorio Run. El comando se detalla a continuación:

```
Linux# cfgmaker --global 'WorkDir: /home/httpd/mrtg' \
        --global 'Options[_]: bits,growright' \
        --output /home/httpd/mrtg/gw1.cfg   \
         comunidad@200.81.135 .75
```

Ejemplo 8.4 Comando Para Configurar el MRTG

Luego editamos el archivo Gw1.fcg :

```
### Global Config Options##
Workdir: /var/www/html/gateways/gw1/
Htmldir: /var/www/html/gateways/gw1/
Logdir: /var/www/html/mrtg/

RunAsDaemon: Yes
Refresh: 300

### Llamadas activas
PageTop[_]:Llamadas Activas del gateway N°1
Options[_]: gauge, growright,nopercent, avgpeak
Ylegend[_]: Llamadas activas
ShortLegend[_]: Activas

Target[200.81.135.75_1]: .1.3.6.1.4.1.9.10.19.1.1.4.0 &
.1.3.6.1.4.1.9.10.19.1.1.4.0:public@200.81.135.75

Title[200.81.135.75_1]: Llamadas activas
```

Ejemplo 8.5 Definición Para Editar el archivo Gw1.fcg

Después de editar el archivo de configuración Gw1.cfg , creamos la pagina web Index.html, que será la página de inicio para monitorear el Gateway, para lo cual nos valemos del script **Indexmaker** ubicado en el directorio Run.

```
Linux# indexmaker /var/www/html/gateways/gw1/gw1.cfg >
/var/www/html/gateways/gw1/index.html
```

Ejemplo 8.6 Definicion Para Crear La Pàgina Index.html

Una vez creado el archivo de configuración y creado la pagina index.html, se inicia el proceso MRTG. Este proceso corre en forma de demonio, y hace consultas cada 300 segundos y actualizando el grafico en formato Png, listo para publicar en la web.

```
Linux#/usr/local/mrtg-2.9.0pre31/bin/mrtg /home/httpd/mrtg/gw1.cfg
```

Ejemplo 8.7 Definiciòn Inicio Proceso MRTG.

Este procedimiento se aplica a todos los Gateways y Routers de la plataforma que se desea monitorear. Sin embargo para una mejor presentación y más fácil acceso, se genera una pagina web principal con cualquier editor HTML, y se crean los enlaces correspondientes a los equipos.

8.9 PRESENTACIÓN FINAL DE LA APLICACIÓN

8.9.1 Dirección Web

Se puede acceder a la siguiente dirección web pública:

http://200.81.133.23/intal/

User : intal

Password : intal

8.9.2 Snapshots o Fotos Instantáneas

8.9.3 Llamadas Activas en los Gateways

La pagina de inicio muestra las llamadas activas de todos los gateways, además los enlaces para las otras paginas y para mas detalles.

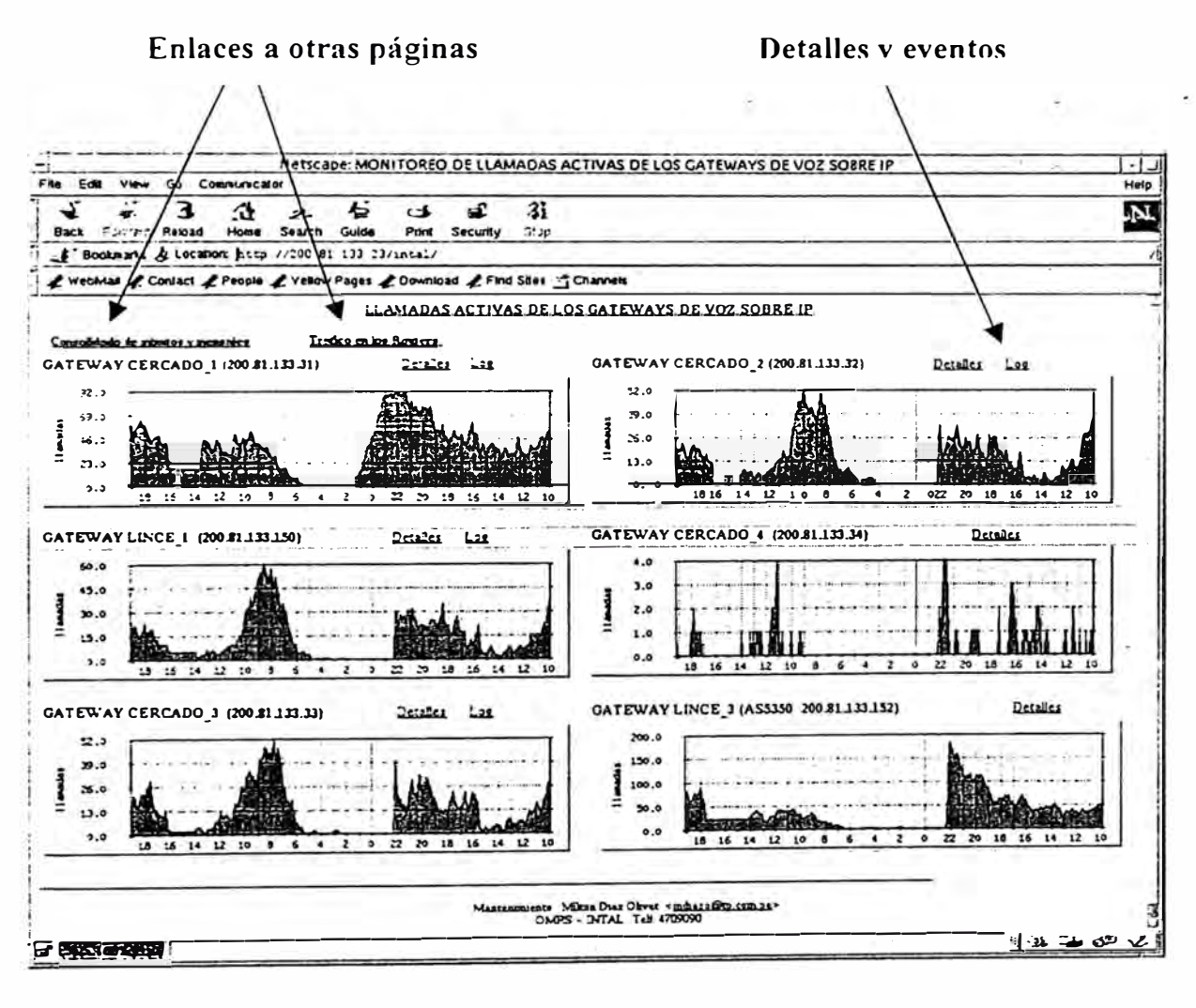


Figura 8.3 Llamadas Activas en los Gateways.

8.9.4 Monitoreo del Tráfico en los Routers

Esta página nos permite visualizar el trafico Ip para todas las interfaces de los routers, tanto para el Nodo de Cercado como para el de Lince.

Las subinterfaces del nodo ATM son graficados en detalle, de tal manera que se puede visualizar el tráfico con cada país independientemente. El color celeste nos representa trafico entrante en bits por segundo y la línea marrón nos representa el tráfico saliente también en bits por segundo. Cada gráfico representa una interface y para visualizar el consolidado por minuto, semanal, mensual y por año se accede al link detalles.

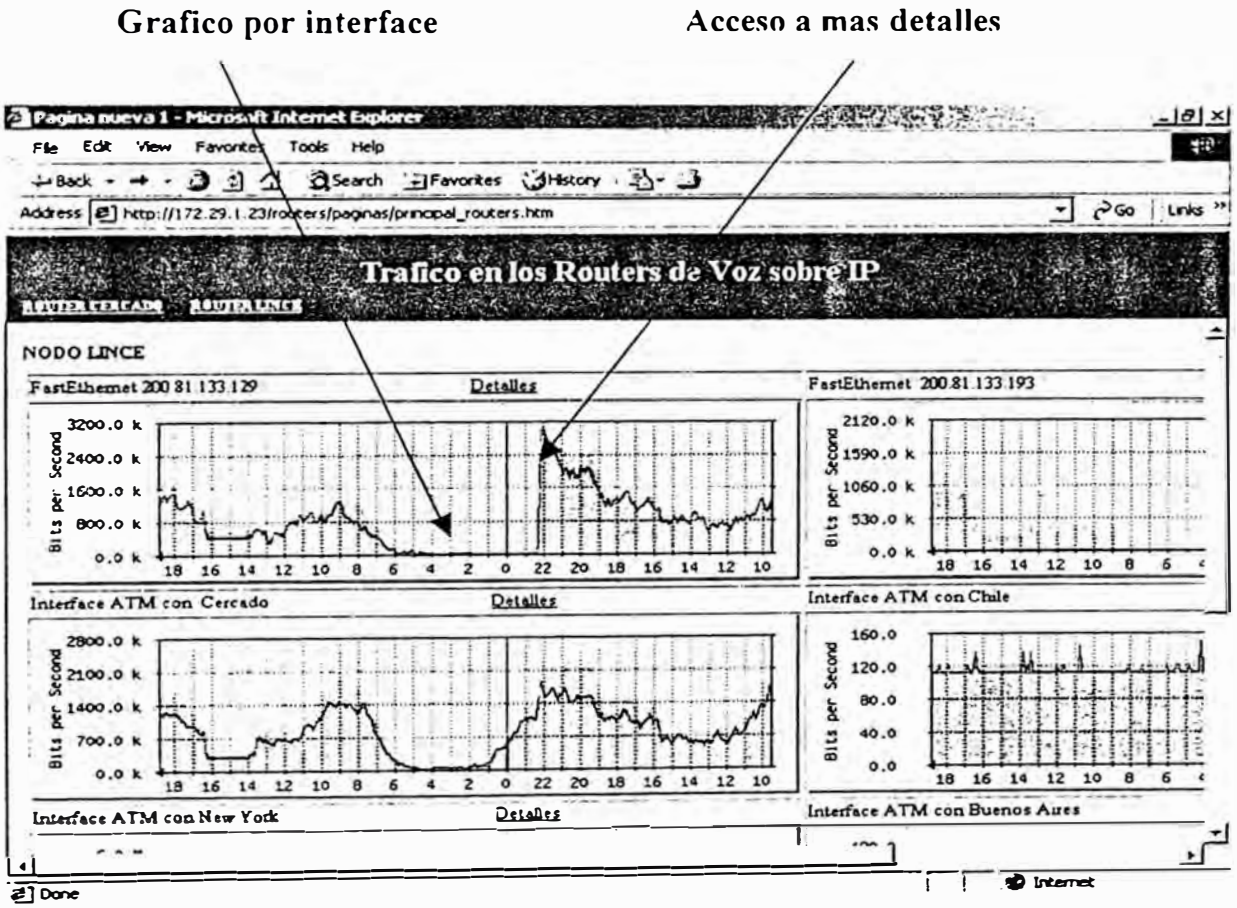


Figura 8.4 Monitoreo del Tráfico en los Routers.

8.9.5 Resumen de Eventos de la Plataforma

Esta página nos muestra un resumen de los eventos de la plataforma de voz sobre Ip. Estos mensajes son recolectados de los archivos da tarificación y son procesados para generar una pagina web que muestra el total de minutos y un resumen de eventos detallado por operador.

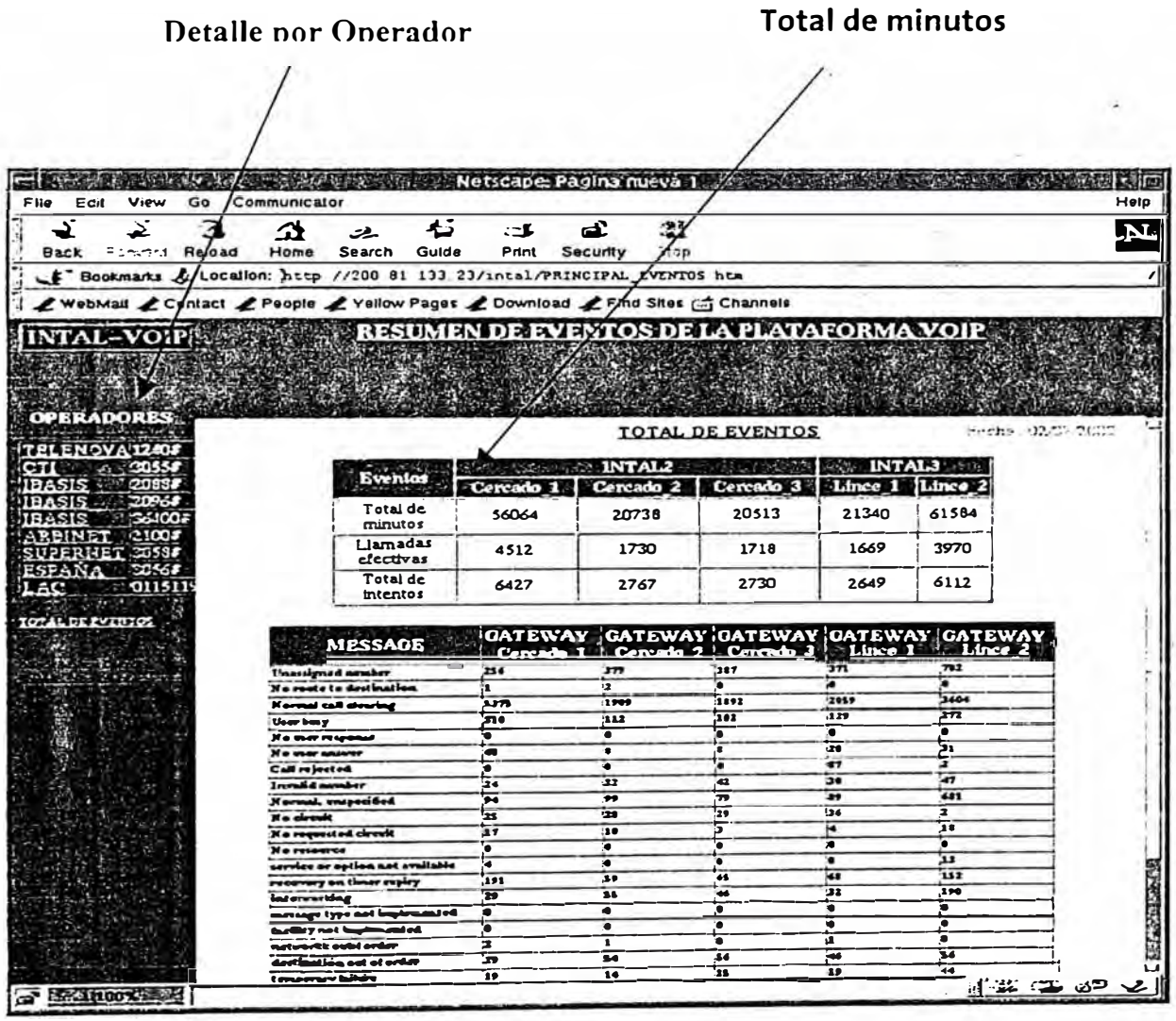


Figura 8.5 Resumen de Eventos de la Plataforma.

8.9.6 Detalle y Consolidado de Llamadas Activas

Esta pagina nos muestra un consolidado de las gráficas por hora, por semana, por mes y por año. Esta página es de gran utilidad porque nos muestra un histórico del perfil de tráfico y de los días y semanas de mayor carga.

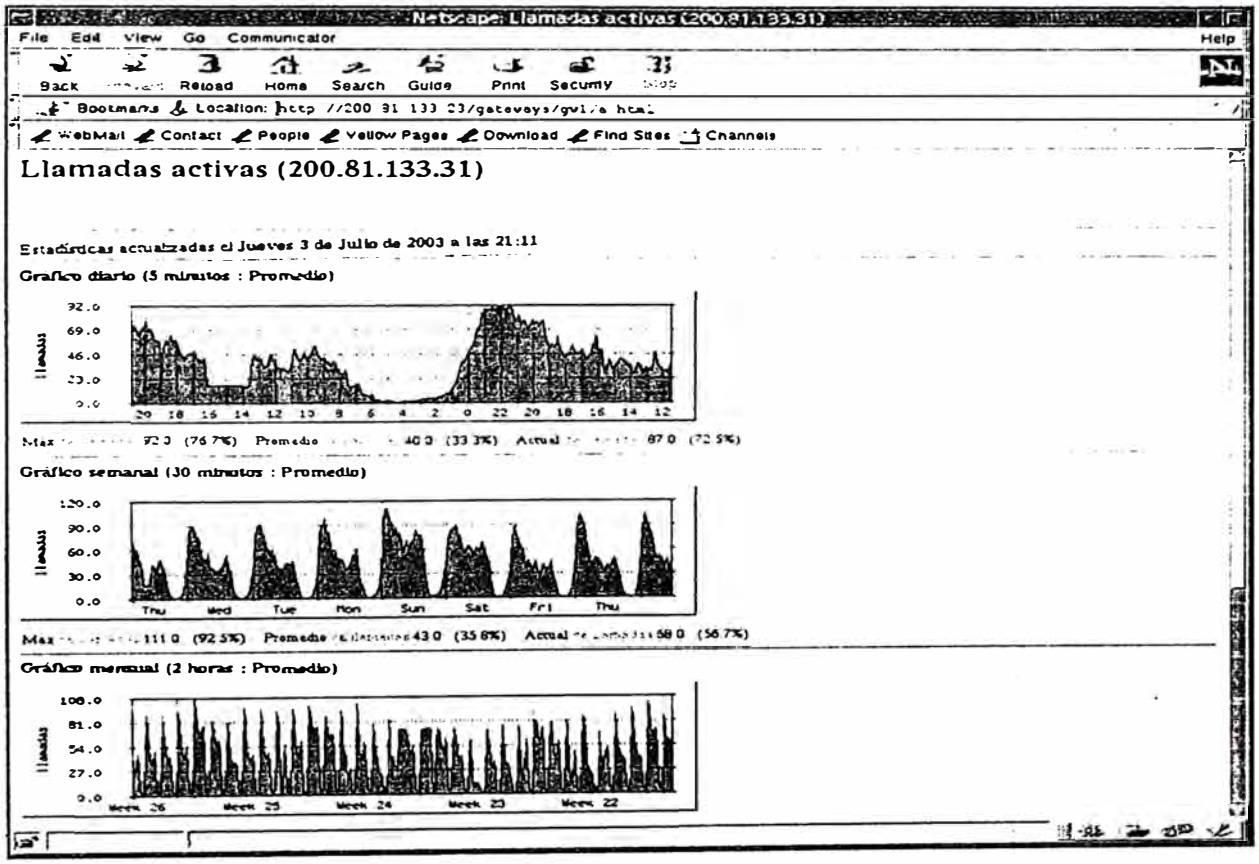


Figura 8.6 Detalle y Consolidado de Llamadas Activas.

8.9.7 Log de Eventos

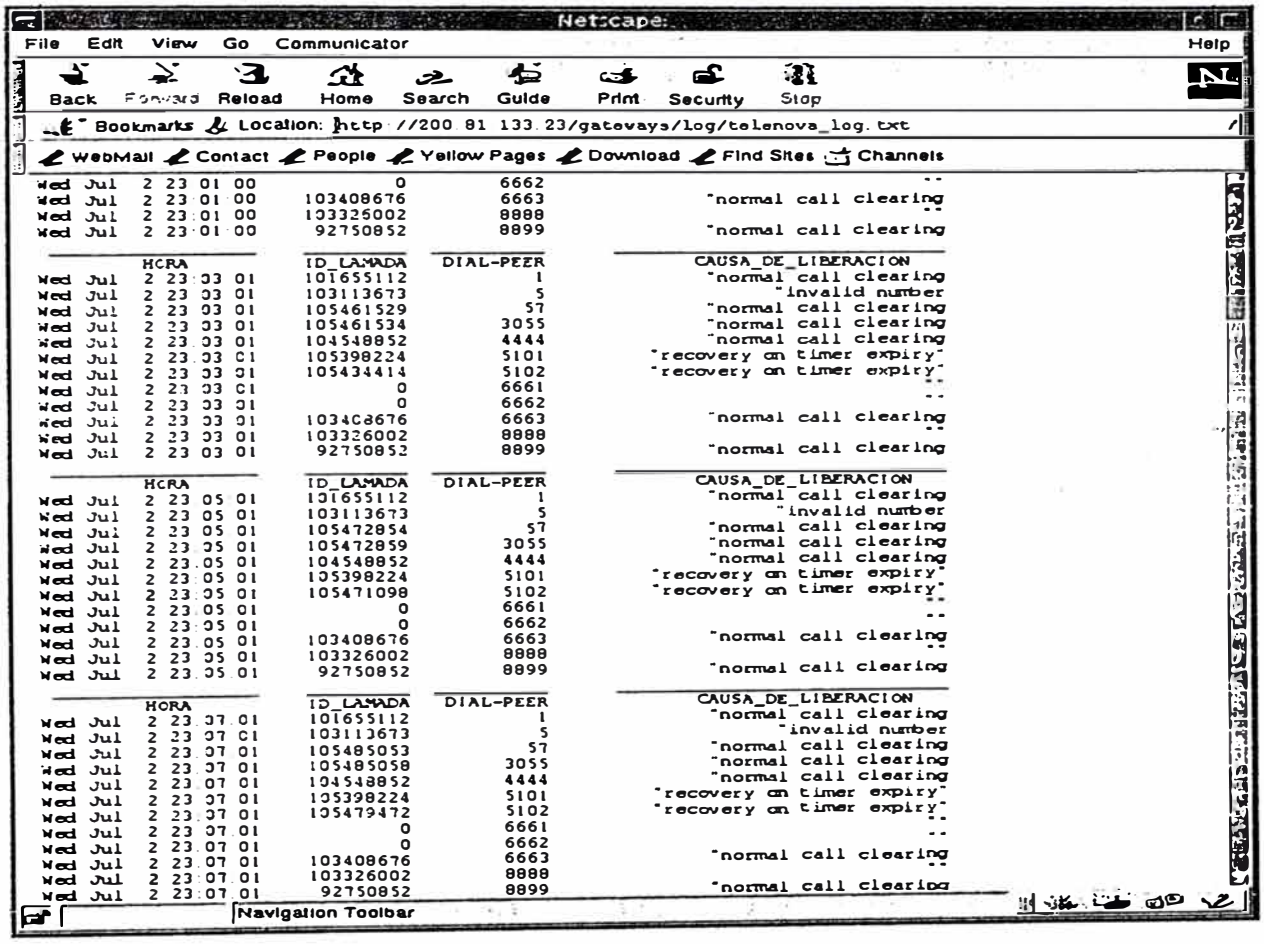


Figura 8.7 Log de eventos de la plataforma de VOIP.

CONCLUSIONES

1. Respecto al futuro de SNMP, la versión de actual SNMPv3 nos proporciona funciones de seguridad con respecto a la original SNMPv1 y esta reorientada a la gestión no sólo de las redes, sino también de los sistemas y aplicaciones que trabajan en ellas. La actual versión incluye, entre otras prestaciones de seguridad, autenticación, autorización, control de accesos y confidencialidad.

2. La ventaja principal de SNMP está en su sencillez. Se puede decir que su lema es el trabajo realizado con un mínimo coste de desarrollo. Esta es la principal característica que a permitido que SNMP se convierta en el protocolo de facto en la red de Internet. Un agente típico de SNMP necesita de 10kb de memoria, mientras que su análogo CMIP de OSI es un protocolo más complejo consume 40kb de memoria.

3. Entre su principal inconveniente se detallo el echo de que al SNMP acusan entre otras cosas de ser demasiado "hablador" y, por tanto, generar mucho

tráfico, remitiendo así a su principio de funcionamiento por "sondas". Este inconveniente puede llegar a traducirse rápidamente en incompatibilidad con amplios entornos de red. Así, pues, el desarrollo de las arquitecturas cliente/servidor y la extensión geográfica de las empresas analizan este tipo de configuración.

4. SNMP es un protocolo del nivel de aplicación que se apoya en los protocolos TCP/IP y el UDP, del nivel de transporte, enlace y red, respectivamente.
   CMIP, sin embargo, descansa sobre el protocolo TP4 de OSI, en el nivel de transporte. Pero existe la posibilidad de que utilice TCP/IP, en cuyo caso el protocolo se llama CMOT (CMIP over TCP/IP). Por lo demás, son similares.
   El protocolo de más bajo nivel TCP/IP es el más utilizado en este tipo de aplicaciones. Su éxito reside, en gran parte, en su madurez y probada eficacia tecnológica para la conexión de múltiples redes. Esta es una de las ventajas de SNMP respecto a otras tecnologías ya que descansa sobre TCP/IP que es un protocolo ampliamente difundido.

5. El presente informe muestra una aplicación basada en el protocolo SNMP, para el monitoreo de una red de voz sobre ip (VOIP). Estas aplicaciones muestran claramente la sencillez con que se puede hacer uso de esta tecnología de gestión.

6. La aplicación práctica es actualmente de gran utilidad en la gestión de la plataforma de voz sobre ip, porque muestra los tráficos de voz (llamadas activas) y de datos en tiempo real los cuales pueden se visualizados desde Internet accediendo a la pagina web pública **http://200.81.133.23/intal/**

7. La página web tiene un link que muestra un consolidados de eventos producidos durante el día como por ejemplo total de minutos, llamadas completadas, ocupadas, causa de liberación de la llamada etc.

8. Esta aplicación practica que se basa en el protocolo SNMP para la gestión remota es constantemente actualizada y adaptada a nuestros requerimientos particulares. Actualmente esta monitoreando los routers y gateways del Pots de Peru, pero en un futuro cercano se piensa monitorear algunos equipos ubicados en otros países.

ANEXO I

RELACIÓN DE FIGURAS TABLAS Y EJEMPLOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE EJEMPLOS