12
2ej

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

DETERMINACION DEL MODELO DE ADMINISTRACION PARA LA RED HETEROGENEA DEL PARTIDO REVOLUCIONARIO INSTITUCIONAL (PRI)

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N :

ALEJANDRO ARAIZA MARTINEZ
GERARDO HERRERA CORONA
RAUL LARA CISNEROS

DIRECTOR DE TESIS. M. en I. LAURO SANTIAGO CRUZ

MÉXICO, D.F. **1996**

TESIS CON
FALLA DE ORIGEN

Alejandro

Para mi esposa Izabel y al hijo(a) que esperamos, por los motivos que me dan para superarme y por el amor que les profeso.

A mi madre Irene, por su apoyo y amor incondicional de toda la vida.

A mi hermana Angelez y mi abuelita Micaela, por el cariño que me han brindado.

A todas las personas que de una u otra forma han intervenido en mi formación profesional, con una mención especial para mi amigo Héctor.

Gerardo

A mi madre Martha Corona, quien siempre me apoyó en mis estudios, que me ha alentado a superarme y que me ha brindado su cariño y apoyo.

A mis hermanos Carlos, René, Alejandro, Sonia y Sandra que de alguna forma me alentaron a terminar.

A mi novia Norma que me apoyó durante el desarrollo de este trabajo.

A todas aquellas personas que de una u otra forma me brindaron su apoyo y me alentaron a continuar.

A mis amigos.

A mi cuñada Ariadna, quien constantemente me instó a terminar.

Raúl

Dedico este trabajo a todos aquellos que contribuyeron a mi formación profesional pero en especial a:

A mis padres Raúl y María Luisa por todo el apoyo y confianza que siempre me han demostrado y por ser unos padres maravillosos.

A mis hermanos Gerardo, Luis y Xochitl, por su paciencia y por haber soportado toda la presión e incomodidades que este trabajo genero demostrando una vez mas su compañerismo y amor.

A todos aquellos que de alguna manera me aportaron su ayuda durante el desarrollo de este trabajo.

A mi Facultad y Universidad por haberme dado tanto.

TESIS

COMPLETA

INDICE

INTRODUCCION

Es innegable el auge que ha experimentado las redes de computadoras, debido a los múltiples beneficios que aporta el tener interconectados equipos de cómputo, tales como compartición de recursos costosos, reducción de información repetida, mayor consistencia de la información, aumento de la potencialidad de los equipos, etc.

En la actualidad las redes crecen para dar servicio a docenas, cientos o incluso miles de usuarios, los cuales requieren compartir información y recursos mediante la interconexión de equipos, protocolos y medios de comunicación distintos, formando de esta manera grandes redes híbridas.

De aquí que la necesidad de tener un control eficiente de la red, es vital. Mantener el sistema funcionando, monitorear, contabilizar y controlar las actividades y recursos en redes grandes, y más aún heterogéneas, representa un gran reto, que sin una herramienta estándar y transparente a los medios y protocolos de la red, sería imposible de resolver para cualquier administrador.

En los últimos años han surgido propuestas de estándares de tres organizaciones internacionales para resolver el problema de la administración de redes, dichos organismos y sus respectivas propuestas son los siguientes:

- ISO: CMISE y CMIP
- INTERNET: SNMP y CMOT.
- IEEE: CMOL

En este trabajo se hace un estudio de estos estándares a lo largo de sus capítulos tres, cuatro y cinco respectivamente, para finalmente, elegir el producto en el mercado basado en el modelo de administración más viable y capaz de analizar y monitorear el comportamiento del flujo de información y de los elementos que componen a la red heterogénea de una Institución Política.

Es importante resaltar la importancia que tiene una administración confiable en la red, que le permita a esta Institución mantener segura y consistente su información.

Para el logro de esta meta, el presente trabajo se ha estructurado en seis capítulos. En su capítulo inicial se abordan tópicos que son básicos en redes de computadoras y que servirán para la mejor comprensión de los temas posteriores. En su segundo capítulo se trata un tema medular para el desarrollo de los temas subsecuentes que forman el cuerpo principal de este trabajo: la clasificación y breve estudio de los protocolos de redes más representativos en la actualidad. En los capítulos tres, cuatro y cinco se desarrolla un análisis de los tres estándares de administración de red.

Finalmente, en el capítulo seis, basándose en los conocimientos expuestos en los capítulos anteriores, se hace un estudio y selección del producto de administración de red más viable para la red de la Institución Política.

CAPITULO

1

CONCEPTOS GENERALES

1. CONCEPTOS GENERALES

Este capítulo tiene como propósito fundamental, proporcionar algunos de los elementos que consideramos como básicos, para poder visualizar y entender más fácilmente todo lo que involucra la administración de una red de computadoras.

1.1 CONCEPTOS BASICOS

1.1.1 Redes LAN, MAN y WAN

Una red de computadoras no es más que un conjunto de equipos conectados entre sí y que tiene como finalidad la compartición de recursos (físicos y lógicos).

Aun cuando las redes se pueden clasificar de diferentes maneras, en este apartado las clasificaremos por su extensión geográfica.

Redes de Area Local (*Local Area Network*, LAN)

Este tipo de redes se ha hecho muy común en nuestro país y en el resto del mundo, debido en buena parte, a la proliferación de computadoras personales y al abaratamiento del *hardware* en general. Una de las principales propiedades de una red LAN es que está limitada en cuanto a extensión geográfica, pero no siendo la única característica, pasaremos a explicar brevemete las demas :

- Su cobertura es de algunos metros a cinco kilómetros, siendo esto variable.
- Alta velocidad de transmisión, la cual fluctúa en la actualidad entre 1 y 100 Mbps, aunque las nuevas tecnologías de redes públicas pueden alcanzar estas velocidades.
- El canal de la red suele ser privado, es decir, propiedad de la institución dueña de la red.

Redes de Area Metropolitana (*Metropolitan Area Network*, MAN)

Las redes de Area Metropolitana, son las redes a nivel ciudad y dentro de las cuales pueden estar incluidas varias redes LAN, formando una gran red.

Redes de Area Extendida (*Wide Area Network*, WAN)

Las redes WAN son las redes con más historia y su cobertura es prácticamente ilimitada, usan principalmente como medio de transmisión las líneas telefónicas o cualquier otro medio público. Como un ejemplo de esto podemos mencionar a la red de la comunidad Internet.

1.1.2. Arquitectura *peer-to-peer* y Cliente-Servidor

En el mundo de los redes se han desarrollado dos tipos de arquitectura: la igual a igual (*peer-to-peer*) y la cliente-servidor. A continuación se explicará brevemente su principio de operación.

- Arquitectura *peer-to-peer*: Esta arquitectura está basada en la filosofía de que cada computadora que forma parte de la red tiene la misma autoridad que cualquier otra para tener acceso al canal de comunicación, debido a esta característica es que recibe este nombre.

- Arquitectura Cliente-Servidor: Esta arquitectura está basada en una computadora que se encarga de mantener el control de la red y atender las peticiones hechas por las demás computadoras de la red, a esta computadora se le conoce como Servidor o *server* y a las que hacen peticiones se les denomina Clientes, el cual sitúa parte del procesamiento en las estaciones de trabajo, distribuyendo así, el trabajo en varios nodos de la red. A este mismo concepto en telecomunicaciones se le conoce como maestro-esclavo o primario-secundario.

  Esta arquitectura es de las más usadas comercialmente y por la misma razón cuenta con una gran cantidad de aplicaciones , sobre todo en bases de datos.

1.1.3 Topologías

Empecemos por definir el término topología, el cual se refiere a la forma que toma un conjunto de objetos al ser agrupados. Transportando esta idea al ámbito de las comunicaciones y sobre todo al de las redes de computadoras, diremos que la topología de una red se refiere a la forma de conectar los elementos que la conforman, de tal manera que se distinguen cinco tipos de topologías básicas, siendo éstas enunciadas y explicadas en las siguientes líneas:

- Topología de *Bus*: En esta topología el medio de comunicación es único y es compartido por todos los nodos de la red, por lo que si falla éste falla toda la red. Este tipo de topología se ilustra en la Figura 1.1.3.a.

- Topología de Anillo: Se le denomina de anillo por que la ruta que sigue la comunicación es un círculo, aúnque físicamente en algunas ocasiones no lo parezca. En esta topología cada computadora recibe el mensaje y verifica si la dirección que trae corresponde a la suya, de ser así lo toma, en caso contrario procederá a retransmitirlo y así será hasta que finalmente éste llegue a su destino, ver Figura 1.1.3.b.

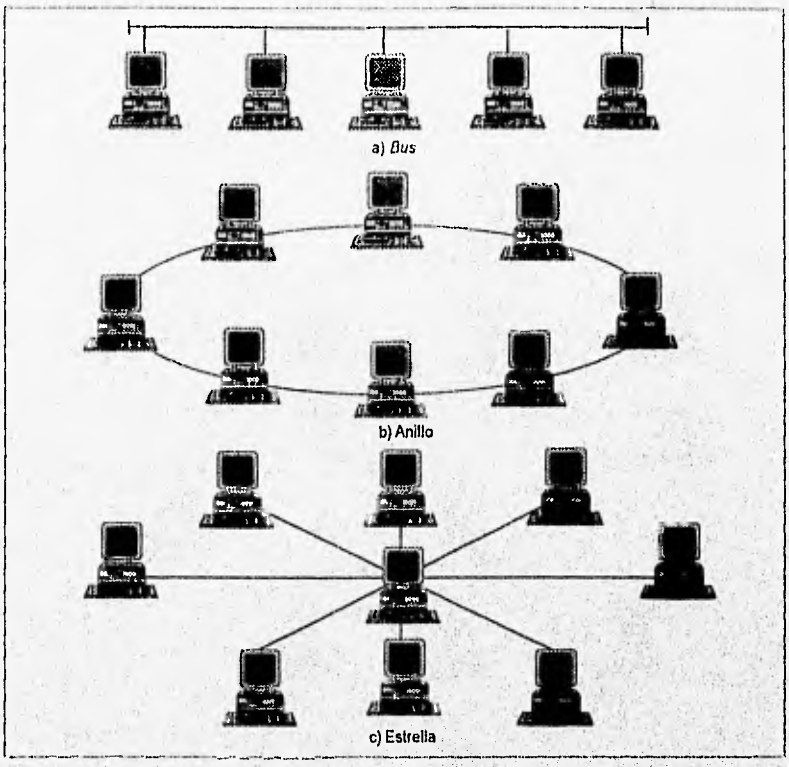


Figura 1.1.3 Topologías de redes

- Topología de Estrella: La topología de estrella se caracteriza por contar con un nodo central, el cual es el encargado de controlar la comunicación entre todos los nodos de la red, como se puede ver en la Figura 1.1.3.c. Si este nodo central falla la red entera fallará.

- Topología de Arbol: La topología de árbol está fundamentada en la jerarquía que tienen los nodos de la red, los cuales forman un árbol, donde los nodos que están en la posición más alta tienen también una jerarquía superior, siendo el de mayor autoridad el nodo que se encuentre en la parte superior del árbol. Este nodo es el encargado de mantener el control de la red, por lo que si éste falla toda la red fallará, esta topología se ilustra en la Figura 1.1.3.d.

- Topología de Malla: A este tipo de conexión también se le conoce como conexión de todos contra todos, pues cada nodo mantiene comunicación con todos, por lo que las rutas son muy variadas y el tráfico en la red muy complejo. Este tipo de topología se le conoce como red distribuida, en donde todos los nodos comparten recursos. Si una computadora falla, el sistema continuará funcionando, ver Figura 1.1.3.e.

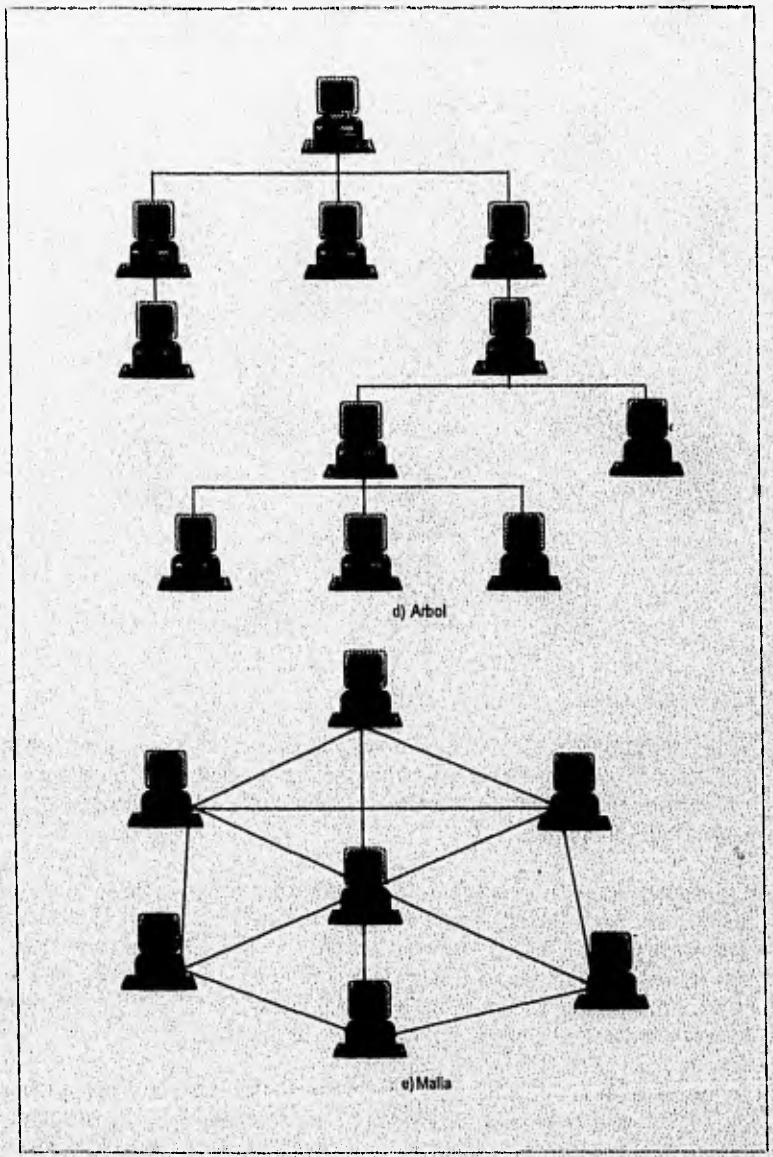


Fig. 1.1.3 Topologías de redes

1.1.4 Organismos de normalización

Normalizar es crear un patrón al cual todos los fabricantes se deberán ajustar, lo cual no es una tarea fácil y es la razón por la que surgen organismos internacionales que tienen como principal función unificar criterios a través de normas que se establecen con las opiniones de los interesados. Dentro de estos organismos podemos citar a los más destacados o que han tenido mayor influencia en el desarrollo de los estándares en el área de las comunicaciones.

- Comité Consultivo Internacional de Telefonía y Telegrafía (*Comite Consultatif Internacionale Telegraphique et Telephonique*, CCITT): Este comité es miembro de la Unión Internacional de Telecomunicaciones, el cual pertenece a las Naciones Unidas. Dentro de este organismo juegan un papel muy importante las principales compañías telefónicas del mundo.

- Organización Internacional de Normalización (*International Standard Organization*, ISO): Este organismo está formado por todos los organismos normalizadores, dentro de los cuales influyen los comités de usuario y por supuesto los fabricantes.

- El Instituto Americano Nacional de Estándares (*American National Standard Institute*, ANSI): Esta es la principal organización normalizadora de los Estados Unidos y tiene representación en ISO.

- El Instituto de Ingenieros Eléctricos y Electrónicos (*Institute of Electrical and Electronics Engineers*, IEEE): Este es un instituto de profesionales de la ingeniería, en cuyo seno no sólo acoge a ingenieros sino a profesionales que se encuentren relacionados con las áreas de la electricidad, electrónica, computación y telecomunicaciones. Tiene tal importancia que los estándares establecidos por el IEEE son tomados por el ANSI como propios.

- La Comunidad Internet: Esta comunidad no es propiamente un organismo internacional de normas, más sin embargo ha desarrollado estándares muy importantes con respecto a la administración de redes, los cuales serán examinados posteriormente a lo largo de este texto.

1.1.5 Estándares en redes LAN

Como hemos podido ver, la necesidad de establecer estándares en la industria es muy importante, por tal motivo este punto lo dedicaremos a hablar de tres estándares que en la actualidad son muy populares dentro del ambiente de redes locales, por supuesto estamos hablando de Ethernet, Arcnet y Token Ring.

- Ethernet: Este estándar fue desarrollado por Xerox Co., tomó su nombre por analogía con el concepto de "eter", definido por los antiguos griegos como la sustancia que llenaba a todo el universo y por lo tal estaba en contacto con todas las cosas. Ethernet fue usado primeramente con fines académicos en el proyecto Aloha en Hawaii.

Ethernet trabaja con el protocolo CSMA/CD y bajo la topología de *bus*, las velocidades típicas actuales de transmisión para este método son de 10 Mbits/seg y se encuentra regido por la norma del IEEE 802.3.

- Ethernet II: Ethernet II es una modificación del Ethernet original, el cual tiene como característica fundamental la capacidad de conectarse no sólo en topología de *bus* sino que también en topología de anillo.

- Arcnet: Este estándar fue desarrollado por Datapoint Co., en 1982 fue hecho público el protocolo que emplea y que se denomina *token passing*, funciona bajo una topología de *bus* y se rige bajo la norma 802.4 de IEEE, siendo las velocidades típicas actuales de 2.5 MBits/seg.

- Token Ring: Creado por IBM, lógicamente tiene una alta conectividad con equipos IBM, usa como protocolo el *token passing* y tiene un buen desempeño, como desventaja su cableado es complejo y su velocidad de transmisión típica actual es de 4/16 Mbits/seg.

1.1.6 Elementos de conectividad

Conectividad es la capacidad de conexión de una red a otra para dar paso a una inter-red, para conseguir esto existen varios dispositivos los cuales veremos a continuación:

- *Gateways*: Un *gateway* es un dispositivo que tiene como función comunicar dos redes que manejan protocolos diferentes, en pocas palabras, un *gateway* lo que hace es traducir de un protocolo a otro.

- Puentes (*Bridges*): La finalidad de un puente es comunicar dos redes, pero a diferencia del *gateway*, éste opera sobre los niveles inferiores del modelo de referencia OSI.

- Repetidores: Un repetidor tiene como propósito fundamental el recibir la señal y amplificarla para que llegue a su destino, se usa para dar mayor cobertura a la red, es decir el uso de un repetidor nos permite alcanzar distancias mayores. Aunque su uso está limitado, ya que si usamos varios de estos dispositivos indiscriminadamente, el ruido puede ser grande, pues al aumentar la señal aumenta el ruido que es amplificado también. Dentro de los repetidores encontramos los repetidores pasivos y activos.

- Ruteadores (*Routers*): Un ruteador es un dispositivo cuya finalidad es la de encontrar o seleccionar una ruta adecuada para conectar o comunicar dos nodos. Actualmente existen dos tipos, el de nivel 1 que rutea los datos dentro de una sola red y el de nivel 2 que lo hace entre redes.

- Puente-Ruteador (*Brouter*): El *Brouter* es un dispositivo combinado pues realiza las funciones de un puente pero selecciona la ruta para comunicar dos nodos en redes diferentes.

Este dispositivo se encarga de "puentear" algunos protocolos, dejándolos pasar solamente, mientras que a otros (los que lo requieren), los traduce y los encamina a su destino.

I.2 SISTEMAS OPERATIVOS DE RED

Un Sistema Operativo de Red (*Network Operating System*, NOS) es un conjunto de programas que se encargan de controlar los niveles de seguridad en la red, proporcionar los elementos para la interface con el usuario, controlar la compartición de recursos y proporcionar la comunicación entre los diferentes usuarios de la red.

Cabe aclarar que en este punto nos ocuparemos únicamente de los Sistemas Operativos para LANs.

De la definición anterior podemos desglosar las funciones más importantes de un Sistema Operativo de Red de la siguiente manera:

- Compartir recursos: Tales como discos duros con su información (datos y programas) con capacidad de bloqueo de registros y archivos en bases de datos, impresoras, graficadores y otros dispositivos.

- Niveles de seguridad: Control de límites de acceso, con manejo de jerarquías y asignación de derechos (*Read Only, Delete,* etc.) basados en *logines* y *passwords.*

- Otras facilidades: Tales como correo eléctronico de datos y envío de mensajes.

A lo largo de la evolución de los Sistemas Operativos de Red, éstos han utilizado distintos métodos para la compartición de datos y programas almacenados en el disco duro de la red, llámese a esta compartición leer, escribir, borrar o crear. Estos métodos son los siguientes:

- Como Servidor de Discos: Hasta hace algunos años el Sistema Operativo de Red sólo hacía que el usuario de la red "viera" al disco duro de la red como un *drive* local más.

- Como Servidor de Archivos. Hoy en día los Sistemas Operativos de Red poseen un *software* especializado que se encarga de administrar el acceso a los archivos del disco duro de la red, haciendo que el control de acceso sea centralizado por el *server* y su utilización sea multiusuario. De esta manera se garantiza la integridad de los datos de la red, tal como lo hacen las minicomputadoras y los *mainframes.*

- Como Servidor de Base de Datos: Esta es una forma más reciente de comportamiento, que a diferencia del anterior, el *software* que controla el acceso

a los archivos es aún más especializado, siendo capaz de proporcionar el servicio de un Sistema de Administración de Base de Datos (*Data Base Management System*, DBMS).

Con los avances en el desarrollo de componentes electrónicos, el crecimiento del mercado de redes de computadoras y el crecimiento de las redes corporativas, los Sistemas Operativos de Red actuales, en su proceso de evolución, presentan las siguientes características:

- Interconexión de redes: La interconexión de redes es la capacidad de unir diferentes equipos de *hardware* de red, para formar una red que sea transparente, una "inter-red".

  Debido a la gran variedad de equipos de *hardware*, provistos por múltiples vendedores, ha surgido la necesidad de que las redes con distintas características puedan interconectarse en una "inter-red". Lo anterior hace importante la capacidad de interconectar sistemas diferentes.

  La clave para la interconexión está en la independencia del *hardware* de los Sistemas Operativos de Red, lo cual ayuda a evitar la obsolecencia del *hardware*.

- Modo protegido de operación (Memoria Virtual): Esta característica está cubierta actualmente por casi todos los sistemas al contar con microprocesadores 80286 o superiores, con ello el sistema es capaz de direccionar una cantidad de memoria mucho mayor; lo anterior permite mejorar significativamente la eficiencia del servidor, por ejemplo NetWare ya desde su versión 3.11 aprovecha esta característica.

- Integridad de datos: A medida que las redes crecen en demanda, el punto de la integridad de datos se hace más crítico, pues la red debe incluir protección contra fallas de sistema, que causan pérdida de datos y tiempo. Las protecciones que un Sistema Operativo de Red debe tomar en cuenta son: contra fallas del sistema, fallas del medio magnético y corrupción de datos.

En suma, el Sistema Operativo de Red es considerado el componente más importante de una red.

De los Sistemas Operativos de Red más importantes en el mercado actual podemos mencionar:

- LANManager
- NetWare
- LANtastic
- 3+Open
- Apple-Share

- Vines
- Windows NT
- UNIX

1.3 ADMINISTRACION DE UNA RED

1.3.1 Definición de la administración de una red

Diversas organizaciones tienen distintas opiniones acerca de la administración de red y por ello emplean diferentes definiciones del término. Quizá lo conveniente sea tomarla de una definición académica.

Se acepta hoy en día que la administración involucra la planeación, organización, monitoreo, contabilidad y el control de actividades y recursos. Esta definición puede ciertamente ser aplicada a la administración de red, sin embargo, las estructuras de la administración de una red propuestas por OSI e Internet se enfocan principalmente al monitoreo, la contabilidad y el control de las actividades y recursos de la red. Los otros dos aspectos de la administración de red, la planeación y organización, no son contemplados en el esquema OSI/Internet, ya que se refieren más bien a una etapa previa a la de operación. Es claro que si la red no está planeada y organizada adecuadamente, ningún monitoreo, contabilidad y control es factible.

1.3.2 Importancia de la administración de una red

Las redes proporcionan verdaderos beneficios al permitir un acceso casi instantáneo a los datos (generalmente de una importancia crítica en una organización) o compartir recursos de cómputo, sin importar el tiempo o el espacio. Cabe señalar que la fragilidad de este proceso es grande y que eventos aparentemente triviales no relacionados entre sí, pueden hacer que la red se desplome o que el funcionamiento degenere en forma drástica.

De ahí que la necesidad de tener un control eficiente de la red es vital. Mantener el sistema funcionando, monitorear, contabilizar y controlar las actividades y recursos en redes heterogéneas representa un gran reto, que sin una herramienta estándar y transparente a los medios y protocolos de la red sería prácticamente imposible de resolver para cualquier administrador.

1.3.3 Tres estándares para la administración: Internet, Modelo OSI e IEEE

Muchas compañías operan en un ambiente heterogéneo y utilizan una gran variedad de componentes de *hardware*, así como de medios y protocolos de comunicación. Las organizaciones se enfrentan con un gran problema técnico en la construcción de la interconexión de sus equipos provenientes de distintos fabricantes. A este cuadro hay que agregar mayor complejidad cuando las redes crecen en voz y datos dentro y entre organizaciones.

En los años recientes han surgido propuestas de estándares, fundamentalmente de tres organizaciones internacionales para resolver el problema de administración de redes, dichos organismos y sus propuestas son las siguientes:

- ISO: CMISE y CMIT.
- INTERNET: SNMP y CMOT.
- IEEE: CMOL

Por una parte, los estándares ISO están siendo usados crecientemente en la industria de las telecomunicaciones, sin embargo, los protocolos de Internet son los más empleados actualmente en la industria para definir sistemas de comunicación híbridos. Los esfuerzos de IEEE se concentran en la forma de estandarizar los sistemas LAN y MAN. Estos estándares no solamente permiten interfaces entre diferentes computadoras, terminales, multiplexores, PBXs, etc. , sino también dan al usuario de la red más fexibilidad en la selección de equipo y *software,* ya que definen interfaces y protocolos estándar entre diferentes equipos. De esta manera, el comprador potencial tiene menos problemas en el momento de adquirir componentes, en el caso de que el equipo de un proveedor podría no interconectarse con el componente de otro vendedor.

De igual importancia es el hecho que estos estándares permiten una plataforma común para el desarrollo y soporte de programas de administración de red. Esta característica simplifica las interfaces entre un control centralizado y los recursos de la administración de red.

Por otra parte, el estándar de administración de OSI está basado en las técnicas del Diseño Orientado a Objetos (*Object Oriented Design*, OOD). El OOD es una herramienta excelente para el manejo de dispositivos de multivendedores en una ambiente de un control centralizado de la red, que da al supervisor considerable flexibilidad a la hora de esquematizar todos los componentes de una red.

La aceptación y el uso de un estándar normalmente nos lleva al abatimiento de costos, ya que permite la producción en serie de diferentes elementos -incluso chips VLSI- que contribuyan a controlar una red. Este hecho deja en libertad a la industria para usar este recurso como una plataforma en el diseño e implementación de Procesos de Valor Agregado (*Value-Added Processes*, VAPs).

1.3.4 Consideraciones en la administración de una red

Para fundamentar la administración de red basada en los estándares OSI/Internet/IEEE, es importante tomar en cuenta las siguientes consideraciones:

- La administración de una red es necesaria para reducir el costo de la interconexión de los diferentes sistemas.

- Se requiere uniformidad en el intercambio de información entre los equipos provenientes de multivendedores.

- Tener una administración integral en donde se defina el mismo nivel de servicio mediante un protocolo de comunicación común.

- La integración y uniformidad no imposibilitan los procesos de valor agregado (VAPs) dentro del contexto de estos estándares.

1.3.5 Administración integral de una red

Como se mencionó en el punto anterior, una meta importante de la administración de una red heterogénea es dar soporte a la administración integral de la misma. Esto no significa que los servicios sean los mismos para todos los usuarios, sino que sea el mismo nivel de servicio de administración para las definiciones de rendimiento en una red: contabilidad, configuración, seguridad, criterio de fallas y el intercambio de unidades de datos mediante un protocolo común.

Para ilustrar como se puede alcanzar esta meta considere el ejemplo de la Figura 1.3.a., en donde el control de una red se encuentra monitoreando dos diferentes *gateways*. El *gateway* A pertenece a un fabricante y el B a otro, además tampoco utilizan un protocolo común. Los mensajes, por ejemplo, mensajes de alarma, se mandan al control de la red por cada *gateway*, para interpretar apropiadamente estos mensajes el controlador de la red deberá:

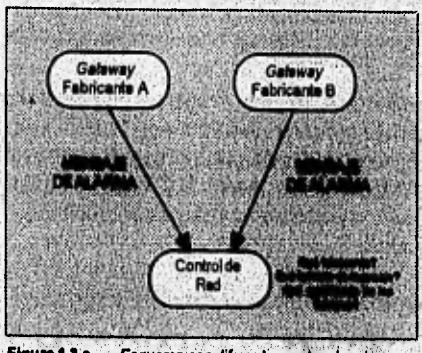


Figura 1.3.a Esquema con diferentes protocolos de distintos fabricantes.

- Saber de que máquina provienen los mensajes de alarma.

- Después de determinar lo anterior, el control deberá ejecutar la rutina de *software* correspondiente para decodificar los bits y campos del mensaje, ya que para cada máquina existe un formato propietario.

- Después de que los bits y campos hayan sido decodificados, otra rutina específica de cada fabricante deberá ser ejecutada para determinar el significado de los campos del mensaje.

Los campos del mensaje pueden ser los mismos o pueden no serlo. Por ejemplo, el mensaje de alarma podría tener un código 3, el cual podría significar algo distinto para el fabricante A del significado del mensaje del fabricante B o también para el control de la red.

Considere la alternativa ilustrada en la Figura 1.3.b. Si el control de la red recibe mensajes estandarizados provenientes de los *gateways* A y B, solamente se necesita ejecutar una sola rutina estandarizada para codificar los bits y los campos del mensaje, así, la interpretación de estos mensajes permiten una administración integral ya que le permite al control de la red determinar (con una mínima cantidad de código) qué mensaje es, cuál es su forma y qué significa.

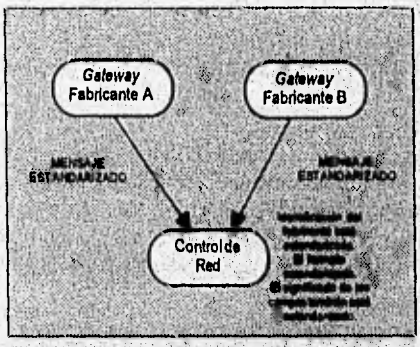


Figura 1.3.b Esquema con protocolos estandarizados de distintos fabricantes.

Para resaltar el potencial de lo anterior, imagine el empleo de las características de una administración común de red con redes Ethernet, TCP/IP, voz mediante teléfono, fibra óptica y las Redes Digitales de Servicios Integrados (*Integrated Service Digital Network*, ISDN). La clave en los estándares de administración de red es desarrollar un conjunto integral de procedimientos y estándares que se apliquen de igual manera a través de los diferentes tipos de redes.

1.3.6 Componentes en la administración de una red

Los estándares de administración de red OSI, Internet e IEEE definen conceptos tales como "proceso de administración" (llamado también "sistema de administración de red" en algunos productos comerciales), "agente de administración" (también conocido como "agente de proceso"). En un sentido estricto, un sistema de administración de red realmente contiene solamente protocolos que conducen información hacia y desde diversos elementos de la red a través de varios agentes en un sistema y en un proceso de administración.

Existe otro elemento en el sistema de administración llamado Base de Información de Administración o biblioteca (*Management Information Base*, MIB). A continuación se realiza la síntesis de los tres conceptos más importantes de la administración de redes:

- Agente: Reporta al proceso de administración el estado de los elementos de la red y recibe instrucciones del mismo sobre las acciones que se realicen sobre estos elementos.
- Proceso de Administración: Dirige las operaciones del agente.
- MIB: Es empleada tanto por el agente como por el proceso de administración para determinar la estructura y el contenido de la información relativa a la administración que se está dando.

¿Dónde residen estos componentes? Los estándares de administración OSI, Internet e IEEE no requieren que sean colocados en un sitio en particular de la red. Sin embargo, en la Figura 1.3.c se muestra una configuración típica en una LAN, dejando ver la localización del proceso de administración, el *software* que sirve de agente, y la MIB. En implementaciones recientes, el agente (*software*) es usualmente colocado en componentes tales como servidores, *gateways*, puentes y ruteadores. Normalmente una Estación de Control de Red (*Network Control Station*, NCS) actúa como el proceso de administración. La MIB es normalmente colocada en el NCS y la parte de la MIB que le pertenece al agente es también contenida dentro del propio agente.

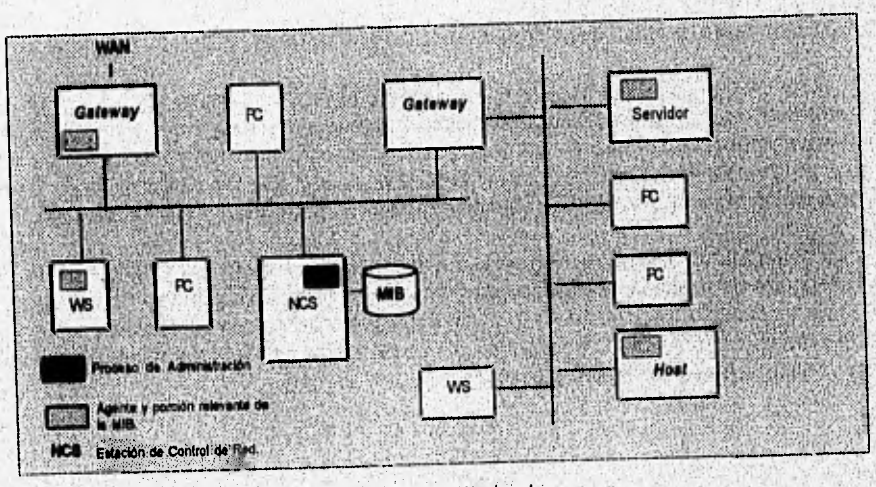


Figura 1.3.c Localización de los componentes de administración de red.

1.3.7 Filtros y mediciones

Filtros

El concepto de filtro es muy utilizado sobre todo en sistemas OSI de administración de red, a diferencia de las administraciones de Internet e IEEE en las cuales dicho concepto prácticamente no existe. Los filtros se emplean para determinar que eventos en una red deban ser reportados. Ellos son vitales en esquemas de administración ya que previenen de una sobrecarga a la red de reportes innecesarios y de eventos triviales, de esta manera sólo permiten el reporte de información crítica.

La administración OSI de redutiliza el término filtro para describir afirmaciones acerca de la presencia de valores de atributos en un objeto manejado. Un filtro puede contener más de una afirmación, en este caso las afirmaciones se agrupan juntas con operadores booleanos. La prueba de filtrado ocurre si ésta es cierta, al mismo tiempo el objeto manejado es seleccionado por la invocación de una operación.

Mediciones

El término medición se puede emplear en conjunto con el de filtro, la medición describe la selección de un conjunto de objetos manejados en un Arbol de Información de Administración (*Management Information Tree*, MIT) al cual se le ha aplicado un filtro.

La medición permite un proceso de administración para especificar el nivel de detalle de los objetos en el árbol. Dicho árbol se define como un "árbol de contenido". Así los protocolos permiten que identificadores de objetos sean pasados para describir el nivel en que se afecta éste por la operación de administración. Esto pudiera ser solamente el objeto base o la MIT entera.

1.3.8 Concepto de Base de Información de Administración (MIB)

A pesar de que ya se ha mencionado a la MIB en puntos anteriores, es necesario establecer de una manera formal la definición de este importante concepto. Este objeto conceptual es actualmente una base de datos que contiene la colección de datos relativos a la administración de la red y que es compartida entre administradores y agentes para proveer información acerca de los elementos de la red administrados. Ver Figura 1.3.d.

La MIB además, define una estructura (varía dependiendo del estándar OSI, Internet ó IEEE) para almacenar los datos recopilados desde la red y para información derivada, como por ejemplo, un conteo de los paquetes con diferentes características tales como de tamaño inferior, conteo de colisiones, almacenamiento de paquetes completos y/o parciales para su análisis posterior, etc.

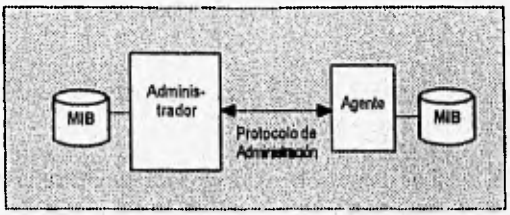


Figura 1.3.d Ubicación de la MIB.

1.4 DISEÑO ORIENTADO A OBJETOS

El Diseño Orientado a Objetos surgió en los años 70's y tiene como finalidad definir entes que guardan ciertas características que los hacen únicos y que les dan el carácter de objetos.

1.4.1 Objetos manejados

Los Objetos que se manejan pueden variar dependiendo de quién los define, por lo que para analizarlos y ver como están definidos, tomaremos los tres estándares existentes:

Objetos manejados por OSI

Los Objetos manejados definidos por OSI son los siguientes :

- **Atributos** (*Attributes*): Los atributos son las características del objeto.
- **Operaciones** (*Operations*): Son las operaciones que pueden ser ejecutadas por el objeto.
- **Notificaciones** (*Notifications*): Son las notificaciones que reportan lo que ocurre con el objeto.
- **Comportamiento** (*Behavior*): Es el rendimiento en cuanto a las operaciones efectuadas por el objeto.

Objetos manejados por Internet

Los objetos manejados por la comunidad Internet se enuncian en los párrafos siguientes:

- **Sintaxis**: La sintaxis en este modelo se refiere al tipo de dato que se emplea y que puede ser de los siguientes tipos: *integer*, *octet string*, *sequence* o *sequence of*.

- *Access*: Es la forma en que un objeto puede ser manejado y tiene cuatro operaciones que son permitidas y son: *read only*, *read write*, *write only* y *not accesible*.

- *Status*: El *status* puede tomar tres estados los cuales son: *mandator*, *optional* y *obsolete*.

- *Name*: Es el nombre por medio del cual será identificado el objeto.

Objetos Manejados por IEEE

En este caso los objetos son los mismos que se definieron por OSI, pero con la variante de que las operaciones son diferentes, las cuales se muestran en las siguientes líneas:

- Operaciones *Get* : Se utilizan para obtener un valor de un objeto determinado.

- Operaciones *Set*: Estas operaciones son usadas para poner un valor en un objeto determinado.

- Operación *Set* y *Compare*: Con este tipo de operación el objeto es sometido a un conjunto de pruebas y si las pasa, entonces le es asignado un valor particular.

- Operaciones *Action*: Esta conjunto de operaciones tienen como finalidad aportar información del objeto en caso de que suceda una falla.

- Operaciones *Event*: Estas operaciones no están definidas dentro de un servicio de usuario, sino que se encuentran localizadas dentro de las entidades administrativas.

1.4.2 Encapsulación

Un objeto se encuentra encapsulado, cuando otro objeto no puede saber lo que ocurre dentro de él. Un objeto se comunica con otro sólo a través de mensajes emitidos. La encapsulación es entonces la capacidad de los objetos de realizar sus funciones respactivas sin que los demás objetos conozcan de cómo las lleva a cabo, pues sólo recibe y envía mensajes. Ver Figura 1.4.a.

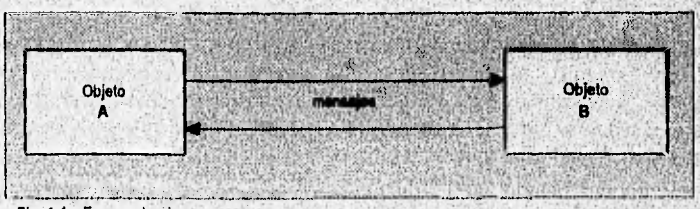


Fig. 1.4.a Encapsulación

I.4.3 Clases

Una clase es un grupo abstracto de objetos administrados, los cuales tienen características en común.

El diseñador del modelo de información de una red deberá definir el conjunto de clases de objetos que lo forman. Como ejemplos de clases en una red podemos mencionar las siguientes: nodo, interface, enlace, modem, puerto, tarjeta de comunicación, protocolo, alarma, mensaje, servicio ,etc.

Para ejemplificar más el concepto tomemos a la clase "nodo"; esta clase representa cualquier elemento de usuario final en la red, tales como: estación de trabajo, terminal, *host*, PBX o nodo X.25, todos ellos objetos administrados (MOs) con características comunes. Así mismo, dentro del OOD se definen las "subclases" como aquellas clases que forman parte de una clase mayor o "super-clase"; por ejemplo, la clase "interface" es una subclase de la clase "nodo" y que representa a aquellos dispositivos que nos sirven para interconectar subredes, tales como repetidores, puentes, ruteadores, *brouters* y *gateways*.

1.4.4 Herencia

Del contexto de OOD, la herencia significa la compartición de atributos entre objetos. Ella provee la función de volver a utilizar los atributos, ya que los nodos del nivel inferior en un árbol de información de administración deberán mostrar el mismo comportamiento de su padre. En la terminología de OOD, esta idea se refiere a la encapsulación de objetos de nivel inferior dentro de los objetos padres de un nivel superior.

El diseño orientado a objetos usa el término de "herencia de clase" para definir como heredan las nuevas clases las propiedades de las clases ya existentes. Puede haber una sola herencia o herencias múltiples. En el caso de una sola herencia, un objeto hereda atributos de una sola clase padre. Una forma más compleja de la herencia es la herencia múltiple, la cual permite heredar a una subclase más de una clase padre. No todos los sistemas orientados a objetos permiten la herencia múltiple; a pesar de ser una característica poderosa, crea una considerable complejidad en el *software*.

I.4.5 Polimorfismo

Polimorfismo es la capacidad de las operaciones y funciones de poder aplicarse a más de un objeto.

Cuando hablamos de herencia de clase nos referimos a que las operaciones que se aplican a una clase padre se deben aplicar también a las clases hijas, es aquí donde el polimorfismo y la herencia se relacionan.

CONCLUSIONES

A través del capítulo se ha podido constatar de la gran diversidad de tópicos que envuelven a la administración de redes. Se han abordado conceptos fundamentales en el ámbito general de redes tales como topologías, elementos de conectividad, sistemas operativos de red, etc., así como conceptos básicos en el ámbito de administración de redes, tales como principales componentes en la administración de redes y diseño orientado a objetos. Tales conceptos permitirán alcanzar una mejor comprensión en los capítulos subsecuentes, pues son herramientas importantes para la definición de la estructura de un administrador de red.

CAPITULO

2

PROTOCOLOS DE COMUNICACION
TCP/IP, NOVELL, APPLETALK, DECNET y SNA

2. PROTOCOLOS DE COMUNICACION: TCP/IP, NOVELL, APPLETALK, DECNET y SNA

2.1 INTRODUCCION

En este capítulo se pretende explicar algunos de los principales protocolos que se utilizan actualmente para enlazar redes en todo el mundo, pues siendo éstos piezas medulares de las telecomunicaciones, es necesario conocerlos bien. Comencemos por definir lo que es un protocolo, un protocolo es el conjunto de reglas o normas que se establecen para cumplir o entablar una comunicación, es importante señalar que un protocolo no es un elemento de *software*, sino un conjunto de especificaciones a las que éste se debe apegar. Existen protocolos de bajo nivel, de nivel medio y de alto nivel, los primeros se encargan del *hardware*, los siguientes generalmente son servicios de red como el transporte de paquetes y finalmente los últimos son aquellos utilizados por el usuario.

En primer lugar se hace una clasificación de los protocolos dependiendo de la arquitectura empleada para su desarrollo, posteriormente se analiza lo que es una arquitectura de protocolos estratificada o *stack* de protocolos. Cabe resaltar la importancia que reviste a este punto, pues es piedra angular para lograr una mejor comprensión de los diferentes protocolos que se tratan en esta sección. Finalmente, se hablará de las últimas tecnologías en cuanto a la transmisión de datos y de los adelantos y ventajas que éstas representan.

2.2 CLASIFICACIONES

A los protocolos de comunicación se les ha tratado de clasificar de múltiples maneras, entre las que se cuentan:

- Por sus métodos de conmutación: Conmutación de paquetes o conmutación de circuitos.
- Por su tipo de conexión: Orientados a la conexión o sin conexión.
- Por sus métodos de comunicación: Maestro-esclavo, *peer-to-peer* o híbrido.

En la Figura 2.2.a se muestra un árbol que inicialmente clasifica a los protocolos por su método de comunicación, para luego subclasificarlos por otras características que se describirán adelante.

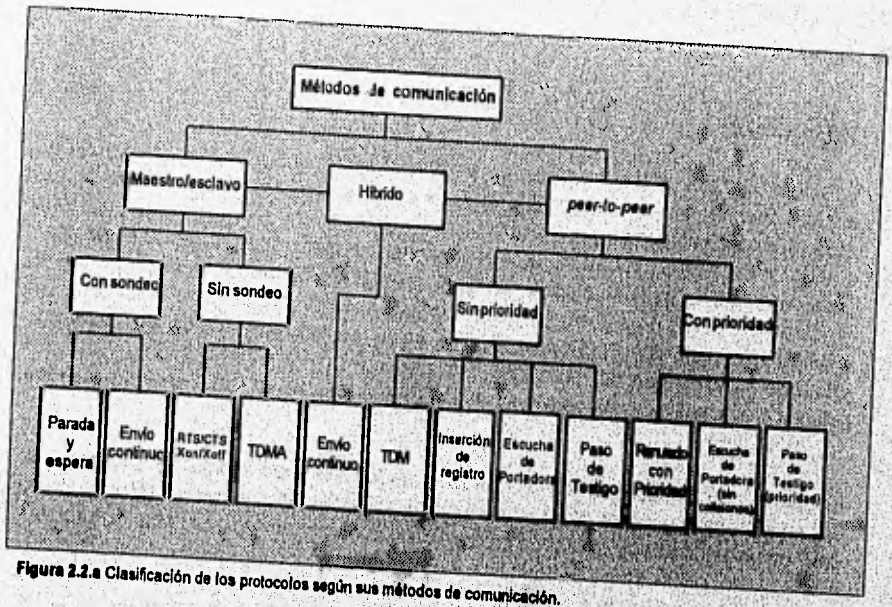


Figura 2.2.a Clasificación de los protocolos según sus métodos de comunicación.

Como se podrá percibir a lo largo de los apartados siguientes, en que se explican cada uno de los conjuntos de protocolos más utilizados, prácticamente todos los procesos de comunicación siguen los siguientes pasos:

- Establecimiento del enlace: Una vez que la conexión física se ha conseguido, el nodo origen pregunta al nodo destino si es que se encuentra preparado para la comunicación, si es así el nodo destino responde afirmativamente efectuándose el enlace, si la respuesta es negativa se espera y de nueva cuenta pregunta si esta listo o no.

- Transferencia de información: Una vez que los dos nodos se encuentran listos, se realiza el intercambio de información. El nodo destino verifica la integridad de la información recibida y notifica sus resultados al origen.

- Liberación del enlace: Cuando los nodos no deseen transmitir, entonces se da por terminada la comunicación.

Para llevar a cabo esta comunicación existen dos métodos básicos, los cuales ya es trataron en el apartado 1.1.2, por lo que omitiremos la definición.

- Maestro-esclavo, primario-secundario o cliente-servidor.

• *Peer-to-peer* (de igual a igual).

Como se aprecia en la Figura 2.2.a, ambos métodos utilizan diferentes tecnologías para su implemetación. A continuación se hará una breve explicación de dichas tecnologías.

Maestro-esclavo, primario-secundario o cliente-servidor

Estos sistemas pueden ser con sondeo o sin sondeo. En las siguientes líneas se describirán dichos sistemas y sus variantes.

Sistemas con Sondeo

Esta técnica funciona de la misma manera cuando se trata de comunicar dos nodos maestros o un maestro con uno esclavo, y consiste de dos operaciones básicas:

- Sondear: Es transmitir datos del nodo esclavo al nodo maestro. El nodo maestro constantemente está "preguntado" a sus nodos esclavos si es que desean enviarle datos (esto básicamente es sondear), si es así, el nodo esclavo enviará sus datos al maestro y este último checará errores en los mismos; si no existen, enviará un mensaje de reconocimiento (AK), en caso contrario, enviará un mensaje de rechazo (NAK). Cuando el nodo emisor ya no desee transmitir datos, entonces enviará un mensaje de fin de transmisión (EOT).

- Seleccionar: Es transmitir datos del nodo maestro al nodo esclavo. Cabe señalar que en la actualidad la mayoría de los protocolos no realizan esta operación, ya que el nodo maestro almacena información necesaria para saber cuando poder enviar información.

Debido a que el nodo maestro tiene que sondear a cada uno de los nodos esclavos, el tiempo y recursos perdidos son muchos, si consideramos que muchos son los nodos que no deseen transmitir. Los métodos de sondeo han cambiado a:

- Sondeo selectivo: Se maneja una tabla de prioridades para los nodos esclavos, de manera que cada vez que un nodo responda al sondeo negativamente, se le disminuirá su prioridad y con ello se le sondeará menos; por el contrario, cada vez que un nodo responda afirmativamente al sondeo se le incrementará su prioridad y con ello se le sondeará más.

- Sondeo de grupo: Es muy utilizado en topologías de anillo. Aquí el sondeo se hace por grupo de nodos, de manera que si alguno de los nodos del grupo contesta afirmativamente al sondeo, entonces éste envía sus datos al nodo contiguo, para que si dicho nodo también desea transmitir, agregue su información a la cadena, y lo pase al nodo contiguo, y así sucesivamente hasta que llegue al nodo maestro.

Existen dos técnicas básicas de sondeo, las cuales se describirán a continuación:

- Parada y espera: Es la forma más sencilla de transmitir y consiste en que un nodo envía un mensaje al otro y espera, cuando recibe una respuesta entonces envía, espera a que el otro nodo envíe sus datos y luego él envía, y así sucesivamente. Este mecanismo es *half-duplex*, lo que quiere decir que la comunicación se puede realizar en ambos sentidos pero no a la vez. Para no perder el control de los datos que envían los nodos, cada uno de ellos se encarga de poner un número de secuencia a cada paquete que envía, así el nodo receptor sabrá cual es el orden de los datos y si es que falta alguno.

- Envío continuo (ventanas móviles): A diferencia de la técnica anterior, la técnica de envío continuo (*Allowed to Request*, ARQ) puede trabajar con un mecanismo de *full-duplex*, por lo cual maneja el concepto de "ventanas" de transmisión y de recepción. Una ventana es la cantidad de recursos asignados a una comunicación.

  En este tipo de comunicaciones *duplex*, el secuenciamiento de las tramas es muy importante, debido a la gran cantidad de flujo de información entrante y saliente al mismo tiempo. Por ello, las ventanas de transmisión y de emisión se controlan por contadores, que verifican el secuenciamiento de las tramas enviadas y recibidas.

Sistemas sin sondeo

Como se aprecia en la Figura 2.2.a, forman parte de este grupo los métodos: Requerimiento de Transmisión (*Request to Send*, RTS)/Permiso de Transmisión (Clear to Send, CTS), *Xon/Xoff* y Acceso Múltiple por División de Tiempo (*Time Division Multiple Access*, TDMA).

- RTS/CTS: Estos tipos de protocolos son muy poco empleados, aunque su principal uso está en la interface a nivel físico, con la interface RS-232-C. Una aplicación típica de este protocolo es la conexión de un equipo terminal con un modem.

- *Xon/Xoff*. *Xon y Xoff* son dos caracteres ASCII empleados por equipos periféricos como impresoras, graficadores o trazadores, para indicar al equipo terminal su disponibilidad (*Xon*) o su indisponibilidad (*Xoff*), lo cual indica al equipo terminal si puede o no enviar datos.

- TDMA: En esta técnica el nodo maestro espera a que lleguen las solicitudes de los nodos esclavos. Estas solicitudes son enviadas dentro del paquete de información en curso en un campo de control especial. Cada cierto tiempo, el nodo maestro envía una trama de control que indica que nodos pueden utilizar el canal y por cuanto tiempo. Una vez que un nodo esclavo recibe una trama de autorización, éste inicializa su reloj para enviar su información por el tiempo asignado.

Peer-to-peer

Los sistemas *peer-to-peer* pueden ser con o sin prioridad. En las siguientes líneas se describen dichos métodos y sus distintas implementaciones.

Sistemas sin prioridad

Dentro de la segunda rama de la clasificación, sistemas *peer-to-peer*, se encuentra una subclasificación: sistemas sin prioridad y sistemas con prioridad. Dentro de los primeros se encuentran Multiplexado por División de Tiempo (*Time Division Multiplexing*, TDM), inserción de registro, escucha de portadora y paso de testigo, los cuales se describen a continuación:

- TDM: En este protocolo, a cada nodo se le asigna un tiempo en el cual puede transmitir, de esta manera el canal queda compartido por todos los nodos en el tiempo y de forma equitativa (sin prioridad).

- Inserción de registro: Es un método muy usado en topologías de anillo, en la que todo nodo puede transmitir, ya que va anexando sus datos a la trama que recorre el anillo.

- Escucha de portadora: Este protocolo consiste en que los nodos que deseen utilizar el canal deben "escuchar" para saber si el canal está desocupado o no; si el canal se encuentra desocupado, el nodo espera un tiempo aleatorio y entonces envía sus datos. Debido a que los nodos pueden tratar de accesar el canal en instantes imprevistos, en este tipo de protocolos, es muy común que ocurran colisiones, cuando sucede que dos o más nodos intentan accesar al canal en el mismo instante; este problema no se puede evitar, pero sí se puede reducir su incidencia y remediar cuando ocurre. Cada vez que ocurre una colisión, los nodos involucrados tendrán que dejar de transmitir y esperar otro tiempo aleatorio para reiniciar su transmisión.

- Paso de testigo: El paso de testigo es más conocido por su nombre en inglés *token passing*. Este protocolo es empleado generalmente en redes con topologías de *bus* o de anillo. El paso de testigo consiste en que una pequeña trama llamada testigo o *token* se encuentra circulando a lo largo del anillo o del *bus*, si un nodo desea transmitir, tendrá que esperar a que el *token* lo "visite", entonces verificará que éste esté vacío, y si es así, anexará sus datos a la trama del *token*, si no es así, tendrá que esperar la próxima visita del *token*.

Sistemas con prioridad

En estos sistemas se utilizan técnicas como las anteriores solo que modificadas. La principal modificación es desde luego el uso de prioridades. Los protocolos de ranurado con prioridad, escucha de portadora (sin colisiones) y paso de testigo con prioridad, forman parte de este grupo.

- Ranurado con prioridad: Este protocolo es semejante al de TDM, sólo que aquí los tiempos a transmitir de cada nodo son asignado según su prioridad.

- Escucha de portadora (sin colisiones): En este protocolo a los nodos se les asigna una prioridad, de tal forma que el tiempo que esperan para tomar el canal, una vez que éste está libre, es variable según la prioridad.

- Paso de testigo con prioridad: Este protocolo es semejante al descrito anteriormente. Aquí los nodos tienen asignada una prioridad, si ésta es mayor que la que trae el *token*, entonces tendrá que esperar a la siguiente vuelta y el *token* actualizará la prioridad, para que a la siguiente vuelta le toque transmitir al nodo visitado.

2.3 STACK DE PROTOCOLOS EN EL MODELO OSI

En el capítulo anterior ya hablamos de la importancia que tienen los estándares y de las organizaciones que los emiten, atendiendo a esto se examinará un estándar, en este caso el *stack* de protocolos OSI.

La razón más importante para que exista un modelo de *stack* de protocolos es desarrollar un conjunto de procedimientos, tal que sea común a todos los fabricantes. En la arquitectura de protocolos estratificados cada estrato o nivel aporta un servicio el cual puede estar formado por varias funciones de servicio. En sí, la idea es que cada nivel aporte un componente que pueda ser usado por los demás niveles, este modelo está formado por siete niveles los cuales se tratarán a continuación:

- Nivel Físico: En este nivel se activa y desactiva un circuito físico entre los nodos que entablan la comunicación, aquí se especifican las características de *hardware* para la conexión física de un nodo a la red (voltaje, tipo de código y la forma de accesar al medio).

- Nivel de Enlace: En este estrato se realiza la transferencia de los datos a través del medio de comunicación, en este nivel se verifica que los datos lleguen sin errores y si éstos se dan, se tratará de recuperar los datos , si es que son recuperables. En este estrato se define el formato de los paquetes de datos o *frame*.

- Nivel de Red: Se encarga de llevar acabo las operaciones de ruteo por la red y la comunicación entre distintas redes.

- Nivel de Transporte: Proporciona la comunicación con los tres niveles superiores, especifica como se realiza la comunicación *peer to peer* segura y eficiente entre procesos antes de que ésta ocurra, por lo cual provee servicios para que se establezcan comunicaciones orientadas a la conexión. El nivel de transporte proporciona fundamentalmente tres tipos de servicios:

  - Servicios orientados hacia el establecimiento de una conexión.

  - Servicios orientados hacia la realización de transacciones.

  - Servicios orientados hacia la difusión de información a múltiples destinatarios.

  A los entes de este nivel se les denomina estaciones de transporte o puntos finales del bloque de transporte.

  Las operaciones de intercambio de información entre estaciones de transporte se realizan mediante protocolos denominados de "transporte entre puntos finales" (*end-to-end transport protocols*).

- Nivel de Sesión: El nivel de sesión funciona como interface del nivel de trasporte, normaliza el proceso de una sesión local y su terminación. En este nivel se identifica a los usuarios que están participando de la sesión. Se encarga del acceso a terminales remotas (conexiones virtuales). El objetivo de los elementos situados en este nivel es proporcionar un soporte a la comunicación entre los niveles del nivel de presentación. Los entes del nivel de sesión utilizan a su vez los servicios del nivel de transporte. En sí una sesión no es más que una relación de cooperación entre dos nodos para permitir la comunicación entre ellos. Cada ente del nivel de sesión se identificará mediante una dirección, asociada a un elemento capaz de almacenar la información que se intercambia.

- Nivel de Presentación: El objetivo de los elementos situados en este nivel es proporcionar un conjunto de servicios . Dichos servicios están orientados principalmente a la interpretación de la estructura de la información intercambiada por el proceso de aplicación. Se asigna una sintaxis a los datos, es decir, se le da una forma a éstos sin importar su significado semántico, esto es con el fin de asegurar la comunicación, si ésta usa diferentes representaciones. Las funciones asignadas a los niveles de aplicación y presentación son de la misma naturaleza y en cierto modo complementarias.

- Nivel de Aplicación: Este nivel se encarga de atender a las aplicaciones del usuario, en éste importa la semántica -el significado- de los datos. Este estrato maneja terminales virtuales, soporta el uso de datos remotos, técnicas de transferencia de archivos y distribución de actividades de bases de datos.

La Figura 2.3.a nos muestra el proceso de aportación que cada nivel hace al *frame*. En ella podemos ver cómo cada nivel aporta un *header* a dicho *frame*, y por el lado receptor, ascienden los datos por las siete capas, quitándoles sus *headers* correspondientes, a este proceso se le conoce como encapsulación y desencapsulación.

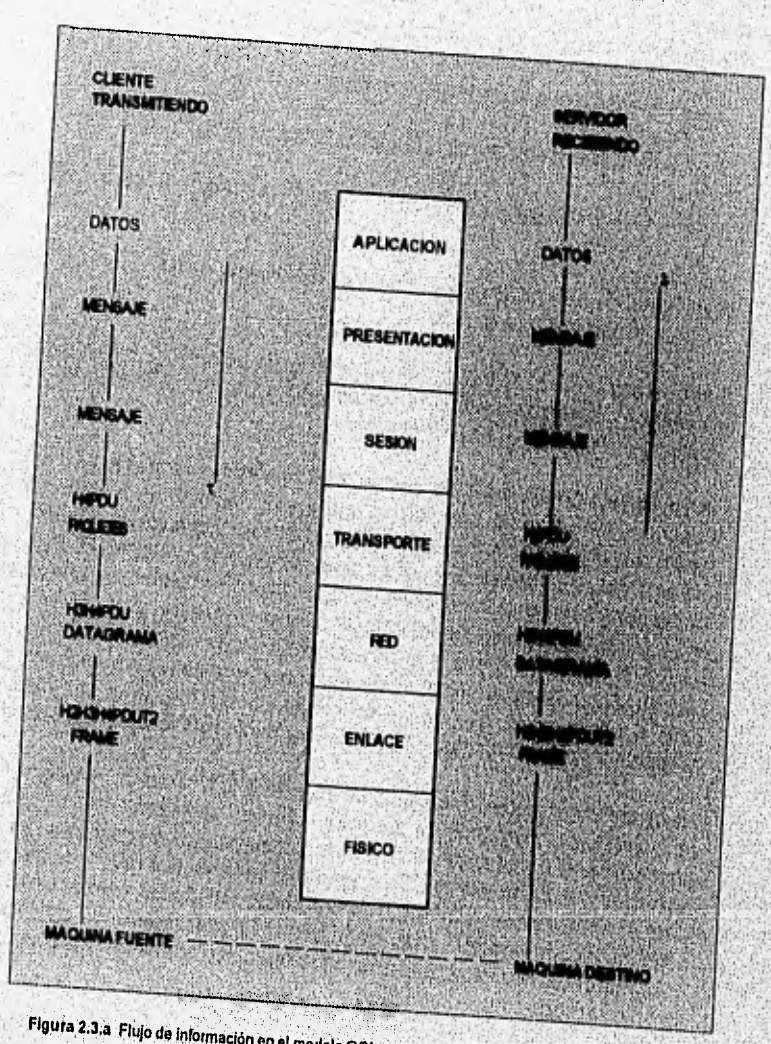


Figura 2.3.a Flujo de información en el modelo OSI

Este proceso de encapsulación y desencapsulación es diferente al manejado en el OOD, cuyo concepto ya fue examinado. La idea es semejante pero el proceso en sí es diferente. Pues en este caso se agregan encabezados o *headers* dando forma así, al *frame* que finalmente será transmitido.

2.4 TCP/IP

El protocolo TCP/IP fue desarrollado originalmente por el gobierno de los Estados Unidos (el departamento de la defensa) con una finalidad netamente militar, sus iniciales significan Protocolo de Control para la Transmisión y Protocolo de Inter-red (*Transmission Control Protocol/Internet Protocol*, TCP/IP). Este protocolo fue lanzado comercialmente después, y en poco tiempo fue distribuido a compañías privadas, universidades y centros de investigación, con lo cual rápidamente captó la atención de los principales centros de desarrollo de ese país y pasó a ser la columna vertebral de lo que en la actualidad es la red más grande del mundo, hablamos de la red de la Comunidad Internet.

TCP/IP ofrece tres servicios básicos, correo electrónico, transferencia de archivos y sesiones remotas. El modelo TCP/IP varía del modelo OSI, ya que el primero tiene cuatro niveles solamente; la razón de esto es que este modelo fue desarrollado antes de que surgiera el modelo normalizado OSI. En la siguiente figura se muestran los niveles Internet TCP/IP.

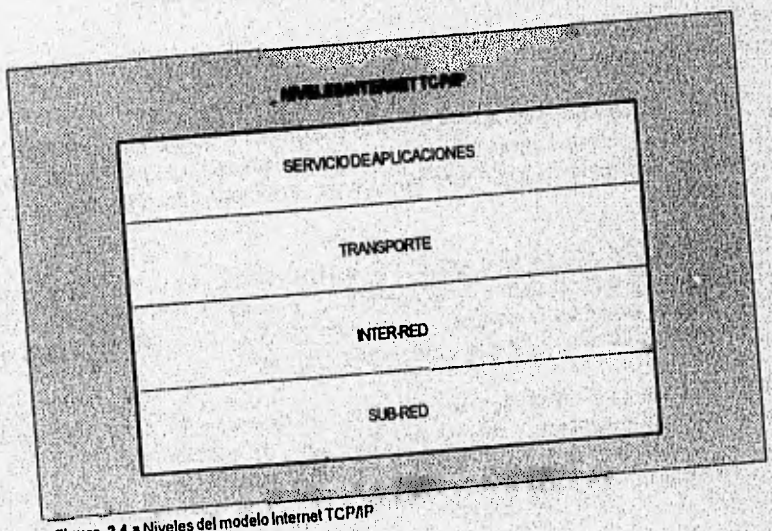


Figura. 2.4.a Niveles del modelo Internet TCP/IP

- Nivel de Sub-red: Este nivel contiene una interface para sub-redes como *Tymnet, Transpac, Arpanet* y *Ethernet*. En lo que respecta a la interface a redes se requiere de una máquina que esté conectada a un *gateway*. Realiza el intercambio de datos entre un dispositivo y la red a la que se conecte. También rutea datos entre dispositivos en la misma red.

- Nivel de Inter-red: Este nivel nos ofrece las funciones necesarias para establecer las conexiones con otras redes y *gateways*, esta capa es responsable del encaminamiento correcto de los datos desde el nodo origen al nodo destino. Además de que en este nivel se encuentra alojado el Protocolo de Internet (*Internet Protocol*, IP) y el Protocolo Controlador de Mensajes Internet (*Internet Control Message Protocol*, ICMP) que es un protocolo que auxilia a IP en cuanto a ruteo y mapeo de direcciones.

- Nivel de Trasporte: Este estrato es el responsable de entablar la comunicación *peer-to-peer* además de contener TCP y el Protocolo de Datagrama de Usuario (*User Datagram Protocol*, UDP), se encarga de regular el flujo de información y de que los datos lleguen adecuadamente, sin errores y en secuencia.

- Nivel de Servicios de Aplicación: Este nivel soporta directamente la interface con el usuario final y soporta los servicios básicos que se mencionaron al inicio de esta sección. Contiene al Protocolo de Transferencia de Archivos (*File Transfer Protocol*, FTP).

Cabe hacer notar que el nombre de los paquetes varía dependiendo del nivel en que se manejen, éstos pueden ser mensaje, paquete, datagrama y *frame*.

En la Figura 2.4.b se muestran los protocolos que forman el *stack* TCP/IP ajustados a los siete niveles del modelo OSI.

- IP: Es la plataforma de los servicio básicos de TCP/IP, se encarga de definir la unidad mínima de transferencia, también denominado datagrama, además realiza funciones de ruteo y determina las condiciones para la entrega de paquetes.

  El tamaño del datagrama usado por IP está en función de la red a la que se conecta, pero si el tamaño del datagrama excede el valor límite, éste se fragmenta y se envía seccionado. Cabe señalar que cada fragmento mandado tiene la misma estructura que el datagrama original.

  El IP hace uso de datagramas para la transferencia de datos, pero las direcciones manejadas por IP se diferencian de entre las computadoras conectadas a la red y de los programas de aplicación que hacen uso del protocolo, por lo que IP está limitado para el uso de direccionas múltiples.

Este protocolo como ya se mencionó antes, ofrece varios servicios: servicio de aplicación, servicio de transporte confiable y servicio de entrega de paquetes sin conexión, servicio de entrega de paquetes no confiable de mejor esfuerzo sin conexión.

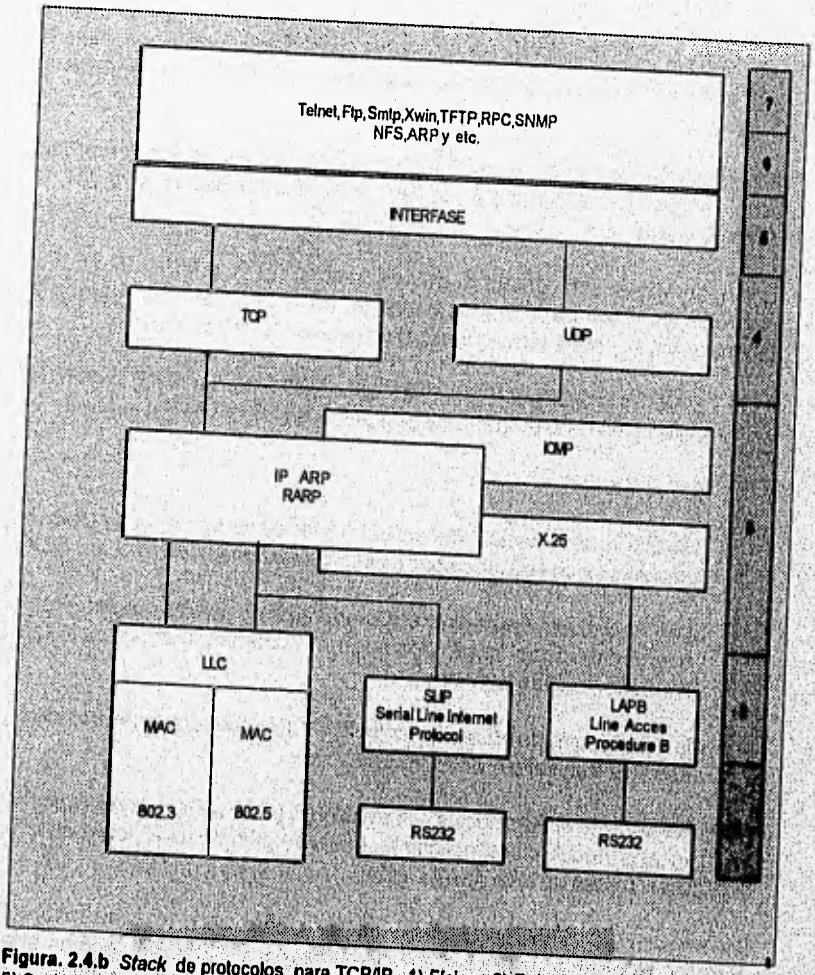


Figura. 2.4.b *Stack* de protocolos para TCP/IP, 1) Físico 2) Enlace 3) Red 4) Transporte 5) Sesión 6) Presentación 7) Aplicación.

- TCP: Este protocolo nos provee de servicio de comunicación orientada a la conexión, transportación *peer-to-peer* entre múltiples procesos, multiplexado en el acceso a puertos múltiples, orientación a flujo de datos, conexión de circuito virtual, *buffer* de transferencia, conexión *full-duplex*, así como flujo de datos no estructurado.

- UDP: Tiene como objetivo crear una interface que permita comunicarse con puertos externos, por lo que UDP hace uso de IP. A los mensajes de UDP se les denomina datagramas de usuario y están constituidos por un encabezado y un área de datos, el encabezado del datagrama se divide en cuatro campos de 16 bits.

- ICMP: Las funciones básicas de este protocolo son: generar reportes de errores en el proceso de datagrama y proveer de algunos mensajes y *status* de administración

El ICMP reside en un *host* o en un *gateway* junto con el IP. ICMP verifica que la dirección de IP sea correcta y de no ser así, emitirá un mensaje de error. A continuación se presentan algunos aspectos importantes de ICMP:

- ICMP es usado por IP. El IP encapsula los datos de ICMP dentro del datagrama de IP para ser transportado a través de una red Internet.

- IP usa a ICMP.

- ICMP no puede hacer a IP confiable, pues su función es la de reportar errores, ya que es el encargado de los niveles altos del protocolo TCP.

- ICMP reporta errores dentro del datagrama de IP, pero no puede reportar errores dentro de sí mismo, lo cual en un momento puede provocar que se hagan reportes erróneos.

- Si el datagrama de IP se fragmenta, ICMP sólo puede reportar errores en el primer fragmento y no en los siguientes.

- ARP: El Protocolo de Resolución de Direcciones (*Adress Resolution Protocol*, ARP) es un protocolo cuya finalidad es la conversión de direcciones de IP a direcciones físicas para los niveles altos.

Generalmente IP trabaja con tablas de mapeo, dichas tablas ofrecen un mapeo entre las direcciones de IP y las direcciones físicas. En una LAN, ARP toma la dirección de IP y busca la dirección correspondiente en la tabla de mapeo, si la dirección no se encuentra, se envía un *broadcast* a toda la red el cual es denominado ARP *request*. Si una máquina recibe el *brodcast* reconociendo la dirección IP, ésta emite una réplica de ARP para el *host*. Este *frame* contiene la

dirección física del *hardware* que es requerida por el *host*. Una vez recibido el *frame* por el *host*, éste incorpora la dirección dentro de ARP. El proceso se ilustra en la figura 2.4.c.

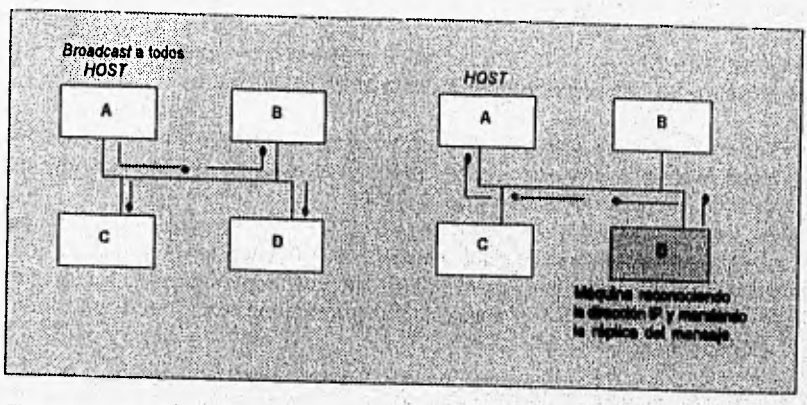


Figura 2.4.c. Petición de ARP y respuesta a la misma.

- RARP: El Protocolo de Reversa de Resolución de Direcciones (*Reverse Adress Resolution Protocol*, RARP) trabaja de manera similar al ARP, excepto que el nombre nos indica que trabaja en reversa pues el *host* manda un *broadcast* a toda la red preguntando la dirección de IP y recibe por respuesta el mensaje de una máquina, quien devuelve la dirección IP del *host*. A la máquina que responde se le conoce como servidor de RARP. En la siguiente figura se ilustra dicho proceso.

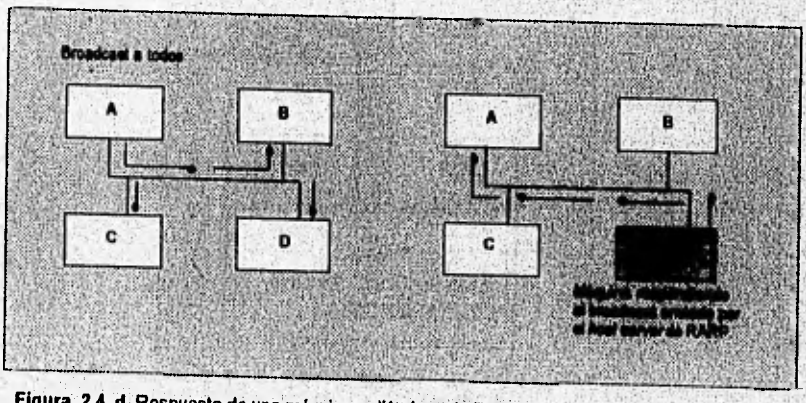


Figura 2.4. d. Respuesta de una máquina valiéndose de RARP.

2.5 XNS, IPX y SPX

El Sistema de Red de Xerox (Xerox Network System, XNS), el cual fue desarrollado en Xerox Palo Alto Research Center (PARC), ha sentado las bases para varias redes populares, como por ejemplo VINES de Banyan o 3+ de 3Com. Sin embargo, de entre ellas, la más conocida es Novell con su sistema operativo NetWare.

Propiamente, Novell es una derivación de la forma en que comunica a los equipos de cómputo el XNS, por lo cual existen varias similitudes entre ellos. En este apartado se tratarán de una manera muy cercana a XNS y Novell.

Al igual que cualquier sistema de comunicación, XNS y Novell están definidos por una arquitectura de *stack* de protocolos, el cual a su vez, se basa en los siete niveles del modelo OSI. En la Figura 2.5.a se muestra la construcción del modelo de comunicación para ambos sistemas.

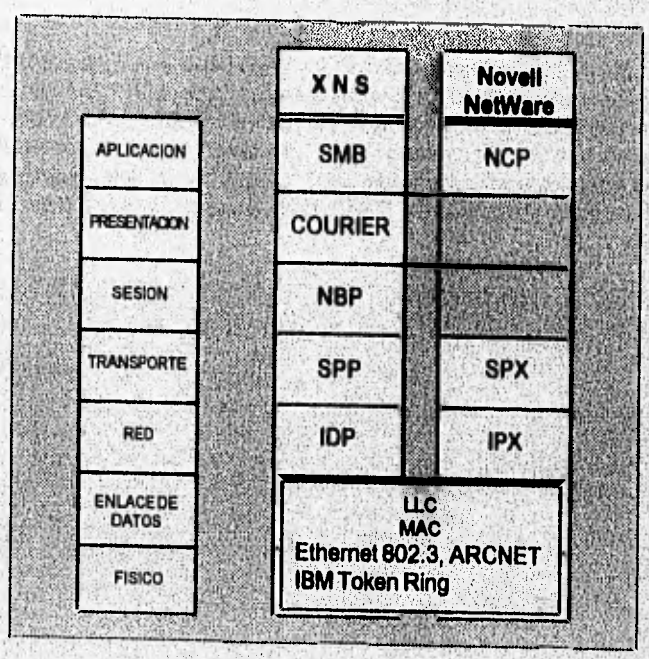


Figura 2.5.a Modelo de comunicación para XNS y Novell.

De la figura podemos hacer las siguientes consideraciones generales:

- Novell ha adoptado la porción baja de la arquitectura de XNS y la ha incorporado a su propia arquitectura. El nivel de red de XNS, el Protocolo de Datagrama de Inter-red (*Internetwork Datagram Protocol*, IDP) ha sido transladado casi íntegramente a Novell con el nombre de Protolo de Intercambio de Paquetes de Inter-red (*Internetwork Packet eXchange*, IPX).

- También, Novell usa variantes del Protocolo de Paquetes Secuenciados (*Sequenced Packet Protocol*, SPP), incorporando sus características en el Intercambio de Paquetes Secuenciado (*Sequenced Packet eXchange*, SPX).

- Varios protocolos de los niveles superiores del OSI no han sido adoptados por Novell. Para cubrir las funciones propias de estos niveles, esta compañía emplea a los Protocolos Núcleo de NetWare (*NetWare Core Protocols*, NCP).

Nivel Físico y de Enlace de Datos

El XNS y Novell soportan a los tres estándares de adaptadores que se utilizan principalmente en redes LAN y MAN: Arcnet, Ethernet (versión 2.0 y la especificación de IEEE 802.3) y Token Ring. En Arcnet, el enlace de datos y su adaptador (nivel físico), son provistos por desarrollos propietarios de Datapoint Co. Para el caso de Ethernet 802.3 y Token Ring emplean a dos subniveles, el Control de Enlace Lógico (*Logical Link Control*, LLC) y el Control de Acceso al Medio (*Medium Access Control*, MAC).

El subnivel MAC se responsabiliza de controlar el proceso de mover un paquete de datos hacia un nodo destino. Este subnivel en conjunto con el nivel físico envían los datos por el medio de transmisión hacia el MAC del nodo destino, y éste a su vez presenta dicho paquete a su correspondiente LLC.

El LLC entonces, sube el paquete al nivel de red, por ejemplo al IPX. Pareciera que el LLC no aportara nada a la comunicación, pero no es así, ya que este subnivel provee independencia entre los adaptadores del nivel físico -o de subniveles como el MAC- y el nivel de red. Compañías como Novell están en libertad de desarrollar rutinas que sirvan como interface entre su nivel de red y el LLC, de esta manera, si existe un nuevo tipo de medio por el que se comuniquen datos, no se altera la forma de subir el paquete a los niveles superiores del LLC.

Nivel de Red

Es en este nivel en donde se encuentra mayor similitud entre el XNS y Novell. Como IPX e IDP son dos protocolos prácticamente idénticos, lo que se trate en este apartado sobre uno de ellos tiene completa aplicación sobre el otro.

El IPX es un protocolo sin conexión (*connectionless*) o datagrama. El término *connectionless* significa que cuando una aplicación de la estación de trabajo usa al IPX

para comunicarse con otra estación de trabajo, no se establece ninguna conexión directa o se entuba (*pipe*) la comunicación entre las dos estaciones. Así, los paquetes del IPX conteniendo datos son direccionados y mandados a su destino, pero no se garantiza ni se verifica que sean recibidos. El término "datagrama" significa que cada paquete es tratado como una entidad individual, sin tener relación de una secuencia lógica con algún otro paquete.

El IPX realiza tareas que incluyen el direccionamiento, ruteo y conmutación de los paquetes de información para obtener paquetes individuales de una localidad a otra dentro de una red.

El ser un protocolo sin conexión proporciona algunas ventajas pero también desventajas. A continuación se enumeran algunas de ellas:

- No es necesaria una disponibilidad simultánea del que manda y el que recibe, puesto que no existe una conexión predeterminada. Sin embargo, el que manda no recibe ninguna verificación o garantía de que el otro haya recibido completo el mensaje.

- La flexibilidad en el ruteo de paquetes es mayor puesto que no se necesita un ruteo predeterminado. La desventaja es que los paquetes podrían llegar en cualquier secuencia.

- Los paquetes pueden ser mandados a múltiples destinos, simplemente duplicando el paquete y cambiándole la dirección destino.

El paquete de IPX, al igual que el de IDP de Xerox, consiste de dos partes, un *header* de 30 bytes y una porción de datos que puede ir de 0 a 546 bytes. La longitud mínima del paquete es de 30 bytes (el *header*) y la longitud máxima es de 576 bytes. En la Tabla 2.5.b se resume la estructura de un paquete IPX, y se establece las pequeñas diferencias con respecto a IDP.

Nivel de Transporte

En este nivel se ubican los protocolos SPP y SPX de Novell. Al igual que los protocolos del nivel de red, SPP y SPX son prácticamente idénticos, por lo mismo, la descripción que se realice de uno se puede aplicar al otro.

El SPX es idéntico al IPX excepto que tiene el *overhead* adicional del nivel de transporte. Estas tareas adicionales al nivel de red hacen del SPX un protocolo orientado a la conexión (*connection-oriented protocol*). Esto significa que, antes de mandar un paquete SPX, una conexión o *pipe* entre el emisor y receptor se establece. El SPX realiza las tareas de garantizar la transmisión, el secuenciamiento de paquetes, detección y corrección de errores y supresión de paquetes duplicados.

Desplazamiento	Contenido	Tipo	Comentario
0	Checksum	BYTE[2]	Se aplica solo al IDP. En IPX siempre es 0xFFFF
2	Longitud	BYTE[2]	Longitud del paquete completo. Mínimo 30 bytes, máximo 576.
4	Ctrl de Transp.	BYTE	Se usa en puentes. IPX lo pone en 0 antes de mandar el paquete.
5	Tipo de Paquete	BYTE	Indica el tipo de servicio ofrecido o solicitado. Para IPX es 0 o 4, para IDP va de 0 a 31.
6	Red destino	BYTE[4]	El núm. de la red a donde va el paquete.
10	Nodo destino	BYTE[6]	El núm. del nodo a donde va el paquete. FFFFFF va a todos los nodos
16	Socket destino	BYTE[2]	El *soket* rutea al paquete a procesos dentro de un nodo.
18	Red fuente	BYTE[4]	El núm. de la red de donde viene el paquete.
22	Nodo fuente	BYTE[6]	El núm. del nodo de donde proviene el paquete.
28	Socket fuente	BYTE[2]	Contiene el proceso que lleva el paquete.
30	Datos	byte [0 A 546]	Información pura.

NOTA: Todos los campos son alto-bajo, lo que significa que el byte más significativo es el primero.

Tabla 2.5.b Estructura del paquete de IPX.

El contar con una transmisión asegurada de la información es una gran característica, sin embargo, también tiene sus desventajas. El siguiente, es un resumen de los beneficios y problemas del SPX:

- Transmisión garantizada: Una conexión es establecida antes de que la información

sea mandada y una verificación de la transmisión regresa al emisor. La desventaja es que si el receptor no está disponible, los paquetes no pueden ser mandados. La propagación del mensaje hacia múltiples estaciones es dificultosa. También, algunas aplicaciones no necesitan garantizar la transmisión para todos los paquetes.

- Secuenciamiento garantizado: El secuenciamiento de los paquetes está garantizado, de esta manera se sabe cuantos paquetes de un mensaje necesitan llegar y su secuencia apropiada.

- Eliminación de paquetes duplicados: Durante el proceso de garantizar la transmisión (el cual incluye el mandar nuevamente paquetes presumiblemente perdidos), es posible que lleguen paquetes duplicados al lado receptor. SPX descarta tales paquetes así es que la aplicación recibe solamente una copia de los datos mandados por el emisor.

El paquete de SPX es idéntico al paquete del IPX, excepto que éste tiene 12 bytes adicionales en su *header*. El paquete consiste de dos partes: un *header* de 42 *bytes* y una porción de datos que va desde 0 hasta 534 *bytes*. La longitud mínima del paquete es de 42 *bytes* (el *header*) y la longitud máxima es de 576 *bytes*.

Los campos del paquete SPX que van desde el *checksum* hasta el *socket* fuente tienen exactamente el mismo significado de sus correspondientes en el IPX, con las siguientes excepciones:

- El campo de Tipo de Paquete es siempre 5 para el SPX.

- El campo de Nodo Destino podría no contener una dirección de propagación. La propagación de un mensaje hacia varias terminales no es permitida en SPX.

En la Tabla 2.5.c se muestra la estructura de SPX.

Niveles Superiores: Aplicación, Presentación y Sesión.

Como se pudo apreciar en la Figura 2.5.a, varios protocolos de niveles superiores de XNS no han sido adoptados por Novell. Por ejemplo, el XNS define un protocolo llamado *Clearinghouse*, para encontrar nombres de servidores sobre la red, en lugar de esto, Novell desarrolló su propio Protocolo de Servicio de Avisos (*Service Advertisement Protocol*, SAP).

Existen varias capacidades de XNS como las de intercambio e impresión que no han sido implementadas en Novell. En algunos casos, la industria que desarrolla aplicaciones sobre esta red adoptan protocolos equivalentes, por ejemplo se usa normalmente el PostScript en lugar del estándar Interpress de XNS para obtener servicios de impresión. XNS emplea el Servicio de Manejo de Mensajes (*Message Handling Service*, MHS) de *Action Technologies*. Para el procedimiento de llamadas remotas, Novell usa al Procedimiento de Llamadas Remotas de Netwise en lugar del protocolo Courier de XNS.

Desplazamiento	Contenido	Tipo	Comentario
0	Checksum	BYTE[2]	
2	Longitud	BYTE[2]	
4	Ctrl de Transp.	BYTE	
5	Tipo de Paquete	BYTE	Es siempre 5 en SP
6	Red destino	BYTE[4]	
10	Nodo destino	BYTE[6]	No es permitido el mensaje a todos los nodos (FFFFFFFF).
16	Socket destino	BYTE[2]	
18	Red fuente	BYTE[4]	
22	Nodo fuente	BYTE[6]	
28	Socket fuente	BYTE[2]	
30	Ctrl. de Conexión	BYTE	Controla el flujo bidireccional
31	Tipo de flujo	BYTE	Tpo de dato encontrado en el paquete.
32	Id. de conexión f.	BYTE[2]	Núm. de conexión asignada por SPX a la fuente del paquete.
34	Id. de conexión d.	BYTE[2]	Núm. de conexión asignada por SPX al destino del paquete.
36	Núm. Secuencia	BYTE[2]	Es un contador de paquetes intercambiados en una sola dirección.
38	Reconocimiento	BYTE[2]	Es el núm. de secuencia del siguiente paquete SPX que se espera recibir
40	Núm. Localidad	BYTE[2]	Es el número de buffers "escuchados" en una dirección de la comunicación
42	Datos	byte (0 A 534)	Información pura.

NOTA: Todos los campos son alto-bajo, lo que significa que el byte más significativo es el primero.

Tabla 2.5.c Estructura del paquete de SPX.

No quiere decir que sea malo para Novell adoptar solamente una parte de la arquitectura de XNS. De hecho son pocas las compañías las que han implementado versiones completamente apegadas a los sistemas de XNS. Por lo mismo, esta forma de comunicación se emplea en la industria, solamente para sentar las bases en la construcción de servicios de red, tales como el acceso de datos, impresión ó recursos de comunicación.

Novell sustituye las funciones de los niveles superiores con su Protocolo Núcleo de NetWare (*NetWare Core Protocol*, NCP). El NCP le permite al usuario accesar datos remotos, archivos para impresión, y realiza otras operaciones básicas. Sin embargo, su función primordial es permitir la creación de una conexión con el servidor.

En el intercambio de datos, la estación de trabajo primero obtiene su propia dirección por medio de uno de los protocolos de subnivel llamado Protocolo de Ruteo de Información (*Router Information Protocol*, RIP), ya que se tiene la dirección, se envía al servidor para crear una conexión con él. La máquina que funge como servidor NCP puede mantener varias conexiones activas, una a cada estación. El NCP asigna un número para identificar a cada conexión.

Cada paquete que entra al *socket* de NCP contiene cuatro piezas de información que ayudan a identificar la operación. El número de conexión indica cual de las conexiones que el NCP está manteniendo en el servidor deberá trabajar con este paquete.

El tipo de requisición indica si el paquete es una petición o un respuesta. La forma de operación del envío de paquetes se parece mucho a jugar ping-pong, ya que primero se envía un paquete de petición y el receptor contesta con un paquete de respuesta, y así sucesivamente.

El número de secuencia asegura que después de un envío de paquete de petición, el siguiente paquete deba ser de respuesta.

Finalmente, se requiere un número de tarea. Con él, NCP identifica que todas las partes de una transacción sean terminadas completamente. Por ejemplo, la creación y configuración de una nueva conexión son partes de la misma tarea, al final de la conexión el NCP enviará la petición de fin de tarea.

2.6 APPLETALK

AppleTalk es el nombre dado por Apple a su arquitectura de red propietaria. AppleTalk es todo un *stack* de protocolos definido en los siete niveles del modelo OSI.

En el presente punto, primeramente se hará una descripción general del *stack* AppleTalk, para posteriormente entrar en detalle de cada uno de los protocolos que lo forman, ambas descripciones se harán en base al modelo de referencia OSI.

AppleTalk clasifica sobre el modelo OSI sus protocolos, de acuerdo a sus funciones (Fig. 2.6.a), de la siguiente manera:

- Funciones físicas y de enlace de datos: Contienen los Protocolos de Acceso a Enlace (*Link Access Protocol*, LAPs) y en este grupo se concentra el *hardware* necesario para las interfaces y la conexión al medio.

- Funciones de flujo de datos *peer-to-peer*: Aquí se concentra el nivel de red y parte del nivel de transporte, en los que sus protocolos son los encargados de garantizar el flujo de datos de un *socket* de red a otro (*socket* es un término usado en AppleTalk y en otras redes para designar a cualquier entidad direccionable

en un nodo). Este grupo no se hace responsable de la confiabilidad de los datos liberados.

- Funciones de entidades nombradas: Ubicado en una parte del nivel de transporte y otra parte del nivel de sesión. Debido a que el grupo anterior no es capaz de recordar las direcciones origen y destino de un enlace, AppleTalk lo soluciona asignando nombres a todos los dispositivos y servicios de la red. Dichos nombres son manipulados por los protocolos de este grupo.

- Funciones de liberación de datos confiables: Ubicados básicamente en los niveles de sesión y una parte del de transporte, y se encargan de asegurar la liberación confiable de datos sin importar el origen ni el destino de los mismos.

- Servicios de usuario final: Se encuentran localizados en el nivel de presentación y constituyen una parte del de aplicación. Proporcionan los servicios que desea un usuario final. Dichos servicios se pueden resumir en compartición de archivos y compartición de impresoras.

A continuación se describen a detalle cada uno de los protocolos que forman el *stack* AppleTalk, para lo cual se hará uso de la Figura 2.6.b.

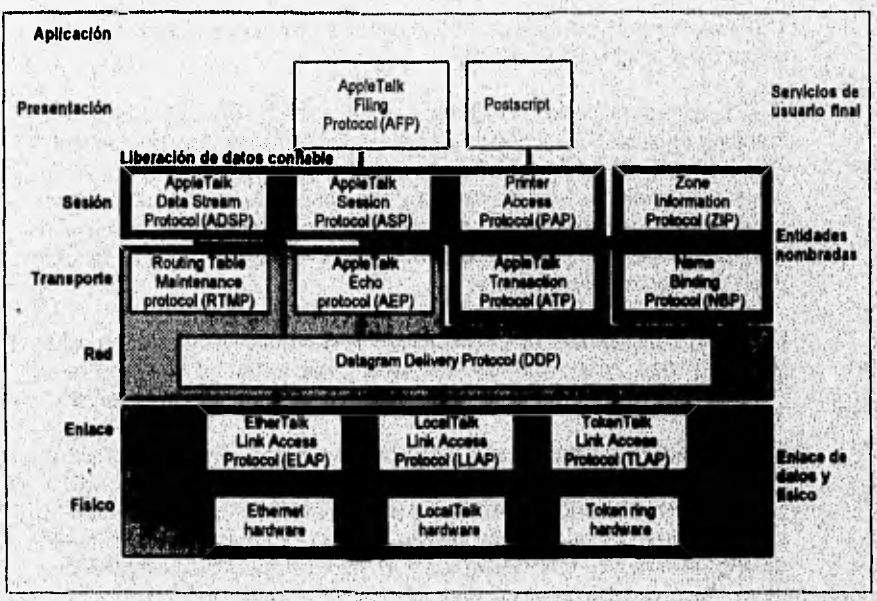


Fig. 2.6.a *Stack* de protocolos AppleTalk agrupado por funciones.

- Nivel Físico: Como se puede apreciar en la Figura 2.6.b, el nivel físico en el *stack* es el responsable del manejo del *hardware* de la red. Debido a que AppleTalk se basa en el modelo OSI, la arquitectura AppleTalk puede utilizarse en estándares de red como Ethernet y Token Ring, sin embargo AppleTalk también ha definido su propio *hardware* de red llamado LocalTalk, el cual utiliza como interface un bus síncrono RS-422A con velocidad de transmisión de 230,400 bps.

- Nivel de Enlace: Como se sabe el nivel de enlace es el encargado de llevar a cabo la interface del sistema con el *hardware* de la red. En este nivel se encuentran los Protocolos de Acceso al Enlace (*Link Access Protocols*, LAPs). En sus inicios Apple llamó Protocolos de Acceso de Enlace de AppleTalk (*AppleTalk Link Access Protocol*, ALAP) a su protocolo del nivel de enlace, con el paso del tiempo éste fue cambiado a Protocolo de Acceso de Enlace Local (*Local Link Access Protocol*, LLAP). Para una interface Ethernet, este protocolo es llamado Protocolo de Acceso de Enlace de Ethernet (*EtherTalk Link Access Protocol*, ELAP), mientras que para Token Ring es llamado Protocolo de Acceso de Enlace de Token Ring (*Token Link Access Protocol*, TLAP).

  Los LAPs manejan la asignación dinámica de ID (identificador) de nodo llamada *dynamic node ID assignament*, lo cual significa que cada nodo toma un ID en el momento que el nodo empieza a trabajar en la red. Debido a lo anterior, cada nodo de la red puede tener un ID distinto cada vez que éste se conecta a la red.

  LLAP es un protocolo del tipo CSMA/CD, lo cual significa que evita las colisiones por medio de contensión en la red.

  LLAP puede escoger entre enviar paquetes de información a un solo nodo en una "transmisión directa" o a todos los nodos en una "transmisión radial" (*broadcast*), en ambos casos LLAP envía un paquete de *Request-to-send* (lapRTS) al nodo destino. Si la transmisión es directa, LLAP espera un paque de respuesta *Clear-to-send* (lapCTS) del nodo destino; en caso de una transmisión radial, el nodo fuente sólo espera un tiempo máximo de 200 ms, conocido como *interframe gap* (IFG), y entonces envía el paquete.

  Los LAPs para Ethernet y Token Ring tienen el mismo funcionamiento que LLAP y los tres hacen que los niveles superiores (nivel de red hacia arriba) sean independientes del *hardware* utilizado, sea Ethernet, Token Ring o AppleTalk.

- Nivel de Red: En el Stack AppleTalk sólo hay un protocolo en el nivel de red, el Protocolo de Liberación de Datagrama (*Datagram Delivery Protocol*, DDP). Este protocolo es el encargado de realizar la comunicación entre dos *sockets*, toma los datos del nivel de transporte y los convierte en datagramas a 536 bytes de longitud e incluye un *checksum* para que el nodo destino verifique la integridad de los datos, con esto se cumplen las funciones del nivel de red.

  Cada paquete DDP contiene un conteo de salto, el cual le permite direccionar el número de ruteadores por los cuales el paquete viaja, del nodo destino al nodo

fuente. AppleTalk define como máximo conteo de salto 15.

- Nivel de Transporte: En este nivel se encuentran cuatro protocolos, dos de ellos en el grupo de flujo de datos *peer-to-peer* (Figura 6.2.a), uno en el grupo de liberación de datos confiable y el otro en las entidades nombradas. A continuación se describen cada uno de ellos:

  - Protocolo de Mantenimiento de la Tabla de Ruteo (*Routing Table Maintenance Protocol*, RTMP): Contiene la información de direccionamiento y conexión entre redes. Define las reglas para el intercambio de información entre ruteadores. Las tablas de ruteo contiene una entrada para cada red que pueda ser alcanzada por un datagrama dentro de los 15 saltos mencionados en el nivel de red.

    La información que se mantiene en las tablas de ruteo es: el rango de red, la distancia en saltos a la red destino, el número de puerto de la red destino, el ID de nodo del próximo ruteador o "entrada de estado", y el *status* de cada puerto.

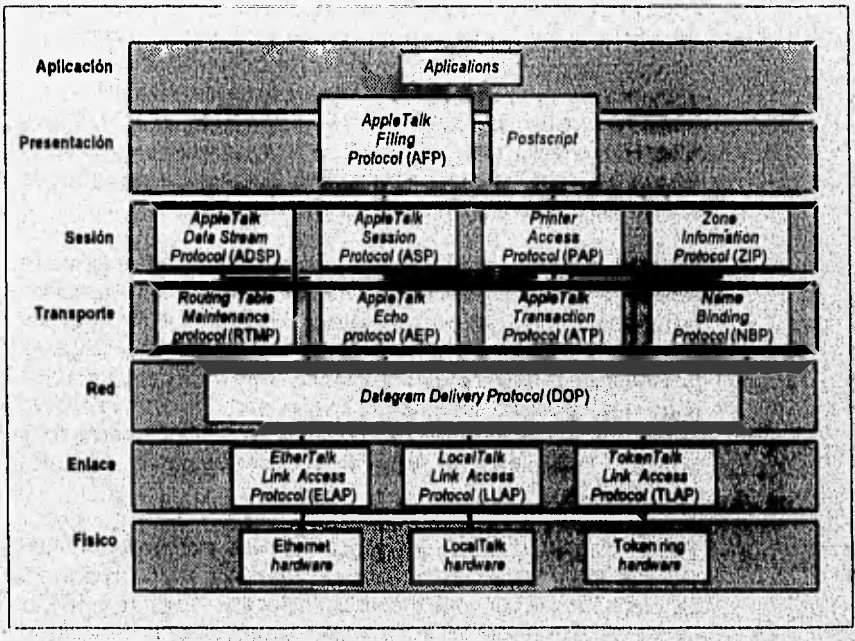


Figura 2.6.b *Stack* AppleTalk en el modelo de referencia OSI.

En el presente trabajo no se ahondará en la información, ya que no es el objetivo del mismo, sólo se debe considerar que dicha información es la necesaria para rutear los datagramas a lo largo de una red.

- Protocolo de Eco de AppleTalk (*AppleTalk Echo Protocol*, AEP): Este protocolo es el encargado de hacer que todo datagrama enviado tenga un regreso del nodo destino al emisor, lo cual es conocido como "eco". Para que suceda esto, cada socket está dotado de un generador de eco.

  AEP maneja dos tipos de paquetes de información: el requerimiento de eco o *echo request* y la respuesta de eco o *echo replay*.

- Protocolo de Transacción AppleTalk (*AppleTalk Transaction Protocol*, ATP): Este protocolo es el encargado de asegurarse que los paquetes lleguen a su destino sin pérdida alguna; en otras palabras, cada vez que a ATP se le solicita para enviar un paquete, lo que se le conoce como Requerimiento de Transacción (*Transaction Request* TReq), el *socket* destino debe enviar un reporte, al que se le conoce como Transacción de Respuesta (*Transaction Response*, TResp), así el ATP establece la comunicación entre dos *sockets*.

  Al tratar de establecer la comunicación entre *sockets*, ATP puede toparse con errores como; que un requerimiento de transacción se pierda en la red, que una respuesta de transacción se pierda o se retrase en la red, o que el *socket* que responde se vuelva inalcanzable. Para determinar cuando ocurre uno de estos errores ATP utiliza un *timer*, si éste expira antes que se reciba una respuesta, ATP retransmite su requerimiento original. ATP continúa retransmitiendo hasta que se reciba una respuesta o hasta que se alcanza el máximo de intentos establecido.

- Protocolo de Liga de Nombre (*Name-Binding Protocol*, NBP): Como se sabe, la red para mantener las comunicaciones opera con direcciones numéricas de inter-red, pero los usuarios de AppleTalk trabajan con entidades nombradas (*named entities*). AppleTalk se encarga de representar de manera interna cualquier entidad nombrada, las cuales pueden ser servicios tales como un servidor de archivos o un dispositivo de red, como una *Mac*. El NBP realiza las translaciones necesarias entre la dirección numérica de inter-red y los nombres de entidades alfanuméricas.

  El NBP mantiene una tabla que mapea los nodos (direcciones de inter-red) y las entidades nombradas (nombre del *socket* cliente), la cual reside en un nodo. Por medio de esta tabla, un *socket* que requiera un servicio, tendrá primero que consultar la localidad de dicho servicio.

• Nivel de Sesión: Como en el nivel de transporte, el nivel de sesión contiene cuatro protocolos, tres de estos protocolos forman parte del grupo de liberación de datos confiable, mientras que el último forma parte del grupo de entidades nombradas. A continuación se describen cada uno de ellos:

- Protocolo de Flujo de Datos AppleTalk (AppleTalk data Stream Protocol, ADSP). Es el encargado de la liberación de una transmisión de datos entre dos nodos.

ADSP , a diferencia de ATP, permite transmisiones *full-duplex*, lo que significa que la transmisión entre dos computadoras pueden ocurrir en ambas direcciones y en el mismo instante. ADSP también incluye control de flujo, de tal manera que un emisor rápido no podrá abrumar a un receptor lento. En una transmisión así, el nodo receptor reporta la cantidad de *buffer* disponible al nodo destino para de esta manera, negociar la transmisión.

ADSP se encarga de asegurarse de que la secuencia de los paquetes recibidos sea la correcta. Todos los paquetes de datos ADSP contiene un número que identifica la secuencia de transmisión, el nodo receptor compara este número con su contador interno y si los números coinciden el paquete es aceptado, en caso contrario, el paquete se rechaza.

- Protocolo de Sesión AppleTalk (*AppleTalk Session Protocol*, ASP). Este protocolo es el encargado de realizar la comunicación entre una estación de trabajo y un *server*, asegurando que el orden en que los comandos sean enviados por la estación de trabajo sea el mismo en que se reciben los resultados.

ASP se vale de dos protocolos del nivel de transporte para realizar su trabajo, el NBP, para obtener la dirección del *socket* que maneja la sesión del servidor, y ATP para proveerse del servicio de transporte para sus paquetes.

ASP ejecuta cuatro procesos: abrir la sesión, cerrar la sesión, "escoltar" a los comandos desde la estación de trabajo hasta el servidor y regresar las respuestas del mismo, y la admistración de la sesión.

- Protocolo de Acceso a la Impresora (*Printer Access Protocol*, PAP). Es el encargado de mantener la comunicación entre las estaciones de trabajo y una impresora. Debido a ello, las funciones del PAP son la iniciación y mantenimiento de una conexión, así como la transferencia de los datos.

PAP también se vale del NBP para encontrar las direcciones de las entidades nombradas, así como de ATP para el envío de los datos.

PAP cubre cinco procesos básicos: apertura de una conexión, transferencia de datos, cerrado de una conexión, determinación del estado del servicio de impresión y filtrado de requerimientos duplicados.

- Protocolo de Información de Zona (*Zone Information Protocol*, ZIP). Una característica que presenta AppleTalk es el agrupamiento de redes en zonas. Dichos agrupamientos son presentados al usuario final como nombres. El ZIP en conjunción con el RTMP, ayudan a los ruteadores a mantener un mapeo de los números de las redes para organizarlos en zonas dentro de la inter-red.

  ZIP crea y mantiene una Tabla de Información de Zona (*Zone Information Table*, ZIT) en cada ruteador. Dicha tabla contiene números de redes y sus respectivos nombres.

- Nivel de Presentación: Dos protocolos forman parte de este nivel: el Protocolo de Archivado AppleTalk (*AppleTalk Filing Protocol*, AFP) y PostScript. El AFP es el encargado de proveer acceso remoto a archivos en la red, mientras que PostScript es utilizado por gran variedad de redes para la definición de páginas en las impresoras.

  AFP trabaja tanto con estaciones de trabajo Machintosh como con las que no lo son. AFP emplea un traductor de AFP para convertir comandos de sistemas de archivos nativos en llamadas AFP, para que puedan ser entendidas por el *server*.

  AFP se vale de ASP para abrir y mantener una sesión entre una estación de trabajo y el servidor.

- Nivel de Aplicación: AppleTalk no tiene definidos protocolos en este nivel, en realidad muy pocos sistemas lo hacen.

En sus inicios de Apple en el campo de las redes, sus diseñadores definieron protocolos para redes pequeñas que podían ser fácil de instalar y mantener, AppleTalk se volvió muy popular pero con el paso del tiempo una gran variedad de problemas comenzaron a surgir en las grandes redes con múltiples ruteadores e incorporación de *backbones*, además del uso creciente de Ethernet. Para resolver los inconvenientes del AppleTalk original, Apple liberó AppleTalk fase 2 en 1989, con un rediseño completo de sus protocolos. Dentro de las novedades más importantes que AppleTalk presenta en su fase 2 podemos mencionar las siguientes:

- Introducción del Protocolo de Ruteo Basado-en-Actualización AppleTalk (*AppleTalk Update-Based Routing Protocol*, AURP), el cual fue diseñado para facilitar a los usuarios la conexión de LANs en redes de área amplia, reduciendo las actualizaciones de la tabla de ruteo sobre el enlace WAN.

- En los niveles físico y de enlace Apple desarrolló un nuevo protocolo para usarse en enlaces de telecomunicaciones de usuarios remotos, el Protocolo de Acceso Remoto de Apple (*Apple Remote Access Protocol*, ARAP).

- Soporte al Protocolo Simple de Administración de Red (*Simple Network Management Protocol*, SNMP), el cual se verá ampliamente en secciones posteriores.

2.7 DECNET

La empresa Digital Equipment Corporation (DEC) emplea a DECnet para comunicar sus equipos con los de los demás fabricantes. Al igual que las formas de comunicación vistas en los puntos anteriores, DECnet también se define por una arquitectura de *stack* de protocolos llamada Arquitectura de Red Digital (*Digital Network Arquitecture,* DNA).

La DNA ha tenido una evolución en la manera en la que realiza la interconexión de los elementos de una red de computadoras, dicha evolución se ha dividido en fases: Fase I (1976), Fase II (1978), Fase III (1980), Fase IV (1984) y actualmente Fase V. En la Fase IV se incorporan características importantes, tales como, la comunicación con LANs mediante Ethernet, redes WAN, servidores de comunicaciones y *gateways* con Arquitectura de Sistema de Red (*System Network Architecture*, SNA) de IBM.

En la Fase V de la DNA se integran completamente los niveles del OSI con los del DNA, de esta manera DECnet forma la comunicación con el modelo de la fig. 2.7.a.

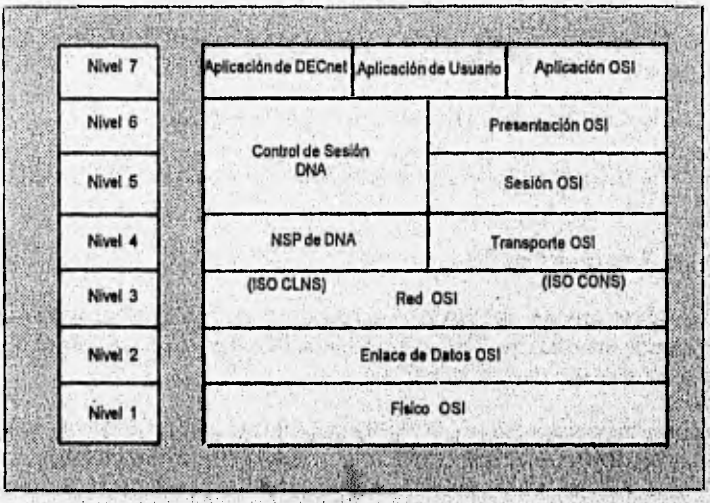


Figura 2.7.a Modelo de Referencia de la Fase V del DNA.

De la figura anterior debemos hacer las siguientes consideraciones:

- Los niveles del 1 al 4 corresponden plenamente al modelo OSI.
- Los tres niveles superiores están divididos en los niveles propietarios del DNA y los propietarios del OSI.

- Las aplicaciones sobre los mismos equipos DEC usan el Nivel de Control de Sesión de DNA. Los protocolos de aplicación de OSI proveen funciones de los niveles de aplicación, presentación y sesión de OSI, tales como el Acceso y Administración de Transferencia de Archivos (*File Transfer Access and Management*, FTAM) y el Elemento de Servicio para el Control de la Asociación (*Association Control Service Element*, ACSE).

Nivel de Control de Sesión de DNA

El nivel de control de sesión provee al usuario de DNA las siguientes funciones.

- Realiza la selección y resolución de direccionamiento. En la resolución de direccionamiento localiza las aplicaciones remotas sobre la red por medio de asignación de nombres, y mantiene información de las direcciones para los nodos locales. La selección de dirección homogeiniza a los protocolos de comunicación para asegurar la comunicación entre las aplicaciones del tipo *peer-to-peer*.

- Realiza un control de la conexión. Homogeiniza las conexiones con las aplicaciones que se reciben y comienza con el proceso de enviar y recibir datos.

- Provee transparencia a los protocolos de transporte.

- Le da soporte a interfaces de programación tales como $IPC y $QIO.

Nivel de Transporte de DNA

El nivel de transporte de DNA provee soporte al protocolo de transporte de OSI, al protocolo de Servicios de Red (*Network Services Protocol*, NSP) y servicios de *peer-to-peer* entre sistemas de comunicación.

El nivel de transporte asegura la transferencia de datos, cada protocolo de transporte provee el establecimiento de la conexión de transporte, el cual acarrea un flujo bidireccional del tráfico entre dos procesos sobre los mismos o diferentes sistemas. Una conexión de transporte es una conexión lógica temporal que normalmente existe hasta que uno de los procesos termina con la conexión. En este nivel se ofrece una interface al servicio de usuario en el nivel de sesión. Las aplicaciones de DNA pueden usar las interfaces de programación $IPC o $QIO para hacer una petición de conexión a través del nivel de Control de Sesión al nivel de transporte. Las aplicaciones de OSI puede hacer requisiciones de conexión al nivel de transporte a través del ACSE, del nivel de presentación o por medio del nivel de sesión. De esto podemos concluir que el nivel de transporte del DNA soporta ya sea el NSP de Decnet y los protocolos de transporte propietarios de OSI.

Nivel de Red de DNA

El nivel de red de la DNA provee compatibilidad con los estándares de ISO de los formatos de paquetes y el direccionamiento de red: ISO 8473 (protocolo de inter-red ó protocolo de nivel de red inactivo), ISO 8208 (protocolo del nivel de paquete de X.25), ISO 8878 (X.25 para proveer servicios de red orientados a la conexión), ISO 9542 (protocolo de intercambio en ruteadores ES-IS).

De acuerdo a estos estándares se permite que los sistemas DECnet funcionen en redes multivendedor e incrementen su capacidad de direccionamiento en redes grandes.

El nivel de red en la arquitectura de DECnet también provee:

- Inicialización al nivel de enlace de datos.
- Servicios de Red Orientado a la Conexión (*Connection-Oriented Network Service*, CONS). En este punto podemos encontrar que el CONS ofrece una conexión específica entre dos usuarios permitiéndoles realizar múltiples transferencias de datos sin tener la necesidad de especificar el destino en cada transferencia.
- Servicio de Red Sin Conexión (*Connectionless Network Service*, CLNS). En este punto se encuentra una completa compatibilidad con el direccionamiento de la Fase IV, además de tener la capacidad de autoconfiguración de los sistemas DECnet y conexión con redes LANs.

Nivel de enlace de datos de DNA

El nivel de enlace de datos de DECnet soporta actualmente a los protocolos de enlace de datos de la ISO y a los protocolos del mismo nivel de LANs. También mantiene una completa compatibilidad con la Fase IV tal como el Protocolo de Mensajes de Comunicación de Datos Digitales (*Digital Data Communications Message Protocol*, DDCMP) síncrono y asíncrono y el protocolo de enlace de datos de los niveles 2 y 3 de X.25.

Este nivel de DECnet provee una ruta de comunicación entre los sistemas conectados de manera directa sobre una red. Controla el movimiento de información entre sistemas, incluyendo la transmisión y recepción de datos. Para mandar los datos a un nodo adyacente, el nivel de enlace de datos realiza las siguientes funciones:

- Establecimiento del enlace (inicialización y condicionamiento de la línea).
- Recuperación y detección de errores.
- Control de la construcción del paquete de datos.
- Control del secuenciamiento de paquetes.

En la Fase V del DNA, el nivel de enlace utiliza los siguientes protocolos los cuales controlan la transmisión de datos a través de un medio determinado.

PROTOCOLO DE ENLACE DE DATOS	MEDIO
CSMA/CD	Conexión con todos los nodos
HDLC	Síncrono
DDCMP	Síncrono y Asíncrono
X.25 LAPB	Síncrono

El nivel de enlace es usualmente implementado en *hardware, firmware* y *software*. Las partes de *software* de este nivel son escritos parcialmente como *drives* de dispositivo de la plataforma VMS. Los protocolos HDLC Y DDCMP, son provistos por el componente WANDD de VMS. Asimismo el protocolo X.25 LAPB lo provee la misma VAX.

Nivel Físico de DNA

El nivel físico de DECnet proporciona control de las conexiones físicas que haya entre los sistemas interconectados. También da soporte a las interfaces CSMA/CD sobre redes LAN, usa un módulo de conexión por modem para soportar conexiones síncronas y asíncronas y continúa dando soporte a las interfaces físicas que existían en la fase IV (CSMA/CD, RS 232-C, RS 423, RS 449, CCITT X.25 y CI bus).

El nivel físico es responsable de la transmisión y recepción de datos sobre un medio que conecta a los sistemas. Mueve los datos transparentemente entre el sistema y las rutas de comunicación con los demás equipos. Además, puede incluir parte del *driver* del dispositivo de las interfaces del mismo, modems y líneas de comunicación.

2.8 SNA

La Arquitectura de Sistemas de Redes (*System Network Architecture*, SNA) fue desarrollada en la década de los 70´s y lanzada al mercado en 1974, la filosofía de SNA en un principio fue permitir un acceso remoto a las grandes computadoras *(mainframes)*.

La arquitectura SNA surgió antes que el modelo de referencia OSI, por lo que ambas propuestas presentan algunas diferencias entre sí, en las siguientes líneas se examinarán los siete niveles de SNA.

Como ya se mencionó, la primera versión de SNA salió en 1974 y se denominó SNA2, después en 1976 se anuncia la versión SNA3 o ACF/SNA, la siguiente versión fue anunciada en 1979 y se llamó SNA4 o ACF/SNA. Como podemos ver la arquitectura SNA es un producto dinámico, que con el tiempo mejora y que con cada liberación se agregan nuevas características. A continuación se revisará brevemente la estructura de la arquitectura de redes SNA.

Estructura de SNA

SNA está organizada en seis niveles o estratos, estos niveles pueden ser agrupados en tres niveles: Aplicación, Administración de Funciones y Transporte, como se muestra en la siguiente figura.

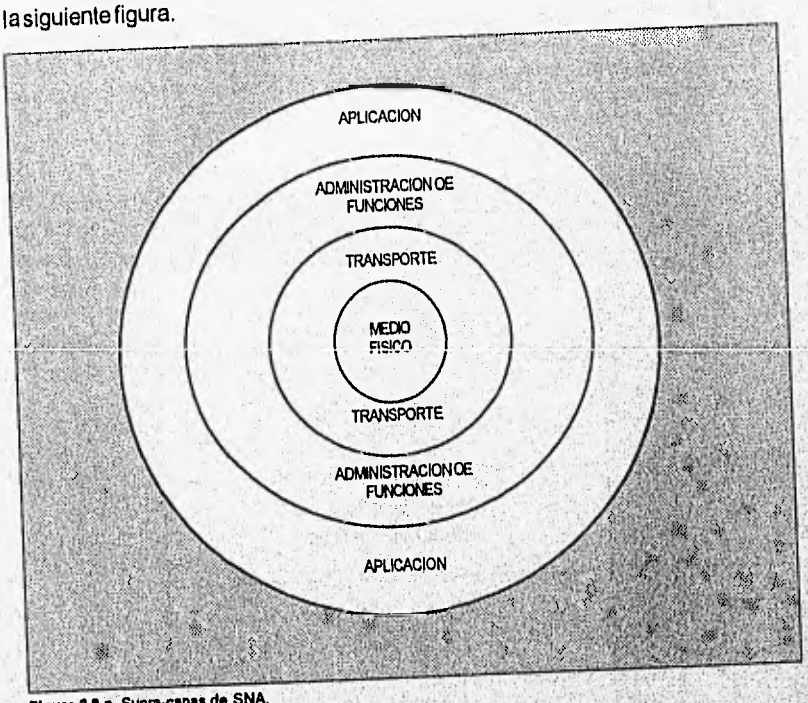


Figura 2.8.a Supra-capas de SNA.

La capa de aplicación es la encargada de recibir los datos provenientes de los programas de usuario. La capa de administración de funciones brinda los servicios de sesión entre las partes en comunicación. El subsistema de transporte se encarga de trasladar los datos. En cuanto al médio físico es el encargado de trasmitir los datos através del medio de comunicación. Desglosando estas tres supra-capas anteriores (el medio físico no se considera para este efecto) en los seis niveles nos queda la siguiente estructura:

- Capa de Control de Enlace de Datos (*Data Link Control*, DLC): Este nivel determina las reglas en las cuales se establece la comunicación entre dos nodos adyacentes. El protocolo que se maneja en este nivel es el denominado Control de Enlace de Datos Síncrono (*Synchronous Data Link Control*, SDLC) o un protocolo para el canal de E/S, en este nivel podemos encontrar un componente por cada línea de comunicación del nodo.

- Capa de Control de Encaminamiento (*Path Control*, PC): Este nivel tiene como objetivo el ruteo de los nodos, seleccionando la ruta más adecuada o más apropiada.

Dentro de este nivel se manejan dos tipos de rutas: ruta explícita y ruta virtual. En cada nodo, el Control de Encaminamiento se encarga de seleccionar el nodo al que se le enviarán los datos y el enlace que será usado y las direcciones de red, así como una tabla de ruteo.

Los mensajes son transmitidos con una longitud determinada, si el mensaje es mayor a esta longitud, entonces se fragmenta y se manda en partes. En la siguiente figura se puede apreciar este nivel, así como las demás capas que conforman SNA.

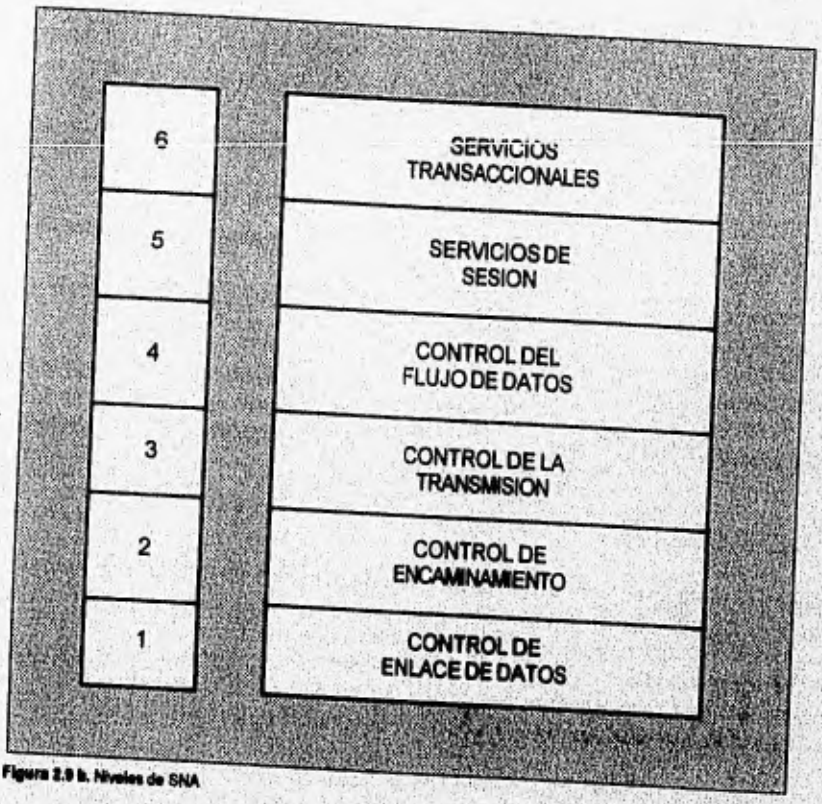


Figura 2.9 b. Niveles de SNA

- **Capa de Control de Transmisión (*Transmition Control*, TC): Este nivel administra la tasa de transferencia de mensajes, previendo sobrecargas y optimizando el uso de la línea de transmisión, también administra el secuenciamiento de los mensajes de datos, dentro de una sesión, rutea los mensajes hacia el usuario final o hacia otras capas de control (DFC y TC), el TC tiene la capacidad de cifrar o descifrar los datos si es que se especifica por el usuario, esto se le denomina criptografía a nivel de sesión.**

- Capa de Control de Flujo de Datos (*Data Flow Control*, DFC): Se encarga de mantener la integridad de los datos que son transmitidos en una sesión de comunicación, sin embargo, sus principales tareas son las siguientes:

  - Modos de envío/recepción
  - Encadenamiento
  - Tipos de respuesta

- Servicios de Sesiones: Estos servicios se encuentran en los Puntos de Control de los Servicios del Sistema (*Service System Contol Point*, SSCP), en las Unidades Lógicas (*Logic Units*, LU) y en las Unidades Físicas (*Physical Units*, PU). Los servicios que se ofrecen son los siguientes:

  - Servicios de Configuración: Este servicio es el responsable de controlar los recursos asociados con la configuración física de la red SNA, se incluyen la activación y desactivación de enlaces entre nodos. Permite el cambio de la configuración al administrador de la red.

  - Servicios al Administrador de la Red: Facilita las comunicaciones entre el SSCP y los operadores de la red. Provee de comandos para "arrancar" y detener la red SNA, activando y desactivando los recursos y llevando bitácoras de errores en los nodos.

  - Servicios de Sesión: Activan y desactivan sesiones cuando se solicita, transforma los nombres de los elementos que inician una sesión, en direcciones.

  - Servicios de Gerencia y Mantenimiento: Permite que un SSCP ejecute varias pruebas para determinar si un nodo o un enlace han fallado y la causa del problema. Lleva estadísticas de errores ocurridos en los nodos.

- Servicios Transaccionales: Este nivel se encarga de proveer los servicios usados en el intercambio de datos entre usuarios finales, provistos por las Unidades Lógicas. Estos servicios se dividen en dos y se enuncian en las siguientes líneas:

  - Servicios de Presentación: Definen el puerto de acceso a la red por usuario final en términos de requerimientos de traducción de códigos y comandos, formatos de pantalla, atributos de video, compactación de datos.

  - Servicios de Aplicación a Aplicación: Estos son servicios definidos para aquellas sesiones que vinculan dos sistemas de procesamiento transaccional. Son accedidos desde los programas de aplicación y permiten comunicaciones entre sí.

2.9 NUEVAS TECNOLOGIAS: FRAME RELAY y ATM

Sin lugar a dudas el avance más importante en el campo de las telecomunicaciones en los últimos años ha sido el desarrollo y la comercialización de las fibras ópticas. Con la fibra óptica se consiguieron tres mejoras importantes:

- Mayor distancia de transmisión: Al eliminar el ruido e interferencias al máximo.
- Mucho mayor velocidad de transmisión: Al utilizar como señales pulsos ópticos en lugar de pulsos eléctricos.
- Mucho mayor ancho de banda: Esto hace posible la transmisión de varias señales, incluso de distinto tipos (voz, datos, imagen, etc.).

Estas características acualmente son aprovechadas por las nuevas tecnologías. Dichas tecnologías están básicamente orientadas a las redes de área amplia, aunque cabe señalar que varias compañías están trabajando en la fabricación de productos con estas tecnologías para su operación en redes LANs.

No podemos hablar de estas tecnologías sin mencionar el concepto de RDSI (Red Digital de Servicios Integrados) la cual podemos definir como la red de área amplia o metropolitana con medios digitales de comunicación de nodo a nodo, y con un ancho de banda suficiente para transmitir información de distintas aplicaciones, como son voz, datos, imagen, etc., todas en la misma red. Las RDSI pueden ser de banda ancha BA-RDSI (B-ISDN sus siglas en inglés) o de banda estrecha BE-RDSI (ISDN sus siglas en inglés).

En la siguiente figura se esquematizan los protocolos de banda ancha y los de banda angosta. En la cual se incluyen tanto los recientes como los utilizados desde hace años.

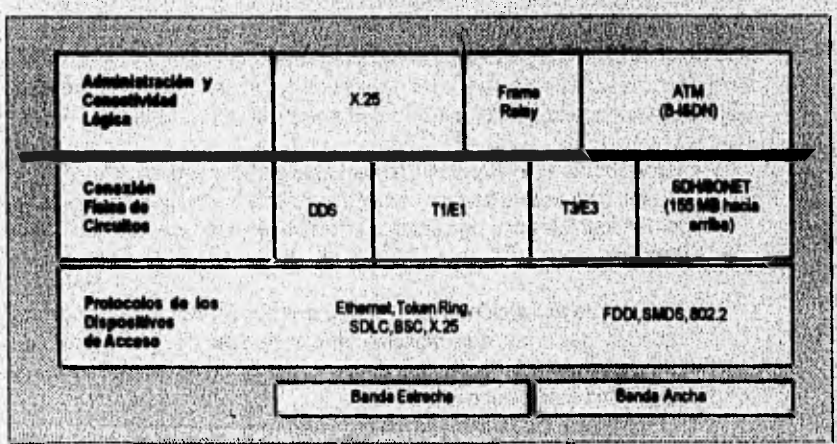


Figura 2.9.a Categorización de los protocolos de banda ancha y banda estrecha.

Protocolos como X.25, Frame Relay y Modo de Transferencia Asíncrono *(Asyncronous Transference Mode,* ATM), funcionan con conmutación de paquetes: protocolos como DDS, T1/E1, T3/E3 y Jerarquía de Datos Síncrono (*Syncronous Data Hyerarchy,* SDH)/ SONET, funcionan con conmutación de circuitos, finalmente los protocolos como Ethernet, Token ring, SDLC, BSC, X.25, FDDI, SMDS y 802.2 son los encargados del nivel físico.

En el presente punto sólo describiremos brevemente algunos, ya que sus aplicaciones están estrechamente ligadas a las redes telefónicas, con conceptos y términos de esta área, lo cual nos desviaría significativamente del propósito de este trabajo.

ATM

En 1990 CCITT en su recomendación para la B-ISDN designa a ATM como el modo de transferencia óptimo para hacer que la B-ISDN sea realidad. ATM utiliza una técnica de multiplexación y conmutación de paquetes de bajo retardo, y se encarga de subdividir los paquetes de información en unidades de longitud fija llamadas "células", las cuales tienen un encabezado que contiene una etiqueta para identificar la dirección del destino o el identificador de canal para el enrutamiento de la "célula". Debido a que las células son asignadas en función de la demanda y el volumen de información a transferir pueden "acomodarse" eficientemente en el canal de transmisión. Servicios de múltiples velocidades binarias en un enlace simple de transmisión de alta velocidad pueden ser suministrados.

ATM es una tecnologías *Cell Relay,* que por ser de banda ancha soportan aplicaciones de voz, datos y video.

Frame Relay

Es un protocolo que trabaja en nivel 2 a base de *frames* y cuyas características más sobresalientes son:

- No retransmisiones: A diferencia de otras tecnologías en que un paquete que llega con errores a su destino debe ser retransmitido, Frame Relay no pide retransmisiones.

- Menor procesamiento: Frame Relay no procesa errores ni integridad, por lo que una trama es aceptada como llegue.

Estas características hacen que Frame Relay sea un protocolo de alta velocidad, confiado a la seguridad de la fibra óptica; también hace que las tramas de información sea mucho más cortas al eliminar bits de chequeo de errores y de retransmisión. Esto se hace muy claro al comparar una trama de X.25, que consta de 27 bits, mientras que una trama Frame Relay se constituye de sólo 7.

Frame Relay permite conectar una LAN a una WAN Frame Relay como muestra la Figura 2.9.b.

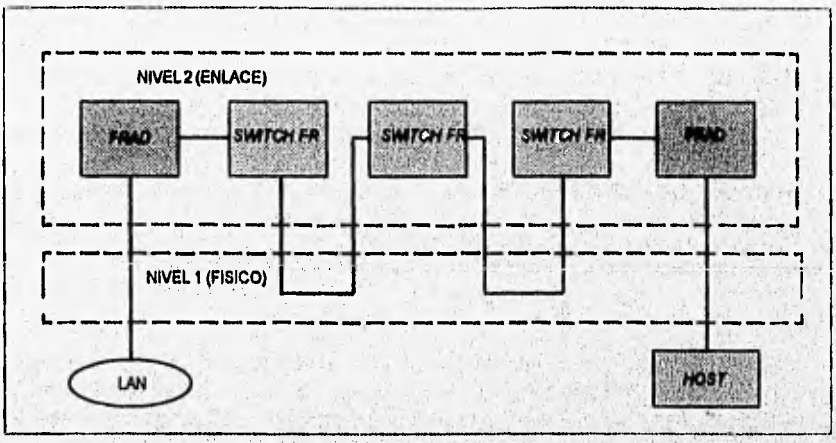


Figura 2.9.b Interconexión LAN-WAN Frame Relay.

En la Figura 2.9.b aparecen dos tipos de dispositivos:

- Dispositivo de Acceso Frame Relay (*Frame Relay Access Device*, FRAD): Son los dispositivos que conectan una LAN a una red Frame Relay, los cuales no tiene capacidad de conmutación y soportan un número limitado de conexiones. Pueden ser ruteadores, puentes o *brouters*.

- Switch FR (*Switch Frame Relay*): Soportan muchos circuitos físicos, tienen capacidad de conmutación y únicamente aceptan datos con formato Frame Relay.

CONCLUSIONES

En este capítulo se han establecido las bases primordiales para llevar a cabo el proceso de comunicación de los elementos que integran una red a través del modelo de referencia OSI.

El apartado de Clasificaciones de Protocolos permitió lograr un entendimiento de las características y comportamientos que presentan los diversos protocolos en función de sus métodos de comunicación. Ello sirvió como plataforma para que, junto con el apartado de *Stack* de Protocolos, se logrará la comprensión del funcionamiento general de los *stacks* de protocolos más representativos en el mercado: TCP/IP, Novell, Appletalk, DecNet y SNA, lo cual nos permite sentar las bases para el entendimiento de los protocolos orientados a la administración de red.

CAPITULO

3

MODELO OSI PARA LA ADMINISTRACION DE REDES (CMISE y CMIP)

3. MODELO OSI PARA LA ADMINISTRACION DE REDES (CMISE Y CMIP)

En secciones anteriores se han introducido conceptos y términos que son fundamentales en la administración de redes. En este capítulo nos enfocaremos al estándar OSI. En primer lugar abordaremos el proceso por el cual pasa un estándar en ISO para ser creado y posteriormente aprobado. Se analizará la estructura del modelo, así como los elementos que lo forman (protocolos, servicios y áreas de administración) además de la forma en que éstos interactúan para lograr una administración eficiente.

3.1 LA ARQUITECTURA Y EL PROCESO PARA LA CREACION DEL ESTANDAR OSI

El proceso por el cual pasa un estándar en ISO para ser creado y aprobado no es nada fácil, por el contrario es un proceso muy largo y complejo, en el cual el modelo es sometido a revisiones y votaciones constantes para su aprobación. En la siguiente figura se muestran las diferentes etapas que componen a este proceso.

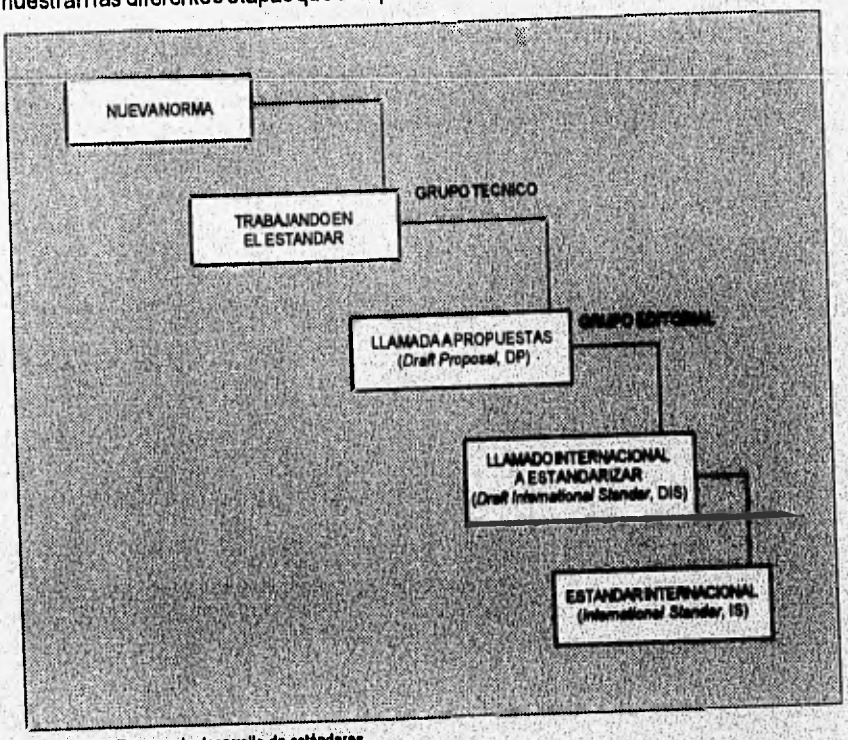


Figura 3.1.a Proceso de desarrollo de estándares.

En la parte superior de la figura podemos ver el cuadro que simboliza la llegada del problema a normalizar, el cual es tomado por un grupo técnico que se encarga de

trabajar en el nuevo estándar, siendo un grupo que labora en éste, reescribiéndolo y revisándolo una y otra vez, . Finalmente, cuando tiene una propuesta, ésta se pasa a un grupo editorial que lo analiza y lo registra. Una vez que esto termina, se convierte en una propuesta estándar y se somete a votación de un comité internacional normalizador; si la propuesta es aprobada, queda aceptada finalmente como un estándar más. Este proceso lleva de tres a cuatro años, lo cual lo hace un evento lento y complejo.

En cuanto al estándar de administración de redes de ISO, podemos decir que consta de cinco áreas funcionales para la administración, las cuales se examinarán en los apartados siguientes.

3.2 ORGANIZACION Y ESTRUCTURA DEL ESTANDAR

El estándar OSI para la administración de redes está formado por varios documentos que forman un conjunto de normas, y se encuentran concentrados en un documento denominado Estructura de Información de Administración (*Structure of Management Information* , SMI) en el cual se incluye información acerca del modelo de administración OSI. El modelo en este documento está dividido en cinco áreas funcionales. En la siguiente figura se muestra la estructura de la organización del estándar OSI para la administración de redes.

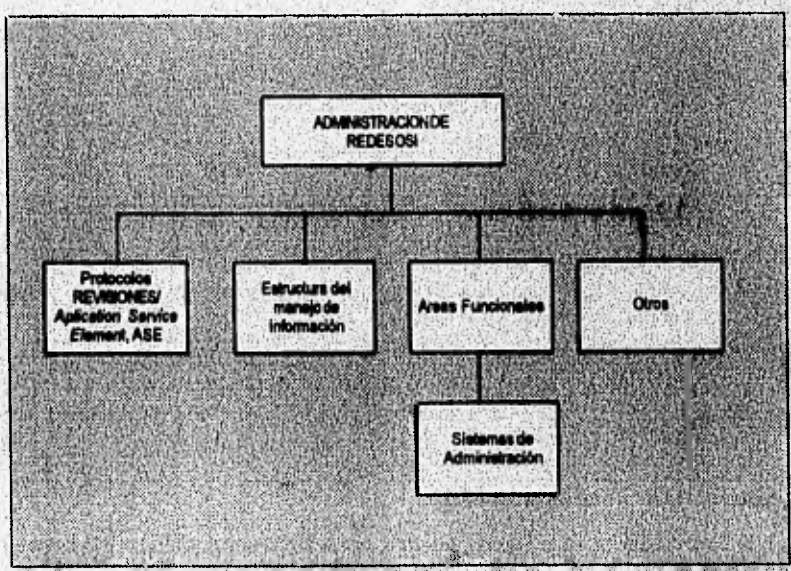


Figura 3.2.a La organización del estándar OSI para la administración de redes, Elementos de Servicio de Aplicación (*Aplication Service Element*, ASE).

En la Figura 3.2.b se muestran las cinco áreas funcionales en las que OSI divide su modelo de administración de redes.

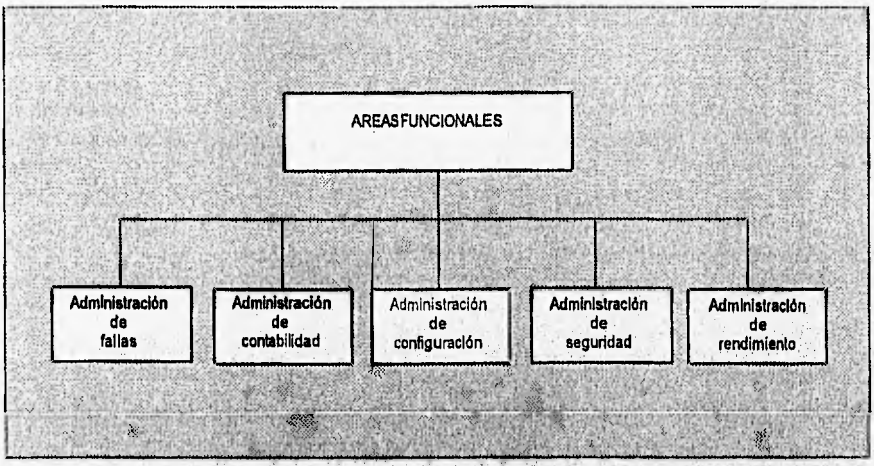


Figura 3.2.b Areas funcionales del modelo OSI de Administración de redes.

- **Administración de fallas: Esta área tiene como finalidad detectar, reparar, rastrear y corregir problemas, así como la posibilidad de hacer diagnósticos.**

- **Administración de contabilidad: Esta área se encarga de hacer un conteo de los recursos de la red que están siendo compartidos, así como de llevar un registro del nivel de utilización de recursos en la red.**

- **Administración de configuración: Esta área es empleada para identificar y controlar el manejo de objetos, es decir, se controla la inicialización, las operaciones y el cerrado de objetos, además de la reconfiguración de los mismos.**

- **Administración de seguridad: Su meta es la protección de los objetos manejados, esto se lleva a cabo por medio de reglas para validar procesos como el mantenimiento de rutinas de control de acceso y el manejo de autorizaciones, entre otras.**

- **Administración de rendimiento: La función de esta área es determinar si el sistema está siendo aprovechado eficientemente, lo cual se determina por medio de parámetros de rendimiento, como son: el tiempo de respuesta en la red, su nivel de carga y el *throughput*.**

Las cinco áreas funcionales de OSI se analizarán con más detalle en el apartado 3.10.

Como se mencionó antes, el modelo de administración es complejo y está formado por varias normas. En la siguiente figura se muestra como se encuentran ubicadas estas normas.

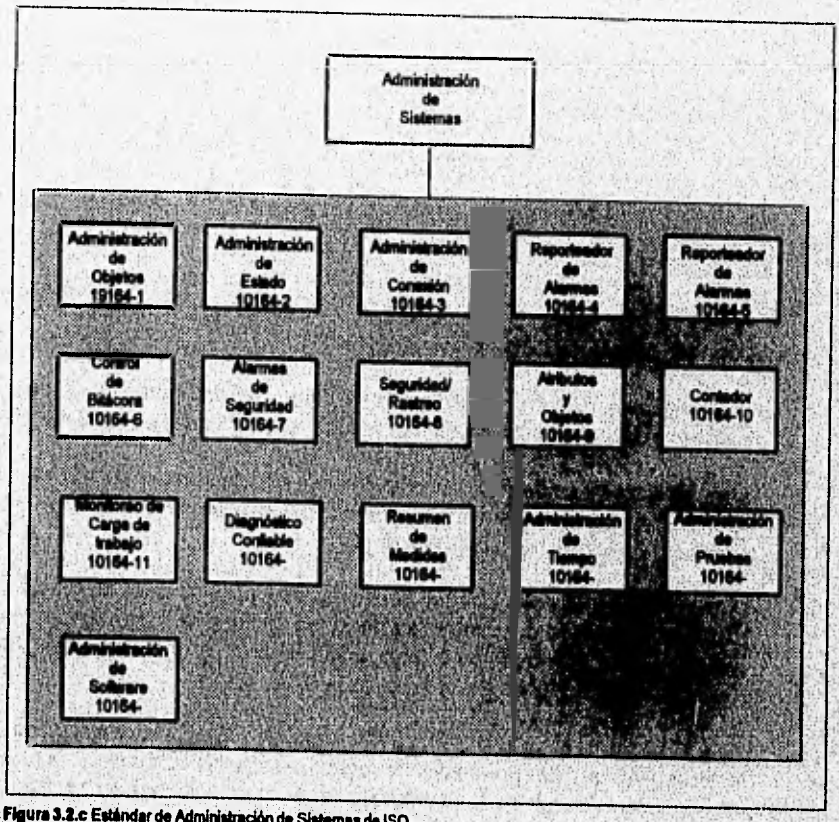


Figura 3.2.c Estándar de Administración de Sistemas de ISO.

Si se desea conocer más acerca de estos documentos, refiérase al apéndice A de este trabajo.

3.3 LOS SERVICIOS DE OSI EN LA ADMINISTRACION DE REDES

La ISO definió una serie de servicios proporcionados en un sistema de administración de red. Como se sabe, las comunicaciones de administración OSI requieren de un enlace orientado a la conexión (proporcionado en el nivel de transporte), además de confiabilidad en el ambiente del nivel de aplicación. En este nivel los agentes y administradores son vistos como aplicaciones iguales que utilizan los servicios de un Elemento de Servicio de Información de Administración Común (*Common Management Information Service Element*, CMISE) para intercambiar la información administrada. CMISE utiliza los servicios del Elemento de Servicio de Control y Asociación (*Association Control Service Element*, ACSE) y del Elemento de Servicio de Operaciones Remotas (*Remote Operations Service Element*, ROSE) para soportar dichos servicios. En general, la administración de red OSI se vale de algunas definiciones de servicios y protocolos en los niveles superiores; no se asiste de los niveles inferiores ya que es base fundamental del OSI el que la administración de un red no deba tomar en cuenta el cómo estos niveles proveen servicios a los niveles superiores. En los subsiguientes puntos trataremos algunos conceptos primordiales en la comprensión del estándar OSI, además se tratarán con mayor detalle a CMISE, ACSE y ROSE.

3.3.1 Comunicaciones Horizontal y Vertical

Las comunicaciones en un sistema OSI (Figura 3.3.a) pueden ser:

- Horizontales: Son comunicaciones que se dan entre el nivel de un nodo y el mismo nivel pero de otro nodo. En la Figura 3.3.a se puede apreciar que el nivel N+1 de la máquina A se comunica de une manera lógica con el nivel N+1 de la máquina B, estos niveles pueden ser el de una aplicación de alarma. Una comunicación semejante ocurre entre el nivel N de la máquina A y el nivel N de la máquina B.

- Verticales: Son aquellas que se dan entre niveles contiguos dentro de un mismo nodo, utilizando el paso de parámetros. En la Figura 3.3.a se aprecian las comunicaciones verticales entre los niveles N+1 y N, tanto en la máquina A como en la B.

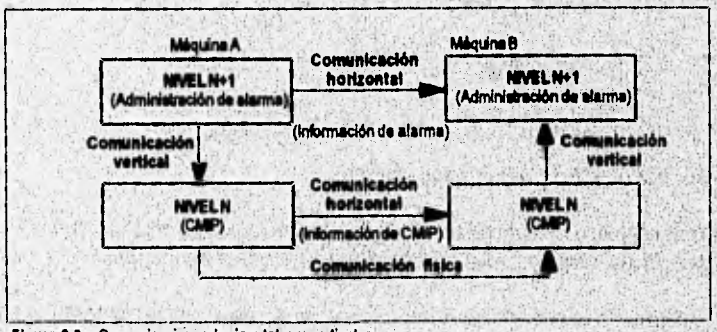


Figura 3.3.a Comunicaciones horizontales y verticales.

Como se puede deducir, toda comunicación horizontal, en realidad se lleva a cabo a través de comunicaciones verticales, debido al proceso de encapsulación y desencapsulación; ya que la comunicación entre dos nodos parte del nivel N+1 del nodo emisor y con una comunicación vertical llega su nivel físico para ser transmitido por el medio; cuando llega al nodo receptor, en él se lleva a cabo una comunicación vertical para llegar al nivel N+1 del mismo.

3.3.2 Componentes de las Comunicaciones Estratificadas

Las comunicaciones estratificadas constan de tres componentes básicos:

- Unidad de Servicio de Datos (*Service Data Unit*, SDU): Consiste de los datos del usuario y de la información de control que se crea en los niveles superiores, la cual se transfiere por el nivel N al N-1. El SDU se guarda en uno de los nodos finales de una comunicación horizontal.
- Información de Control de Protocolo (*Protocol Control Information*, PCI): Es la información que se intercambia entre los niveles iguales de distintos nodos. Son los *headers* y los *trailers*.
- Unidad de Datos de Protocolo (*Protocol Data Unit*, PDU): Es la combinación de un SDU y un PCI.

Para comprender mejor las ideas anteriores, en la Figura 3.3.b se muestra a un PDU en un nivel N, que pasa a un nivel N-1 para convertirse en un SDU, el mismo nivel N-1 se encarga de anexarle su *header* correspondiente (PCI), para convertirlo en un PDU, que para el nivel inmediato inferior será un SDU.

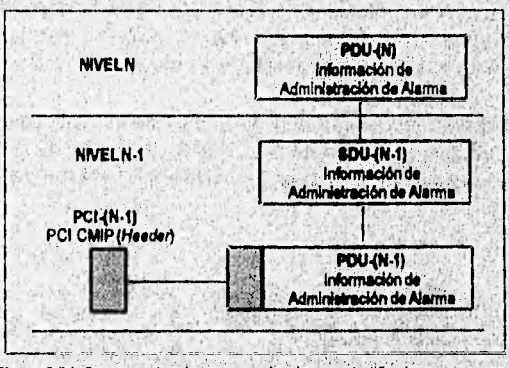


Figura 3.3.b Componentes de las comunicaciones estratificadas.

3.3.3 Encapsulación y desencapsulación

En el primer capítulo tratamos el concepto de encapsulación dentro del Diseño Orientado a Objetos, dicho concepto es diferente al que trataremos a continuación.

Los conceptos de encapsulación y desencapsulación están estrechamente ligados con el apartado anterior, como se verá a continuación:

- Encapsulación: Este proceso comienza desde el nivel de aplicación, en el cual a los datos a enviar se le anexa el PCI correspondiente, y los pasa al nivel inmediato inferior, dicho nivel los recibe como un paquete de datos más a tratar, sin conocer del PCI de nivel anterior (se dice entonces que el nivel recibe los datos encapsulados). Este proceso se repite de nivel en nivel hasta el nivel físico, en el cual no se coloca PCI alguno, y sólo se envían al nodo destino.

- Desencapsulación: Este proceso se lleva a cabo en el nodo que recibe la información. Los datos son recibidos por medio del nivel físico, para que sean tomados por el nivel de enlace, éste quita los PCI que se colocaron en nivel de enlace del nodo origen, y pasa los datos al nivel inmediato superior (se dice entonces que el nivel desencapsula los datos). Dicho proceso se repite hasta que los datos llegan al nivel de aplicación y éste los muestra al usuario final.

3.3.4 Comunicación entre niveles

Cuando un usuario desea un servicio, requiere hacerlo a través de la dirección o identificador de éste o bien a través de un Punto de Acceso al Servicio (*Service Access Point*, SAP). La comunicación entre usuarios y servicios se lleva a cabo por medio de primitivas, como son:

- Requerimiento: Esta primitiva es utilizada por el usuario para solicitar un servicio.

- Indicación: Es utilizada por el proveedor del servicio, ya sea para invocar una función, o para indicar que un función fue requerida en un SAP.

- Respuesta: Es empleada por el usuario del servicio como contestación a una indicación del requerimiento antes solicitado en el SAP.

- Confirmación: Se utiliza por el proveedor del servicio para completar una función antes solicitada en el SAP.

En la Figura 3.3.c se puede ver cómo se lleva a cabo la comunicación entre usuarios a través de las cuatro primitivas antes descritas.

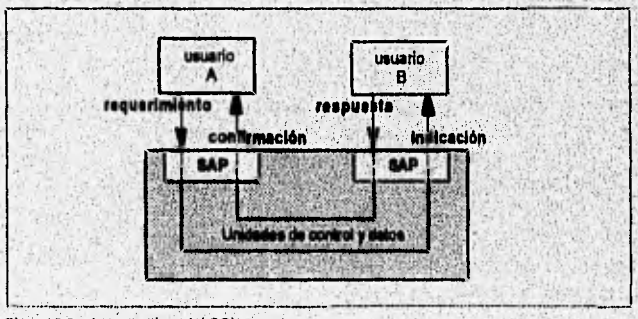


Figura 3.3.c Las primitivas del OSI.

Las primitivas manejan un formato especial. Por ejemplo, una primitiva de requerimiento sería:

N-CONNECT request (called address, calling address, quality of service parameters, user data)

Donde N se refiere a *Network* (red). La primitiva corresponde a un requerimiento de conexión, que como se puede observar, debe ir acompañada de los parámetros de direcciones: de quien llama y a quien se llama, calidad del servicio y los datos del usuario.

3.3.5 Definición de servicios y especificaciones de protocolo

Tanto la CCITT como la ISO se valen de dos términos básicos para la descripción de muchas de sus recomendaciones:

- Definiciones de servicios: Es la definición de los servicios proporcionados entre los niveles a través de las primitivas. Se refiere a los servicios que proporcionan comunicación vertical.

- Especificaciones de protocolo: Son las acciones tomadas en los niveles y entre niveles iguales a partir de las definiciones de los servicios. Se refiere a los servicios que proporcionan comunicación horizontal.

Las definiciones de servicio generalmente son mostradas por medio de diagramas de transición de tiempo. En la Figura 3.3.d se aprecia que una primitiva de requerimiento se utiliza antes de una primitiva de indicación. El uso de las primitivas de confirmación y respuesta está basado en las primitivas de requerimiento e indicación prematuras.

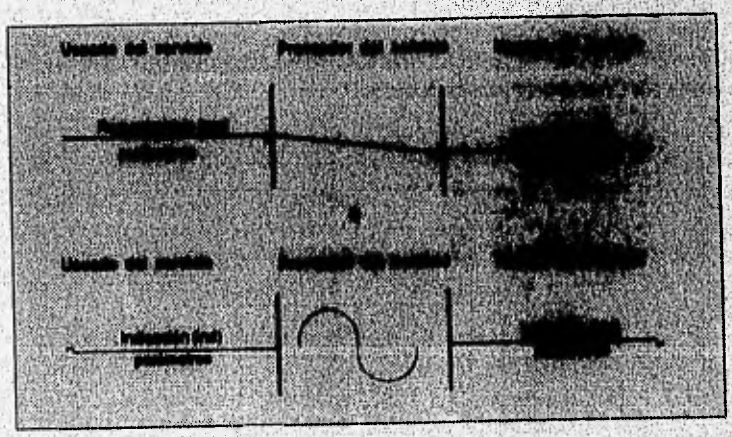


Figura 3.3.d Definiciones de servicio.

3.4 ESTRUCTURA DE OPERACIONES DE LA ADMINISTRACION OSI

Los Objetos Administrados o manejados (*Management Object* , MO) representan generalmente a entidades activas (ruteadores, *gateways*, etc), las cuales muestran un cierto comportamiento y tienen un estado a través del tiempo. Como tal, es necesario habilitar el control de las entidades representadas por los MOs mediante operaciones realizadas sobre los mismos. Puesto que los MOs están encapsulados, las operaciones realizadas sobre ellos solamente deberán hacerse mediante mensajes. Estos mensajes deberán contener todos los parámetros y la información necesaria para determinar la naturaleza de la operación, la manera en que ésta deberá ser llevada a cabo (por ejemplo, si es sincronizada) y las condiciones en las que se ejecute.

En forma similar, los resultados de una operación se pueden conocer solamente cuando el MO emite una indicación con el resultado obtenido. En general, se requiere que se indique si la operación fue exitosa o falló, los parámetros que intervengan en ello y las causas de la falla (si la hubo).

Por otra parte, a los atributos de los objetos administrados se les puede definir como una pieza de información o característica que describe parte de un objeto. El atributo tiene un valor, el cual está contenido en el objeto y puede ser modificado. Los cuatro tipos de atributos son:

- Identificación: Identifica de manera única a un elemento de la red.
- Características: Permite el control de los parámetros de un elemento de la red. En general, este atributo toma valores por omisión cuando el elemento es creado, sin embargo, después pueden ser modificados por la operación que se realice.
- Estado: Permite la inspección del estado actual de un elemento u objeto. A diferencia del atributo de característica, el estado puede cambiar sin que haya alguna operación de administración.
- Contador: Son valores que indican el número de veces que haya sido realizada una operación de administración por un objeto, o por la detección de una condición particular del mismo.

Las operaciones de administración de OSI están categorizadas en dos grandes grupos:

- Operaciones orientadas a los Atributos de los Objetos.
- Operaciones orientadas a los Objetos Enteros.

Antes de explicar este tipo de operaciones, es importante hacer notar que la operaciones orientadas a los atributos obedecen al principio de encapsulación.

Operaciones orientadas a los Atributos

Generalmente estas operaciones permiten la recuperación de los atributos de un objeto y están sujetas a restricciones de acuerdo a la seguridad y privilegios de acceso que se tengan. Dichas operaciones son las siguientes:

- *M-GET*: Recupera el valor de uno o más atributos.
- *M-SET* (con primitiva *replace*): Sobreescribe el valor de uno o más atributos con valores específicos.
- *M-SET* (con primitiva *set to default*): Sobreescribe los valores existentes de uno o más atributos con valores predefinidos por omisión.
- *M-SET* (con primitiva *add values*): Agrega un miembro a un conjunto de valores de atributo.
- *M-SET* (con primitiva *remove values*): Borra un miembro a un conjunto de valores de atributo.

Operaciones orientadas a un Objeto Entero

Esta segunda categoría de operaciones impacta al comportamiento completo de un objeto. Estas operaciones tienen las siguientes funciones:

- *M-CREATE*: Crea e inicia una clase de objeto que va a ser administrado.
- *M-DELETE*: Borra una clase de objeto administrado.
- *M-ACTION*: Especifica una acción para que sea realizada por un objeto.

Asimismo, al recibir una operación el objeto administrado normalmente emite una notificación del resultado de dicha operación. Es decir, como los MOs representan a elementos activos, los eventos internos y externos a los mismos deberán producir efectos en su comportamiento. De tal manera que estos eventos se pueden comunicar a sistemas de administración locales mediante notificaciones que produzcan los MOs.

3.5 LOS NIVELES SUPERIORES PARA LA ADMINISTRACION OSI

Es necesario tener un conocimiento de las operaciones de los niveles de Presentación y Sesión y de los varios elementos que intervienen en el nivel de Aplicación para poder comprender la filosofía de administración de redes que propone OSI. De hecho, la

administración de red se basa en los servicios de estos tres niveles. Pasemos a dar una revisión a estos niveles, comenzando con el nivel de Presentación, para tener un mejor entendimiento de la contribución de ellos a este tipo de administración.

3.5.1 Contribución del nivel de Presentación

El nivel de presentación realiza los servicios asociados con la descripción y presentación de la estructura de datos de un mensaje. Esto no concierne al significado (la semántica) de los datos, sino a cómo se identifica al tipo de los datos y cómo están acomodados los *bits* dentro de un campo de datos, a esto se le llama sintaxis. Con la ayuda del nivel de presentación, el centro de control de red puede decodificar fácilmente mensajes cuando éstos se encuentran estandarizados.

El nivel de presentación puede aceptar varios tipos de datos de una aplicación de un usuario (tales como enteros, cadenas ASCII, booleanos, etc.) y si es necesario, negociar con el otro nivel de presentación sobre cómo los datos serán presentados durante el proceso de comunicación. Para realizar estos servicios, los dos niveles de presentación se relacionan con:

- La sintaxis de los datos de la aplicación enviada.
- La sintaxis de los datos de la aplicación recibida.
- La sintaxis usada entre las entidades de presentación que soportan las aplicaciones enviadas y recibidas.

A este servicio lo llamaremos "sintaxis de transferencia", el cual se negocia entre las entidades de presentación. Cada entidad de presentación elige la mejor sintaxis para presentársela a la aplicación del usuario. Entonces intenta negociar el uso de esta sintaxis con la otra entidad del nivel de presentación. De esta manera, las dos entidades del nivel de presentación deberán estar de acuerdo con la sintaxis de transferencia, antes de que los datos sean intercambiados. Incluso, puede ocurrir que sea necesario para el nivel de presentación cambiar la sintaxis de los datos para que los usuarios se puedan comunicar.

La ISO y CCITT han desarrollado una sintaxis de presentación y transferencia, para que puedan ser usados por los protocolos del nivel de presentación. Tanto la norma ISO 8824 titulada Notación de Sintaxis Abstracta (*Abstract Syntax Notation*, ASN.1), como la ISO 8825 llamada Reglas de Clasificación Básicas (*Basic Encoding Rules*, BER) proveen un conjunto de reglas para desarrollar una descripción de los datos. ASN.1 describe una sintaxis abstracta para los tipos y valores de los datos, mientras que BER describe la presentación que deben mantener los datos durante el proceso de transferencia.

Si se desea profundizar en las estructuras que propone ASN.1, se puede consultar el apéndice B de este trabajo.

3.5.2 Contribución del nivel de Sesión

En este nivel se establece el cómo las aplicaciones pueden establecer una sesión (para el nivel de sesión, un protocolo de nivel superior como por ejemplo ACSE se considera como una aplicación), cómo se puede intercambiar y llevar un conteo del intercambio de datos mediante puntos de sincronización, cómo se pueden usar señales (*tokens*) para llevar a cabo un diálogo entre los nodos y finalmente, cómo se puede terminar con la conexión.

El nivel de sesión provee a la comunicación entre los nodos de los siguientes servicios.

- Coordinar el intercambio de datos entre aplicaciones por medio de conexiones lógicas y diálogos entre las mismas.
- Proveer puntos de sincronización (también llamados puntos de chequeo) para estructurar el intercambio de datos.
- Asignar una estructura sobre las interacciones que realicen las aplicaciones.
- Si es necesario, asignar un orden para que las aplicaciones tomen su turno en el intercambio.
- Usa un punto de sincronización para asegurar que todos los datos hayan sido recibidos por una aplicación, antes de que se concluya con la sesión.

Los elementos que intervienen en la operación del nivel de sesión en una administración OSI son los siguientes:

- *Tokens*: Los *tokens* dan el derecho exclusivo de usar un servicio, el cual se asigna dinámicamente a una aplicación, a la vez de permitírsele utilizar ciertos servicios. Por ejemplo, la aplicación de un nodo puede pedir mediante un *token*, que la aplicación de otro nodo le proporcione uno o más servicios que contenga el segundo nodo.
- Servicios de Sincronización: Se utilizan para coordinar el intercambio de los datos entre los nodos. Estos servicios son puntos de chequeo en el comienzo de una trasferencia, por ejemplo de un archivo. Los puntos de chequeo se usan junto con las siguientes unidades y actividades de diálogo:
  - Punto mayor de sincronización: En este punto se estructura el intercambio de datos en una serie de unidades de diálogo. La recepción de cada uno de estos puntos deberá confirmarse por la aplicación receptora.

- Unidad de diálogo: Es la unidad de comunicación, el punto mayor de sincronización establece el comienzo y el final de esta unidad.

- Punto menor de sincronización: Este punto estructura el intercambio de los datos dentro de una unidad de diálogo. Es más flexible que el punto mayor, puesto que no es necesario que se obtenga una confirmación de recepción de los datos.

- Actividad: Es un conjunto lógico de tareas que consiste de una o más unidades de diálogo.

• Reglas de la actividad: El nivel de sesión restringe a la conexión a una sola actividad a la vez, sin embargo, en una sesión varias actividades consecutivas pueden ser solicitadas.

• Unidad funcional: Se utiliza para definir un grupo de servicios ofrecidos por el nivel. Este grupo consiste actualmente de 12 unidades:

- *Kernel*: Provee 5 servicios no negociables.
- *Half-duplex*: Comunicación en dos sentidos, uno a la vez.
- *Full-duplex*: Comunicación en dos sentidos, al mismo tiempo.
- Tipo de datos: Transfiere datos sin restricción de *tokens*.
- Excepciones: Reporta situaciones especiales en la comunicación.
- Desconexión: Terminación de una sesión por medio de *tokens*.
- Sincronización menor: Invoca al punto menor de sincronización.
- Sincronización mayor: Invoca al punto mayor de sincronización.
- Resincronización: "Resetea" la conexión y la restablece.
- Datos expeditos: Transfiere datos que están libres de restricciones, por medio de los *tokens*.
- Actividad de administración: Provee varias funciones dentro de una actividad.
- Capacidad de intercambio: Intercambia una cantidad limitada de datos mientras no se encuentren en una de las actividades anteriores.

3.5.3 Contribución del nivel de Aplicación

En este apartado se identifica la intervención de varios servicios que le dan soporte a la administración de red de OSI, dichos servicios y sus relaciones se muestran en la Figura 3.5.a.

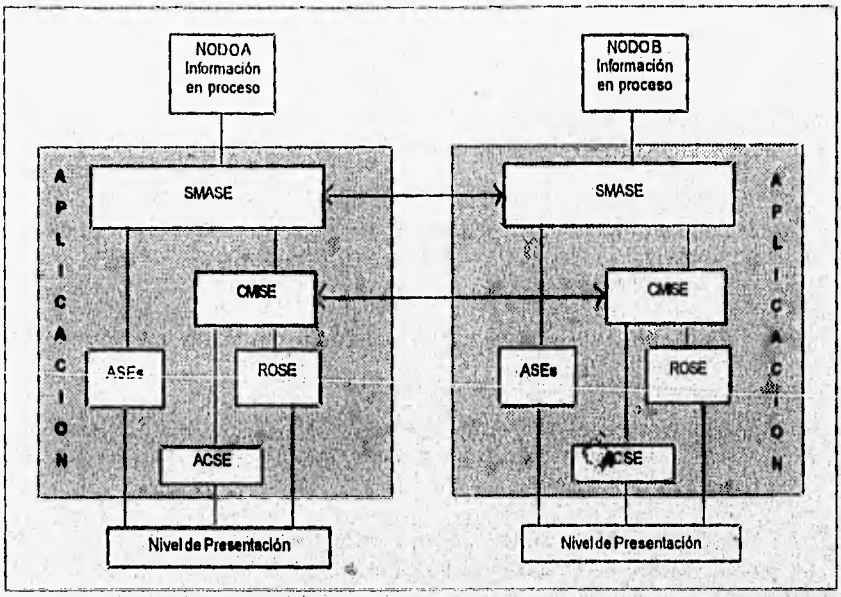


Fig. 3.6.a El nivel de aplicación y los servicios en la administración de red OSI.

Más adelante se enfatizará la descripción de estos servicios, principalmente en el ACSE y en el ROSE, puesto que la administración de red OSI basa su operación en ellos. Por otro lado el Elemento de Servicio de Aplicación (*Application Service Element*, ASE) define a un conjunto de servicios que son necesarios para las aplicaciones. En esencia, los ASEs son subrutinas de *software* que han sido desarrolladas por la ISO para que los programas de administración de red que se escriban, aprovechen estas subrutinas y las vean como una biblioteca de funciones orientadas a la administración. Así mismo, los Elementos de Servicio de Aplicación del Sistema de Administración (*System Management Application Service Elements*, SMASEs) integran propiamente al sistema que dirige las operaciones de administración. CMISE será descrito ampliamente en un apartado posterior.

Elemento de Servicio para el Control de la Asociación (ACSE)

El ACSE provee de varias funciones básicas e importantes a la Entidad de Aplicación (*Application Entity*, AE). Proporciona servicios necesarios entre las aplicaciones que son independientes de cualquier necesidad específica de las mismas. En otras palabras, le da servicios comunes a todas las aplicaciones. Por lo cual, la principal tarea de ACSE es realizar los siguientes cuatro servicios:

- *A-ASSOCIATE*: Instala las asociaciones entre las aplicaciones de dos nodos, valiéndose de los procedimientos del ASE, los cuales están identificados por un nombre. Este servicio se describirá más ampliamente en el apartado de CMIP.

- *A-RELEASE*: Realiza un borrado de las asociaciones que haya entre los elementos.

- *A-ABORT*: Este servicio ocasiona la terminación de las conexiones en los niveles de aplicación, presentación y sesión. Dicho servicio lo inicia la aplicación del usuario.

- *A-P-ABORT*: Tiene la misma función de *A-ABORT*, sin embargo, éste, a diferencia del primero, se inicia por el nivel de presentación.

Para ilustrar lo anterior, suponga que existen dos aplicaciones ejecutándose en dos nodos distintos y no se ha establecido una asociación entre ellos. Una aplicación intenta enviar datos erróneos -posiblemente por problemas en el código- hacia la otra aplicación. ACSE no permitirá que los datos se transfieran, sino que solicitará que se establezca una asociación entre dichas aplicaciones antes de que la transferencia pueda tomar lugar. Además, haciendo uso de ACSE se tiene la posibilidad de rechazar las peticiones que hagan las aplicaciones cuando éstas no estén conectadas por una asociación.

En ASN.1, todas las asociaciones se identifican como *OBJECT IDENTIFIER*. Finalmente, el control de la asociación identifica la sintaxis abstracta de los PDUs de la aplicación por medio del *OBJECT IDENTIFIER*, ya que pudieron ser acarreados por el resultado del uso de ASEs.

Elemento de Servicio para Operaciones Remotas (ROSE)

En el proceso de comunicación de aplicaciones, es probable que éstas provengan de sistemas (nodos) distintos, por lo cual es posible que se involucren operaciones remotas, de tal manera que la ISO y la CCITT establecieron estándares para este tipo de operaciones agrupándolas en el ROSE.

El ROSE permite dos modos de operación:

- Modo síncrono: El cual requiere que la aplicación invocadora establezca comunicación con la aplicación contestadora antes de realizar alguna operación remota.

- Modo asíncrono: En donde se permite a la aplicación invocadora llamar a cualquier operación remota, sin la necesidad de que la aplicación contestadora le avise si ha realizado las operaciones o no.

ROSE en su operación, puede hacer uso de los servicios de ACSE. Sin embargo, él mismo contiene cinco servicios para que las aplicaciones puedan interactuar con dicho elemento de operaciones remotas:

- *RO-INVOKE*: Este servicio le permite a la aplicación que invoca a ROSE hacer una petición de operación. Este servicio comienza cuando ROSE recibe una solicitud de un usuario (el invocador) para que se ejecute una operación, dicho servicio se clasifica como no confirmado.

- *RO-RESULT*: Este servicio también es no confirmado. Lo utiliza el usuario de ROSE para notificar a la aplicación que generó *RO-INVOKE* que la operación se realizó completamente.

- *RO-ERROR*: Su propósito es notificar a la aplicación que generó *RO-INVOKE* que la operación no se pudo completar. Este servicio es del tipo no confirmado.

- *RO-EJECT-U*: Su propósito es volver a invocar los servicios *RO-INVOKE*, *RO-RESULT* o *RO-ERROR*, si es necesario. Asimismo, este servicio tiene la clasificación de no confirmado.

- *RO-EJECT-P*: Se emplea para informar a la aplicación usuaria de ROSE acerca de la ocurrencia de un problema.

Por otra parte, ROSE emplea cuatro macros en su operación, las cuales están definidas en el documento 9072-1 de ISO. A continuación se presenta un breve resumen de ellas:

- *BIND*: Permite la especificación de los tipos de datos que se intercambiarán cuando las aplicaciones se encuentren asociadas o comunicadas.

- *UNBIND*: Permite la especificación de los tipos de datos para que se termine con la asociación de las aplicaciones.

- *OPERATION*: El tipo de notación en esta macro, permite la especificación de una operación o de operaciones "hijas" que resulten de la operación "padre".

- *ERROR*: Permite la operación en donde se especifique una respuesta negativa de las aplicaciones.

3.6 ADMINISTRACION DE LA MIB (*Management Information Base*)

La MIB es una base de datos que almacena información relacionada a los objetos administrados. Aunque tanto las MIBs de los modelos de la ISO, de Internet y de la IEEE, almacenan la información en una estructura de árbol, la MIB de la ISO utiliza un Arbol de Información de Administración (*Management Information Tree*, MIT) dinámico. Por ello, en este apartado nos referiremos a la MIB de la ISO como MIT.

En un sistema de administración de red cada agente administra una MIT. La MIT modela a un grupo de MOs para representar LANs, interfaces o modificar MOs en la MIT, invocar acciones o recibir notificaciones de eventos.

La MIT contiene instancias de MOs, cada instancia incluye atributos que sirven como su Nombre Distintivo Relativo (*Relative Distinguishing Name*, RDN). En la Figura 3.6.a se esquematiza el manejo de una MIT, en ella se aprecia que un número de identificación de puerto puede usarse como RDN para identificar puertos de un MO interface dado. Concatenando los RDNs en la trayectoria de la MIT, desde la raíz hasta un nodo dado, se obtiene un nombre Distintivo Unico (*Distinguishing Name*, DN). Este DN es utilizado por el Protocolo de Información de Administración Común (*Common Management Information Protocol*, CMIP) para identificar a un nodo y accesar su información administrada. El protocolo CMIP se tratará con detalle en apartados posteriores.

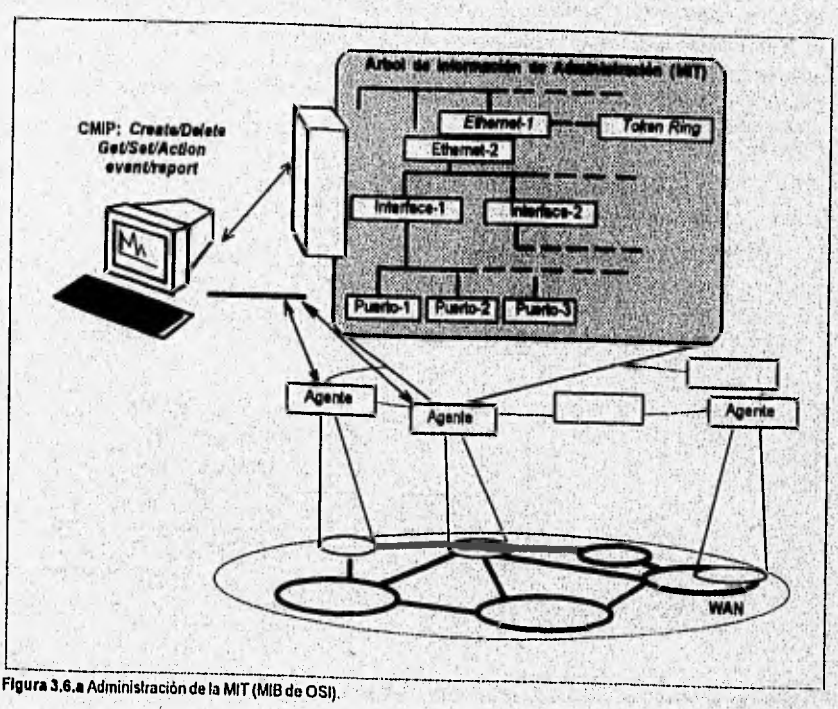


Figura 3.6.a Administración de la MIT (MIB de OSI).

Como se mencionó antes, la MIB de la ISO, a diferencia de la MIB de los otros modelos, es una base de datos dinámica, pues CMIP provee primitivas *CREATE* y *DELETE* que permiten cambiar a la MIT dinámicamente.

Una base de datos dinámica provee flexibilidad y eficiencia en el acceso de la información administrada. Administrando las entidades se pueden controlar los contenidos y estructura de la base de datos. La base de datos también puede organizarse flexiblemente para reflejar configuraciones de dispositivos específicos.

A pesar de que una base de datos dinámica presenta múltiples ventajas, también presenta complejidad en su implementación: los recursos necesarios para almacenar y procesar la información administrada no se pueden predecir al momento del diseño. Los administradores pueden extender a la MIT sobre recursos disponibles de agentes. Los cambios en la MIT pueden crear corrupción de la base de datos. Por ejemplo, un MO puede ser borrado mientras que otros MOs contienen apuntadores a él. Los diseñadores de *software* de aplicación no pueden compartir un modelo único del contenido de la MIT, ya que cada aplicación necesita de un modelo para construir y mantener su propio subconjunto de MIT.

En la Figura 3.6.b se muestra la estructura de la MIT. Las instancias de los MOs y sus atributos, operaciones y notificaciones de eventos se muestran en las figuras sombreadas. El agente OSI provee funciones de selección para localizar los registros de MOs accesados por los SAPs *Get/Set/Action* de CMISE. Los agentes también proveen detección de eventos y chequeo hacia adelante de notificaciones para el manejo de entidades involucradas para recibirlas.

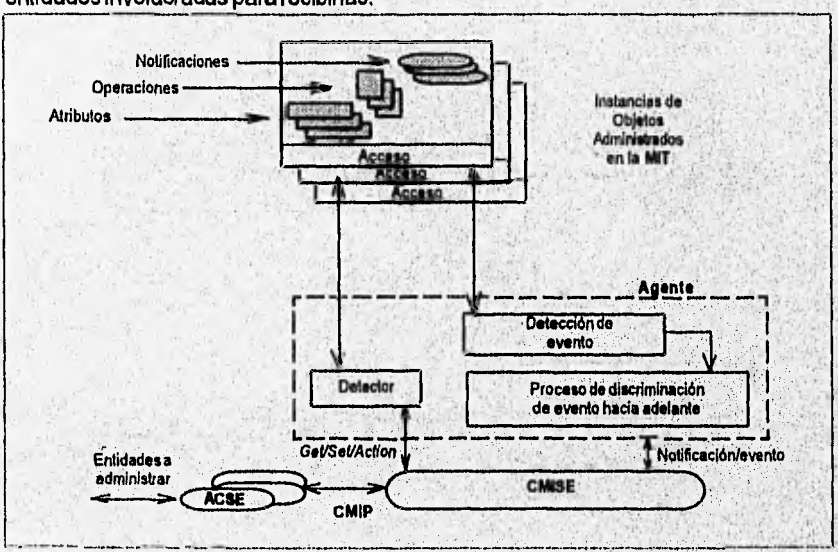


Figura 3.6.b Arquitectura de comunicación en la información administrada.

3.7 ELEMENTOS DE SERVICIO DE INFORMACION PARA LA ADMINISTRACION COMUN (CMISE)

3.7.1 Introducción

El CMISE se define en el documento 9595 de la ISO, donde se establecen los servicios elementales o primitivas que forman al mismo. El CMISE está basado en dos tipos de servicios básicos que son :

- El administrador de notificaciones: Este servicio es utilizado para reportar los eventos alrededor de los objetos que CMISE utiliza.

- El administrador de operaciones: Este servicio tiene como objetivo definir las operaciones para crear, recuperar, modificar, borrar o ejecutar cualquier otra acción con el manejo de objetos.

Como podemos observar, el CMISE es un conjunto de servicios de protocolo que provee al usuario de sintaxis y semántica entre las unidades de datos de protocolo que son intercambiados entre nodos iguales (*peer*). El protocolo está basado en un proceso de pelición-respuesta al invocar una operación que actúe sobre un objeto determinado. Para cada intercambio, el sistema de peticiones actúa en la administración y el sistema receptor juega el papel de agente. Para la ejecución de las operaciones incluidas en el manejo de objetos se tiene una estructura de árbol y el proceso de agente está provisto de una interface externa de comunicación. Las operaciones de administración incluyen intercambio de protocolos para la creación, eliminación, lectura y modificación de la información administrativa y la ejecución de las operaciones especificadas en el manejo de objetos.

En las siguientes líneas se emplearán los términos modo confirmado y modo no confirmado, refiriéndose éstos al modo de operar del servicio, siendo para el primer caso el modo en que se espera una respuesta de confirmación mientras que en el segundo no existe tal opción.

3.7.2 Uso de servicios y niveles

El CMISE, como se mencionó en el párrafo anterior, es un conjunto de servicios que pueden ser invocados desde locaciones remotas, estos servicios constan de varios argumentos o parámetros. El CMISE está formado por los siguientes tipos de servicios:

- *M-EVENT-REPORT*: Este servicio se usa para reportar un evento a un servicio de usuario. Las operaciones de entidades de red están en función de las especificaciones de los objetos manejados, estos eventos no están definidos por el estándar, varios eventos pueden influir alrededor de un objeto, dando este servicio un reporte de los eventos y de los tiempos de ocurrencia en el momento en que se den. En el modo confirmado, los parámetros manejados con la solicitud de servicio son: número de secuencia, el modo que puede ser usado en el envío del reporte, el reporte del evento por el objeto manejado y el tipo de evento que inicia el reporte. Para el modo confirmado la respuesta positiva es un *acknowledgment*, el cual contiene el número de secuencia, así como los parámetros del mandato. La identificación del objeto, el tipo de evento y el tiempo de respuesta pueden ser incluidos opcionalmente en la respuesta. Un error en la respuesta también puede generarse por un error dentro de la sección del servicio.

- *M-GET*: Este servicio se utiliza para recuperar información desde un nodo. El servicio usa información acerca del administrador de objetos para obtener y regresar un conjunto de atributos de identificación y valores del manejo de un objeto o de una selección de objetos manejados. Esto puede usarse únicamente en un modo confirmado y esperar una respuesta. Los parámetros asociados al mandato con petición del servicio son un número de secuencia y el objeto manejado. Cualquiera de los atributos de referencia de un objeto manejado pueden recuperarse o el objeto puede usarse en la selección de otro MO contenido por el primer objeto. Si los atributos no son especificados, esto implica que los valores de todos los atributos del objeto son solicitados. Si se seleccionan múltiples objetos, se emite una señal individual para cada objeto. El número de secuencia de la petición original se usa en la respuesta para enlazar las respuestas múltiples generadas por la solicitud, esto se refiere al parámetro ID de enlace. Para completar la respuesta múltiple es necesario que las salidas de las peticiones sean concluidas completamente, si la operación no fue terminada totalmente se considera como errónea, en caso de que la operación sea haya terminado parcialmente, la respuesta contendrá errores en la información y los parámetros podrán ser restaurados en su totalidad.

- *M-CANCEL-GET*: Este servicio se invoca para atender a una petición o cancelar una petición previamente hecha por el servicio *M-GET*. Puede usarse únicamente en el modo confirmado y al igual que en el caso anterior, espera una respuesta. Los parámetros manejados en este servicio son: número de secuencia para la petición y el número de secuencia de la petición que será cancelada.

- *M-SET*: Este servicio se usa cuando se hace una petición de modificación de atributos. Esta petición puede usarse en los modos confirmado o no confirmado, si es confirmada se espera una respuesta. Los parámetros manejados en este servicio son: número de secuencia, el MO y el modificador del operador,

usando el modificador de operador es posible reemplazar los valores de los atributos de la petición, sumar un nuevo valor al conjunto de valores de los atributos y éstos se sustituyen por el valor que por defecto toma el MO. Este servicio funciona también para múltiples llamadas de objetos cerrando los servicios que para éstos se hubiesen solicitado.

- *M-ACTION:* Este servicio es usado por un usuario para solicitar a otro la ejecución de algún tipo de acción sobre un MO. Este servicio puede solicitarse sobre los modos confirmado y no confirmado, esperando una respuesta si es confirmado. El tipo de acción se define por el objeto manejado al igual que la acción y los detalles de ejecución de la misma, los parámetros asociados al mandato son: el número de secuencia, el objeto manejado y el tipo de acción, los parámetros opcionales incluyen información de la acción específica y de la seguridad del control de información.

- *M-CREATE*: Este servicio se emplea para crear la representación de una instancia diferente de un MO, siendo esto a la larga, asociado con los valores del administrador de información. Este servicio sólo puede operar en el modo confirmado y como ya se sabe se espera una respuesta. Este servicio nos puede permitir copiar un objeto existente en otro con nombre diferente, en la copia, los valores que pueden ser solicitados se sobreescriben, al nuevo objeto se le asigna un nombre usando cualquiera de los siguientes métodos:

  - Nombre explícito: Especificado en la petición
  - Identificación única: Asignado relativamente al superior especificado en la petición.
  - El nombre es completamente asignado por el sistema creador del objeto.

- *M-DELETE*: Este servicio ejecuta la operación inversa de M-CREATE y se usa para borrar una instancia de un objeto manejado, siendo factible su uso únicamente en el modo confirmado, por lo que se espera una respuesta. Los parámetros manejados dentro de este servicio son: número de secuencia para la petición y el objeto manejado.

Una vez habiendo revisado los servicios de CMISE se enunciarán los servicios de ROSE que son usados por CMISE para mantener la comunicación entre niveles (los servicios de ROSE se ya se explicaron más a detalle en el apartado 3.5.3):

1. *RO-INVOKE*
2. *RO-RESULT*
3. *RO-ERROR*
4. *RO-EJECT*

En la siguiente figura se ilustra la relación entre ACSE, CMIP, ROSE y CMISE en el nivel de presentación.

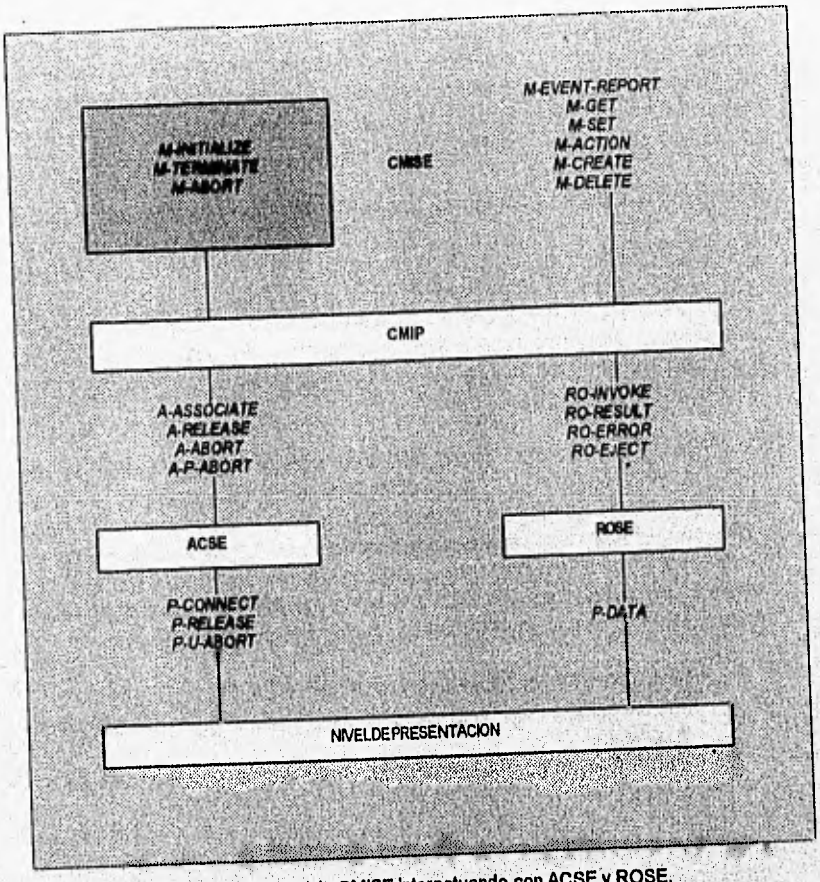


Figura 3.7.2.a Elementos de servicio CMISE interactuando con ACSE y ROSE.

En esta figura se muestra cómo ROSE hace uso del servicio *P-DATA*, el cual pertenece al nivel de presentación, también se observa la relación existente entre CMISE y CMIP con respecto a ACSE, ROSE y el Nivel de Presentación.

3.7.3 Unidades funcionales

El CMISE está basado en el concepto de unidades funcionales, que son mecanismos que describen y definen las comunicaciones entre las entidades y se refiere a una asociación. Un conjunto de servicios puede estar agrupado para el uso de un grupo que negocia los servicios para la asociación.

Las unidades funcionales de CMISE están divididas en unidades funcionales del *kernel*,

Adicional, Selección de Objeto Múltiple , Filtrado y Servicio Extendido.

- *Kernel* :Hace la identificación de los parámetros que no pueden ser usados en una respuesta múltiple de la unidad funcional, además de obtener el alcance y los parámetros del objeto de la unidad funcional que se ha seleccionado.

- Adicional: Esta opción nos permite hacer uso de los parámetros que no son permitidos dentro de la unidad de servicios *kernel.*

- Selección de Objeto Múltiple: Este servicio hace disponibles el alcance y los parámetros de sincronización para el uso de la unidad funcional del *kernel.* No se encuentra disponible para los servicios *M-EVENT-REPORT* y *M-CREATE.*

- Filtrado: Este servicio hace disponibles los parámetros para el filtrado en el *kernel.* Esta unidad no está disponible para las operaciones *M-EVENT-REPORT* y *M-CREATE.*

- Réplica Múltiple:Este servicio hace disponible el enlace de los parámetros de identificación dentro de la unidad funcional del *kernel.* No se encuentra disponible para las operaciones *M-EVENT-REPORT* y *M-CREATE.*

- Servicio Extendido: Este servicio hace disponible la presentación de otros servicios como el *P-DATA.*

En la siguiente tabla podemos observar la unidades funcionales definidas para CMISE, sus características y operaciones con las que mantienen relación.

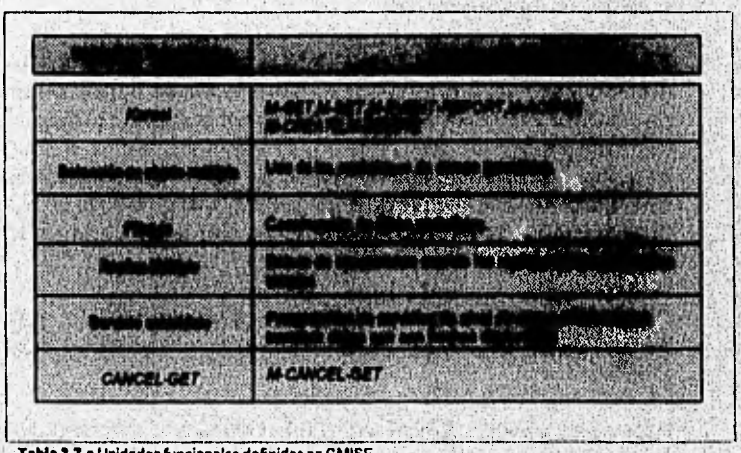


Tabla 3.7.a Unidades funcionales definidas en CMISE.

En la figura anterior se pueden ver las primitivas involucradas en este servicio. El *M-EVENT-REPORT* se usa para recibir los reportes de alarmas entre *gateways* y redes, para que las primitivas hagan uso de la información sus parámetros son los siguientes:

- Identificador de Invocación (*Invocation Identifier*, II): Este parámetro se usa durante la operación de reportes y no se encuentra definido en el estándar. Este parámetro es claro y diferente de los otros identificadores que se puedan estar usando. Este parámetro se usa en todas las primitivas de este servicio.

- Modo (*Mode*, M): Este parámetro se utiliza para identificar si la operación de reporte es confirmada o no confirmada. Se emplea en la petición e indicación de primitivas.

- Clase de Objeto Administrado (*Managed Object Class*, MOC): Se especifican las clases de los objetos administrados sobre el objeto administrado durante el inicio del reporte del evento. Este parámetro es requerido en la solicitud e indicación de las primitivas, siendo opcional en la respuesta y confirmación de las primitivas.

- Instancia de Objeto Administrado (*Managed Object Instance*, MOI): Es el encargado de especificar la instancia del objeto administrado durante el evento. Este parámetro es requerido para la petición e indicación de las primitivas y es opcional para la respuesta y confirmación de primitivas.

- Tipo de Evento (*Event Type*, ETy): Este parámetro identifica el tipo de evento que está siendo reportado. Sin embargo, su identificación depende del contexto del objeto administrado. El tipo de evento es requerido en la solicitud e indicación de primitivas y existe en la respuesta y confirmación de las mismas, sólo si se incluye el parámetro réplica de evento.

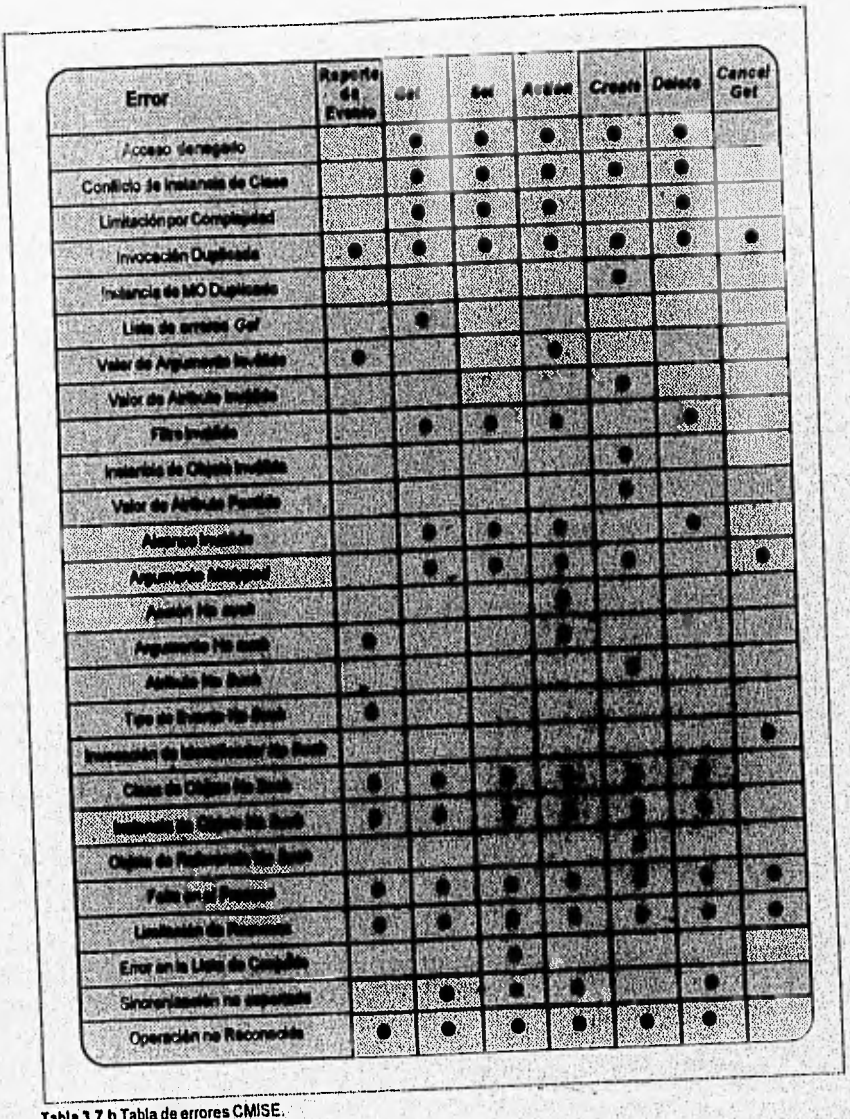
- Tiempo de Evento (*Event Time*, ET): Este parámetro contiene el tiempo del evento que se generó. El tiempo de evento es opcional en la petición e indicación de primitivas y no reside en la solicitud y confirmación de primitivas.

- Información de Evento (*Event Information*, EI): Contiene información del servicio de usuario que está ocurriendo. Este parámetro está definido en el estándar, pues es una aplicación específica y es opcional en la respuesta e indicación de primitivas.

- Tiempo Actual (*Current Time*, CT): Este parámetro contiene el tiempo de generación de respuesta y no está incluido en la respuesta e indicación de primitivas.

- Reemplazo de Evento (*Event Replace*, ER): Este parámetro contiene la respuesta del reporte de evento, si se completo o no. La Réplica de Evento es condicional dentro de la respuesta y confirmación de las primitivas dependiendo de las primitivas de confirmación sobre el usuario seleccionado.

- Errores (*Errors*, E): Este parámetro contiene información de diagnóstico sobre la operación del evento en que hubo error. El estándar CMISE permite reportar los errores contenidos en la siguiente tabla:



Error	Reporte de Evento	Get	Set	Action	Create	Delete	Cancel Get
Acceso denegado		●	●	●	●	●	
Conflicto de Instancia de Clase		●	●	●	●	●	
Limitación por Complejidad		●	●	●		●	
Invocación Duplicada	●	●	●	●	●	●	●
Instancia de MO Duplicada					●		
Lista de errores Get		●					
Valor de Argumento Inválido	●			●			
Valor de Atributo Inválido					●		
Filtro Inválido		●	●	●		●	
Instancia de Objeto Inválida					●		
Valor de Atributo Faltante					●		
Acceso Inválido		●	●	●		●	
Argumento Inválido		●	●	●	●		●
Acción No Hallada				●			
Argumento No Hallado	●			●			
Atributo No Hallado					●		
Tipo de Evento No Hallado	●						
Inconsistencia de [illegible] No Hallado							●
Clase de Objeto No Hallada	●	●	●	●	●	●	
Instancia de Objeto No Hallada	●	●	●	●	●	●	
Objeto de Referencia No Hallado					●		
Falta en el Acceso	●	●	●	●	●	●	●
Limitación de Recursos	●	●	●	●	●	●	●
Error en la Lista de Cancelar			●				
Sincronización no Soportada		●	●	●		●	
Operación no Reconocida	●	●	●	●	●	●	

Tabla 3.7.b Tabla de errores CMISE.

El *M-GET* es usado por un servicio de usuario CMISE obteniendo información de un servicio de usuario igual (*peer*). Típicamente este servicio se utiliza para recuperar los indicadores de *status* sobre el manejo de objetos en la red o información alrededor de los objetos añadidos o removidos. Este es un servicio confirmado, ver la Figura 3.7.c.

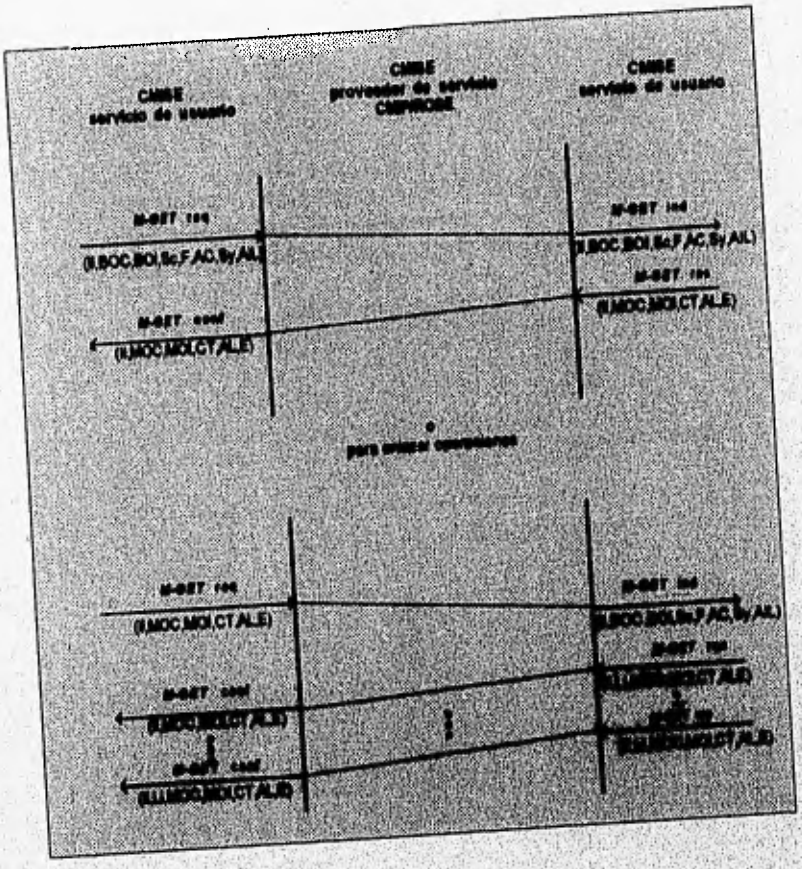


Figura 3.7.c El servicio M-GET.

Los parámetros que se aprecian en la figura son usados por las primitivas, los cuales se explican en las siguientes líneas, algunos de ellos ya se explicaron anteriormente, por lo que sólo se mencionarán:

- Identificador de Invocación (II).

- Identificador de Enlace (*Link Identifier*, LI): El identificador de enlace se usa únicamente si hay múltiples respuestas que son transmitidas por esta operación. Este valor es alguno de los parámetros de identificador invocados en la indicación de las primitivas.

- Clase de Objeto Base (*Base Object Class*, BOC): Este parámetro especifica la clase del objeto manejado cuyos valores de atributo pueden recuperarse durante la operación *M-GET*. Dichos valores se utilizan como punto inicial para seleccionar el objeto administrado para la operación de filtrado.

- Instancia del Objeto Base (*Base Object Instance*, BOI): Este parámetro especifica la instancia de un objeto administrado, además esto puede usarse en un punto inicial sobre el filtro aplicado.

- Alcance (*Scope*, Sc): Identifica el sub-árbol para la búsqueda y recuperación. Se permiten tres niveles de búsqueda y son los siguientes:

  - Sólo Objeto Base.
  - El n-ésimo nivel subordinado del Objeto Base.
  - El Objeto Base y todos sus subordinados..

- Filtro (*Filter*, F): Este parámetro especifica las pruebas de condiciones para la búsqueda. El filtro utiliza operadores booleanos (*AND, OR, NOT*). El trabajo de filtrado se efectúa sobre la prueba de condición encadenada con los operadores booleanos.

- Control de Acceso (*Control Access*, AC): El valor para este parámetro no está definido en el estándar, pero CMISE lo usa para permitir las operaciones sobre las funciones de control durante la asociación. El parámetro de control de acceso no se encuentra residente dentro de la respuesta y confirmación de primitivas que responden a CMISE.

- Sincronización (*Syncronization*, Sy): Este parámetro permite a la invocación del usuario determinar el tipo de sincronizacón que se utilizará en las operaciones de recuperación. Dos métodos de sincronización son usados:

  - Atómico. Determina que todas las recuperaciones o ninguna se ejecuten.

  - Mejor esfuerzo. Las recuperaciones se llevan a cabo sólo si es posible, esto es, si alguna de ellas no puede ejecutarse, las demás continuan intentándose.

- Lista de Identificadores de Atributo (*Atribute Identifier List*, AIL): Este parámetro es opcional y es usado por el servicio de respuesta de CMISE. Este servicio de respuesta examina los IDs para determinar qué valor de atributo se haya respondido. Si el parámetro se omite, todos los identificadores de atributo se asumen. El uso de esta lista depende de la implementación y especificación seleccionada por la MOC.

- Clase de Objeto Administrado (MOC).

- Instancia de Objeto Administrado (MOI).

- Tiempo Actual (CT).

- Lista de Atributos (*Attribute List*, AL): Este parámetro contiene los identificadores de parámetros y algunos valores que pueden regresarse durante la ejecución de CMISE. Este puede ser o no ser incluido, lo cual estará en función del tipo de falla de la operación.

- Errores (E): Estos errores son listados en la tabla 3.7.b.

El *M-CANCEL-GET* se utiliza por un servicio de usuario de CMISE para obtener información de un servicio de igual. Este servicio se emplea para obtener el indicador de *status* sobre un objeto manejado en la red o para obtener información alrededor del borrado o añadido de objetos. Este servicio es del tipo confirmado y se muestra en la siguiente figura.

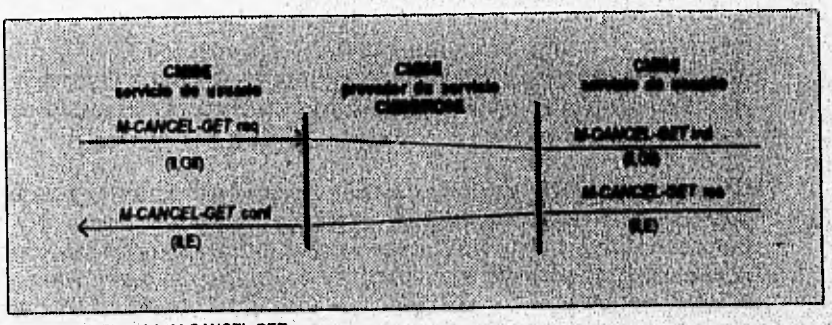


Figura 3.7.d El servicio M-CANCEL-GET.

La figura anterior nos muestra la operación *M-CANCEL-GET* y los parámetros que están involucrados en ella, los cuales se explicarán en las siguientes líneas, algunos de ellos ya se explicaron anteriormente, por lo que sólo se mencionarán:

- Identificador de invocación (II).

- Identificador de Invocaciones *GET* (*Invocation Identifier GET*, IIG): Este parámetro identifica que la operación *M-GET* puede ser cancelada.

- Errores (E): Estos errores aparecen en la tabla 3.7. b.

El *M-SET* se utiliza por una invocación CMISE. Se usa para solicitar el cambio de los valores de los atributos para la ejecución de un servicio CMISE. Este puede ser usado por un servicio confirmado o no confirmado, El servicio *M-SET* se muestra en la siguiente figura.

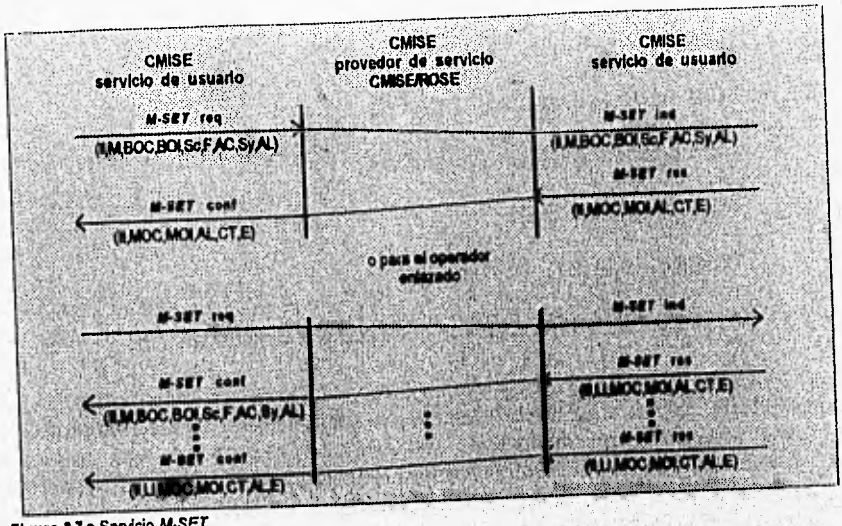


Figura 3.7.e Servicio *M-SET*.

Cuando el servicio es no confirmado tenemos la siguiente figura.

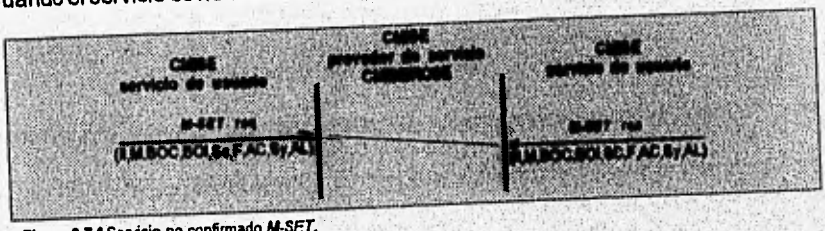


Figura 3.7.f Servicio no confirmado *M-SET*.

Los parámetros involucrados en el servicio *M-SET* son los siguientes, algunos de ellos ya se explicaron anteriormente, por lo que sólo se mencionarán:

- Identificador de invocación (II).
- Identificador de Enlace (LI).
- Modo (M).
- Clase de Objeto Base (BOC).
- Instancia de Objeto Base (BOI).
- Alcance (Sc).
- Filtro (F).
- Control de Acceso (AC).

- Sincronización (Sy).
- Clase de Objeto Administrado (MOC).
- Instancia de Objeto Administrado (MOI).
- Lista de atributos (AL).
- Tiempo Actual (CT):.
- Errores (E): Estos errores se pueden apreciar en la tabla 3.7.b.

El *M-ACTION* es utilizado por la invocación de servicio para solicitar la ejecución del servicio de usuario CMISE que opera sobre un objeto manejado, es decir, la ejecución de una acción sobre un objeto manejado. *M-ACTION* se usa para un servicio confirmado y/o no confirmado. En la siguiente figura se muestran los parámetros que influyen en este servicio y posteriormente son explicados, algunos de ellos ya se explicaron anteriormente, por lo que sólo se mencionarán:

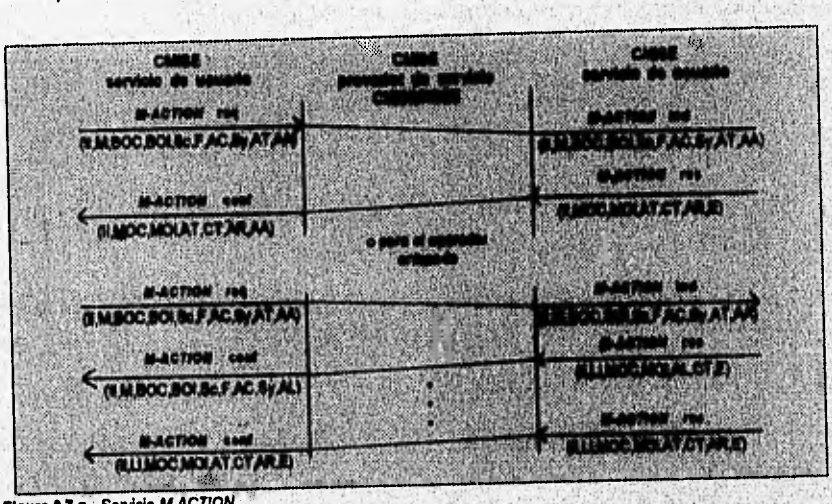


Figura 3.7.g Servicio *M-ACTION*.

- Identificador de Enlace (LI).
- Modo (M).
- Clase de Objeto Base (BOC).
- Instancia de Objeto Administrados (MOI).
- Alcance (Sc).

- Filtro (F).
- Clase de Objeto Base (BOC).
- Instancia de Objeto Administrado (MOI).
- Control de Acceso (AC).
- Sincronización (Sy).
- Tipo de Acción (*Action Type*, AT): Describe el tipo de acciones que son ejecutados sobre el objeto manejado.
- Argumento de Acción (*Action Argument*, AA): Este parámetro provee información extra alrededor de la acción . Esto es opcional para la solicitud e indicación de primitivas y no existe en la respuesta y confirmación de primitivas.
- Tiempo Actual (CT).
- Resultado de Acción (*Action Result*, AR): Este parámetro contiene el resultado de la acción.
- Errores (E): Los errores soportados por M-ACTION se listan en la tabla 3.7 b.

El *M-CREATE* es requerido por la invocación de CMISE. Se utiliza para solicitar una ejecución de CMISE, usado para crear la representación de una nueva instancia de objeto. Este debe completar la identificación y el valor asociado con la próxima información de administración. Se utiliza únicamente en el servicio confirmado. La operación *M-CREATE* se ilustra en la siguiente figura y sus parámetros se explican en las siguientes líneas. Algunos de ellos ya se explicaron anteriormente, por lo que sólo se mencionarán.

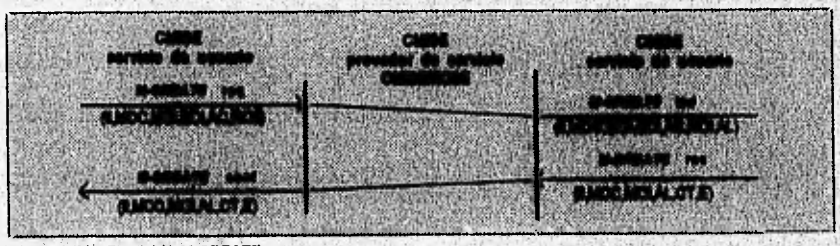


Figura 3.7.h El servicio *M-CREATE*.

- Identificación de Invocación (II).
- Clase de Objeto Administrado (MOC).
- Instancia de Objeto Administrado (MOI).

- Instancia de Objeto Superior (*Superior Object Instance*, SOI): Este parámetro especifica una existencia MOI se diseña en el superior de la nueva instancia del objeto.

- Control de Acceso (AC).

- Instancia de Objeto de Referencia (*Reference Object Instance*, ROI): Estos parámetros especifican la instancia del objeto administarado durante el evento, el cual es requerido por la solicitud e indicación de primitivas, no existe en la respuesta y confirmación de primitivas.

- Listas de Atributos (AL).

- Tiempo Actual (CT).

- Errores (E): Los errores espacificados para este servicio se expresan en la tabla 3.7.b.

El servicio *M-DELETE* es usado para borrar la representación de un MOI y es usado solamente por servicios confirmados. Los parámetros involucrados en este servicio se muestran en la siguiente figura, y se explican después de ésta. Algunos de ellos ya se explicaron anteriormente, por lo que sólo se mencionarán.

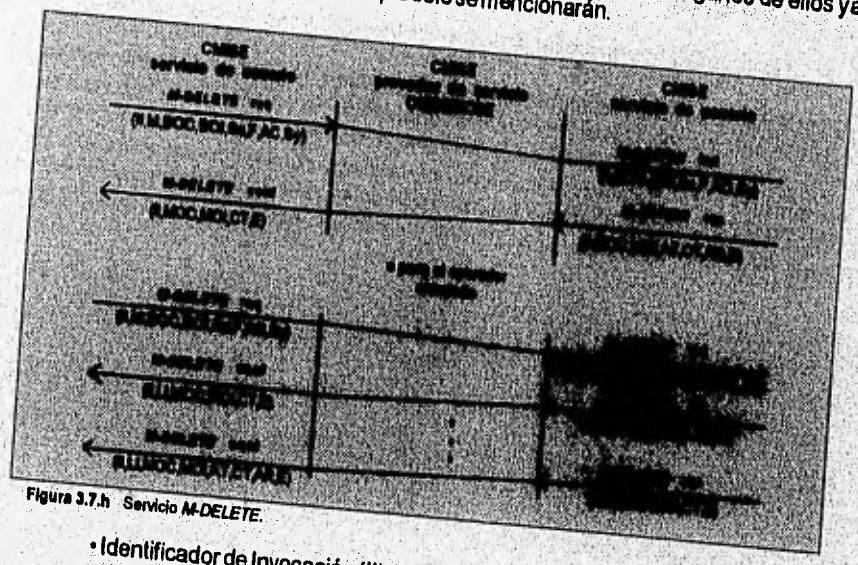


Figura 3.7.h Servicio *M-DELETE*.

- Identificador de Invocación (II).

- Identificador de Enlace (LI).

- Clase de Objeto Base (BOC).

• Instancia de Objeto Base (BOI).

• Alcance (Sc): El valor por defecto de este parámetro es un objeto base único.

• Filtro (F).

• Control de Acceso (AC).

• Sincronización (Sy).

• Clases de Objetos Administrados (MOC).

• Tiempo Corriente (CT).

• Errores (E): Los errores soportados por este servicio son listados en la tabla 3.7.b

3.8 PROTOCOLO DE INFORMACION PARA LA ADMINISTRACION COMUN (CMIP)

3.8.1 Introducción

La norma ISO 9596 establece la especificación para el Protocolo de Información para la Administración Común (*Common Management Information Protocol*, CMIP). CMIP soporta los servicios de CMISE explicados en la sección previa. Recuerde que algunos de los servicios son confirmados, no confirmados, o tienen la opción de ser una operación confirmada o no confirmada; estos servicios permiten a la administración OSI iniciar acciones sobre objetos administrados, para cambiar los atributos de ellos y para reportar el estado de los mismos.

Como otros protocolos de OSI, CMIP deberá seguir las reglas en la composición y el intercambio de PDUs. Todos los PDUs de CMIP se definen por medio de la notación ASN.1 (consultar apéndice B). Por otra parte, el CMIP es independiente de ROSE, por lo cual no contiene tablas de estado, listas de eventos, predicados ni tablas de acción.

3.8.2 Información del usuario CMIP para el servicio *A-ASSOCIATE*

El usuario CMIP es el responsable de iniciar una asociación antes de que alguna operación de administración pueda comenzar. Este procedimiento se realiza bajo los servicios de ACSE. Uno de estos servicios, el *A-ASSOCIATE*, sirve para establecer dichas asociaciones, además deberá contener la siguiente información vista en el apartado de asociaciones: unidades funcionales, control de acceso e información del usuario.

La Figura 3.8.a nos muestra la codificación en ASN.1, para que esta información sea incluida en el servicio *A-SSOCIATE* (el código está abreviado para propósitos de simplificación). Recuerde consultar las reglas que establece ASN.1, las cuales se encuentran en el apéndice B.

```
Informaciónde lusuarioCMIP ::= SEQUENCE
                               {Versióndelprotocolo      [0] IMPLICIT
                               Unidadesfuncionales       [1] IMPLICIT
                               Controldeacceso           [2] EXTERNAL OPTIONAL
                               Informacióndeusuario      [3] EXTERNAL OPTIONAL }

Unidadesfuncionales      ::= BIT STRING
                               {Selecciónde objetomúltiple    (0),
                               filtro                         (1),
                               Repeticiónmúltiple             (2),
                               Servicioextendido              (3) }
```

Figura 3.8.a Codificación en ASN.1 del usuario de CMIP para el servicio *A-ASSOCIATE*.

3.8.3 Unidad de datos del protocolo CMIP

Al igual que para cualquier protocolo de comunicación, CMIP también requiere establecer unidades o bloques de datos durante su operación, a dichos bloques se les conoce individualmente como Unidad de Datos del Protocolo (*Protocol Data Unit*, PDU). En la Figura 3.8.b se muestra la codificación para la creación de PDUs de CMIP, empleando la operación *M-CREATE*.

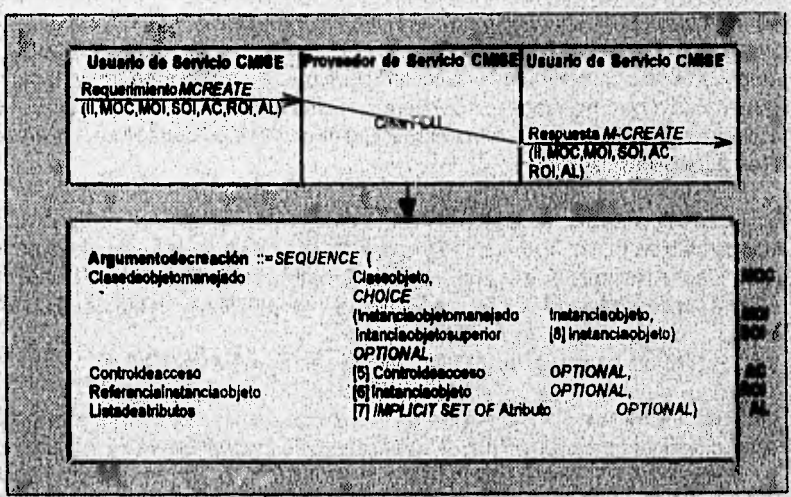


Figura 3.8.b Creación del PDU de CMIP, codificado en ASN.1.

Esta figura nos muestra la primitiva de petición *M-CREATE* y su relación con el PDU *M-CREATE*. En ella se muestra la petición que hace la primitiva *M-CREATE* al ser invocado por el servicio del usuario de CMISE, los parámetros de la primitiva se utilizan para crear el PDU, el cual se manda a que sea procesado por el servicio CMISE. La parte inferior de la figura aumenta la porción del contenido del PDU. El ASN.1 define los tipos de los

campos en el PDU. Para ser exactos, las iniciales al margen del recuadro se usan para representar los parámetros en la primitiva CMISE. Con una excepción, todos los parámetros de la primitiva se mapean dentro del PDU. Para invocar al identificador (II) no se toma en el PDU, sino lo usa el ROSE para el soporte de la identificación.

Como se puede observar, el MOI y un objeto superior (SOI) se definen con *CHOICE*. El Control de acceso del objeto, la instancia del objeto y la lista de atributos son todos parámetros opcionales, sin embargo muchas redes los utilizan hoy en día.

Asimismo, en la Figura 3.8.c se muestra la respuesta que produce el servicio de CMISE para la creación del PDU de CMIP. La actividad de este servicio produce como resultado una primitiva de respuesta *M-CREATE*, los parámetros se muestran dentro del paréntesis en la figura. Como anteriormente se dijo, las primitivas se mapean dentro del PDU y se mandan a iniciar el servicio CMISE. La codificación revela el contenido de

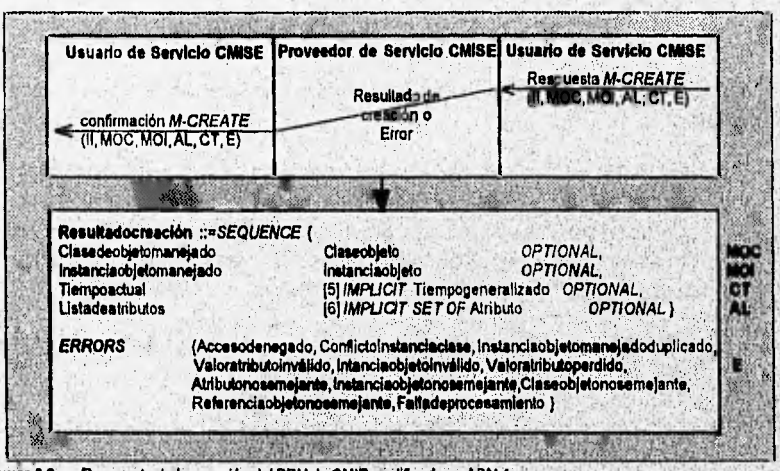


Figura 3.8.c Respuesta de la creación del PDU de CMIP, codificado en ASN.1.

una porción del PDU. Las iniciales de las primitivas y sus relaciones se colocan en el borde derecho de la parte inferior.

3.8.4 Operaciones CMIP

A las entidades que soportan CMIP y que interactúan entre sí en el proceso de administración son conocidas como Máquinas CMIP (*CMIP Machines*, CMIPMs). La Figura 3.8.d intenta unir las piezas del rompecabezas, es decir, muestra las operaciones entre los servicios de usuarios CMISE, la máquina CMIP (el protocolo) y el elemento del servicio ROSE.

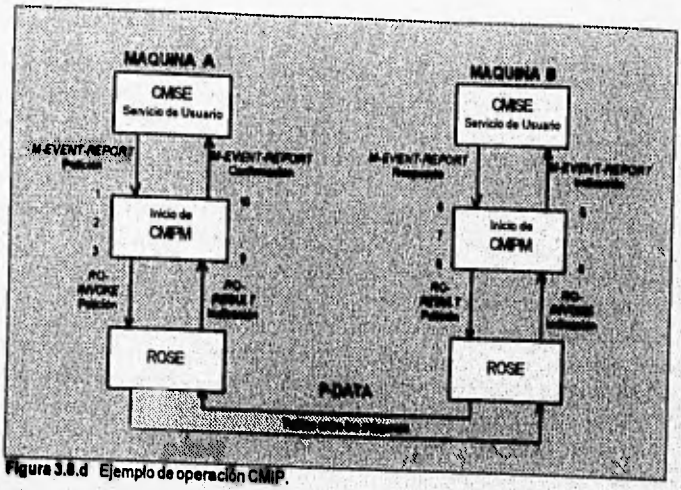


Figura 3.8.d Ejemplo de operación CMIP.

El servicio CMISE y el ROSE siguen los pasos numerados en la figura y descritos a continuación:

1. El CMIPM recibe como petición una primitiva *M-EVENT-REPORT* del servicio CMISE. Los parámetros en la primitiva contienen valores para identificar esta operación de otras operaciones soportadas por CMISE, para obtener información del objeto manejado, el tipo de evento que se reporta, el tiempo de la generación del evento, y la información acerca del mismo.

2. El CMIPM examina la primitiva y construye un PDU de aplicación (*Aplication* PDU, APDU) llamado *M-EVENT-REPORT*.

3. Entonces se manda este APDU hacia el ROSE a través de la primitiva de petición *RO-INVOKE*. ROSE envía esta unidad al nivel de presentación a través de la primitiva *P-DATA*.

4. La parte receptora de CMIPM recibe el APDU por medio de la primitiva *RO-INVOKE* de ROSE.

5. Si la unidad de datos es aceptable, se produce una indicación *M-EVENT-REPORT* para ser recibida por CMISE. De esta manera, el lado receptor obtiene la información necesaria para tomar decisiones de acuerdo al reporte sobre el objeto manejado.

6. El servicio CMISE envía una respuesta *M-EVENT-REPORT* a su CMIPM. Esta primitiva contiene valores para identificar esta operación de otras soportadas por el CMISE, dar información acerca del objeto manejado, dar el tipo de evento que está siendo reportado, dar el tiempo de la generación de esta respuesta y dar el resultado del reporte del evento. Si esta primitiva tiene una respuesta de falla, se incluye un parámetro de error para describir la naturaleza del error.

7. El CMIPM emplea estos parámetros para construir un nuevo APDU.

8. Manda este APDU al ROSE por medio de una primitiva *RO-RESULT*.

9. El CMIPM emisor recibe el APDU por medio del *RO-RESULT*.

10. Si el APDU es aceptable (bien formado), se produce una primitiva de confirmación *M-EVENT-REPORT* hacia el usuario de CMISE. Esta primitiva contiene la información creada por el otro usuario en el paso No. 6.

11. El procedimiento de reporte queda completo.

3.8.5 Párametros asociados con los servicios CMIP

Se deben examinar los parámetros asociados con estos servicios, ya que su análisis permitirá clarificar el uso de CMIP y dará la oportunidad de examinar algunas respuestas importantes, las cuales deben ser tomadas en cuenta por el administrador. A continuación se enlistan estos parámetros y se explica su uso:

- Invocación de identificador (II): Este campo se usa para identificar de manera única una operación específica de crear o borrar y que distingue a ella de otras operaciones que el proceso administrativo pudiera tener en operación. El valor del identificador también se usa para determinar las unidades de datos asociadas con la operación de control de red. Esto es, el campo se emplea en las operaciones *M-SET*, *M-GET* y otras. La administración de red de OSI no especifica la sintaxis, ni el formato del identificador, así es que el administrador queda en libertad de desarrollar el esquema que elija. No obstante, algún tipo de identificación de red y convención para asignar nombres se deberán considerar cuidadosamente.

- Instancia de Objeto Superior (SOI): Este parámetro especifica un MOI existente, el cual se designa como el superior de un objeto nuevo.

- Clase de Objeto Manejado (MOC): El identificador de un objeto manejado se coloca en una MIB con un conjunto de atributos, los cuales describen las características del mismo, todo ello forma a la clase de objeto manejado. Por ejemplo, un circuito de comunicaciones puede describirse por sus atributos tales como la velocidad en la línea, si es dedicado o se le puede conmutar, si es

half-duplex o *full-duplex*, etc. Este conjunto de atributos se identifica como la clase de un objeto manejado. Un objeto manejado que pertenece a la misma clase contiene la misma lista de atributos. El campo se emplea por el cliente de la red o el centro de control de la red para identificar los atributos del recurso que están siendo creados. Los valores como la velocidad de la línea, acceso dedicado, etc. son almacenados en la MIB para propósitos de control de la información.

- Control de Acceso (AC): Este parámetro no está definido concisamente en OSI, sino que permite las funciones del control de acceso para ser invocadas entre los clientes de la red y el control de la red.

- Instancia de Referencia de Objeto (ROI): Una instancia de objeto de referencia es la forma de identificar a un objeto manejado existente. Por ejemplo, un circuito de comunicación activo tiene sus atributos definidos en la MIB. Consecuentemente, si un cliente desea crear un elemento de otro circuito, es decir, hacer del conocimiento del sistema de administración de la red el circuito, no es necesario que sean definidos sus atributos, en lugar de esto, un objeto que ya exista (un enlace de comunicación) puede referenciarse, y sus valores son entonces los valores por omisión del nuevo circuito.

- Lista de Atributos (AL): Supongamos a dos usuarios de la red, por ejemplo, un cliente de la red y un centro de control de la misma, los cuales se crean como objetos manejados. En términos de OSI, la parte que desea crear el objeto es la invocación del usuario, y la parte que recibe la unidad de datos con *M-CREATE* es la parte en funcionamiento. La lista de atributos es usada por la parte que invoca para asignar los atributos al objeto. Estos valores sobreescriben cualquier valor previo así como los valores del parámetro de objeto referenciado. De esta manera la actividad del usuario regresa una confirmación *M-CREATE*, el cual contiene una lista de parámetros. Estos pueden o no ser los mismos de la invocación. En esencia, OSI da a los usuarios de CMISE la oportunidad de negociar los parámetros -las características- de un objeto manejado.

- Tiempo actual (CT): Este parámetro es simplemente la hora en curso en la que ocurre un evento en la red.

- Errores (E): Este parámetro puede contener muchos valores o ninguno. Su empleo requiere que la red y sus clientes usen mecanismos de reporte de errores estándar basados en OSI.

- Unidades funcionales: Las unidades funcionales proveen al administrador de un significado adecuado de la categorización de los servicios que están establecidos entre la red y sus usuarios.

- Información del usuario: Los valores de este campo no están definidos en ninguno de los estándares de administración de OSI. Ellos se adecuan a la aplicación específica, la cual en términos de OSI, significa que el valor depende de los acuerdos que se hayan tomado en las partes de la comunicación. Esta distinción es importante ya que se enfatiza que los estándares OSI son interfaces en la comunicación entre redes y no asignan los valores de la información del usuario que se tengan en este campo. Sin embargo, debería recordarse que el campo de datos del usuario podría tener valores de un protocolo de nivel superior.

- *M-ABORT* fuente: Se emplea si una unidad de datos CMISE/CMIP es un aborto de una operación. Una red puede usar a éste como un proveedor de servicio, o un cliente de la red lo puede usar como un servicio de usuario.

- Modo: Este parámetro se utiliza para indicar si la operación entre dos dispositivos es confirmada o no confirmada.

- Tipo de Evento: Los estándares permiten una operación entre dispositivos para ser clasificada por su tipo. El tipo actual depende de la naturaleza o contexto del recurso de la red (objeto manejado). Podría ser empleado por una red para estandarizar nombres e identificadores para todas las operaciones de red.

- Tiempo de Evento: Contiene el tiempo de la generación del reporte de un evento.

- Argumento del evento: El dispositivo que responde con una unidad de datos que reporta un evento.

- Resultado del evento. Típicamente se utiliza para reportar el éxito o falla de una operación.

- Identificador ligado: Algunas operaciones de red requieren que un objeto manejado o el proceso de administración envíe más de un PDU en respuesta a una consulta del estado. Este parámetro permite encadenar las unidades de datos de las partes de la comunicación.

- Objeto base: Este parámetro se usa para determinar al elemento de la clase de objeto base.

- Clase de objeto base: Se utiliza para especificar los valores del atributo de una clase manejada o para filtrar operaciones.

- Medición: Se utiliza para identificar el subárbol en donde se comience la búsqueda de objetos manejados. Se permiten tres niveles de búsqueda: solamente el objeto base, el nivel n debajo del objeto base y el objeto base con todos sus subordinados. El valor por omisión para éste es solamente el objeto base.

- Filtro: Muchos sistemas de control de red utilizan el concepto de filtros, algunas organizaciones usan este término para describir el borrado de información en una unidad de datos antes de desplegarlo a un operador. Para OSI, el filtro es una afirmación o un conjunto de ellas acerca de los atributos de un objeto manejado. En términos simples, es la forma para seleccionar operaciones. Las operaciones de filtrado pueden codificarse en C, ASN.1, etc., para seleccionar parámetros en la MIB. Los filtros usan operadores booleanos convencionales tales como *AND, NOT* y *OR*.

- Sincronización: Se usa por el usuario para notificar como cambia la sincronización con los objetos. El estándar define dos modos de operación: el "atómico", donde todos los cambios propuestos se checan primero para determinar si pueden ser realizados, si alguno no se puede llevar a cabo ninguno se ejecuta. El segundo modo es el de "mayor esfuerzo", en donde las modificaciones se hacen en cualquier orden, si las modificaciones no se pueden realizar, no afecta a los cambios que puedan ser hechos.

- Tipo de acción: Este parámetro es usado para describir la acción específica que un objeto va a realizar. El tipo es dependiente del contexto del objeto manejado.

- Argumento de la acción: Si es necesario, este parámetro se usa para definir la naturaleza de la acción.

3.9 RELACION ENTRE LAS PRIMITIVAS CMISE Y LAS OPERACIONES CMIP

Como se ha visto hasta ahora, los servicios de CMISE y las operaciones CMIP guardan una estrecha relación entre sí. Ambos son escenciales para el proceso de administración de red que propone OSI, ya que si falta alguno de ellos, no se podría completar dicho proceso. En la tabla 3.9.a de la página siguiente se hace un resumen de las primitivas CMISE y de las operaciones CMIP, así como la correspondencia -si la hay- de cada una de ellas.

3.10 AREAS FUNCIONALES DE LA ADMINISTRACION DE REDES OSI

La administración de redes OSI se encuentra agrupada en cinco áreas funcionales: alarmas, rendimiento, configuración, contabilidad y seguridad; las cuales ya se describieron brevemente en el apartado 3.2 y se explicarán con más detalle posteriormente en este apartado. Las descripciones de las áreas funcionales se derivan de las especificaciones de estándares para administración de redes OSI, principalmente de los documentos ISO 10164.

Dichas áreas hacen referencia a las Areas Funcionales de Administración Específica (*Specific Management Functional Areas*, SMFAs), las cuales identifican:

Primitiva CMISE	Modo	Liga	Operación CMIP
M-EVENT-REPORT req/ind	NC	NA	m-EventReport
M-EVENT-REPORT req/ind	C	NA	m-EventReport-Confirmed
M-EVENT-REPORT rsp/conf	NA	NA	m-EventReport-Confirmed
M-GET req/ind	C	NA	m-Get
M-GET rsp/conf	NA	A	m-Get
M-GET rsp/conf	NA	P	m-Linked-Replay
M-SET req/ind	NC	NA	m-Set
M-SET req/ind	C	NA	m-Set-Confirmed
M-SET rsp/conf	NA	A	m-Set-Confirmed
M-SET rsp/conf	NA	P	m-Linked-Replay
M-ACTION req/ind	NC	NA	m-Action
M-ACTION req/ind	C	NA	m-Action-Confirmed
M-ACTION rsp/conf	NA	A	m-Action-Confirmed
M-ACTION rsp/conf	NA	P	m-Linked-Replay
M-CREATE req/ind	C	NA	m-Create
M-CREATE rsp/conf	NA	NA	m-Create
M-DELETE req/ind	C	NA	m-Delete
M-DELETE rsp/conf	NA	A	m-Delete
M-DELETE rsp/conf	NA	P	m-Linked-Replay

Tabla 3.9.a Primitivas de CMISE, operaciones CMIP y sus relaciones. A=Ausente, NA=No Aplicable, C= Confirmada, NC=No Confirmada, P=Presente.

- Un conjunto definido de funciones.
- Un conjunto definido de procedimientos que se asocian a cada función.
- El servicio para soportar dichos procedimientos.
- Los servicios básicos para soportar a los SMFA.
- La clase de MOs que se afectarán con las operaciones del SMFA.

Por otro lado, los protocolos de área funcional se valen ampliamente del uso de filtros. La definición de los filtros se lleva a cabo utilizando ASN.1, y en ellos se designan una serie de afirmaciones acerca de los atributos y sus valores. Es posible definir más de una afirmación utilizando los operadores lógicos *AND*, *OR* y *NOT*. Como se trató en el apartado 3.3, los protocolos de administración de redes utilizan el DN para identificar a un MO, adicionalmente, a este DN se le asigna un filtro.

3.10.1 Alarmas

El área funcional de alarmas también se conoce como administración de fallas. Las funciones básicas de la administración de fallas son las siguientes:

- Detección de fallas: Las fallas se detectan por monitoreo o por generación de reporte de error.

- Diagnóstico de fallas: Las fallas se diagnostican en los MOs por medio de la reproducción del error, el análisis del error, o por la recepción de reportes desde los mismos MOs.

- Corrección de fallas: La corrección de fallas se lleva a cabo con el uso de otras facilidades, tal como la administración de la configuración.

Sin embargo, para realizar estas funciones básicas, la administración de fallas se vale de otras funciones o servicios, que se definen en los estándares de OSI:

- Reporteo de eventos: Soporta la transferencia de reportes de error y de evento.

- Chequeo de diagnósticos y confianza: Soporta los medios para determinar si un MO es capaz de ejecutar su función.

- Control de "entrada" (*Log control*): Es una función común que soporta actividades como el manejo de *logs* de eventos, restricción de acceso a MOs, así como de la salida.

- Reporteo de alarma: Soporta el uso de alarmas en el sistema.

En la Figura 3.10.a se muestra un ejemplo de los servicios de reporteo de alarma. Este ejemplo fue tomado de la especificación de Administración de Fallas del Protocolo de Administración de Redes de AT&T.

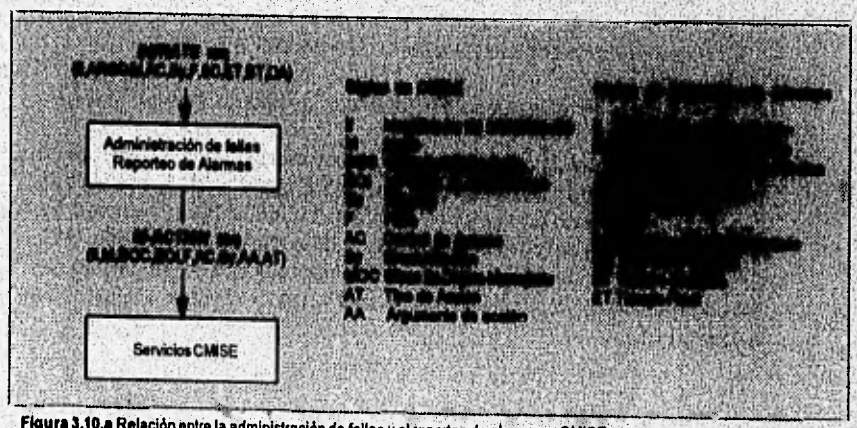


Figura 3.10.a Relación entre la administración de fallas y el reporteo de alarmas a CMISE.

En el ejemplo, la entidad de reporteo de alarma pasa una primitiva de requerimiento *INITIATE* a CMISE. CMISE reconoce que el servicio *INITIATE* requiere un servicio de confirmación CMISE. *INITIATE* se mapea tanto en el parámetro tipo de acción (actiontype, AT), como en el parámetro modo (*mode*, M) de CMISE. En esta operación el parámetro *mode* se debe poner a un servicio de confirmación.

El II mapea directamente al II de CMISE, el cual a su vez mapea al II de ROSE. Por otro lado, los parámetros de instancia y clase de separado de reporte de alarma (*alarm report sieve*), ARSC&I, se utiliza para identificar el separado de reporte. Este parámetro mapea a los parámetros de instancia de objeto y clase de objeto de CMISE.

Los parámetros AC, sincronización y filtro de la primitiva *INITIATE*, mapean directamente a su respectivas contrapartes en CMISE. Los siguientes cuatro parámetros de la primitiva *INITIATE*: SC, DA, BT y ET, mapean al parámetro argumento de acción (*Action Argument*, AA) de CMISE.

El parámetro SC identifica la fuente de la alarma y atributos de la misma que serán reportados.

El parámetro DA identifica la entidad de aplicación que es utilizada para recibir el reporte de alarma.

Los parámetros BT y ET especifican cuándo empezará y cuándo terminará la función de reporteo.

3.10.2 Rendimiento

El estándar de administración del rendimiento se basa en ISO 10064-11, para definir los requerimientos y criterios para medir la ejecución, así como un número de parámetros respecto a los cambios en: carga de trabajo, el *throughput* (número de procesos ejecutados en el tiempo), tiempo de espera de los recursos, tiempo de respuesta, retardo de propagación, disponibilidad y cualquier calidad de servicio (Quality of Service, QOS). La actividad de rendimiento se modela como un protocolo de monitoreo, el cual constantemente monitorea los recursos para medición del rendimiento del sistema, ajuste de los criterios de medición, así como determinar si el rendimiento es satisfactorio.

En la Figura 3.10.b se muestra cómo OSI usa la administración del rendimiento, basada en un modelo de sintonía y monitoreo.

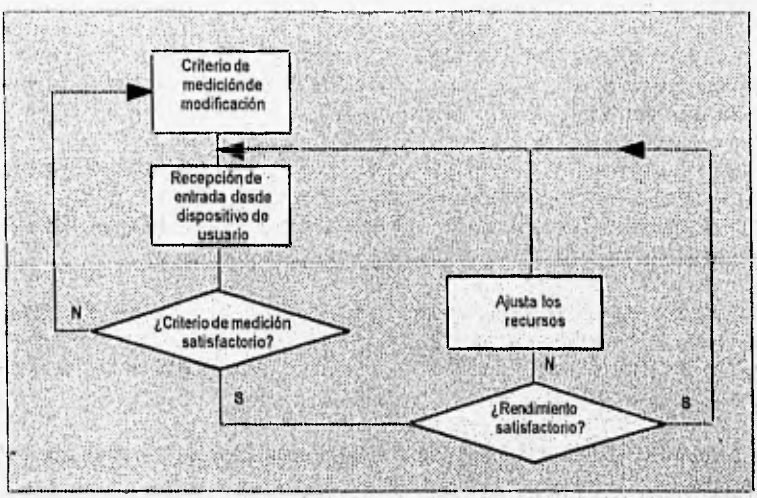


Figura 3.10.b Modelo de sintonía y monitoreo de la administración del rendimiento.

La administración del rendimiento tiene tres funciones primarias: monitoreo, análisis y ajuste. En la siguiente figura se muestra la forma en que se encuentran organizadas las funciones de la administración del rendimiento.

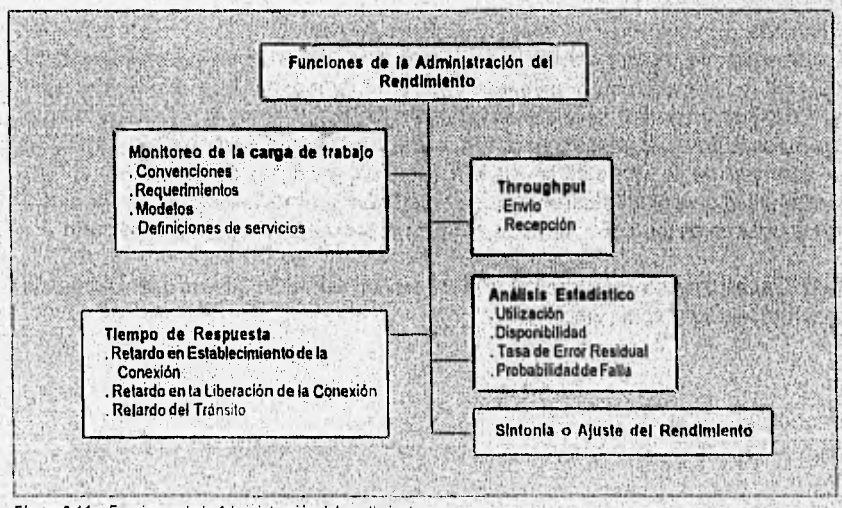


Figura 3.11.c Funciones de la Administración del rendimiento.

Las funciones de la administración del rendimiento se describen en las siguientes líneas:

- Monitoreo de la carga de trabajo: El estándar define un modelo para el monitoreo del rendimiento y para la medición de la carga de trabajo de un MO. El modelo de monitoreo se organiza en torno a los requerimientos, los modelos de monitoreo, así como de las definiciones de los servicios. A su vez, las especificaciones de los requerimientos se organizan en: aviso prematuro de carga de trabajo, limpiado del aviso prematuro de carga de trabajo, sobrecarga, rechazo de aviso, y limpiado de rechazo de aviso. Por otro lado, las especificaciones de la función de monitoreo se organizan en: modelo de carga de trabajo, umbral de carga de trabajo, limpiado de umbral de carga de trabajo, umbral de sobrecarga, modelo de sobrecarga, umbral de carga de trabajo, umbral de pérdida, y limpiado del umbral de pérdida. Por último, las especificaciones de la definición de servicio se organiza en: servicio de reporte de alarma de la carga de trabajo, servicio de reporte de alarma de pérdida, y el servicio de iniciación del umbral de la administración del rendimiento.

- *Throughput*: El monitoreo del *throughput* se realiza con el fin de establecer el mismo dentro de un circuito de comunicaciones o dentro de un nodo de red. En cada dirección de una comunicación se define tanto el *throughput* de envío como la tasa de PDUs transmitidos exitosamente, dentro de una secuencia de unidades de datos provistas en una tasa máxima en el tiempo entre el primero y el último requerimiento para la unidad de transferencia. Ello asume que todas las unidades de medida se transmiten sin errores.

  El *throughput* de recepción es la tasa de PDUs transmitidos exitosamente, durante una secuencia de unidades de datos de servicio, provista en el tiempo, entre el primero y el último requerimiento.

- Tiempo de respuesta: Se define como el tiempo entre la terminación de un requerimiento y la recepción de una indicación o la recepción de una confirmación. El tiempo de respuesta consta del tiempo de establecimiento de la conexión, el tiempo de tránsito y del tiempo de liberación de la conexión.

- Análisis estadístico: Consta de un amplio grupo de actividades utilizada para monitoraer registros y determinar el rendimiento de los MOs. Dentro de esta función se encuentran otros operaciones QOS, como: la utilización, la disponibilidad, la tasa de error residual y la probabilidad de falla.

- Ajuste del rendimiento: Se utiliza para medir el rendimiento de la cola de espera y de los tiempo de espera en la misma.

3.10.3 Configuración

Todo sistema de comunicaciones debe ser capaz de soportar el medio ambiente dinámico. Constantemente ocurren cambios en multiplexores, conmutadores y funciones de modems.

El estándar de administración OSI establece facilidades para reportar en configuraciones físicas o lógicas en un medio ambiente OSI. La administración de configuración es la responsable de los siguientes servicios, por medio de manipulación de la MIB:

- Identificar cualquier MO y la asignación de nombres a los objetos.
- Definer cualquier MO nuevo.
- Colocar los valores iniciales para los atributos de los objetos.
- Administrar las relaciones entre los MOs.
- Cambiar las características operacionales de los MOs y reporta sobre cualquier cambio en el estado de los mismos.
- Borrar MOs.

La especificación de la administración de configuración ISO se vale de muchos otros estándares de administración ISO para la definición de sus funciones.

Los cinco principales estándares del modelo OSI para la administración son usuarios de CMISE. La administración de configuración utiliza los servicos de CMISE que se muestran en la Figura 3.10.d.

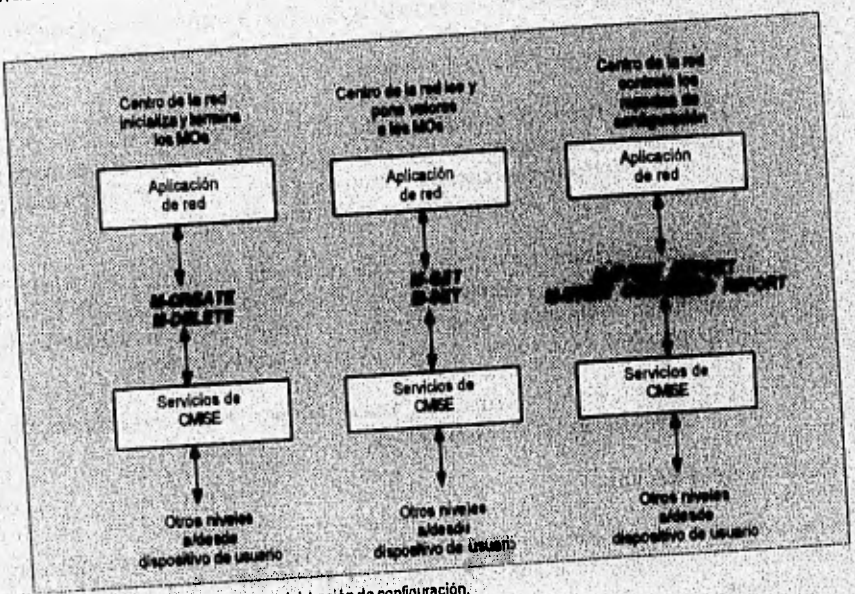


Figura 3.10.d Uso de CMISE en la administración de configuración.

Las aplicaciones de red que se muestran en la figura anterior son *software* escrito para conformar el estándar de administración de configuración.

Los estándares de administración de configuración definen los estados operacionales de los MOs. La administración de configuración relega en ISO 10064-2 para las definiciones de estados.

Se definen cuatro estados operacionales para una administración de la configuración. La Figura 3.10.e muestra las relaciones de los estados operacionales y las transiciones permisibles entre estos estados. Los estados operacionales son:

- Habilitado: El recurso (MO) no se encuentra en uso, pero es operable y se encuentra disponible.
- Deshabilitado: El recurso no está disponible o es dependiente de otra fuente que no está disponible.
- Activo: El recurso está disponible para su uso y tiene la capacidad de aceptar servicios desde otra fuente.
- Ocupado: El recurso está disponible, pero no tiene la capacidad de servicios adicionales.

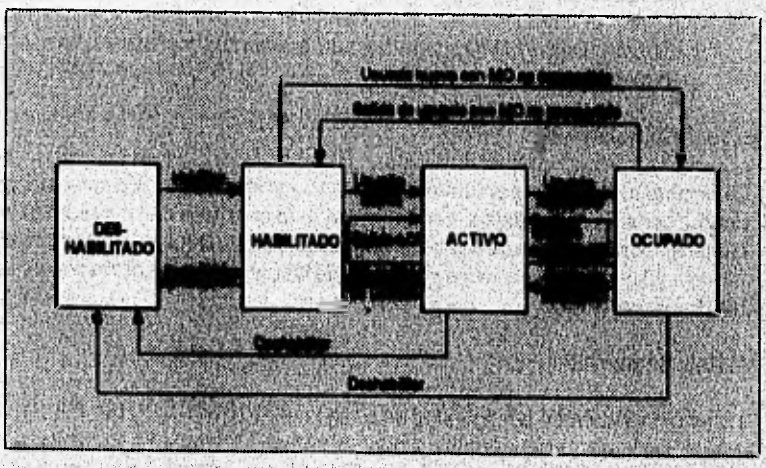


Figura 3.10.e Estados operacionales de la administración de la configuración.

Es importante señalar que no en todos los MOs se presentan los cuatro estados. Por ejemplo, para aquellos MOs que no tienen un límite de usuarios, el estado "ocupado" puede no ser exhibido. Analizando la figura anterior se pueden identificar ciertas reglas en los estados operacionales. Por ejemplo, la transición a un estado "habilitado" significa el llevar a cabo ciertas acciones, las cuales hacen al MO operable. Como se ve en la figura, la transición a "habilitado" solamente puede ocurrir de un estado operacional "deshabilitado". Sin embargo, como también se puede apreciar, existen ciertas acciones que también hacen que un MO pase a un estado "habilitado", como la salida de un usuario por medio del uso de un objeto no compartido desde un estado "ocupado", o la salida de usuario desde un estado "activo".

El estado operacional de "deshabilitado" ocurre en aquellos MOs que se consideren inoperables. Cuando un componente de red presenta algún defecto, entonces se declara inoperable, por ejemplo, un software con errores se puede considerar inoperable, lo que lo convierte en un MO en estado operacional "deshabilitado".

Las siglas CI y CD de la figura, significan Incremento de Capacidad y Decremento de Capacidad, respectivamente.

Por otro lado, la administración de la configuración también se vale de ISO 10064-2 para describir tres estados administrativos:

- Desbloqueado: El MO puede usarse.
- Bloqueado: El MO no puede usarse.
- *Shutting Down*: El MO puede usarse únicamente por el usuario en curso y no por nuevos usuarios.

3.10.4 Contabilidad

La administración de la contabilidad se define en ISO 10164-10, provee un modelo de operaciones de contabilidad, así como una guía en el uso de las unidades de carga, contabilidad de entradas, registros de operación, tasa de cambio, identificadores destino y fuente. El estándar para la administración de la contabilidad es el que menor avance ha tenido dentro de ISO.

3.10.5 Seguridad

La administración de la seguridad se basa en ISO 10164-7, 10164-8 y 10164-9 para establecer los requerimientos de seguridad en la red, tales como: liberación de alarmas, análisis de selección, detección de eventos y organización de estas operaciones. En la siguiente figura se muestran los mecanismos de seguridad.

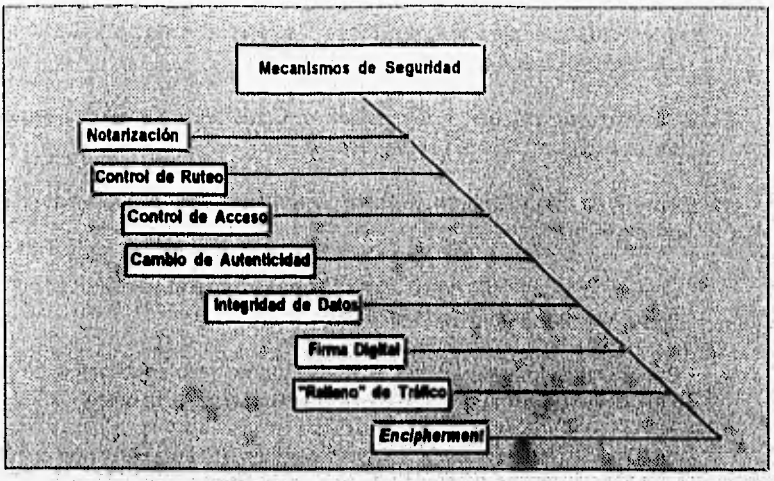


Figura 3.10.f Mecanismos de seguridad.

Dentro de los mecanismos de seguridad encontramos:

- Notarización: Este servicio se encarga de que el *software* de aplicación utilizado garantice la exactitud de la información, tanto en contenido, como en tiempo y liberación.

- Control de ruteo: Contiene reglas que permiten mecanismos de relevación para evitar enlace de comunicaciones de datos o redes específicos con fines de seguridad.

- Control de acceso: Se utiliza para prevenir accesos no autorizados a recursos o prevenir usos de recursos no autorizados.

- Interambio de autenticidad: Se encarga de verificar la identidad de una parte antes de que un acceso a un recurso sea permitido.

- Integridad de datos: Se encarga de asegurar que los datos no se hayan dañado o alterado de alguna manera no autorizada.

- Firma digital: Se utiliza para asegurar que el recipiente de datos sea el propio y que la unidad de datos no haya sido alterada.

- "Relleno" de tráfico: Es el mecanismo en el que a los PDUs se les anexan *bits*, octetos o bloques de datos falsos.

- *Encipherment*: Se encarga de encriptar los datos.

A continuación se describen brevemente los servicios de seguridad de la administración de redes OSI:

- Autenticidad de la entidad *peer*: Este servicio es utilizado para asegurar que la asociación con una entidad *peer* es la debida.

- Autenticidad del origen de los datos: Este servicio se encarga de asegurar que los datos recibidos sean los solicitados.

- Servicio de control de acceso: Asegura que un usuario no autorizado no accese los recursos.

- Confidencialidad de la no-conexión: Se encarga de asegurar que la conexión lógica N de un usuario N sea segura.

- Confidencialidad de campo selectivo: Este servicio se encarga de proveer confidencialidad en ciertos datos dentro de un arreglo de datos.

- Confidencialidad de flujo de tráfico: Se encarga de prevenir que el tráfico no pueda ser analizado de ninguna manera por usuarios externos.

- Integridad de conexión con recuperación: Este servicio se encarga de asegurar que los usuarios en una conexión específica no puedan sufrir modificación, borrado o inserción de la misma, así como que tengan una recuperación de la conexión en caso de problemas.

- Integridad de conexión sin recuperación: Realiza la misma función que la anterior pero sin la capacidad de recuperar conexiones con problemas.

- Integridad de conexión de campo selectivo: Asegura la integridad de campos seleccionados dentro de un SDU contra posibles modificaciones, borrado o inserción.

- Integridad de la no-conexión: Asegura la integridad en un SDU único contra posibles modificaciones.

- Integridad de la no-conexión de campo selectivo: Este servicio asegura que los campos seleccionados dentro de un SDU no sean alterados.

- No-rechazo con prueba de origen: Previene que un nodo ambiguamente identificado tenga derecho a rechazar datos.

- No-rechazo con prueba de liberación: Asegura que los datos del usuario sea liberados correctamente.

Los dos últimos servicios mostrados en la figura de servicios de seguridad son específicos de ciertas redes y no están definidos en el estándar.

CONCLUSIONES

Como se pudo apreciar a lo largo de este capítulo, el modelo de administración de redes de la ISO es un modelo muy completo, pues proporciona una gran cantidad de servicios de administración, aunque como consecuencia, su implementación no es sencilla. Lógicamente el modelo se apoya en el modelo de referencia OSI, sin embargo su implementación se basa en los niveles superiores del mismo. CMISE, CMIP, ACSE y ROSE forman la base de este modelo.

El desarrollo de empresas como British Telecom, DEC e IBM, han hecho que este estándar sirviera de base a otros organismos para el desarrollo de su propio estándar de administración, tal es el caso de los estándares de la Internet y de la IEEE, los cuales se estudian en los capítulos subsecuentes.

CAPITULO

4

MODELO INTERNET PARA LA ADMINISTRACION DE REDES (SNMP Y CMOT)

4. MODELO INTERNET PARA LA ADMINISTRACION DE REDES (SNMP y CMOT)

4.1 LA ARQUITECTURA DE NIVELES Y LOS ESTANDARES DE LA COMUNIDAD INTERNET PARA LA ADMINISTRACION DE REDES

En la siguiente figura se muestran los niveles de Internet para los estándares de la administración de redes. Es preciso señalar que los niveles para el conjunto Internet es más simple que el del ISO. El Protocolo de Administración de Redes Simple (*Simple Network Management Protocol*, SNMP) es la parte fundamental de la arquitectura Internet. Las aplicaciones de administración de redes no están definidas en las especificaciones de Internet, dichas aplicaciones constan de módulos de administración específicos de cada fabricante, tales como: administración de fallas, control de entrada (*log*), seguridad, etc. Como se aprecia en la figura, SNMP se apoya en el Protocolo de Datagrama de Usuario (*User Datagram Protocol*, UDP). El UDP en turno descansa sobre el Protocolo de Internet (*Internet Protocol*, IP), el cual a su vez descansa en los niveles bajos (niveles de enlace y físico).

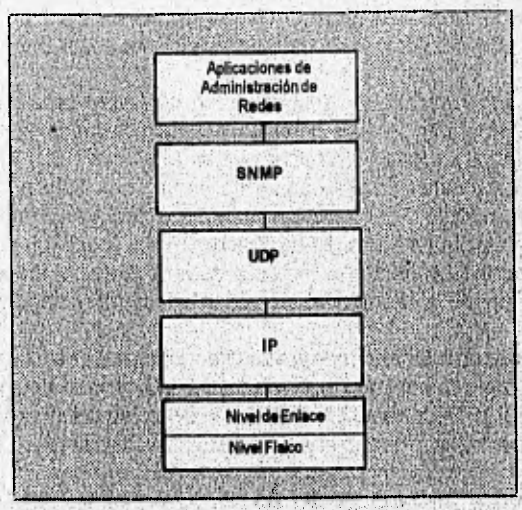


Figura 4.1 Niveles de la administración de redes Internet.

La administración de redes Internet, a diferencia del estándar OSI, únicamente maneja cinco niveles, los cuales obviamente se pueden montar en los siete niveles de OSI. Los niveles físico y de enlace de datos de OSI se agrupan en un solo nivel, mientras que los niveles IP y UDP formarían los niveles de red y transporte del OSI, y el SNMP ocuparía los niveles de transporte, sesión y presentación.

Los niveles UDP e IP son análogos a los tratados en comunicación entre niveles en el estándar OSI.

4.2 REVISION DE LOS PROTOCOLOS DE INTERNET

4.2.1 El protocolo Internet

El protocolo Internet es muy importante, ya que es parte esencial del protocolo TCP/IP, como ya se mencionó en el capítulo dos. Este es un protocolo ruteador y *gateway*, ya que tiene que ver con el tráfico entre redes. IP es un servicio de conexión, el cual permite el intercambio de tráfico entre máquinas, siendo esto posible debido a que el datagrama opera entre dos estaciones o dicho de otra manera entre dos usuarios finales.

Poniendo un ejemplo de la forma en que IP opera, supongamos que tenemos un *gateway* que tiene una longitud de cola máxima, la cual al ser rebasada en su capacidad envía un mensaje de desbordamiento del *buffer*, en cuyo caso el datagrama adicional se descarta en la red; para salvar esta situación, se emplearía un protocolo de un nivel mas alto (UDP) para evitar que se lleguen a descartar los datos. Por otra parte, un IP soporta la fragmentación, es decir que el PDU puede seccionarse en partes pequeñas, para después recuperarse completamente. IP es el encargado de establecer las reglas bajo las cuales se dará la fragmentación, estas reglas se aplican sobre el *gateway*, el cual cumple la función de verificar la compatibilidad de los PDU's, así como de reensamblar los fragmentos del PDU que serán recibidos por el *host*.

El IP hace uso de un mecanismo denominado ruteo fuente (*source routing*) el cual forma parte del algoritmo de ruteo del protocolo. El ruteo fuente permite determinar sobre un Protocolo de Nivel Alto (*Upper Layer Protocol*, ULP) el IP de ruteo del *gateway*. El ULP cuenta con la opción de pasar una lista de direcciones de Internet para el módulo IP. Esta lista contiene el *gateway* intermedio al cual será transmitido el datagrama durante el proceso de ruteo, por lo que si existen varios *gateways* durante la trayectoria, la lista deberá incluir a todos los *gateways* involucrados, siendo la última dirección de la lista la dirección del nodo destino.

Cuando el IP recibe un datagrama que usa las direcciones que se encuentran en el campo de ruteo fuente, para determinar el próximo destino intermedio (al cual deberá de saltar el datagrama), si la dirección no coincide con la dirección del módulo IP, entonces el módulo IP toma la próxima dirección que se encuentre en la lista de destinos del *header* del IP.

El módulo IP es capaz de reemplazar una dirección de la lista de destinos, por alguna de las direcciones que posee. Esta dirección es la que conoce el ambiente cuando el datagrama es recibido. El curso de éste puede incrementarse en un punto para alcanzar la próxima dirección IP en la ruta. Cuando la dirección coincide, el datagrama sigue la ruta fuente determinada por el ULP y además se registra a lo largo del camino, por lo que si se quisiera hacer un recorrido inverso apartir del nodo destino al nodo origen, la información necesaria para efectuarse existiría.

El *gateway* IP toma una decisión de la ruta basándose en la lista de ruteo, en el caso de que el *host* destino se encuentre en otra red, el *gateway* tomará las decisiones adecuadas para encaminar el datagrama a la otra red.

Si la comunicación involucra múltiples destinos, esto complicará la comunicación. Cada *gateway* puede atravesarse y el *gateway* puede tomar decisiones acerca del ruteo.

Cada *gateway* mantiene una tabla de ruteo, la cual contiene la dirección del próximo *gateway*, así como el destino final de esa red. En efecto, la tabla contiene un renglón para cada red. La tabla puede ser dinámica o estática, siendo estipulada dentro del estándar IP para el primer caso. Como se mencionó, cada *gateway* debe de contar con una entrada para cada red, ya que dentro de la tabla de ruteo de éste se maneja un número de red, el cual corresponde a cada salida y la dirección del *gateway* más próximo. El *gateway* vecino es el encargado de hacer las pruebas de ruteo. En caso contrario la lógica del *gateway* del IP establece si la conexión se realiza directamente en esta red.

El IP ruteador se basa en un concepto llamado "distancia métrica". Este valor es realmente el número del último salto entre *gateways* y su valor final. El *gateway* consulta esta tabla de ruteo e intenta igualar la dirección destino contenida en el *header* del IP, con el valor contenido en la tabla de ruteo, si no fueran iguales se les denominará como no encontrada y se descartará el datagrama mandando un mensaje de regreso al IP fuente (*Internet Control Message Protocol*, ICMP). Este mensaje contiene un código destino alcanzable. Sí se encuentra en la tabla de ruteo, el *gateway* de la red direccionará el datagrama hacia el *gateway* de la red a la que corresponda, en caso de que el *gateway* esté conectado al destino directamente, la dirección estará contenida dentro del *header* del IP.

4.2.2 ESTANDARES ADJUNTOS AL IP

En capítulos anteriores se ha explicado la idea de *stack* de protocolos, además de haber examinado algunos de los protocolos más usuales en la industria. En este apartado se tratarán los protocolos que trabajan con IP. En la siguiente figura se presentan dos redes, la red "A" y la "B" las cuales se encuentran conectadas por medio de un *gateway*, tanto en los *hosts* como en el *gateway* se muestran los protocolos que trabajan junto a IP y que como se mencionó, pertenecen a TCP/IP.

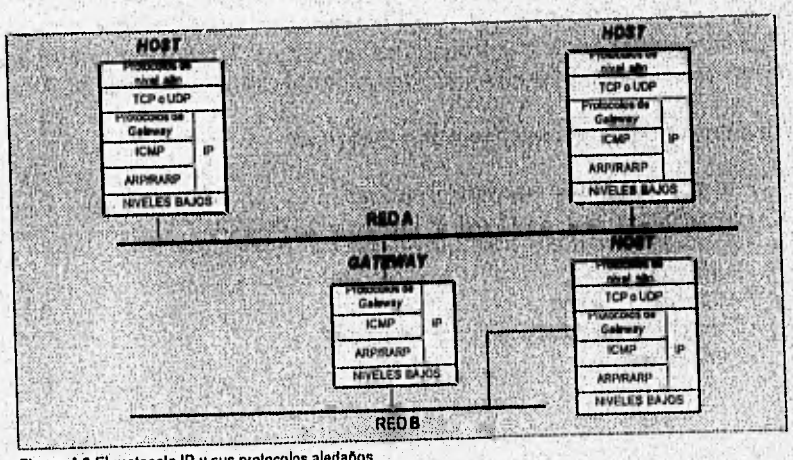


Figura 4.2 El protocolo IP y sus protocolos aledaños

En la figura anterior podemos ver los Protocolos de *Gateway*, ICMP, Protocolo de Resolución de Direcciones/Protocolo de Resolución de Direcciones en Reversa (ARP/RARP).

El IP no participa en el descubrimiento de rutas o la construcción de tablas de ruteo. Por el contrario, el IP usa las tablas provistas por los protocolos de *gateway*. Estos protocolos son responsables de conocer la topología de una red o una inter-red e intercambiar información de ruteo con *gateways* y *hosts*. El protocolo del *gateway* circunda extensamente usando estándares como el Protocolo de Información de Ruteo (*Routing Information Protocol*, RIP), Protocolo de *Gateway* Externo (*External Gateway Protocol*, EGP) y más recientemente, el Protocolo de Primera Trayectoria Abierta más Corta (*Open Shortest Path First Protocol*, OSPF). El ICMP es un compendio de protocolos para el IP. La tarea principal de ICMP es proveer reportes de error y un *status* de información entre el *gateway* y el *host*. ICMP no corrige errores y no recuperá tráfico pasado. ICMP es usado principalmente para reportar problemas o condiciones poco usuales por el módulo IP. El ARP/RARP provee de resolución de direcciones entre el Control de Acceso al Medio físico (*Medium Access Control*, MAC), los cuales contienen un *gateway* o *host broadcasting* sobre un mensaje ARP de estación en la red. El mensaje contiene una dirección destino IP. La estación responde a la dirección MAC y la almacena en la RAM de la estación. Después de esto, dicho mensaje no necesita mandar un *broadcast*, sino que sólo requiere de consultar una tabla de destinos de direcciones IP con la dirección MAC, usando entonces la dirección MAC y el *frame* de la LAN para la transmisión.

El RARP es menos usado, su función es, como se trató en el capítulo dos, opuesta a ARP y se utiliza para obtener una dirección IP a partir de una dirección MAC.

4.2.3 El Protocolo de Control de Transmisión/ Protocolo de Internet (TCP/IP) y el Protocolo de Datagrama de Usuario (UDP)

En la Figura 4.2 se muestra el uso de TCP y el UDP soportados por IP. Como ya se mencionó, el protocolo TCP/IP es un protocolo orientado a la conexión, lo cual proporciona una buena integridad a la inter-red. Una de las funciones de TCP/IP es la de recuperar el tráfico. Esto sólo es posible si el servicio de conexión es invocado por un UDP.

TCP nos provee de los siguientes servicios de alto nivel:

- Administración de conexiones orientadas
- Seguridad en las operaciones de transferencia de datos
- Operaciones de transferencia de datos en flujos-orientados
- Funciones *Push*
- Resecuenciamiento
- Control de flujo (deslizamiento de ventanas)
- Multiplexaje

- Transmisión *Full-duplex*
- Seguridad y precedencia

Debido a que TCP es un protocolo orientado a la conexión, éste mantiene un *status* de la seguridad en el flujo de datos dentro y fuera del módulo de TCP. En este contexto, podemos ver que TCP transfiere datos en una o varias redes, recibiendo las aplicaciones de usuario (o del próximo ULP). Esto se puede entender mejor, si de nueva cuenta observamos la Figura 4.2, en donde el dato enviado entre dos *hosts* es el cruce de una red a otra. Por ser un protocolo orientado a la conexión, TCP es responsable de la seguridad de transferencia de los datos a un nivel alto, por lo que hace uso de números de secuencia positivos y negativos.

Un número de secuencia se asigna a un octeto transmitido. El módulo TCP que recibe los datos utiliza una rutina de chequeo de errores o *checksum*, la cual se encarga de verificar posibles daños en la información durante el proceso de transmisión. Si el dato es aceptado, TCP regresa un ACK. Si el dato se daña, TCP lo descarta y usa un número de secuencia negativo en el NAK. TCP enlaza muchos otros protocolos orientados a la conexión, usa *timers* para asegurar que el lapso de tiempo no sea excesivo antes de la retransmisión del dato o de la transmisión del ACK.

El TCP recibe el dato de un ULP en un flujo orientado. Esta operación contrasta con otros protocolos en la industria de transmisiones orientadas a bloque. Los protocolos orientados al flujo están diseñados para mandar caracteres individuales y no bloques, *frames*, datagramas, etc. Los *bytes* se envían a un ULP sobre un flujo base, *byte* por *byte*. Cuando el flujo llega al nivel TCP, los *bytes* se agrupan dentro de un segmento TCP. Este segmento se pasa al IP (o a otro protocolo de bajo nivel) para su transmisión al próximo destino. La longitud del segmento está determinada por TCP, aunque un desarrollador de sistema también puede determinar cómo TCP tomará las decisiones.

En armonía con la capacidad de flujo de transferencia, TCP soporta el concepto de una función *push*. Esta operación se utiliza cuando una aplicación busca hacer cierta la transmisión de todos los datos a un nivel bajo, en donde TCP es transmitido. Dentro de éste gobiernan los *buffers* administrados de TCP.

Para obtener la función *push*, el ULP manda una primitiva *push* a TCP. La operación requiere que TCP avance todos los *buffers* de tráfico sobre la forma de un segmento o segmentos del destino. El usuario de TCP puede usar una operación de cerrar conexión para proveer la función *push* correctamente.

En suma si usamos el número de secuencia para un ACK, TCP utiliza una resecuencia de segmentos, si es que éstos llegan al destino final en desorden. Puesto que TCP soporta un sistema de conexión, es completamente posible para datagramas sobre una inter-red, TCP entonces elimina las duplicidades de los segmentos.

El módulo TCP es también capaz de controlar el flujo de datos mandados para evitar el desbordamiento de los *buffers* y una posible saturación de la máquina. El concepto usado en TCP se basa en el concepto de ventanas. El emisor transmite un número específico de *bytes* con esta ventana, después la ventana se cierra y el emisor se detiene, dejando de mandar datos. TCP también tiene gran versatilidad en el multiplexaje de varias sesiones con un *host* y con ULP. Esto se realiza a través de algún nombre simple para puertos y *sockets* en los módulos de TCP e IP.

TCP provee transmisión *full-duplex* entre dos entidades TCP y puede trasmitir simultáneamente en ambas direcciones. TCP provee al usuario de la capacidad de especificar los niveles de seguridad y precedencia (priorización de niveles) para la conexión.

4.2.4 Los niveles superiores del protocolo Internet

Se considera a ULP como un protocolo de nivel de aplicación. Los siguientes protocolos se utilizan en los Servicios de Transferencia de Archivos (*File Transfer Protocol*, FTP), el Correo Electrónico (*Simple Mail Transfer Protocol*, SMTP) y el TELNET, que se emplea para dar soporte a la transferencia de archivos, correo y terminales virtuales.

4.2.5 Las operaciones de encapsulación y desencapsulación

El proceso de encapsulación y desencapsulación es muy similar a los ya explicados en la sección de protocolos, por lo que solamente se indicará que, en contraste con el modelo OSI, la encapsulación y desencapsulación del PDU de *Internet* no contiene los niveles de presentación o sesión.

4.3 LA MIB DE INTERNET

La Estructura de Información de Administración (*Structure of Management Information*, SMI) de *Internet* describe el esquema y estructura de identificación para los MOs en una inter-red. El SMI se relaciona principalmente con las áreas administrativa y organizacional. SMI relega la tarea de la definición de los objetos a los otros Requerimientos de Administración de red para Comentarios (*Requests for Comments*, RFCs). SMI describe los nombres utilizados para definir a los MOs. Estos nombres son identificadores de objetos.

4.3.1 Jerarquización

Los objetos en una inter-red tienen muchas características en común a lo largo de las subredes, sin importar su manufactura. Podrían significar gastos para cada organización el ocupar importantes recursos y tiempo en la codificación de ASN.1 para describir a los mismos. Sin embargo, la MIB de *Internet* proporciona un esquema de registro en donde los objetos se definen y se categorizan en una jerarquía de registro.

La siguiente figura muestra el árbol de registro de *Internet* e ISO para la MIB de *Internet*.

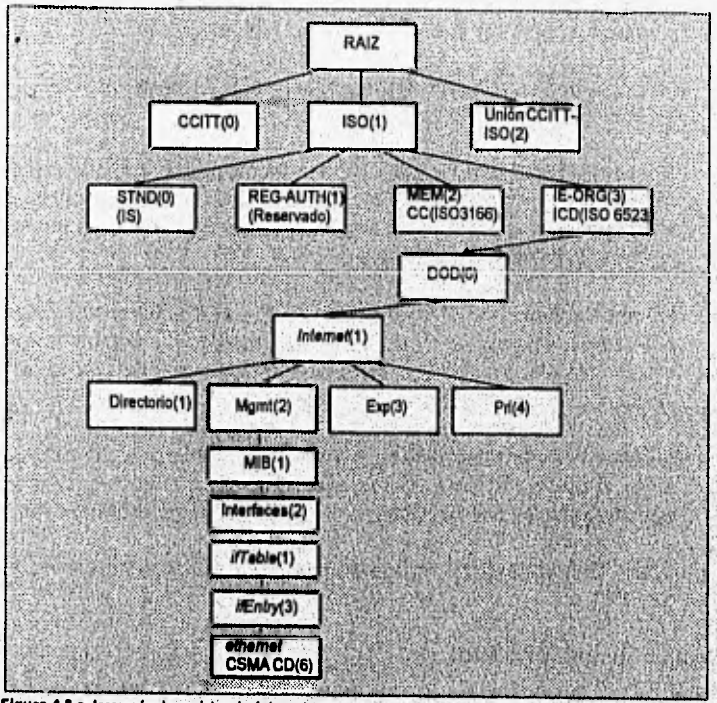


Figura 4.3.a Jerarquía de registro de *Internet*.

En el nivel de red, la jerarquía de registro se identifica, ya sea por CCITT(0), ISO(1) o por la unión CCITT-ISO(2). El ejemplo de la figura muestra la trayectoria desde la raíz hasta la hoja que identifica una interface *Ethernet*. El número de registro completo o valor de Identificador de Objeto para esta hoja es 1-3-6-1-2-1-2-1-3-6. Cuando algún agente genere un reporte acerca de dicha hoja, deberá referirse a ésta por su valor de Identificador de Objeto.

4.3.2 Sintaxis y Tipos

Los estándares de Internet utilizan construcciones ASN.1 en la MIB para describir la sintaxis de los tipos de objetos. Sin embargo, el conjunto completo de instrucciones de ASN.1 no está permitido en *Internet*. Los siguientes tipos de primitivas si están permitidas: *INTEGER*, *OCTET STRING*, *OBJECT IDENTIFIER*, y *NULL*. Además, los tipos de instrucciones constructoras de estructuras de datos permitidas son: *SEQUENCE* y *SEQUENCE OF*.

El estándar SMI define seis tipos mayores de objetos manejados:

- Dirección de Red: Este tipo permite una opción de familia de protocolos *Internet*. El tipo se defino en una notación modificada de ASN.1 como *CHOICE*, el cual permite escoger el protocolo dentro de la familia.

- Dirección IP: Se utiliza para definir la dirección de 32 *bits* de *Internet*. La notación de ASN.1 es un *OCTET STRING*.

- *Ticks* de tiempo: Este tipo representa a un entero no negativo que se utiliza como registro de eventos, tales como el último cambio a un MO, la última actualización a una base de datos, etc. El estándar SMI requiere que se represente un incremento de tiempo de centésimas.

- Medida: La definición SMI para este tipo es de un entero no negativo, el cual puede ser del rango de 0 hasta 2^{32-1}. La medida pueden tomar los valores incrementándose o decrementándose y puede pasar del 0 al valor mayor, o viceversa, pasar del valor mayor a cero.

- Contador: Este tipo definido se describe como un entero no negativo también en el rango de 0 a 2^{32-1}, sin embargo, éste difiere de la medida en que los valores que toma sólo se pueden incrementar y puede pasar del último valor 2^{32-1} a 0.

- Opaco: Este tipo definido permite a un MO pasar cualquier cosa como un *OCTET STRING*.

4.3.3 La Estructura de la MIB

La estructura de la administración de redes *Internet* se encuentra organizada alrededor de grupos de objetos. Actualmente diez grupos de objetos se han definido como miembros de la MIB. En la siguiente figura se muestra la composición de dichos grupos.

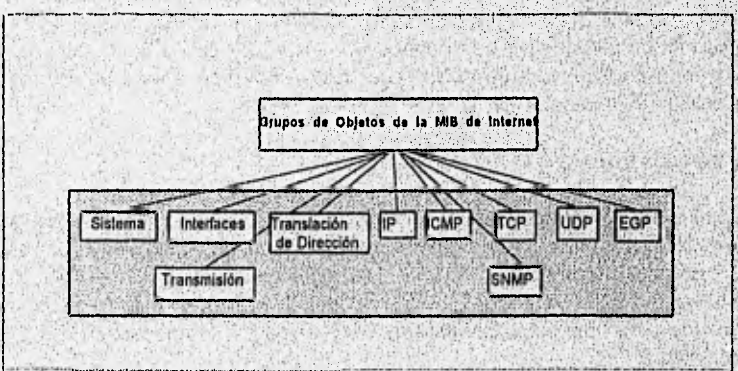


Figura 4.3.b Grupos de objetos de la MIB.

Cada uno de los diez grupos de objetos se definen en detalle en la MIB de *Internet*. RFC 1213 provee una descripción más detallada para cada grupo de objetos.

4.3.4 Grupos y objetos

A continuación se dará una breve descripción de cada uno de los grupos de objetos que forman el estándar *Internet*:

- Sistema: Este grupo describe tanto el nombre y versión del *hardware*, sistema operativo y *software* de red de la entidad, como una indicación de cuándo esa porción de administración del sistema fue reinicializada.

- Interfaces: Describe el número de interfaces de red soportadas, el tipo de interface en operación bajo el IP, el tamaño del datagrama soportado por la interface, la velocidad *bits/s* de la interface, la dirección de la interface, el estado operacional de la interface y la cantidad de tráfico recibido, liberado o descartado, y las razones.

- Translación de Dirección: Este grupo describe las tablas de translación de dirección para una dirección de red a nivel físico, o viceversa.

- IP: Describe si es que la máquina regresa datagramas, el valor de tiempo de vida de los datagramas originados en ese sitio, la cantidad de tráfico recibido, liberado o descartado y las razones, información de operaciones de fragmentación, tablas de dirección y tablas de ruteo, incluyendo dirección destino, distancia métrica, tiempo de la ruta y protocolo desde el cual la ruta fue aprendida (como RIP, EGP, etc.).

- ICMP: Describe el número de los mensajes recibidos y transmitidos, y las estadísticas de problemas contabilizados.

- TCP: Describe el algoritmo de transmisión y los valores máximos y mínimos de retransmisiones, el número de conexiones TCP que la entidad soporta, la información en operaciones de transición de estados, la información en el tráfico recibido y enviado, y el puerto y números IP para cada conexión.

- UDP: Describe la información de tráfico recibido y enviado y la información en los problemas encontrados.

- EGP: Describe información en tráfico recibido y enviado y problemas encontrados, la tabla frontera de EGP, direcciones y fronteras, y el estado EGP para cada frontera.

- SNMP: Este grupo se adicionó a la MIB II. Contiene 30 objetos que se utilizan con SNMP. La mayoría de dichos objetos cuentan con capacidad de reporteo y estadísticas en tráfico SNMP.

4.3.5 Notación para objetos

Los objetos *Internet* se describen utilizando palabras claves reservadas. En *Internet* se utilizan cinco notaciones para describir el formato de los MOs:

- Objeto (descriptor): Describe al objeto en texto ASCII.
- Sintaxis: Describe la representación en trama de *bits* del objeto, sea entero, octeto, etc.
- Definición: Describe al objeto manejado en texto, para que el usuario comprenda mejor la notación.
- Acceso: Este campo describe si el objeto manejado es *read-only*, *read-write* o no accesible.
- *Status*: Se utiliza para describir información, por ejemplo, si el objeto es obligatorio, opcional, o si es obsoleto.

4.3.6 Patrones para definir objetos

Todas las definiciones de objetos se hacen con patrones y código de ASN.1. En este apartado solamente se tratarán los patrones. El formato de los patrones se muestra en la siguiente figura, cuyos campos se definieron en el punto anterior.

OBJETO:

Un nombre para el tipo de objeto con su correspondiente *OBJECT IDENTIFIER*

Sintaxis:

La codificación ASN.1 para describir la sintaxis del tipo de objeto

Definición:

Descripción textual del tipo de objeto

Acceso:

Opciones de acceso

Status:

Status del tipo de objeto

Figura 4.3.c Bases para las definiciones de tipo de objetos de la MIB de *Internet* (IMIB).

Cada uno de los grupos de objetos están definidos en RFC 1213, con este formato de patrón estándar. Para ilustrar más el punto, sobre un ejemplo concreto, a continuación se presenta el patrón del tipo de interface *ifType*, específicamente las interfaces TCP/IP, el cual se mostró en la Figura 4.3.a, como ejemplo de tipo de interface en la jerarquía de registro.

OBJETO:
ifType {*ifEntry* 3}
Sintaxis:
INTEGER {
otro(1),
regular 1822(2),
hdh 1822(3),
ddn-X25(4),
rfc877.X25(5),
ethernet-csmacd(6),
iso88023-csmacd(7),
iso88024-*tokenBus*(8),
iso88025-*tokenRing*(9),
iso88026-man(10),
starLan(11),
proteon-10Mbit(12),
proteon-80Mbit(13),
hipercanal(14),
fddi(15),
lapb(16),
sdlc(17),
t1-*carrier*(18),
cept(19),
Isdnbásico(20),
Isdnprimario(21),
-- y otros más
}
Definición:
El tipo de interface... es decir debajo del nivel IP, en la estructura de niveles de *Internet*.
Acceso:
Read-only
Status:
Obligatorio

La notación de *ifType* {*ifEntry* 3} significa que *ifType* viene siendo la entrada número 3 en el árbol de jerarquización.

La cláusula "Sintaxis" describe el ASN.1 del tipo de objeto. Como la entrada muestra, las interfaces de subred, enlace de datos y físico que existen bajo IP se describen y se asignan

en un valor entero. Así, dos máquinas que intercambian información sobre la interface soportada bajo el nivel IP requieren utilizar estos valores.

4.3.7 El nivel superior de la MIB

En la siguiente figura se muestra la notación RFC 1213 ASN.1 para la MIB. El código puede entenderse en el contexto de los grupos de objetos y del árbol de jerarquización antes vistos.

El enunciado *IMPORTS* designa un cierto número de definiciones que se importan desde el RFC 1155 y 1212. Todos los objetos se designan como *OBJECT IDENTIFIERS* y se definen con otro nombre dentro del árbol de nombres ({mgmt1},{MIB-21}, etc.).

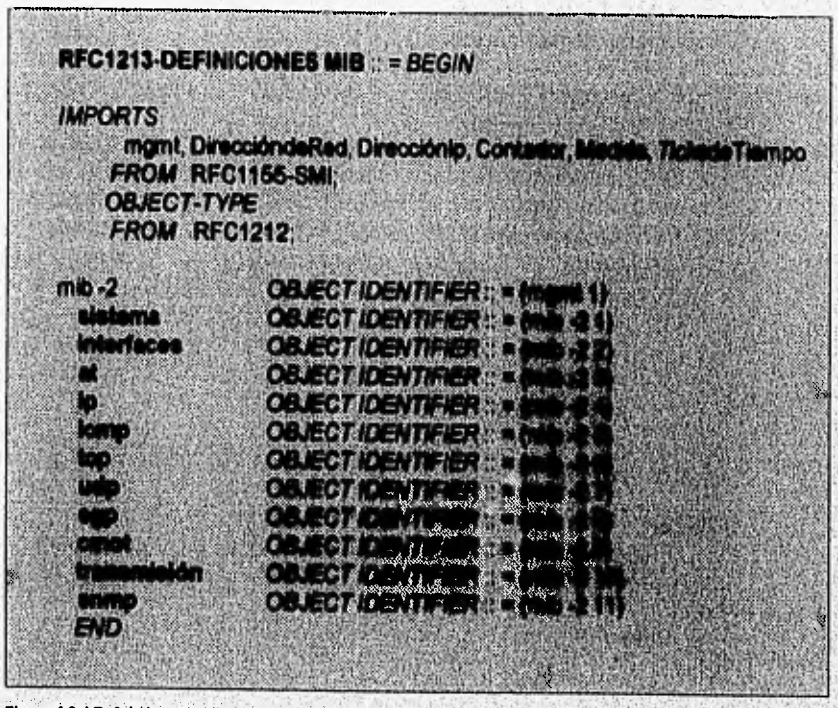


Figura 4.3.d Definición de la MIB de niveles superiores.

4.4 EL PROTOCOLO PARA LA ADMINISTRACION DE RED SENCILLO (SNMP)

4.4.1 Introducción

La arquitectura del Protocolo para la Administración de Red Sencillo (*Simple Network Management Protocol*, SNMP) se encuentra organizada alrededor de los siguientes

conceptos y metas:

- Conservar al *software* agente de administración tan barato como sea posible.
- Soporte de funciones de administración remotas para tomar ventaja del trabajo inter-red.
- Desarrollo de la arquitectura para propiciar un crecimiento futuro, mediante su conservación independiente de *hosts* y *gateways* específicos.

El desarrollo de SNMP comenzó con las investigaciones de varias universidades y laboratorios en los Estados Unidos. Los primeros productos de SNMP aparecieron en el mercado en 1988 provenientes de diferentes compañías dedicadas a conectar redes (Cisco, *Advanced Computer Corporation* y Proteon). Desde entonces, casi todos los fabricantes de productos de interconexión de redes han desarrollado y mercadeado los productos SNMP.

La arquitectura del protocolo está muy relacionada con su predecesor, el Protocolo de Monitoreo de *Gateways* Sencillo (*Simple Gateway Monitoring Protocol*, SGMP), el cual fue desarrollado en 1987. Aunque SGMP sirvió como base para SNMP, es diferente ya que el primero se desarrolló para monitorear *gateways* únicamente, por lo que es muy limitado. Además, SGMP no tuvo un conjunto de unidades de datos de protocolo (PDU's) y no había forma de medir la seguridad y autenticidad de los mensajes.

4.4.2 Relaciones Administrativas

La arquitectura de SNMP utiliza una variedad de términos que se explican en esta sección. Utilizaremos la Figura 4.4.a como punto de comienzo para la descripción.

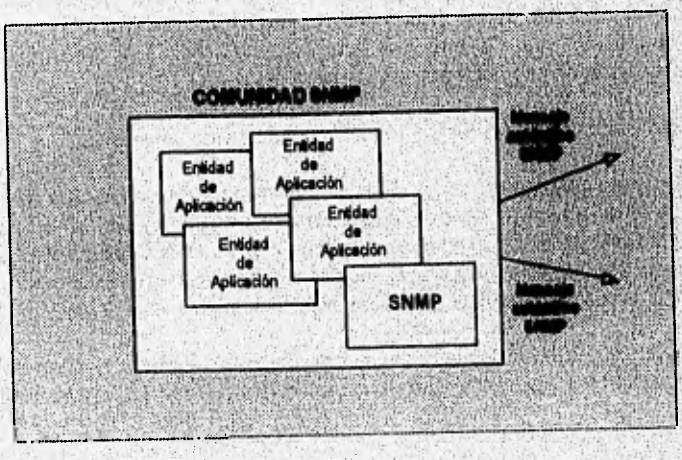


Figura 4.4.a Relaciones administrativas de SNMP.

A las entidades que residen en las estaciones de administración de red y a los elementos de la red que se comunican unos con otros utilizando el estándar SNMP se les llama "entidades de aplicación de SNMP". Al par de entidades de aplicación que forma junto con los agentes SNMP se le llama una "comunidad SNMP". Cada comunidad se identifica por medio de un nombre jeráquico.

Los mensajes de SNMP se originan por medio de entidades de aplicación y el conjunto de ellos forman la comunidad SNMP. A los mensajes que generan las entidades de aplicación se les llama "mensajes auténticos SNMP". Los esquemas de autenticidad se emplean para identificar el mensaje y verificar su autenticidad. A este proceso se le llama "servicio de autenticidad".

Un elemento de SNMP utiliza objetos de la Base de Información de Administración de *Internet* (*Internet Management Information Base*, IMIB). Este subconjunto de objetos perteneciente a este elemento se le conoce coma "vista de MIB SNMP", en cambio, un modo de acceso de SNMP representa un elemento del conjunto (por ejemplo, elementos de sólo lectura, elementos de sólo escritura, etc.). Finalmente, a un par del modo de acceso SNMP con la vista de la MIB se conoce como "perfil de la comunidad SNMP". En esencia, el perfil se utiliza para especificar los privilegios de acceso a la vista de MIB, estas relaciones se determinan por la comunidad SNMP con el desarrollo de perfiles llamados "políticas de acceso SNMP", dichas políticas de acceso establecen las reglas sobre el cómo los agentes SNMP y los elementos de red pueden usar la MIB.

Típicamente, la información de la comunidad y algunos otros datos se almacenan en un archivo de configuración del sistema, el cual contiene información acerca de un usuario y la comunidad del agente. El archivo deberá contener un nombre de comunidad y la dirección del IP de la entidad asociada con la comunidad. Los privilegios de acceso tales como *read-only*, *write-only*, etc., se definen con el nombre de la comunidad. Además, se provee una vista que describe al subconjunto de la MIB que está disponible para este nombre de comunidad. Se incluye una información de trampa; dicha información define a qué comunidad atrapar. También se incorporan los nombres de las variables, ellos llevan la localización y contactos en el sistema.

4.4.3 Estrategia de la administración a través de poleo y trampas

El SNMP opera con dos funciones de administración, la primera es que una entidad interactúa con un agente de administración para recuperar (*get*) variables, en la segunda, una entidad interactúa con un agente para alterar (*set*) variables. La idea de esta simple característica es que ésta limita significativamente las funciones que pueden realizarse con SNMP, el cual como consecuencia, limita la complejidad del *software*.

Estas funciones pueden implementarse por medio de operaciones de poleo. Un administrador SNMP puede programarse para mandar mensajes de poleo periódicamente

hacia los dispositivos de administración en ciertos intervalos, dichos intervalos pueden establecerse a través de la MIB de SNMP.

Este concepto es importante en la evaluación de SNMP por tres razones. Primero, el uso del poleo conserva al sistema relativamente simple. Segundo, como el poleo se controla por medio del administrador SNMP, puede limitar el monto del tráfico de la información de administración que se crea en la inter-red. Tercero, un protocolo de administración de poleo restringe severamente la flexibilidad de los elementos de administración para reactivar las condiciones de una manera adecuada, de esta forma se limita el número de dispositivos que pueden manejarse en la inter-red.

El SNMP no es un protocolo de poleo completo, ya que permite algún tráfico no solicitado llamado trampa, basado en parámetros de restricción. La relación entre el poleo y las trampas se muestran en la Figura 4.4.b.

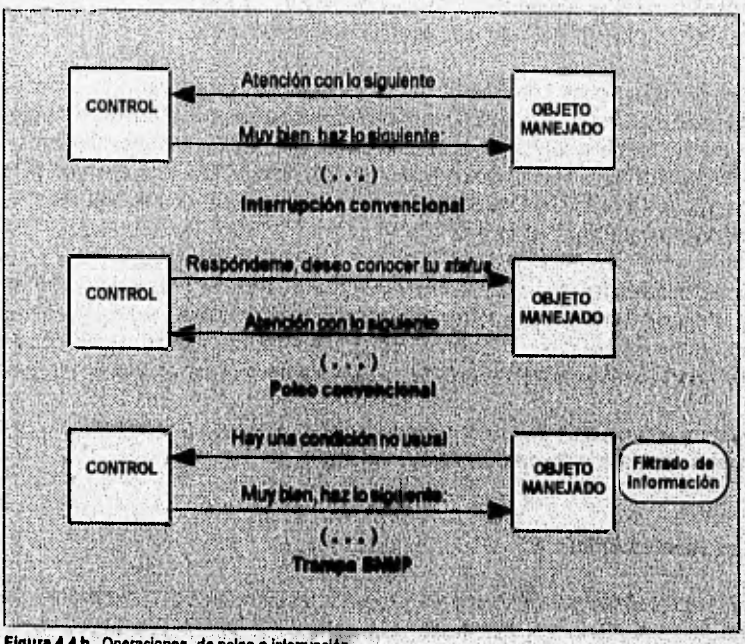


Figura 4.4.b Operaciones de poleo e interrupción.

En esta figura, se muestra un proceso de interrupción convencional, donde el centro de control de la red responde al objeto manejado cuando éste le manda una interrupción. La interrupción es una convención para las comunicaciones de administración entre máquinas. En el dispositivo de control, se recibe el mensaje "Atención con lo siguiente" decide tomar algunas acciones y regresa con el mensaje "Muy bien, haz lo siguiente...".

Las interrupciones no son favorecidas por algunos diseñadores ya que es difícil predecir su resultado. De esta manera, se puede producir un considerable *overhead* en los CPUs de las computadoras de la red, ya que cada interrupción deberá servirse con el uso de ciclos de CPU.

En contraste con el sistema de poleo puro, la máquina de control continuamente envía mensajes (llamados mensajes de poleo o solamente poleo) hacia los objetos manejados. En este ejemplo, vemos el mensaje "respóndeme, deseo conocer tu *status*" el cual forza a una respuesta del objeto manejado.

Las operaciones de poleo (segundo bloque de la Figura 4.4.b) pueden consumir un monto sustancial de *overhead* introduciendo tráfico de datos que no pertenecen al usuario de la red. Estas operaciones son un problema si el poleo se utiliza frecuentemente y no se producen respuestas productivas. Los recursos tales como ancho de banda y ciclos de CPU son consumidos sin ningún efecto positivo sobre la eficiencia de la red. En contraparte, los protocolos de poleo son relativamente sencillos de implementar y permiten al diseñador compilar un perfil del tráfico más predecible.

Otra forma de comunicar el control de la red con uno de sus agentes que usa SNMP es mediante una llamada de interrupción modificada llamada trampa. El agente del objeto manejado es el responsable de realizar chequeos -comunmente llamados filtros- y solamente reporta condiciones que cumplan con un cierto criterio. Por ejemplo, si las colas son grandes al recibir el mensaje "hay una condición inusual", la máquina de control puede decidir tomar ciertas acciones.

Una trampa sigue siendo todavía una interrupción, así es que se han ganado algunas ventajas. Primero la trampa (la interrupción) se realiza sobre pocos eventos críticos y segundo, el mensaje de interrupción es sencillo y pequeño, así, valiosos recursos tales como el ancho de banda y los ciclos de CPU pueden usarse de una manera más eficiente.

Sin embargo, es prudente continuar usando el poleo al menos periódicamente para mantener una actualización de los recursos de la red.

4.4.4 Los Niveles SNMP

El SNMP fue diseñado por *Internet* para usarse sobre inter-redes. Actualmente está diseñado para correr sobre el UDP como se muestra en la Figura 4.4.c. El UDP existe porque no puede ser operado sobre otros protocolos si el desarrollador está dispuesto a realizar algunos módulos interface con el protocolo.

El SNMP usa al UDP como un protocolo sin conexión (*conectionless*). Por lo cual, no hay garantía de que el tráfico de administración se reciba en la otra entidad. Como ocurre con los demás protocolos sin conexión, el *overhead* del procesamiento es reducido, sin embargo, si se necesita una contabilidad, el administrador de red deberá construir operaciones orientadas a la conexión en los niveles superiores de aplicación.

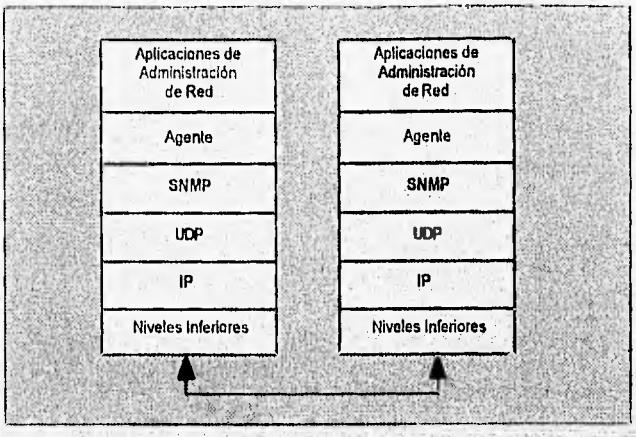


Figura 4.4.c Los niveles SNMP.

4.4.5 Los Protocolos de Datagrama del Usuario (PDUs)

El SNMP utiliza operaciones relativamente sencillas y un número limitado de PDUs para realizar sus funciones. Se han definido en este estándar cinco PDUs, que se muestran a continuación:

- *Get Request*: Se utiliza para accesar al agente y obtener valores de una lista. Contiene identificadores para distinguir las múltiples peticiones, así como valores para proveer información acerca del estado del elemento de la red.

- *Get-Next Request*: Este PDU es similar al *Get Request*, excepto que permite la recuperación del siguiente identificador lógico en un árbol de MIB.

- *Set Request*: Se usa para describir una acción que va a ser realizada sobre un elemento. Típicamente, se emplea para cambiar los valores en una lista de variables.

- *Get Response*: Responde al *Get Request*, al *Get-Next Request*, y al *Set Request*. Contiene un identificador que lo asocia con el PDU previo. También contiene identificadores que proveen información acerca del estado de la respuesta (códigos de error, estado del error y una lista de información adicional).

- *Trap*: Este PDU permite al módulo de administración reportar sobre un evento en un elemento de red o cambiar el estado del elemento de la red.

La Figura 4.4.d muestra el límite entre el *software* de SNMP y el *software* del nivel superior.

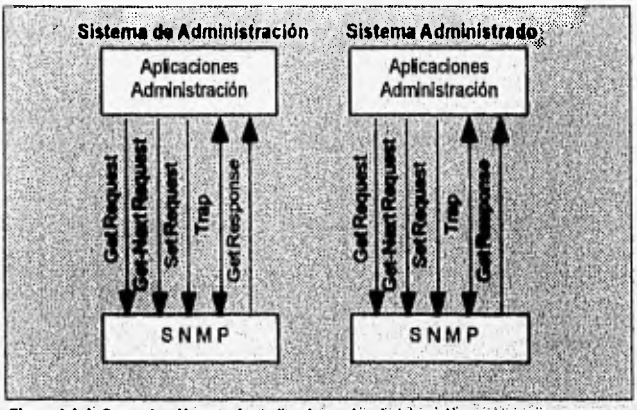


Figura 4.4.d Comunicación entre las aplicaciones de administración y SNMP.

4.4.6 Operaciones entre los Agentes y Administradores de SNMP

La Figura 4.4.e muestra algunas operaciones típicas entre los agentes SNMP y los procesos de administración. Se hace notar que los procesos de administración (*software*) se localiza en la estación de control de red (NCS). La MIB maestra se almacena también en el NCS, sin embargo, ninguno requiere que el *software* de administración y la MIB maestra residan en alguna estación especial sobre la red. Por ejemplo, este *software* y la MIB podrían estar en un *host*.

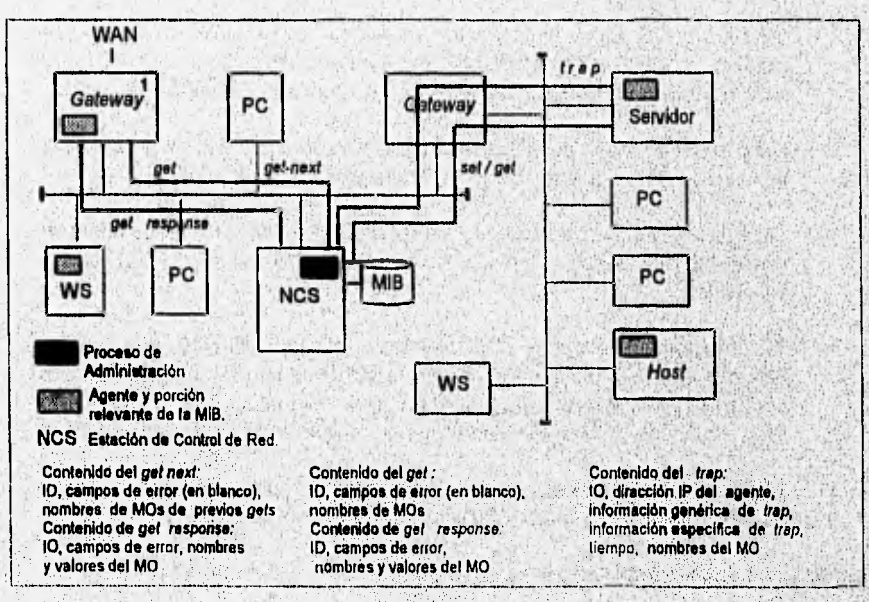


Figura 4.4.e Operaciones de *get, get-next, set* y *trap*.

La figura nos muestra un *get* típico y un *get response*. La NCS envía un *get* al *gateway* 1 de la WAN. El contenido del PDU *get* es un identificador (ID) para determinar de manera única a dicho PDU. También los campos de error están contenidos en el PDU *get*, sin embargo están en blanco para una operación *get*. Los nombres de los objetos manejados (MOs) identifican a la información recuperada en la MIB del *gateway*. Estos nombres deberán codificarse de acuerdo con las definiciones en la MIB.

El *gateway* regresa un *get response*, el cual también contiene el mismo ID con el que fue mandado por el PDU *get*. Los mensajes de error se llenan si algún problema se encuentra; como se dijo antes, estos campos de error se codifican bajo reglas de SNMP. Los valores obtenidos de la MIB son regresados también como los nombres asociados de los MOs.

En la figura anterior también se envía un *get-next*, el cual típicamente se usa para accesar la siguiente instancia de un objeto en la MIB. Este podría accesar al siguiente objeto en la base de datos o podría accesar el siguiente renglón en la tabla. El PDU *get-next* invoca al *get response* en el *gateway*. Las claves para buscar en la base de datos del *get-next* son los nombres de MOs dol previo *get*.

Finalmente se muestra cómo se usa el *trap*. En este ejemplo, un servidor sobre una LAN envía un PDU *trap* hacia el NCS, éste se procesa transparentemente en el *gateway* de la LAN y se pasa al segmento de la LAN sobre la cual está conectado el NCS. El contenido del *trap* consiste de información suficiente para identificar al ID con el propósito de tener un perfil de la comunidad de información, así como la dirección IP del agente que envía el *trap*. Además, la información del *trap* está contenida en el PDU, la cual es una información general perteneciente al *trap* así como específica del mismo. También se incluye al tiempo, el cual representa el momento en el que el *trap* fue mandado. Opcionalmente, cualquier nombre de MO que sea relevante para el *trap* puede reportarse. Como respuesta, el NCS emite un *get* o un *set* para obtener más información acerca de la naturaleza del problema o cambiar algunos parámetros en le MIB del servidor.

4.4.7 Notación para la codificación de PDUs de SNMP

Todos los PDUs de SNMP se codifican basándose en ASN.1, para ejemplificar esto considere las siguientes líneas en donde se muestra el formato común para codificar un PDU.

```
PeticiónID      ::=     INTEGER
EstadoError     ::=     INTEGER {
        sinError(0),
        muyGrande(1),
        nombreNoIdentificado(2),
        valorErróneo(3),
        soloLectura(4),
        errorGenérico(5) }
IndiceError     ::=     INTEGER
VariableInvolucrada ::=   SEQUENCE {
        nombre          NombreObjeto,
        valor           SintaxisObjeto }
ListaVariablesInvolucradas ::= SEQUENCE OF
        VariableInvolucrada
```

Se utiliza al campo PeticiónID para distinguirlo de entre las diferentes peticiones en los PDUs. El campo EstadoError provee una lista para describir el tipo de error que está siendo registrado, dicha lista se accesa por medio del campo IndiceError. El campo sinError simplemente reporta que no hubo errores. El valor muyGrande se usa para reportar que los resultados de una operación no cabrían en una unidad de datos de SNMP. Para indicar al controlador que el nombre de variable no puede identificarse, se emplea al indicador nombreNoIdentificado. Si una variable no puede identificarse o su sintaxis no tiene sentido, entonces se puede hacer uso de valorErroneo. El indicador soloLectura se utiliza para reportar que no puede escribirse en una variable y solamente puede ser leída. Finalmente, el errorGenerico se emplea para reportar cualquier otra evento distinto a los anteriores.

Se identifica el nombre del elemento administrado y cualquier valor asociado a él valiéndose de la secuencia VariableInvolucrada. El ListaVariablesInvolucradas es un conjunto de valores para establecer las relaciones de las variables. Resalta el hecho de que SNMP usa al término "variable" para describir una instancia de un MO, al término VariableInvolucrada para describir la liga de una variable con los valores de dichas variables. El término ListaVariablesInvolucradas simplemente contiene una lista de nombres de variable y sus valores.

La Figura 4.4.f nos muestra la notación ASN.1 así como la representación gráfica de una trampa PDU. Los campos en el PDU llevan consigo la siguiente información:

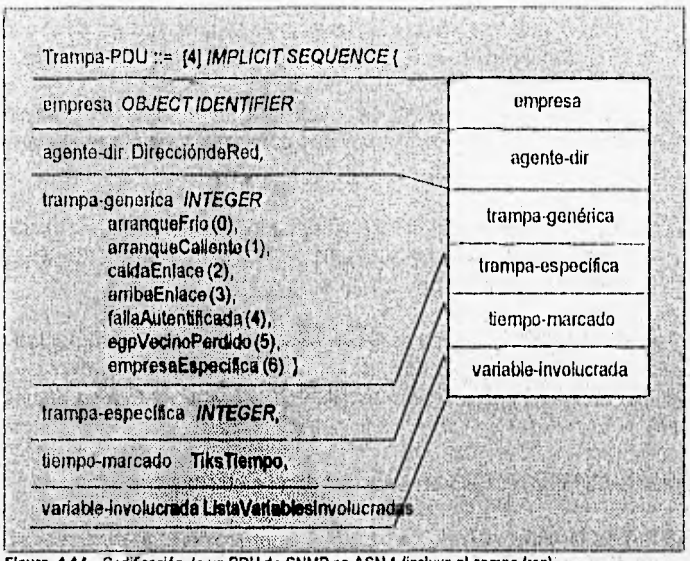


Figura 4.4.1 Codificación de un PDU de SNMP en ASN.1 (incluye al campo *trap*).

- El campo empresa es el tipo del objeto que genera la trampa, se basa sobre el objetoSistemaID.

- El agente-dir contiene el valor de la dirección de red del agente, el cual se utiliza para identificar la dirección del objeto que genera la trampa.

- El trampa-generica contiene información de ciertos eventos:

  – arranqueFrio: Con este campo se indica que la entidad emisora se encuentra reinicializándose.

  – arranqueCaliente: Al igual que con el anterior campo, también se indica que está ocurriendo una reinicialización, sin embargo, éste no afectará las configuraciones.

  – caidaEnlace: Se usa este campo para representar un problema en la comunicación. Los valores que la variable puede tomar son el nombre y el valor de la interface afectada valiéndose de una instancia de *ifIndex*.

  – arribaEnlace: Esta trampa se usa para representar que uno de los enlaces de la comunicación se encuentra disponible para operar. Nuevamente, los valores que puede tomar este campo son el nombre y el valor de la interface con la ayuda de *ifIndex*.

- fallaAutentificada: Este valor de trampa se utiliza para indicar que una dirección de un mensaje no puede ser validado apropiadamente.

- egpVecinoPerdido: Este valor se usa para indicar que un egpVecino fue dado de baja, por lo cual se perdió la comunicación peer que se tenía con él.

- empresaEspecifica: Indica que la entidad emisora reconoció la existencia de un evento. El valor que pueda tomar esta trampa depende del campo específico de la trampa.

El campo trampa-específica se utiliza para identificar el valor que deba tener la trampa empresaEspecífica, dicho valor es cero si las trampas-específicas están inactivas. Finalmente, el campo tiempo-marcado contiene como valor el tiempo del sistema (*sysUpTime*).

4.4.8 Mapeo de SNMP para los servicios del nivel de transporte

El SNMP utiliza los servicios del nivel de transporte, normalmente para colocar a SNMP sobre un nivel de transporte del tipo *connectionless* tal como UDP o un protocolo de OSI en este nivel. Esta operación se muestra en la Figura 4.4.g con un diagrama de secuencia de tiempo. Se pasa al nivel de transporte un PDU de SNMP por medio de la primitiva de petición *T-UNITDATA*. El nivel de transporte se responsabiliza de transportar la unidad de datos hacia la otra máquina llegando al modulo SNMP a través de la primitiva de indicación *T-UNITDATA* la vía *connectionless* requiere de pocos recursos y pocos ciclos de máquina, ya que no necesita establecerse una comunicación directa.

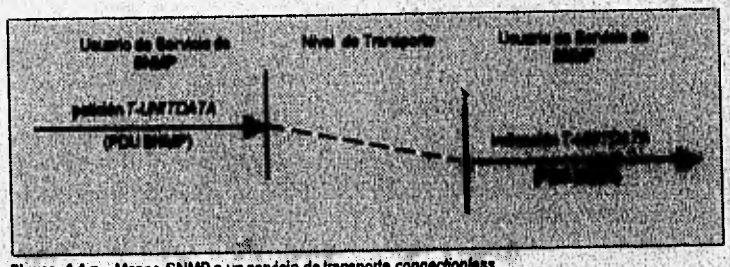


Figura 4.4.g Mapeo SNMP a un servicio de transporte *connectionless*.

Por otro lado, también se usa al SNMP sobre una comunicación del tipo *connection-oriented* del nivel de transporte. Se requiere de primitivas para establecer las conexiones y desconexiones, además deberá existir una conexión entre las dos entidades antes de que las unidades de datos de SNMP sean intercambiadas. Asimismo, este tipo de protocolos requiere de operaciones de desconexión para cualquier recurso que fuere

utilizado durante la sesión entre dichas entidades. Si SNMP desea mandar solamente algunas transacciones, sufrirá un *overhead* y un posible retardo en la obtención de alguna relación entre las entidades.

Ciertamente es posible usar un protocolo orientado a la conexión. Nada impide el establecimiento de una conexión en el nivel de transporte entre dos entidades y simplemente se deja indefinidamente para canalizar el tráfico SNMP. Sin embargo, a pesar de que no se le da demasiado soporte al tipo *connection-oriented*, se tiene la ventaja de permitir al administrador de red obtener una integridad *peer-to-peer*.

En una operación típica, un agente SNMP espera una transmisión de un UDP, TCP o un protocolo de transporte OSI. Cuando se recibe un PDU, se almacena la dirección de la entidad emisora. Su siguiente trabajo es decodificar el datagrama de acuerdo a las reglas ASN.1 y traducirlo a un mensaje SNMP. Se realizan chequeos sobre la versión de los campos y la autenticidad de ellos por medio de un análisis de nombres. Si todo está bien y el PDU es válido y bien formado, entonces el agente usa las variables en la petición para determinar en qué parte de la MIB buscar. Si la operación es un *get*, se turna una respuesta *get* al peticionario empleando ya sea un UDP, TSP o alguna otra entidad del protocolo de transporte.

4.4.9 Otros aspectos de las operaciones en SNMP

Esta sección contiene información adicional sobre las operaciones SNMP. Es importante hacer notar que SNMP no está diseñado para accesar una tabla completa en la MIB, algunas partes del lenguaje y paquetes de acceso permiten al programador accesar todos los renglones y columnas en una tabla con una iteración de un comando *get/put/set*. SNMP accesa un renglón de una tabla a la vez, si el siguiente renglón es necesario, una petición *get-next* se puede usar para obtener el siguiente. El acceso a los datos de un renglón depende del cómo se formulan los argumentos en la petición.

SNMP tampoco fue diseñado para accesar a un nodo dentro de un árbol. Para usar SNMP, el usuario de éste deberá proveer nombres *OBJET TYPE* a un servidor SNMP. Dicho servidor regresa la primera instancia de nombre o nombres *OBJET TYPE* encontrados en la base de datos. Por ejemplo, el comando *get* (tcpConexionEstado) deberá obtener el primer elemento de la variable tcpConexionEstado en la MIB. Como mencionamos anteriormente, los resultados de este valor podrían usarse en la siguiente operación *get-next* para obtener el siguiente elemento en la MIB y así sucesivamente.

SNMP pemite operandos múltiples como argumentos, por ejemplo *get-next* (tcpConexionEstado, tcpConexionDireccionLocal, etc.), puede usarse para obtener múltiples elementos en la base de datos.

Asimismo, se puede usar SNMP para encontrar la localización en la MIB. Por ejemplo, un *get-next* con un argumento que diga ipRuteadorSiguienteSalto.127 recuperará el primer elemento que siga a este valor en la tabla. Si por ejemplo un identificador de red 122 fue almacenado en la MIB y el siguiente identificador fue 127, SNMP podrá recuperar el primer elemento después del 122, precisamente el 127.

Por otra parte, SNMP es un protocolo del tipo *atomic* puesto que realiza acciones completas sobre todas las peticiones o no realiza ninguna de ellas si alguna no la puede hacer.

El acceso que se hace a la MIB se realiza a través de la identificación de instancias de objetos, cada tipo de objeto definido en la MIB se llama por su nombre y se identifica como un *OBJETIDENTIFIER*. Para ilustrar este punto, la Figura 4.4.h nos muestra la MIB para un grupo llamado ip; si deseamos identificar una instancia de ipPoromisionTTL, se podría emplear un esquema de identificación jerárquica con el valor 1.3.6.1.2.1.4.2. Además, si agregamos un cero a este número, SNMP podría entender que es una instancia específica y sólo una instancia ipPoromisionTTL.

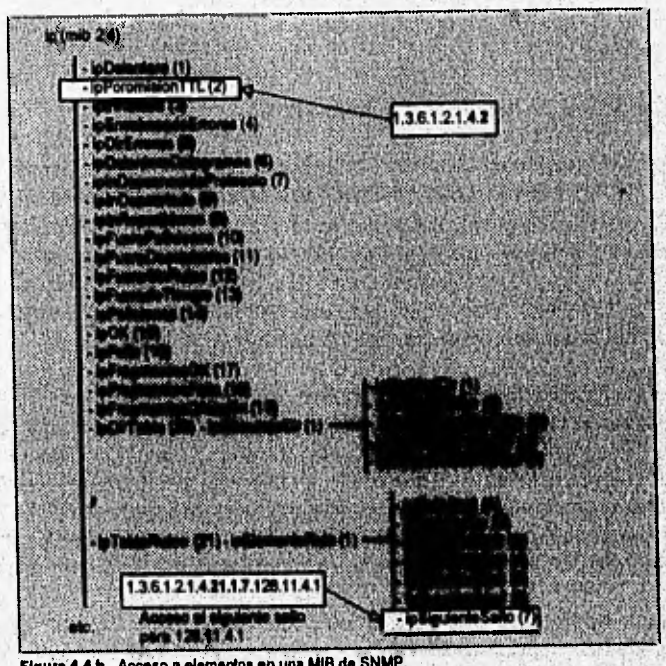


Figura 4.4.h Acceso a elementos en una MIB de SNMP.

La parte inferior de la anterior figura muestra dos formas de identificar una variable. La parte superior del recuadro que esta en la parte inferior de la figura muestra a los nombres

concatenados del nodo en el árbol jerárquico con número Internet conectado como el último número concatenado. Esto pudiera ser empleado para accesar el siguiente salto (la variable ipRutaSiguienteSalto para una dirección internet 128.11.4.1). Si una instancia no existe, la respuesta del *get* regresa un mensaje nohaytalnombre.

Recuerde que si ocurre un error en una petición, el resto de los parámetros en la petición no son procesados. Esto puede crear algunos problemas en el caso de que un usuario deseara obtener la información que haya sido válida. Una solución es usar algún tipo de argumento general para SNMP; un *get-next* (sistObjetoID, sistLocalización) podría buscar repetidamente en la MIB. Como no se definió sistLocalizacion en dicha MIB, la operación no se aborta sino la búsqueda va hacia el siguiente elemento en la base de datos, la cual es siNumero.0. Con un simple chequeo mediante *software* es muy fácil determinar que sistLocalizacion no está disponible, obviamente una búsqueda completa podría producir sistObjetoID.0 y sistLocalizacion.0 tal y como nos lo muestra la siguiente figura.

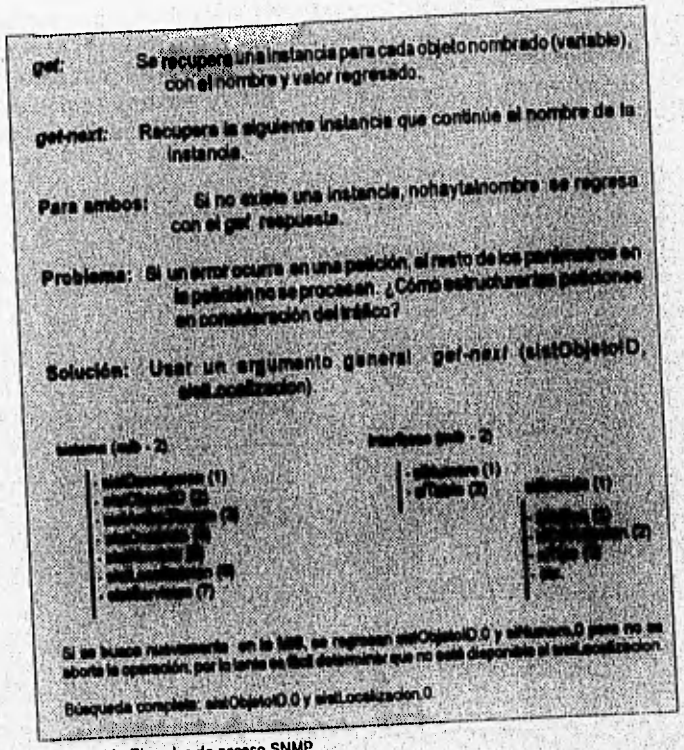


Figura 4.4.1 Ejemplos de acceso SNMP.

4.5 PROTOCOLO COMUN DE INFORMACION Y SERVICIOS SOBRE TCP/IP PARA LA ADMINISTRACION (CMOT)

El Protocolo Común de Información y Servicios sobre TCP/IP para la administración (*Common Management Information Services Protocol Over TCP/IP*, CMOT) fue desarrollado por la Fuerza de Ingenieros de Desarrollo de Internet (*Internet Engineering Task Force*, IETF), y está basado en el estándar de administración de redes de ISO. Este protocolo puede correr sobre el nivel de transporte bajo el concepto de sistema orientado a la conexión. Su principal propósito es describir las reglas de mapeo que pueden usarse con la IMIB y MIB ISO.

4.5.1 Los niveles de CMOT

En la siguiente figura se muestra la arquitectura de CMOT, en ésta podemos ver como ACSE se usa en el nivel de aplicación proveyendo ASEs de servicios al administrador de red y también en este nivel se emplea ROSE.

En la figura se observa un nivel adicional denominado Protocolo de Presentación "*light-weiht*" (*light-weiht Presentation Protocol*, LPP), el cual mantiene comunicación con TCP. TCP y UDP fueron desarrollados bajo la definición de servicios de OSI. La función de LPP es la de interface con los servicios de aplicación de OSI con respecto a los módulos TCP/UDP. El siguiente protocolo es el IP, el cual es transparente al CMOT en los niveles altos.

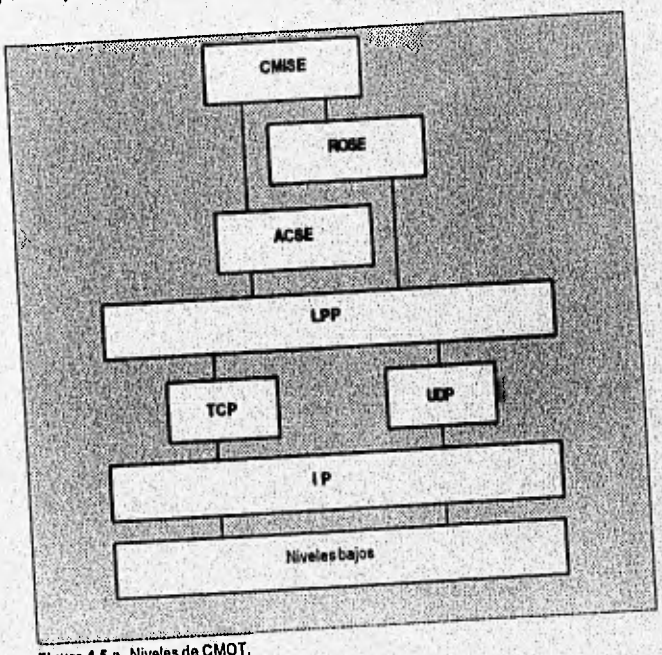


Figura 4.5.a Niveles de CMOT.

4.5.2 El Protocolo de Nivel de Presentación LPP

Como ya se mencionó en el párrafo anterior, LPP funciona como una interface entre OSI y los módulos TCP y UDP. Las especificaciones de este protocolo se encuentran contenidas en la RFC 1085 y se titula " Servicio de Presentación ISO sobre bases *Internet* TCP/IP" (*ISO Presentation Services on Top of TCP/IP-based Internets*). LPP tiene el mapa ROSE y ACSE dentro del nivel de transporte (TCP o UDP). Un camino fácil que se emplea para el protocolo del nivel de transporte en una máquina remota, es mandar un PDU a la máquina remota requiriendo Servicio de Calidad Específico (*Specific Quality of Service*, QOS). Si la máquina no soporta un QOS, puede regresar la primitiva *P-CONNECT* con un valor negativo. Adicionalmente, cuatro números de puertos puede definirse por la interface CMOT con TCP o UDP:

- Administrador CMOT: 163/TCP
- Administrador CMOT:163/UDP
- Agente CMOT:164/TCP
- Agente CMOT:164/UDP

El LPP está diseñado para proveer un nivel mínimo de servicio para las entidades ACSE y ROSE, definiendo cinco servicios del modelo OSI para este fin.

- *P-CONECT*
- *P-RELEASE*
- *P-U-ABORT*
- *P-P-ABORT*
- *P-DATA*

LPP no corre con otros elementos de servicio de aplicaciones, que no sea el Elemento de Servicio de Transferencia de Seguridad (RTSE). En la Figura 4.5.b se muestra la arquitectura para el LPP, y los tres módulos que la componen. Al módulo despachador, que soporta la interface del nivel de presentación del nivel alto, se le denomina *PS-User*. Este usuario es ACSE o ROSE y la interface está definida por ISO 8822 o CCITT X.216.

El módulo de serialización está formado por un serializador y un des-serializador. El serializador acepta un objeto ASN.1 y produce un flujo binario basado en las BER de ISO 8825 o su contraparte de la CCITT X.209. El des-serializador tiene una función contraria al anterior.

El módulo de red provee de interfaces de protocolo de bajo nivel, por tanto maneja un puerto de conexión a TCP o UDP.

Los módulos están ordenados en capas. El LPP no soporta una negociación de servicios, simplemente aprovecha el *software*.

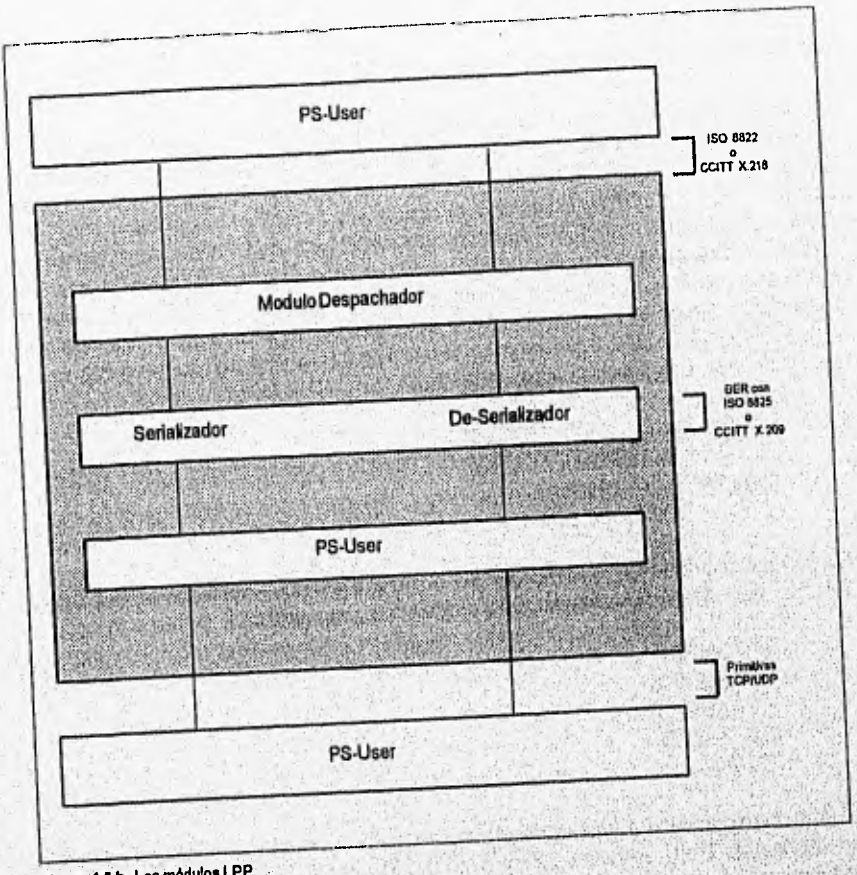


Figura 4.5.b Los módulos LPP

LPP define un subconjunto de parámetros en estas operaciones dentro del nivel de presentación, dichos parámetros son los siguientes:

- Dirección de Presentación
- Lista de Contexto de Presentación
- Datos de Usuario
- QOS
- Versión de Servicio de Sesión

La estructura de la Dirección de Presentación de LPP usa cuatro campos para definir estas direcciones:

1) Direcciones de red
2) Selector de transporte
3) Selector de sesión
4) Selector de presentación

Los tres selectores son valores que son ignorados por LPP. La dirección de red consiste de 32 bits de la dirección IP, 6 bits del número de puerto y un valor de indicación de servicio de transporte para LPP.

La lista del Contexto de Presentación está limitada para IPP, pues sólo tiene dos tipos de contexto: contexto de presentación para el ACSE y el contexto de presentación para ASE.

El Dato de usuario se pasa a LPP por un objeto de nivel alto en ASN.1. Este rubro sólo permite dos identificadores de contexto que son: el contexto de presentación de ACSE y el contexto de presentación ROSE.

El parámetro QOS se encuentra limitado para LPP y sólo se permiten dos parámetros de servicio base TCP y UDP.

La versión del servicio de sesión también puede tener dos valores que pueden usarse con LPP.

CONCLUSIONES

En este capítulo se analizó al modelo de administración de redes de la comunidad Internet con su protocolo SNMP. Este protocolo de administración se apoya en sólo 5 servicios de comunicación, lo cual representa un número menor que los que ocupa el OSI. Su arquitectura es en general más simple que CMIP, ello hace que su implementación sea más sencilla. Esto nos explica la existencia de innumerables productos de administración y análisis de red en el mercado basados en SNMP, pues a diferencia de CMIP/CMISE, SNMP es soportado por cada vez más dispositivos de comunicación (ruteadores, *gateways*, etc.) de distintos proveedores.

CAPITULO

5

MODELO IEEE PARA LA ADMINISTRACION DE REDES (CMOL)

5 MODELO IEEE PARA LA ADMINISTRACION DE REDES (CMOL)

El estándar IEEE se basa en el uso de CMIP/CMISE sobre los siete niveles del OSI. CMISE/CMIP no consume mucha memoria o muchos ciclos de CPU, pero una implementación en los siete niveles OSI, requiere de 200 y 400 *bytes* de RAM. Una aproximación adoptada por IEEE está en sus estándares de administración de redes LAN y MAN (IEEE 802.1B/-15, marzo 1990), en las que se hace una implementación de un protocolo de administración de red sobre los dos niveles bajos del modelo OSI. La IEEE toma una aproximación de CMIP sobre el control de enlace lógico (*Logical Link Control*, LLC), conocido como protocolo de Administración Común Sobre el control de enlace Lógico (*Common Management Over Link*, CMOL). Es importante señalar que el estándar de administración de redes IEEE está incompleto y constantemente surgen cambios en el modelo.

En el presente capítulo se abordarán aquellos tópicos que son de vital importancia en la comprensión del modelo para la administración de redes de la IEEE. En primer término se partirá de un esquema de estratificación de niveles, haciendo analogía con el modelo de referencia OSI. Después se tratarán brevemente los estándares establecidos por el Instituto de Ingenieros Eléctricos y Electrónicos para la administración de redes, así como su estructura, interfaces y servicios, para finalmente abordar el protocolo de administración en sí.

A continuación se presenta una tabla de las siglas que serán utilizadas a lo largo de este capítulo:

Entidad de Administración de Nivel	(*Layer Management Entity*, LME)
Interface de Administración de Nivel	(*Layer Management Interface*, LMI)
Interface de Administración de Nivel en el nivel *N*	(*Layer Management Interface at layer N*, (*N*)-LMI
Administración de Red	(*Network Management*, NM)
Servicio de Datos de Administración de Red	(*Network Management Data Service*, NMDS)
Interface de Servicio de Datos de Administraciones de Red	(*Network Management Data Service Interface*, NMDSI)
Interface de Administración de Red	(*Network Management Interface*, NMI)
Proceso de Administración de Red	(*Network Management Process*, NMP)
Entidad de Protocolo de Administración de Red	(*Network Management Protocolo Entity*, NMPE)
Entidad de Administración de Nivel de Administración de Red	(*Network Management Layer Management Entity*, NM_LME)
PDU de Administración de Red	(*Network Management PDU*, NM_PDU)
Entidad de Protocolo	(*Protocol Entity*, PE)
Requerimiento	(*Request*, RQ)
Respuesta	(*Response*, RSP)

Tabla 5.a Siglas comúnmente empleadas en la administración de redes LAN/MAN en IEEE.

5.1 ARQUITECTURA DE NIVELES EN LA ADMINISTRACION DE RED DE LA IEEE

La arquitectura de niveles de IEEE para la administración de redes se encuentra definida básicamente para redes LAN y MAN. En la siguiente figura se muestra la arquitectura de niveles en la IEEE.

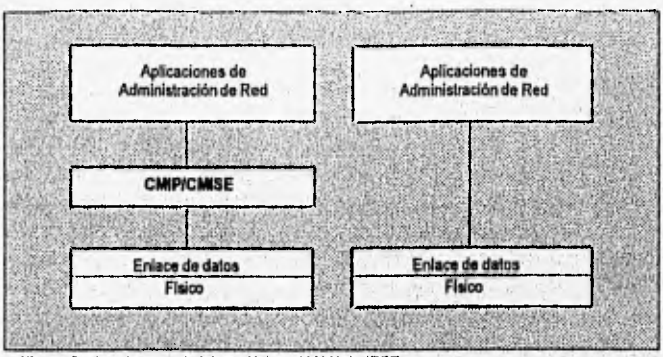


Figura 5.1 Niveles de administración LAN/MAN de IEEE.

El trabajo del estándar de IEEE se encuetra básicamente en la norma IEEE 802.1, y pretende lograr una solución al problema de los productos de multivendedores. El principal atractivo del estándar IEEE es su simplicidad y eficiencia, las cuales se podrán advertir en los apartados posteriores. Como se aprecia en la figura anterior, en el lado izquierdo, la arquitectura de niveles de IEEE es muy sencilla, pues consta básicamente de CMIP montado en los niveles inferiores; para la figura del lado derecho se observa el estándar de IEEE para LAN y MAN, en el cual se aprecian los niveles inferiores (enlace de datos y físico) únicamente, siendo esto alternativo, como se verá posteriormente en los siguientes apartados. Igualmente, cabe señalar que el estándar de la IEEE puede o no utilizar CMIP, y que, aunque está diseñado para operar con sistemas basados en el modelo OSI, también pueden operar en sistemas que no lo estén.

Sobreponiendo esta arquitectura en la del modelo de referencia OSI, podemos deducir que el nivel CMIP/CMISE, corresponde primordialmente a los niveles de red, transporte y sesión.

5.2 LOS ESTANDARES DE LA ADMINISTRACION DEL IEEE

5.2.1 Opciones de conexión con los tipos de LLC 1, 2 y 3

En el ambiente de trabajo de la IEEE, se admite que un sistema orientado a la conexión podría limitar la zona de influencia y el poder de una LAN. Consecuentemente, los estándares de administración del IEEE LAN/MAN usan dos modelos del tipo *connectionless*: modelo *connectionless* sin reconocimiento y modelo *connectionless* con reconocimiento.

Una de las razones para usar estos modelos es que muchas aplicaciones locales no necesitan de la integridad en los datos que provee una red orientada a la conexión. Por ejemplo, un equipo sensor puede perder datos ocasionales ya que las lecturas del sensor se realizan muy frecuentemente, y los datos perdidos no afectan en mucho al contenido de la información. Otro ejemplo de ello son los sistemas de pregunta-respuesta, los cuales,

normalmente,realizan reconocimientos en el nivel de aplicación, dichos sistemas no requieren de servicios orientados a la conexión en los niveles inferiores. Finalmente, una transmisión de voz empacada puede tolerar algún paquete perdido sin que afecte la calidad de reproducción de la voz.

Por otra parte, los procesos de aplicación de alta velocidad no pueden tolerar el *overhead* que se produce en el establecimiento y terminación de las conexiones. El problema es particularmente severo en una LAN, con sus canales de alta velocidad y bajos promedios de error. Muchas aplicaciones de LAN requieren de establecimientos de conexión de unos con otros, otras requieren de comunicaciones muy rápidas entre los DTEs.

Un servicio *connectionless* de reconocimiento es muy utilizado por varias razones. Un protocolo de enlace de datos normalmente mantiene tablas de estado, lo cual podría ser poco práctico para proveer del servicio a todas las estaciones de la red local. Las estaciones de trabajo como las Máquinas de Diálogo Automatizado (*Automated Teller Machines*, ATMs) aún requieren de un proceso de poleo para sus transacciones. La computadora *host* deberá también estar segura de que todas las transacciones sean mandadas y recibidas sin errores. Adicionalmente, el sistema de alarmas necesita algún tipo de reconocimiento para asegurar que la computadora reciba notificación de los mensajes que le envía, pues sería demasiado consumo de tiempo el establecer la conexión antes de mandar el dato alarma.

5.2.2 Clases de servicio

Los estándares de LAN 802 incluyen cuatro tipos de servicio para usuarios de LLC:

- Tipo 1: Servicio *connectionless* sin reconocimiento.
- Tipo 2: Servicio modo-conexión.
- Tipo 3: Servicio *connectionless* con reconocimiento.
- Tipo 4: Todos los demás servicios.

Todas las redes que se basan en la norma 802 deberán proveer servicios *connectionless* sin reconocimiento (tipo 1), aunque de manera opcional también se puede proveer los servicios del tipo 2. Las redes que tienen el tipo 1 proveen de no ACKs, control de flujo y recuperación de errores. Las del tipo 2 proveen administración de la conexión, ACKs, control de flujo y recuperación de errores. Las redes tipo 3 no proporcionan una rutina específica para conexiones o desconexiones, sino un reconocimiento inmediato de unidades de datos. Muchas redes del tipo 1 usan un protocolo de nivel alto (por ejemplo, nivel de transporte) para dar funciones de administración a la conexión.

Los parámetros para esta interface son muy simples y reflejan el deseo de la IEEE de conservar la interface en el protocolo de nivel alto LLC, sólo cuatro parámetros se asocian con cualquiera de las anteriores primitivas. Todo ello forma al CMOL.

5.3 MIBs EN LA IEEE

A lo largo de este trabajo se ha utilizado con cierta familiaridad, el concepto MIB. Se sabe que el estándar OSI ha definido una MIB basada en un árbol jerárquico dinámico, y por ello es conocida como MIT, aunque en algunas referencias se le conoce como Librería de Información de Administración (*Management Information Library*, MIL). El estándar

de *Internet* define su MIB también con una estractura de árbol, y en su primera versión se le conoce como IMIB, mientras que a la segunda versión se le llama IIMIB. Sin embargo, IEEE no ha definido aún una MIB o MIL dentro de su estándar. No obstante la *Internet* toma las normas IEEE 802.5 de *Token Ring* y la IEEE 802.4 *Token Bus* para enriquecer su definición de MIB.

5.4 OPERACIONES DE ADMINISTRACION

Las actuales operaciones de administración de los protocolos del IEEE LAN/MAN son simples. Dichas operaciones son cinco y describen el acceso y manipulación de objetos con un LME, el cual está formado por cinco operaciones ofrecidas en un servicio de administración de redes al NMI. Los objetos no están definidos en la especificación del IEEE 802.1B, pero se encuentran definidas en la sección de administración de la especificación 802. Las cinco operaciones para la administración se muestran en las siguientes líneas:

- *Get operation.* Se usa para obtener un valor de un identificador de objeto.
- *Set operation.* Esta operación se utiliza por el conjunto de valores sobre un objeto.
- *Compare y Set operation.* Se utiliza para ejecutar un conjunto de pruebas y si las pruebas se concluyen, al objeto se le asigna un conjunto de valores particulares.
- *Action Operation.* Esta operación se emplea para ejecutar una secuencia de operaciones sobre un objeto y/o para requerir al objeto transitar hacia un estado identificado. Esta operación requiere de la información que puede regresarse y referirse al suceso de falla de la operación.
- *Event operation*. Esta operación no se inicia por un servicio de usuario, por el contrario, éste es un evento que está localizado inicialmente por el LME, algunas veces enlazado a un mensaje de no solicitado o interrupción.

Existen dos operaciones más, que no requieren de acciones entre NMI, NMP y el usuario final. Estas dos operaciones se especifican en las siguientes líneas.

- *Trace Operation.* Se utiliza para verificar la existencia de una estación remota NMP. Está operación usa PDUs para obtener el servicio. Recibiendo sobre la traza de PDU, el NMP remoto manda una respuesta. La respuesta contiene varios campos de información que son autorizados por NMP, para la identificación de ellos y reportar el *status* de las operaciones. La traza autoriza dos o más estaciones que pueden estar involucrados en la operación. Estos pueden atravesar en cascada una red para reportar su topología.

- *Load Operation.* Esta operación autoriza la transferencia de un bloque de información, usualmente código de máquina, para un servidor remoto de una estación LAN/MAN. En la actualidad la operación de carga es un protocolo más eficiente, llamado sistema de carga de protocolo, esto no se encuentra descrito en el estándar 802.1B del IEEE, pues está tipificado en el estándar 802.1E/D6 del IEEE.

5.5 ESTRUCTURA DE LA ADMINISTRACION LAN/MAN

La estructura de la administración de LAN/MAN (la arquitectura de administración del IEEE para redes LAN/MAN) está dividida en dos partes que son el NMP y los LME´s, esto se puede apreciar en la siguiente figura.

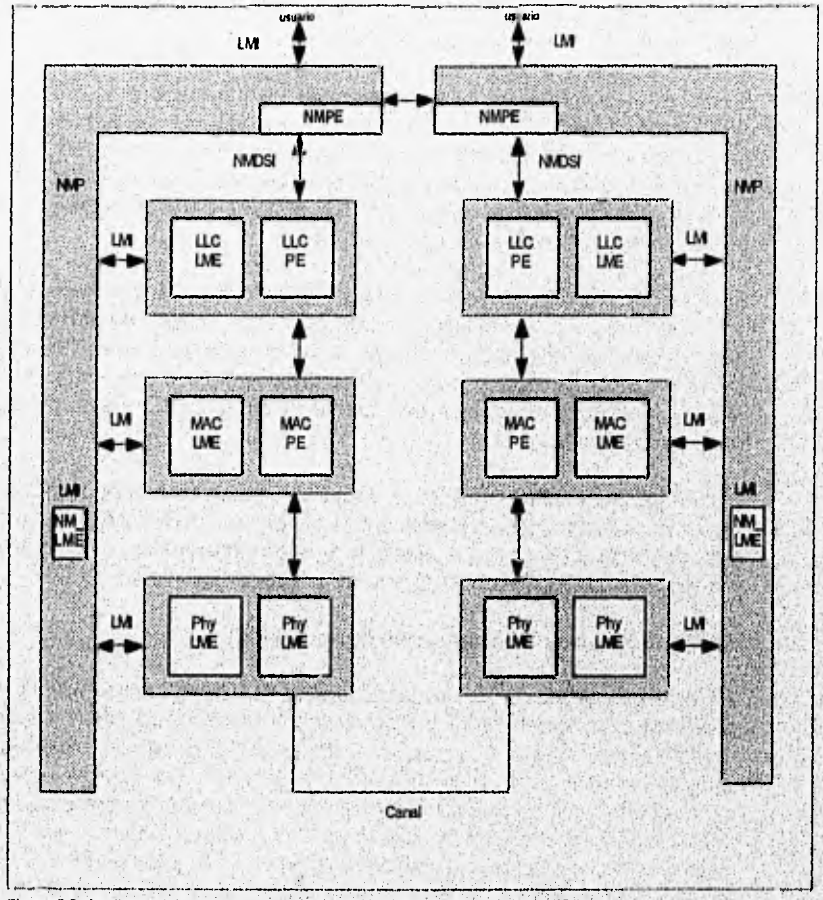


Figura 5.5. Arquitectura de protocolos de administración de redes LAN/MAN del IEEE.

5.5.1 Proceso de administracion de red

El NMP provee el acceso a las operaciones de la administración de red. Estas operaciones reciben la información y ejecución de acciones internas en la estación de trabajo, así como la comunicación con otros NMPs dentro de otros dispositivos sobre la red.

Con respecto a las reglas de administración de redes de OSI, el NMP puede ejecutar el papel de un administrador sobre un agente. Desde el punto de vista del aministrador de red, el agente es local. El servicio solicitado por el administrador mantiene autoridad sobre como ejecutarlo y si la operación se ejecuta actualmente.

El servicio NMI es el puente entre el NMP y el administrador de la red, por lo que el NMI se afecta con la definición de servicios.

El NMP también contiene dos componentes que operan internamente con un NMP, el NMPE y el NMLME, lo cual se puede apreciar en la figura anterior. El NMPE actúa como la entidad de comunicación para una administración de comunicación *peer-to-peer* a través del intercambio de PDUs entre ***NMPs peer-to-peer***. Esta función es similar para un PE, el cual se encuentra descrito en cada uno de los niveles de las estaciones en la LAN.

En la figura anterior se aprecia como NMPE es autorizado para accesar a los servicios de sub-nivel de la estación LAN para comunicación de LLC con el NMDSI. Esta interface está provista con definiciones de servicios (primitivas). Estas primitivas se utilizan para mapear directamente al tipo 1 de LLC *unacknowledged*, información sub-numerada de *conectionless* de las *frames* (*Unumerable Information*, UI).

El NMLME trata sobre la instancia de un LME. Esto se usa para proveer operaciones de administración relativa para el NMP local. El NMLME provee de otras funciones semejantes a un evento de ruteo y operaciones de control de acceso para el administrador de la estación LAN/MAN completa.

5.5.2 La Entidad de Administración de Nivel (LME)

El LME existe en cada nivel dentro de la estación LAN/MAN, lo cual se emplea para proveer funciones de administración de red del nivel específico para la determinación del nivel en el que reside. En el contexto del estándar IEEE, esto se emplea para proveer un administración local de nivel. El principal propósito es proveer información acerca del *status* de la operación del nivel y permitir el ejercicio del control sobre las operaciones dentro del nivel, lo cual provee información alrededor de la existencia de cierta operación con cada nivel. La comunicación de NMP con el LMI, es una interface y está definida como una definición de servicios (primitivas).

5.5.3 La Entidad de Protocolo

En la figura 5.5 se muestra la existencia de una Entidad de Protocolo (*Protocol Entity*, PE) dentro de cada nivel. El PE se utiliza para proveer las funciones específicas de cada nivel, como son la activación de nivel, coordinación de nivel, funciones de control de error, etc. Dichas especificaciones están publicadas por el IEEE en el estándar que define el LLC, Control de Acceso al Medio (MAC), y el subnivel físico descrito en la norma 802.3 y que define el Método de Detección de Colisiones de Múltiple Acceso (*Carrier Sence Multiple Acces Collision Detect*, CSMA/CD) y la 802.5 para *Token Ring*, la cual no está descrita en el estándar 802.1 para la administración.

5.5.4 Relación de los protocolos, interfaces y entidades

En este apartado una vez más haremos referencia a la figura 5.5, en la cual podemos ver que todas las operaciones están constituidas a su vez por dos más pequeñas, denominadas local y remota. Con la operación local, el NMP recibe peticiones del administrador de la red acerca del NMI y decide con el LME local si el servicio es solicitado. Como se puede ver NME realiza operaciones de ruteo con lo seleccionado por LMI. Para operaciones remotas sobre un igual NMP, éste invoca al NMPE, que transfiere un PDU a la interface de NMDSI de la actual estación LAN de nivel. Al recibir el final el PDU pasa a través del nivel físico y de los niveles MAC y LLC al NMPE. El tráfico pasa al NMP, para que éste proceda a tomar una decisión acerca del ruteo de la solicitud respectiva al LMI y obtener la próxima información.

5.5.5 Operaciones entre PEs y LMEs

Las operaciones de administración entre el PE y el LME no se encuentran definidas en el estándar 802.1 del IEEE. Varios de los estándares del IEEE definen un número de operaciones que existen entre estas dos entidades.El efecto de las operaciones entre dos entidades es revelado por el LMI dependiendo de la naturaleza de la emisión de la solicitud para el usuario final y cómo éste es usado por el NMP.

5.6 INTERFACES Y SERVICIOS

Se definen en el estándar IEEE tres interfaces de servicio, el NMI, el LMI y el NMDSI. Estas interfaces contienen varios servicios los cuales se muestran en la tabla Tabla 5.6.

Dichos servicios serán explicados en el siguiente apartado.

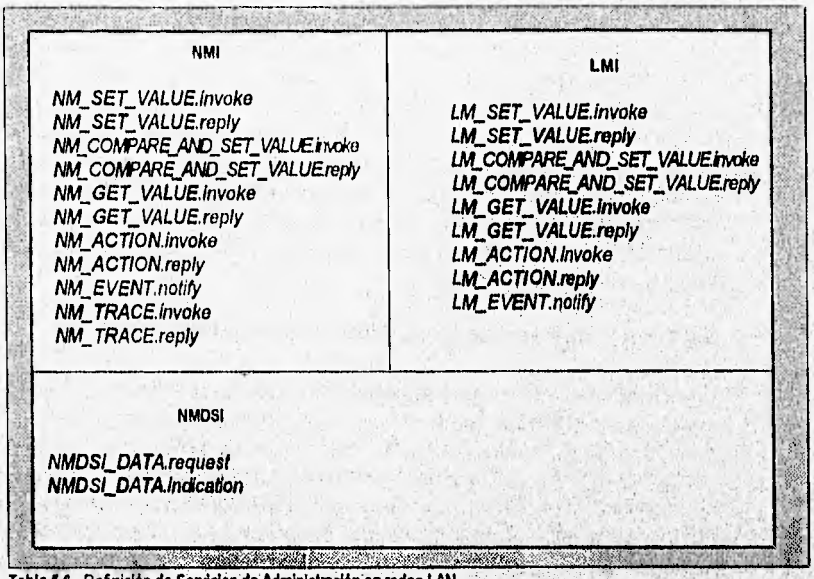


Tabla 5.6 Definición de Servicios de Administración en redes LAN.

5.7 OPERACIONES Y PARAMETROS ASOCIADOS CON LOS SERVICIOS.

Esta sección examina las operaciones y parámetros asociados en cada una de las tres interfaces. Recuerde que los servicios de administración de red se invocan en las interfaces NMI, LMI y NMDSI.

5.7.1 La interface NMI

La única excepción de las operaciones de administración NMI se encuentra en las tres primitivas *NM_EVENT.notify*, *NM_TRACE.invoke* y *NM_TRACE.reply*.

El NMP que sea invocado por una primitiva realizará varias operaciones de edición para tener la certeza de que la llamada es correcta. Entonces se examina la dirección destino, y si es local, invoca los servicios del LME apropiado valiéndose de las primitivas de definición de LMI. La operación LMI se identifica por un parámetro identificador del recurso. Sin embargo, si la dirección destino es remota, el NMP deberá mandar el tráfico hacia otra estación de la LAN. Consecuentemente, el NMP es el responsable de la creación de un PDU apropiado y del envío de este *NM_PDU* a la estación vía NMPE. Como veremos después, el NMPE pasa este PDU al LLC PE por medio de su primitiva de definición en la interface NMDSI. Por el contrario, el NMP manda primitivas réplica hacia el usuario, las cuales se reciben por primitivas del LME a través de la interface LMI o por medio del arribo de un PDU de LLC a NMPE por medio de la interface NMDSI.

Como se mencionó anteriormente, tres de las primitivas de definición de servicios se manejan de manera diferente a la descripción general previa. La *NM_EVENT.notify* es una operación no solicitada. Está dada por un usuario final por medio de un NMP seguido por un PDU de evento de una estación remota a través de la interface NMDSI al NMPE. Alternativamente, esta primitiva podría darse al usuario final ya que un evento local ocurrió y el NMP fue notificado por un LME valiéndose de una primitiva LMI.

El último conjunto de primitivas interactúa con las operaciones de trazo. En la recepción de un *NM_TRACE.invoke*, el NMP formará un PDU trazo y lo transferirá a una estación remota vía el NME. Posteriormente, se envía un *NM_TRACE.reply* al usuario final en el momento de recibir un PDU respuesta trazo de una estación remota.

Con esta información presente, procederemos a examinar las primitivas, sus parámetros y las operaciones asociadas con más detalle.

Varios parámetros están asociados con las primitivas NMI. Para simplificar este análisis, considérese que no todos los parámetros se utilizan en todas las primitivas y los estándares deben ser estudiados si se necesitan más detalles.

- *Access_control_information:* Provee un acceso a la información de control, el cual puede usarse para otorgar o denegar el acceso a la información de administración contenida en el interior de las entidades de administración o bien permitir o negar operaciones para ser realizadas por los LMEs.

- *Actual_quality_of_service* (QOS): Especifica la calidad de servicio que pertenece a la información asociada con la primitiva. Esto es, el QOS indica que fue provisto como un resultado de una operación de petición. Solamente es relevante si el *source_address* es remoto. A este campo se le da el parámetro *quality_of_service* por medio de una primitiva de invocación.

- *Destination_address:* Identifica al NMP si es local para el valor en el *resourse_identifier.*

- *Exchange_identifiers:* Estos parámetros son opcionales, pueden usarse en las primitivas de invocación y en la réplica. Se usan para igualar las peticiones a las respuestas y consiste de un identificador del recurso, el cual se emplea para el origen *NM_user*, y un identificador de transacción que identifica de manera única esta transacción individual.

- *Operation_status*: Se utiliza para indicar la terminación completa o falla de una operación. El parámetro se codifica y se usa por una de las seis acciones de la operación: *GET, SET, C&S, ACTION, EVENT, TRACE.*

- *Parameter_list:* Se usa en una primitiva de invocación para identificar los elementos que serán asociados con la operación de petición. También está presente el *parameter_list* en una primitiva de réplica para indicar las operaciones realizadas. Además, este parámetro contiene información relativa al suceso y la razón de la falla de una operación en relación con un parámetro.

- *Action_list:* Identifica las operaciones sobre cada objeto y está presente en la primitiva *NM_ACTION.invoke.*

- *Action_result_list:* Este parámetro se regresa por la acción de una primitiva de réplica indicando cuál de las operaciones fue realizada.

- *Event_identifier:* Se usa junto con la primitiva notificadora de eventos para almacenar información relativa al estado del recurso que está siendo reportado.

- *Report_address:* Se utiliza con una primitiva invocadora de trazo para proveer información relativa a la dirección del reporte. Esta es un indicador del dónde puede ser enviado el estado del reporte de una operación de trazo.

- *Resource_identifier:* Identifica la entidad que examina la información del recurso específico que está contenido en los parámetros de la primitiva. Este podría actuar sobre la información del control de acceso, el *parameter_list*, acción ID, etc. Además actúa en favor de la dirección destino del NMP.

- *Resourse_required_flag:* Indica si un NMP que se está recibiendo se requiere para responder a una operación de petición.

- *Source_address:* Se utiliza para identificar al originador de la operación.

- *Trace_operator_list:* Se usa en la primitiva invocadora de trazo para proveer información acerca del operador *list*. Este contiene direcciones MAC de las estaciones que están participando en el trazo, así como una descripción de la operación a ser realizar.

- *Trace_report:* Proporciona información en una réplica trazo, el contenido depende de los parámetros en el operador *list* que está contenida en la invocación de trazo.

5.7.2 La Interface LMI

Las operaciones LMI actúan únicamente sobre LME y el NMP. Por lo tanto, las operaciones trazo no se realizan ya que su propósito es obtener información de estaciones remotas. Consecuentemente, las operaciones de LMI a NMP son de sólo naturaleza local. Como se mencionó antes, muchas de las operaciones NMI encuentran camino a sus correspondientes operaciones LMI.

Las funciones de las operaciones mayores, tales como *GET*, *SET*, *COMPARE* y *ACTION* reflejan todas las operaciones de administración de este estándar. Por otro lado, esta sección deberá ayudar a examinar los parámetros específicos asociados con las primitivas. De nueva cuenta es importante notar que no todos los parámetros están asociados con todas las primitivas.

- *Parameter_identifier:* Se utiliza para identificar cada parámetro existente asociado con una primitiva. Este parámetro determina qué objeto manejado tiene una acción realizada sobre él, tal como un *GET* o un *SET*, éste podría asociarse con el *parameter_list* en las primitivas NMI.

- *Access_classes:* Se utiliza para realizar operaciones de control de acceso. Aunque no se describe en el estándar, este parámetro se asocia con el parámetro *access_control_information* en las primitivas NMI.

- *Status:* indica la terminación completa o la falla de la operación. Se asocia con el parámetro *operation_status* en la primitiva NMI.

- *Parameter_value:* Especifica el valor actual del identificador del parámetro *requested layer.* Por supuesto, este valor solamente tiene significado si se completó la operación.

- *Test_parameter_identifier* y *test_parameter_value:* Se utilizan durante las operaciones de prueba en la interface NMI.

5.7.3 La interface NMDSI

Como se vió en la Tabla 5.6 hay sólo dos definiciones de servicio en la interface NMDSI, las de petición e indicación. Estas primitivas se generan cuando al NMPE se le ordena por el NMP, que mande un PDU, o si el NMPE recibe un PDU de otra estación.

La dirección fuente especifica la dirección de enlace de datos del originador NM PDU. La dirección de destino especifica la dirección de enlace de datos a donde está destinado el NM PDU. El campo *priority* indica la prioridad asociada con el elemento QOS de una primitiva NMI asociada. El NM PDU es simplemente la información de administración que se envía a la dirección destino.

5.8 EL PROTOCOLO DE ADMINISTRACION

Como ocurre con todos los protocolos estratificados de OSI e IEEE, el estándar de administración de red LAN de IEEE incluye un protocolo que opera en conjunción con las definiciones de servicios. Su propósito es permitir al NMP transferir PDUs a otro NMP con un *NM_user.* El protocolo define los procedimientos para los administradores (los NMPs que requieren una operación) y los agentes (aquellos NMPs que actúan sobre el requerimiento o que generan un evento).

Este protocolo se encuentra organizado alrededor del concepto de "intercambios de administración" entre administradores y administradores y/o administradores y agentes. Existen cinco procedimientos definidos para cada intercambio de administración que son los siguientes:

- El procedimiento "carga" (*load*): Se utiliza para transferir información (generalmente código de programa) a otra máquina en la red.

- El procedimiento "requerimiento/respuesta" (*request/response*): Se utiliza por los administradores y agentes para informarse uno al otro acerca de las operaciones.

- El procedimiento "privado" (*private*): No se encuentra definido en el estándar.

- El procedimiento "evento" (*event*): Lo emplea un agente para enviar información no solicitada a un administrador.

- El procedimiento "trazo" (*trace*): Lo emplea el administrador para dirigir a un agente y así examinar los recursos en la red.

Para soportar estos procedimientos, los PDUs se intercambian entre los administradores y los agentes. Los PDUs se clasifican ampliamente como sigue:

- *Private*PDU: Este PDU no está definido en el estándar y su implementación es específica. Su inclusión es para poder tener uso de etiquetas privadas.
- *Load*PDU: Se incluye dentro del Protocolo de Carga del Sistema (IEEE 802.1EEE).
- *Request*PDU: Este PDU se envía desde un administrador a un agente para requerir alguna operación.
- *Response*PDU: Lo envía un agente a un administrador en respuesta a un *Request*PDU.
- *Event*PDU: Es un PDU no solicitado, desde un agente para informar a un administrador acerca de algo que ha ocurrido.
- *Event*ACKPDU: Se trata de un PDU de reconocimiento del *Event*PDU.
- *Trace*RQPDU: Es un PDU que envía un administrador a un agente, el cual a su vez puede enviarlo a otro agente. Su función es la de permitir operaciones de diagnóstico y chequeo.
- *Trace*RSPPDU: Este PDU reconoce al *Trace*RQPDU.

Dentro de un NM-PDU a la codificación de una operación se le conoce como nivel-operación PDU. A continuación se presentan los nivel-operación PDUs de los distintos tipos de PDU antes definidos:

*Request*PDUs:	*PrivateRQ, GetRQ, SetRQ, CompareAndSetRQ, ActionRQ.*
*Response*PDUs:	*PrivateRSP, GetRSP, SetRSP, CompareAndSetRSP, ActionRSP.*
*Event*PDUs:	*EventInfo.*
*Event*AckPDUs:	*EventAckInfo.*
*Trace*RQPDUs:	*TraceRQ.*
*Trace*RSPPDUs:	*TraceRSP.*

CONCLUSIONES:

El estándar de administración de redes LAN/MAN de IEEE está basado en la arquitectura IEEE 802 así como en el modelo OSI. Este estándar está diseñado para trabajar con o sin CMIP en el nivel LLC por lo cual se llamará CMOL; sin embargo, su diseño permite hacer uso de la interface CMISE. Por otro lado, mientras no se estabilice este estándar, el SNMP de *Internet* puede operar sobre ellos.

Como se pudo constatar en el presente capítulo, el estándar de administración de redes LAN/MAN de IEEE no se encuentra aún definido en todas sus partes, por lo que prácticamente no existen productos en el mercado que se basen en él, sin embargo fue importante elaborar su estudio.

CAPITULO

6

REVISION DE PRODUCTOS PARA LA ADMINISTRACION DE REDES EN EL MERCADO Y SU APLICACION EN LA INSTITUCION POLITICA

6. REVISION DE PRODUCTOS PARA ADMINISTRACION DE REDES EN EL MERCADO Y SU APLICACION EN LA INSTITUCION POLITICA

6.1 IMPLEMENTACION DE ESTANDARES

6.1.1 Introducción

En el presente capítulo se hará un breve estudio de los principales productos de administración de redes en el mercado, poniendo de relieve sus características más notables. Se presentan plataformas para los tipos de red más representativos, como son TCP/IP, AppleTalk, IBM, etc.. Posteriormente se hará una descripción de las características de la red de la Institución Política, para que en base a ella, finalmente se haga la elección de alguno de los productos.

Con la ayuda de los conocimientos adquiridos en los cinco capítulos anteriores, la comprensión del funcionamiento de cada uno de los productos será más fácil, lo cual nos proporcionará elementos para tener una decisión acertada en la elección de alguno de estos, para ser aplicado en la red de la institución política.

Antes de continuar será importante retomar el cuestionamiento ¿Qué es la administración de red?

- Una entidad que monitorea y reporta el estado de operación de la red.
- Un mecanismo que asiste al administrador en la determinación de problemas, detección de fallas y su corrección.
- Un sistema de administración de red utilizará toda la información recolectada para determinar el rendimiento, *throughput* y sobre todo la eficiencia.

Una red sin un sistema de administración se puede comparar con un avión volando sin instrumentos. Las redes de hoy en día están constituidas de muchos elementos diferentes: cables conectados en diferentes topologías, una variedad de componentes para interconectar los varios segmentos, el uso de redes de área amplia para comunicarse con sitios remotos, y muchos más elementos. La gran pregunta es: ¿Cómo se le va hacer para administrar todos estos elementos de red tan variados? La respuesta a esto es el uso de analizadores y/o monitores.

Los monitores permiten obtener información estadística tales como: tasas de error, tasas de *broadcasts*, utilización de la red, tiempos de respuesta, tráfico global, historiales, etc. Por otro lado, los beneficios que arroja el uso de analizadores son: detección, resolución y prevención de problemas más rápido, reducción de servicios costosos, incremento en la productividad por medio de la administración de red, incremento en el conocimiento de la administración de red, definición de la utilización global y por estación, etc.

6.1.2 Arquitectura de la British Telecom´s Open Network

La British Telecom (BT) es una de las empresas que han incursionado en la administración de redes según el modelo OSI y ha intentado poner en práctica este modelo sobre sus productos, ya que es uno de los miembros fundadores del OSI/NMF. La BT como es de esperarse se mantiene bajo la filosofía de Arquitectura de Sistemas Abiertos (*Open Network Architecture*, ONA), siendo a la vez proveedor y consumidor de redes, lo cual la obliga a poner en práctica los estándares de administración de redes de OSI.

La arquitectura de la BT para la administración de redes, está basada en cuatro sub-arquitecturas como se muestra en la siguiente figura y que se explican a continuación.

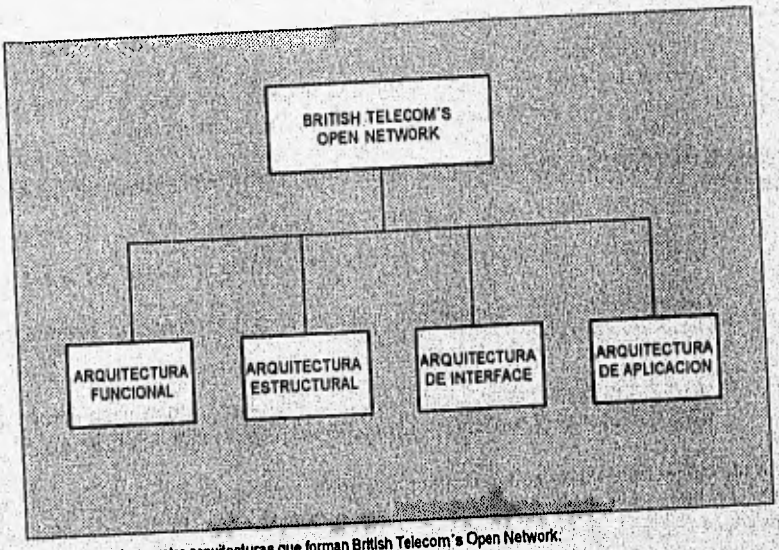


Figura 6.1.a Las cuatro arquitecturas que forman British Telecom's Open Network.

- La Arquitectura Funcional: Esta arquitectura describe las funciones que son ejecutadas por el administrador de red y que están descritas como el sistema administrador dentro de un convenio interactivo mutuo. Esta arquitectura se divide en siete áreas funcionales de administración, dentro de las que estan contenidas las cinco que define OSI, pues como ya se ha mencionado el modelo de BT se basa en este, obteniéndose de esta manera un modelo extendido de siete funciones.

  En la siguiente figura se puede apreciar como algunos de los nombres coinciden con los del modelo OSI pero otros son diferentes.

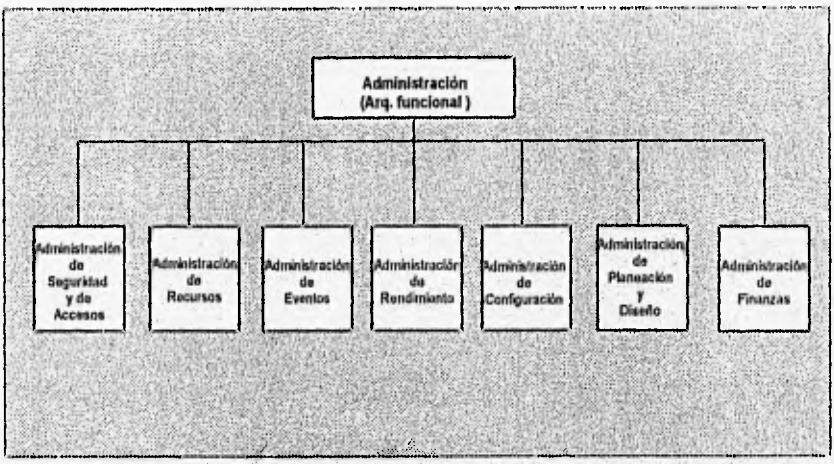


Figura 6.1.b Arquitectura funcional de la British Telecom's

La clasificación de las áreas funcionales es semejante al establecido por OSI, aunque cabe aclarar que existen algunas pequeñas diferencias o adecuaciones, como por ejemplo las áreas de Administración de Rendimiento y Administración de Configuración son similares a OSI . La Administración de Eventos tiene como finalidad controlar el evento sobre la red, como podrían ser pruebas, operaciones de diagnóstico y Administración de Alarmas que están relacionados con la función. La Administración de Recursos se emplea para el manejo de las entidades físicas y lógicas sobre la red. Esta función establece todos los recursos empleados, como son: *hardware, software*, circuitos, pruebas de equipo, etc. La Administración de Finanzas es similar al especificado en OSI (Administración de Contabilidad), pero además tiene las funciones de facturación, depreciación, amortización y mantenimiento de costos. La Administración de Seguridad y Accesos se encarga de validar los accesos, la seguridad, y la encriptación de las operaciones de acceso. Finalmente, la Administración de Planeación y Diseño es usada para determinar la topología de la red, la carga de ella, estrategias de ruteo, operaciones de *fall-back*, etc.

- La Arquitectura Estructural: La arquitectura estructural describe cómo están organizadas las funciones de esta con respecto a la misma arquitectura, cómo los niveles de la administración de la red interactúan con otros y cómo se pasan la información entre sí. La arquitectura estructural está organizada en cuatro niveles jerarquizados, los cuales se muestran en la figura siguiente.

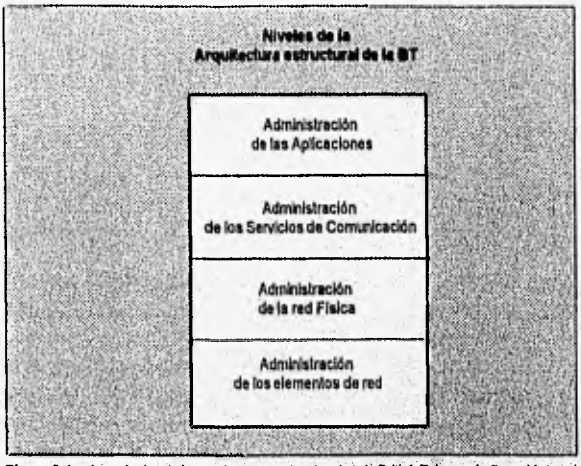


Figura 6.1.c Los niveles de la arquitectura estructural de la British Telecom's Open Network.

Los niveles superiores tienen como función administrar las aplicaciones (Nivel de Administración de Aplicaciones) y proveer los servicios necesarios para establecer las conexiones para lograr tal objetivo (Nivel de Administración de Servicios de Comunicacción), el siguiente nivel (Administración de la red Física) tiene la función de soportar la conexión para usuario remoto, además de administrar las tablas de ruteo y los mensajes de respuesta (*acknowledgment)* y finalmente, la administración de elementos de red, la cual se encarga de los paquetes de conmutación administrados.

- Arquitectura de Interface: La arquitectura de la interface de la BT está de acuerdo a las normas de OSI. los niveles de esta arquitectura se pueden apreciar en la figura de la página siguiente. Dicho esquema nos muestra como la arquitectura de interface está estratificada bajo el concepto de los siete niveles. Estos soportan los sistemas de administración de red, así como los diferentes paquetes de red y de redes LAN. Si analizamos el nivel siete de la arquitectura podremos reconocer los elementos de su estructura (FTAM, ACSE,CMISE, ROSE y ACSE), los cual ya fueron tratadados oportunamente en el capitulo tres.

- Arquitectura de Aplicación: Las aplicaciones que esta arquitectura soporta son muy variadas, sin embargo, las principales son las siguientes :

  - X.11 *Windows*
  - SQL
  - Bases de datos (SQL)
  - Interfaces de Comunicación OSI/NM
  - UNIX V5

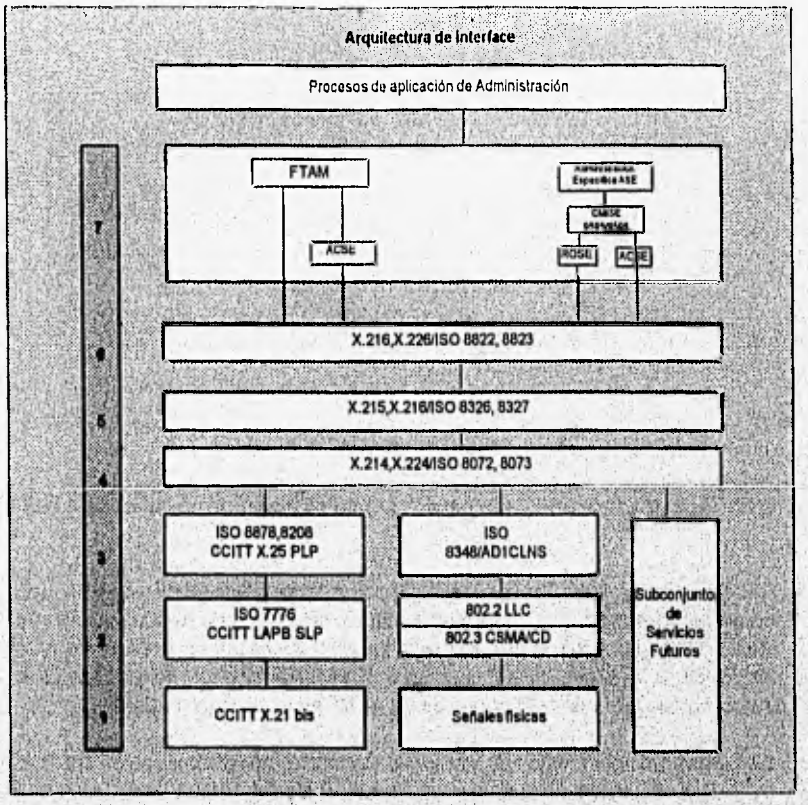


Figura 6.1.d Arquitectura de interface de BT con respecto al modelo OSI.

Una vez explicadas las cuatro arquitecturas de la BT hablaremos de la la arquitectura de sistemas abiertos (*Open Network Architecture,* ONA) que está formada por tres elementos, los que procederemos a explicar en las siguientes líneas y se ilustra en la siguiente figura.

- **Protocolo de Administración de Sistemas**: En este elemento se establecen los *stacks* de protocolos permisibles para soportar la administración de red OSI.

- **El conjunto de mensajes de Administración**: Este componente definen la interface de aplicación para las alarmas de vigilancia, pruebas y funciones de diagnóstico.

- Objeto y Librería de Atributo: Aquí se definen las reglas sobre el uso de patrones y macros sobre las operaciones de administración de redes.

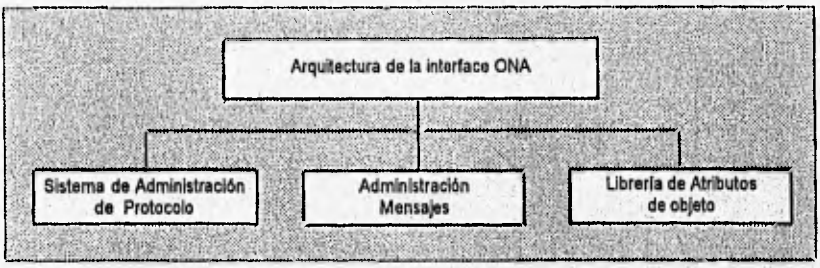


Figura 6.1.e Arquitectura de Sistema Abierto de la BT.

6.1.3 Arquitectura unificada para la administración de redes de la AT&T

AT&T ha desarrollado una arquitectura para la administración de redes basada en en el modelo OSI, la cual se denomina Arquitectura de Administración de Redes Unificada (*Unified Network Management Architecture*, UNMA). En esta arquitectura se fucionan los conceptos y especificaciones de protocolos e interfaces de OSI con la experiencia de AT&T en el manejo de UNIX.

La UNMA de AT&T está basada en el Protocolo de Administración de Red (*Network Management Protocol*, NMP). La función del NMP es proveer una interface para el uso de componentes tales como teléfonos, *modems* y estaciones de trabajo. A estos elementos se les conoce como Sistemas de Administración de Elementos (*Elements Management Systems*, EMSs). El concepto de EMS es muy empleado en el ambiente de telecomunicaciones en Estados Unidos. Hoy en día las telecomunicaciones se pueden dividir en tres áreas según AT&T: cliente local, cambio de red local y acarreos de intercambio de red.

AT&T ha implementado su UNMA alrededor de *ACCUMASTER*, donde este último usa NMP para la administración de la red, por lo que un usuario accesa al sistema de administración de red de AT&T por medio de *ACCUMASTER*.

AT&T ha trabajado conjuntamente con Cincom System Inc. para desarrollar un módulo que le permita al administrador de la red obtener información acerca de redes SNA, dando como resultado un sistema, el cual combina a NetView de IBM con los módulos de *ACUMASTER*. En la siguiente figura se muestra la arquitectura de UNMA, que como se podrá observar es congruente con el modelo OSI.

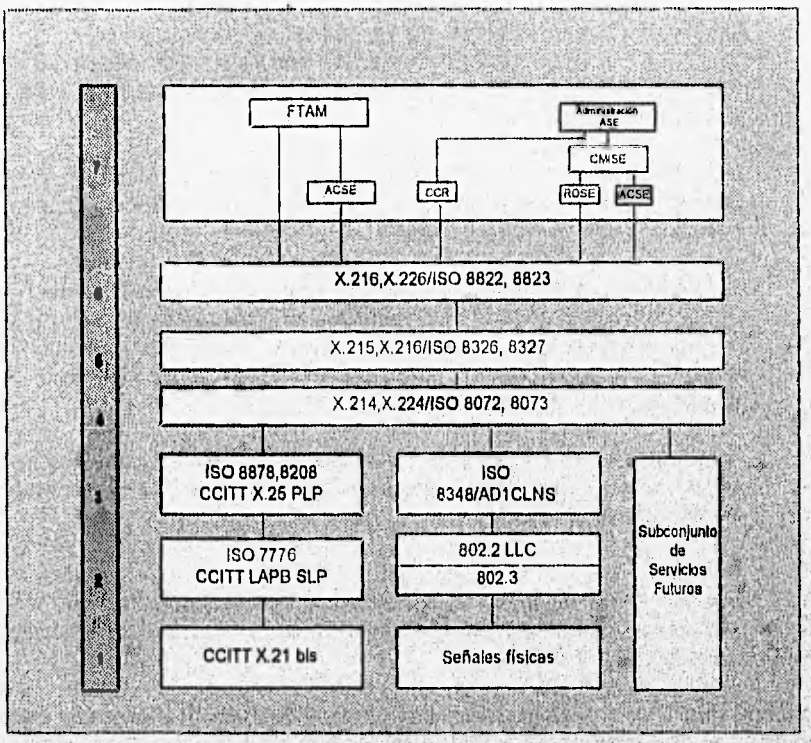


Figura 6.1.e Arquitectura de la AT&T con respecto al modelo OSI.

El UNMA está dividido en seis categorías , donde cinco de éstas se encuentran alineadas con el modelo propuesto por OSI, las cuales procederemos a explicar brevemente en las siguientes líneas.

- Administración de Configuración y Nombre: Proporciona información de los cambios, directorio e inventario administrativo. Este habilita al administrador de la red para añadir y mover objetos, así como obtener información de los directorios, tales como los servicios contratados con el vendedor.

- Administración de Fallas: Esta área engloba varias pruebas que son consideradas como básicas por la BT en la administración de redes, dichas evaluaciones son pruebas de conexión, diagnóstico de problemas, reconfiguraciones, problemas y servicios de reparación. El objetivo fundamental de esta área es la de auxiliar al administrador de la red en el diagnóstico y reparación de fallas, así como reconfiguraciones, además de que se encarga de mantener un registro de fallas.

- Administración de Rendimiento: En este punto se especifican los servicios de monitoreo tales como rastreos y límites de rendimiento. Las mediciones de rendimiento que se hacen son las siguientes: disponibilidad, utilización y actividad completa en la red.

- Administración de Contabilidad: Este rubro se refiere a la facturación de cuentas de clientes, presupuestos, así como la verificación de varias operaciones de facturación y redes.

- Administración de Seguridad: Esta sección se especializa en la seguridad de la red, por lo que se encarga de la seguridad en lo que a accesos a la red se refiere y autorizaciones de niveles de seguridad. Las actividades en materia de seguridad, se refieren a accesos de códigos de autorización, locación geográfica, hora y día de acceso, etc.

- Planeación de red: En esta área se maneja la planeación a futuro de la red o dicho de otra manera el crecimiento de ella, así como errores de resolución y factores de contingencia.

AT&T basa su NMP en los cuatro niveles superiores del modelo OSI siendo esto trasparente al usuario final. Los niveles inferiores del modelo están basados en normas del IEEE así como de la CCITT.

NMP presenta algunas diferencias con respecto al modelo de OSI. Algunas de éstas son las mejorías hechas al servicio *SET*, permitiendo cambiar el valor sobre un atributo multivaluado y estableciendo atributos por defecto para este servicio. Con respecto al servicio *M-GET*, AT&T da libertad de cancelarlo en cualquier momento.

6.1.4 Arquitectura para la administración de redes DEC

Digital Equipment Corporation (DEC) ha sido una firma que durante varios años ha soportado los estándares de OSI, ya que prácticamente todos sus productos se ajustan a este modelo. En el nivel corporativo, esta empresa ha hecho varios pronunciamientos para usar los estándares de OSI, de entre los cuales, su esquema de administración de red, conocida como Arquitectura de Administración de Actividad (*Enterprise Management Architecture*, EMA), contiene esta tendencia.

EMA se basa en los estándares OSI que fueron descritos en capítulos anteriores, como ejemplo de ello es que usa a CMIP y CMISE. Tal como sucede en la administración de red de OSI, EMA se organiza también alrededor de los conceptos de programación de objetos y de objetos administrados o manejados.

La estructura de EMA se divide en un modelo director y en un modelo de entidad, como se ilustra en la Figura 6.1.f. La combinación del director y la entidad define la estructura de las interfaces de administración de DEC y la interacción entre los sistemas de administración (directores) y los objetos manejados (entidades).

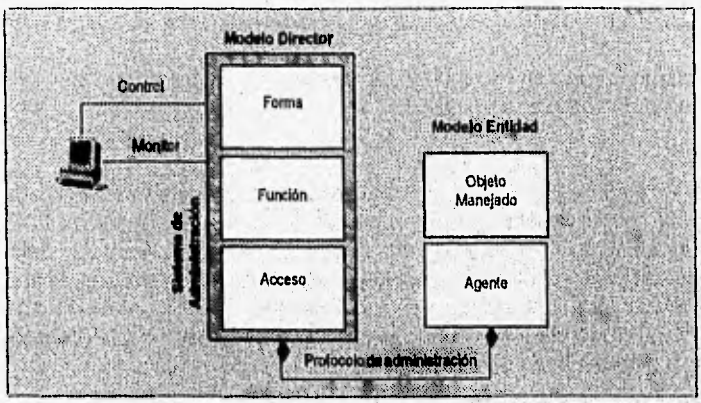


Figura 6.1.f Arquitectura de EMA

El modelo director de EMA provee una arquitectura modular para el diseño de un sistema de administración abierto. La administración que hace DEC para administrar su red es usar al modelo director para dar una interacción sencilla entre los módulos de administración. Como se verá posteriormente, estos módulos de administración se conectan como se necesite al interior de los componentes del director.

El modelo de entidad está diseñado para proveer información de la administración de red y servir como interface de administración sin tener que alterar los objetos manejados. Este es un modelo orientado a objetos, de esta manera encuentra un buen acomodo con la arquitectura de administración de OSI. En armonía con las ideas de OSI, el modelo de entidad provee el significado para definir objetos dentro de clases de objetos, la cual permite la compartición de diversas características dentro de dichas clases y da un conjunto de reglas para la administración de las clases.

El modelo director consiste de tres niveles: forma, función y acceso, y el modelo de entidad consiste de dos niveles (el cual provee operaciones similares al del nivel de acceso): el agente de administración y el objeto manejado.

- Forma: Este nivel soporta la convergencia de información de administración entre un director y sus elementos. Un ejemplo de los elementos son los usuarios.

- Función: En este nivel se describen los servicios ofrecidos por las aplicaciones residentes dentro del director EMA.

- Acceso: En él, se definen las operaciones para el control y monitoreo de las entidades administradas de EMA.

- Agente de administración: Este nivel proporciona una interface remota al objeto manejado y realiza procedimientos en respuesta a la entidad director.

- Objeto manejado: Es una entidad (*hardware* o *software*) que provee de un servicio específico a un cliente.

Los modelos entidad y director contienen varias interfaces externas e internas para soportar las comunicaciones entre los niveles del director y la entidad. DEC ha establecido tres tipos de interfaces para estas actividades, el primer tipo define las comunicaciones entre el objeto manejado y el agente de administración. El segundo tipo lo hace entre el modelo director y el modelo entidad. El tercer tipo es una interface interna que se utiliza entre los niveles dentro del modelo director y la aplicación del usuario o la estación de trabajo del mismo.

La ruta en la figura 6.1.f etiquetada como protocolo de administración, identifica al protocolo de comunicaciones que transporta información entre la entidad agente y el nivel de acceso del directorio. Se pueden usar varios protocolos para llevar esta información, el protocolo preferido es CMIP, sin embargo, DEC establece que se pueden emplear otros protocolos tales como el SNMP, TCP/IP (CMOT), SNA y productos DECnet.

EMA está diseñada por el director para manejar cualquier tipo de entidad. La aplicación más sencilla es de una organización para definir y desarrollar un objeto manejado de acuerdo con el modelo de entidad de DEC. Sin embargo, si las entidades no están diseñadas de acuerdo con el modelo de entidad, se deberán presentar al director por medio del nivel de acceso. En esta situación, el nivel de acceso actúa, como un protocolo de emergencia y deberá manejar la traslación desde el modelo entidad. Como en muchas aplicaciones actuales, esta práctica requiere de interfaces consistentes con el modelo director, no obstante que exista en los niveles de administración.

Por otra parte, el modelo entidad de DEC es un modelo paralelo al de la arquitectura de administración de OSI, ya que soporta consistentemente la Estructura de Información de Administración (Structure Management Information, SMI). El modelo entidad identifica a las clases basadas en atributos, grupos de atributos, directivas y eventos.

- Atributos: Como sucede en SMI, los atributos son piezas específicas de datos que proveen de información de administración proveniente de entidades administradas. Por ejemplo, un atributo puede ser el estado operacional en una tarjeta con valores tales como: deshabilitada, activa u ocupada.

- Grupos de atributos: Los grupos de atributos tienen que ver con el concepto de instancias de SMI. Una instancia es el objeto manejado de la misma clase, es decir, los grupos de atributos contienen atributos los cuales tienen algo en común como identificadores, nombres, etc.

- Directivas: Las directivas describen a las peticiones que realiza un director a una entidad. La entidad recibe una directiva y regresa una respuesta. Esto es similar a las notificaciones y operaciones en el modelo OSI.

- Eventos: Los eventos modelan las notificaciones de OSI, los cuales permiten que se mande información no solicitada a un director desde el agente.

El modelo entidad también permite subordinar entradas en donde una clase de objeto pueda ser un miembro de otra clase de objeto, así la implementación que hace DEC de la administración de redes permite la herencia y la compartición del comportamiento entre objetos y clases de objeto superiores.

El EMA se organiza en base a modelos orientados a objetos, de esta manera, las acciones presentadas por el agente al objeto manejado puede incluir definiciones de servicio de CMISE descritas en capítulos previos, así como las operaciones específicas definidas en otros servicios de la administración OSI.

Con respecto al modelo director, el *role* de éste es coordinar las actividades de administración de red e inicializar operaciones administrativas a petición de las aplicaciones de usuario. Además, también responde a los eventos mandados por el modelo entidad.

El director de EMA se basa en los tipos de módulos implementados en el *software*. Ellos consisten del Depósito de Información de Administración (Management Information Repository, MIR), módulos de administración (presentación, funcionalidad y módulos de acceso) y el ejecutor.

El ejecutor actúa como un coordinador de la ejecución del software de EMA. Proporciona una calendarización del trabajo, de alta las llamadas a procedimientos remotos (RPCs), coordina la atención a las llamadas remotas y locales, etc.

La MIR es un depósito que almacena la información administrativa y está diseñada para almacenar cuatro tipos de información relativa al ambiente de administración.

- Dato clase: Este tipo de dato contiene información de las entidades administradas que comparten las mismas propiedades. Así, se permite las definiciones de herencia en las cuales los subordinados se pueden usar para agrupar clases de entidad relacionadas a través de las aplicaciones de administración.

- Dato Instancia: Contiene las ocurrencias de las entidades que hay en una clase, como pueden ser los identificadores de red, nombres de paquetes y otros identificadores usados por la configuración del sistema. DEC permite que esta información sea usada por la administración de la red para determinar la representación del estado de la red.

- Dato atributo: Este tipo de dato contiene información específica de las instancias. El dato atributo se puede utilizar para dar un perfil de un objeto manejado, típicamente se le almacena a este dato en una conexión en donde se refleje el tiempo y se le pueda analizar.

- Dato Privado: Este tipo de dato es utilizado por el administrador de red, siendo de uso único del administrador y por lo tal privado.

Con respecto a los módulos de administración, éstos dividen los servicios de administración en piezas lógicas (niveles lógicos) por lo tanto se permite que las operaciones y los servicios se relicen separadamente. Por lo cual, existen los módulos de presentación, de funcionalidad y de acceso .

Los módulos de presentación dan soporte a la interface que haya entre los componentes del director y los usuarios finales, por ejemplo una computadora personal y una estación gráfica pueden estar comunicadas con un director por dos módulos de presentación distintos. Por otra parte, los módulos de funcionalidad se responsabilizan de algunos servicios de administración específicos, tales como la configuración, el rendimiento, la contabilidad, la seguridad y el manejo de fallas. Este módulo utiliza registros en la MIR para realizar muchas de sus actividades. Finalmente, los módulos de acceso le permiten al director tener una interface con cualquier modelo entidad (por ejemplo entre equipos de diferentes fabricantes), por medio de una conversión de formatos de los dos sistemas en comunicación y llevándolos a uno solo que típicamente es ASN.1.

6.1.5 La administración de sistemas abiertos de IBM, NetView

IBM ha hecho anuncios de chequeo de su arquitectura de Administración de Red Abierto (*Open Network Management*, ONM). ONM soporta los movimientos estratégicos de IBM hacia el Modelo de Referencia OSI, lo cual incluye el uso de CMISE/CMIP.

ONM no corresponde directamente a la administración de redes OSI. Sin embargo, se pueden encontrar muchas similitudes entre ambas administraciones, como se puede apreciar en la siguiente figura.

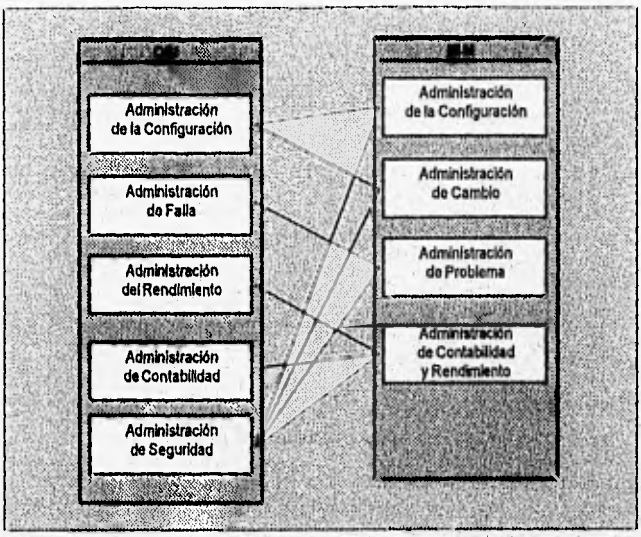


Figura 6.1.g Comparación de las funciones de administración de red de IBM contra las de la administración de redes OSI.

IBM se ha enfocado a NetView en cuanto a la tendencia a la Administración de redes OSI. NetView ha desarrollado algunas pantallas de gráficos muy poderosas y amigables. Más aún, NetView soporta interconecciones TCP/IP. IBM ha desarrollado su Subsistema de Comunicaciones OSI/CS, el cual permite a una máquina IBM comunicarse con una red basada en OSI.

La estructura de NetView está basada en:

- Puntos focales: Permiten al centro de control de la red obtener una vista global de la red. Como en muchos otros productos de administración, los puntos focales de NetView proveen una administración de red centralizada.

- Puntos de entrada: Estos se encargan de enviar información de administración a los puntos focales. Los puntos de entrada se enfocan en el soporte de dispositivos de red del tipo SNA. Por ejemplo, componentes de puntos de entrada podrían incluir los sistemas 36, 38 y 88; computadoras series 8100, etc.

- Puntos de servicio: Sus características permite a NetView "puentear" dentro de componentes no SNA y componentes no IBM. NetView también soporta servicios orientados a voz. Al igual que los puntos de entrade, los puntos de servicio también se encargan de enviar información de administración a los puntos focales, como se puede apreciar en la siguiente figura.

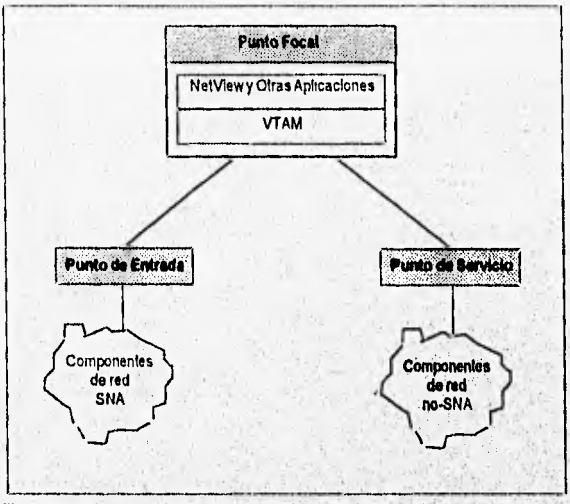


Figura 6.1.h Puntos foceles, puntos de entrada y puntos de servicio.

NetView se vale del Método de Acceso de Telecomunicaciones Virtual (VTAM) para poder interactuar con otros sistemas, tales como X.25, ISDN, otras LANS, SNA y OSI.

6.2 ALGUNOS PRODUCTOS EN EL MERCADO

6.2.1 Network General

Network General Corp. es conocido por su analizador *Sniffer* ("Husmeador") y por su *Watchdog Network Monitor* (Monitor de Red "Perro Guardián"), sin embargo, *Network General* lanzó en 1991 su Sistema "Husmeador" Distribuido (*Distributed Sniffer System*, DSS).

El DSS es un sistema des-centralizado que consiste de servidores remotos y consolas de administración que se aplican en un modelo cliente-servidor para monitorear y corregir fallas en la red. Los servidores *Sniffer*, conteniendo *software* de análisis y monitoreo, pueden colocarse en cualquier segmento de la red que requiera del monitoreo. Una o más consolas *SniffMaster*, localizadas en cualquier punto de la red para controlar a los servidores y visualizar la actividad de la red. Estas características hacen del DSS un sistema que puede ser utilizado,tanto en redes LAN como en redes WAN.

Las consolas y servidores se encuentra disponibles para *Ethernet* o para *Token Ring*. Existe también una versión WAN de servidores para ser conectado con consolas sobre *Ethernet* o *Token Ring*. Para unidades *Ethernet*, el control y comunicaciones entre consolas y servidores pueden estar basados, ya sea en el protocolo TCP/IP o Novell IPX, la elección se configura de fábrica. El soporte para protocolo NetBEUI de IBM o IPX de Novell para consolas o servidores *Token Ring*, también puede ordenarse. No

hay límite en el número de consolas y servidores que pueden usarse en una red. DSS utiliza estos protocolos como protocolos de transporte para comunicar a los servidores *Sniffer* y las consolas *SniffMaster*, ver siguiente figura.

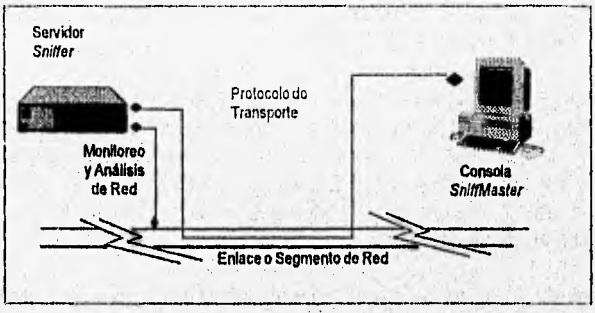


Figura 6.2.a Comunicaciones de Servidor *Sniffer*.

Los servidores *Sniffer* suministran resultados de análisis y monitoreo a las consolas *SniffMaster* y también pueden enviar alarmas directamente a administradores SNMP. Los servidores *Sniffer* se comunican con múltiples consolas *SniffMaster* para permitir a los múltiples administradores de red ver " la misma fotografía de la red" al mismo tiempo, a su vez, las consolas *SniffMaster* pueden "hablar" con múltiples servidores *Sniffer*, para proveer una vista multi-segmento del compartamiento de la red. Estadísticas, alarmas e información de protocolo se almacena en el servidor para minimizar el tráfico de la red y retener eficientemente información de la red muy valiosa. Los servidores se basan en el sistema personal 80386SX, disco duro de 40 *Mbytes*, 1 *Mbyte* de RAM para aplicaciones de monitoreo y cobertura local o 5*Mbytes* para aplicaciones de monitoreo y análisis y hasta coberturas de área amplia, y son lo suficientemente pequeños para poder colocarse en cualquier punto dentro de la red.

En suma, las aplicaciones de los servidores *Sniffer* son tanto de monitoreo como de análisis, como se muestra a continuación:

- Monitoreo de todo el tráfico de la red.
- Mantenimiento detallado de estadísticas de la red y estaciones (monitoreo).
- Comunicación de alarmas de falla a las consolas *SniffMaster* y a las estaciones de administración SNMP (monitoreo).
- Generación de reportes estándar (monitoreo)
- Captura y análisis de toda actividad de la red.

- Análisis comprensivo de los siete niveles.

- Inclusión de 12 módulos de análisis de protocolos, comprendiendo arriba de 120 protocolos diferentes (IBM-SNA, NetBIOS-, Novell NetWare, TCP/IP, Sun NFS, Banyan VINES, X Windows, OS/2 LAN Manager, XNS/MS-NET, ISO/OSI, DECnet, AppleTalk, X.25).

Una característica muy importante es que en un ambiente TCP/IP, los servidores *Sniffer* pueden configurarse para generar un mensaje de trampa de SNMP, en cualquier punto donde alguna alarma se haya generado. Este mensaje puede ser enviado a las múltiples estaciones de administración de red SNMP independientes. El mensaje de trampa contiene toda la información y descripciones encontradas en una alarma de Aplicación de Monitor.

Los beneficios de un sistema distribuido de análisis y monitoreo, se pueden resumir en los siguientes puntos:

- Distribución de las responsabilidades de los problemas encontrados en la red.

- Reducción de la necesidad de viajar para diagnosticar y localizar problemas.

- Observación del tráfico de red a través de puentes y ruteadores.

- Localización de problemas rápidamente y con poca gente.

- Análisis y monitoreo distribuido las 24 horas.

- Minimización del tráfico de administración de red.

Estas características del Sistema *Sniffer* Distribuido se consiguen a partir de la utilización de servidores dedicados a la administración (servidores *Sniffer*), lo cual permite al administrador:

- Colectar y analizar en un segmento remoto tantos datos como sea posible.

- Traer información (no sólo datos) a una consola.

- Correlacionar información en una consola desde múltiples servidores.

- Recibir alarmas generadas por servidores, automáticamente.

- Obtener información en tiempo real.

- Dejar los datos en el servidor de forma que otras consolas puedan accesarlos si fuese necesario.

Por otro lado, los requerimientos mínimos que se recomiendan para las consolas *SniffMaster* son: microprocesador 80386, *4Mbytes* de memoria, disco duro de 60*Mbytes*, monitor VGA color, DOS 4.01 y bus ISA.

Las consolas *SniffMaster*, permiten el monitoreo y control de múltiples servidores *Sniffer* simultáneamente, lo cual permite una vista amplia de la red. Además permiten mostrar alarmas de *log* de todos los segmentos y estaciones de la red.

6.2.2 SynOptics

SynOptics tiene una gran variedad de productos de administración que ofrecen al usuario una gran gama de posibilidades, sobre todo en el campo de la interconectividad a través de cableado telefónico. Todos ellos se basan en los estándares de la industria, concretamente en lo que se refiere a la administración de redes, en el protocolo SNMP.

Más aún, SynOptics es famoso por sus concentradores inteligentes que ofrecen plataformas de conectividad 10BaseT y otros que dan apoyo simultáneo a Ethernet, Token Ring y FDDI. Anexo a esto, esta compañía ofrece aplicaciones administrativas con base gráfica que le permiten monitorear y controlar la red desde una estación de trabajo.

Los agentes que maneja SynOptics tienen la facilidad de manejar una red muy compleja, multiprotocolo y geográficamente dispersa desde una sola estación de administración.

El agente básico proporciona funciones de administración elementales, tales como presentación en pantalla de la topología de la red, y la posibilidad de habilitar y deshabilitar puertos, diagnósticos y estadísticas.

Por otra parte, el agente avanzado provee todas las funciones del básico, pero además funciones tales como: autotopología, alarmas y umbrales, particiones de segmento automáticas, etc.

Con respecto a las facilidades que SynOptics le brinda al administrador de la red, provee software administrador instalable en computadoras personales para dar soporte a diferentes plataformas. Debido a lo extenso de la gama de sus productos, SynOptics permite adaptar el tipo de tecnología utilizada como plataforma de sistemas operativos a la administración de red.

De esta manera, se ofrece software administrador que guarda compatibilidad con las siguientes plataformas: DOS, OS/2, UNIX y Novell.

6.2.3. Novell

Debido a la creciente necesidad de controlar correctamente una red, Netware ha lanzado al mercado algunos productos cuya finalidad es la administración de una red, estos productos son compatibles con el protocolo SNMP. Básicamente son cuatro productos los enfocados a este fin, aunque podemos encontrar algunas herramientas adicionales que tienen el mismo objetivo. La idea de Novell es poner al alcance del administrador los elementos necesarios para la administración, desde una sola consola, la cual contenga toda la información necesaria para conocer el estado de la red. El sistema está diseñado bajo arquitectura distribuida diferente a la arquitectura tradicional de otras administraciones, pues éste aprovecha los mensajes de los agentes de cada dispositivo conectado a la red.

El sistema de Administración de Red de NetWare es capaz de detectar cambios en la red, sin que el administrador tenga que actualizar la consola de administración, ya que ella misma se actualiza, pues al entrar en función manda exploradores (agentes) a que le informen del estado de la red y de su configuración.

Los cuatro productos fundamentales de Novell para la administración de la red son los siguientes:

- Sistema de Administración de red de NetWare (*NetWare Management System*, NMS): La función de este sistema es permitir controlar y monitorear servidores que manejen protocolos IPX, nodos y ruteadores desde una PC, siendo requisito contar con plataforma Windows. El sistema soporta SNMP, lo cual le da el poder de soportar cualquier dispositivo que maneje agentes de este protocolo.

  El NMS está constituido por dos componentes: agentes y consola de administración centralizada. El primero de estos componentes manda información por la red acerca del estado de los dispositivos de la misma. Por otro lado la consola de administración centralizada colecta la información reportada por los agentes, para analizarla y dar un reporte que ayude al administrador de la red a mantener el rendimiento en ésta, además de brindarle información necesaria para determinar lo que pasa en la red. El NSM nos permite actualizar de manera fácil el inventario de la red, ya que detecta los IPX de los dispositivos, así como los IP de los ruteadores, y entrega dicha información en un mapa de localización de ruteadores y dispositivos. El sistema da la facilidad de seleccionar un icono del mapa de red y mostrarnos en otra ventana lo que está ocurriendo en esa sección específica de la red.

  El NMS diagnostica probables problemas en la red e indica los nodos críticos de la misma. Cuando se detecta un problema el sistema hace sonar una alarma que le indica al administrador que ha surgido una complicación. Por ejemplo, si una tarjeta llegara a fallar el administrador lo detecta y toma medidas para minimizar la caida de la red. Si hubiese IPXs o IPs duplicados, el sistema los

detecta e indica los nodos en conflicto y la causa de éste. Por lo anteriormente expuesto, este sistema es una combinación de un control centralizado en una consola y un proceso distribuido con el que recibe información.

Requerimientos del NMS para la consola son:

- Computadora 80386 ó 80486 IBM PC/ AT o compatible.
- Tarjeta VGA/SVGA y monitor.
- 12 MB RAM.
- De 40 a 80 MB libres en disco duro, dependiendo del tamaño de la red.
- MS Windows 3.1 y mouse compatible.
- MS-DOS versión 5 o posterior.
- Tarjeta de red compatible.

Requerimientos de NMS para el Servidor son:

- NetWare 3.11 o posterior.
- Procesador 80386.
- 2MB de RAM.
- 10 MB de espacio en disco duro.

• Agentes de Administración de NetWare (*NetWare Management Agents*, NMA): Este producto está intrínsecamente relacionado con el NMS, pues se encarga de enviar alarmas y estadísticas a la consola NMS. El NMA es un módulo cargable NLM el cual se carga en el servidor de archivos de un segmento de LAN. Este agente es compatible con SNMP, por lo que puede ser recibido por consolas SNMP sin problemas. Sus principales características son:

- Permite conocer las características de la configuración del servidor, incluyendo las actividades de archivos, volúmenes de datos, configuración de las unidades de disco, tarjeta adaptadora y servidor de impresión.

- Hace notificaciones de problemas en el servidor, acciona alarmas en tiempo real. Dichas alarmas incluyen problemas de unidades de disco, de volúmenes, de memoria, bitácora en uso y módulos NLMs.

- Permite el uso de alarmas sobre los parámetros del servidor, incluida la actividad de archivos, uso de memoria y de bitácora en uso.

- Permite configurar objetos designándoles alarmas para SNMP.

- Habilita el NMS y el *NetWare Service Management* para LAN NetView.

Los requerimientos de *hardware* y *software* son:

- Requiere ejecutarse sobre un servidor NetWare 3.x o 4.x siendo necesario para obtener la mayor ventaja del NMS.

- Computadoras 80386 ó 80486 IBM PC o compatibles

- 8 MB de RAM para el servidor NetWare 3.11.

- 10 MB de RAM sobre los servidores NetWare 4.0.

- 2 MB de espacio libre en disco para el volumen SYS.

• Administrador de Servicios de Concentrador (*Hub Services Manager*, HSM): Este sistema nos permite administrar concentradores basados en PCs. Para este efecto Novell ha desarrollado una especificación llamada HMI, la cual permite que los concentradores que cumplan esta especificación puedan ser administrados por NMS. Pero si el concentrador cumple con 10baseT, también será monitoreado. Este producto monitorea cada uno de los puertos del concentrador. El HSM está formado de módulos cargables LMSs que se encargan de administrar los puertos del concentrador en tiempo real. También cuenta con un módulo denominado HUBSNMP, el cual es compatible con consolas de administración SNMP y opera sobre los protocolos IPX e IP; pero solamente puede ser empleado bajo ambientes TCP/IP. El segundo módulo con el que HSM cuenta es para ambiante Netware y se denomina Consola de Administración de Concentrador (*Hub Management Console*, HUBCOM) y soporta SNMP. Sus características son:

  - La interface de usuario cuenta con ayuda en línea, bitácora de eventos seleccionados por usuario, estadísticas en tiempo real.

  - Control sobre puertos para habilitar y deshabilitarlos, pruebas sobre el concentrador y capacidad de inicializarlo, además de generar alarmas durante los eventos.

  - Es compatible con SNMP, hace uso de una MIB, es compatible con las normas del IEEE para el manejo de concentradores.

  - Su capacidad de monitoreo es grande pues monitorea todos los elementos de red conectados a los puertos del concentrador. Es capaz de monitorear múltiples concentradores.

  - Maneja los protocolos IPX, UDP/IP y soporta SNMP. Permite accesos remotos.

- Permite configurar los puertos del concentrador e identifica las direcciones de MAC de cada estación.

- Provee de estadísticas sobre niveles de *throughput*, colisiones y grandes eventos, autoparticiones, enlaces, tramas leídas y *bytes* leídos y errores

Los requerimientos del sistema son los siguientes:

- Computadoras 80386 o 80486 PC IBM o compatibles, tarjetas de concentrador HMI instaladas en el servidor.

- Netware 3.11 o posterior, 200 KB de espacio como mínimo en el disco duro y 250 KB de memoria en RAM.

• LANalyzer Agent: LANalyzer está formado por módulos cargables NLMs y trabaja en conjunto con NMS. Recoge y almacena estadísticas acerca de la red, soporta SNMP y el Monitoreo Remoto (*Remote Monitoring*, RMON). LANalyzer reside en el servidor y es capaz de detectar problemas en la red. Este sistema nos alerta cuando detecta una dirección de IP duplicada, cuando el nivel de utilización de la red está en su límite o cuado ocurren errores. Existen dos tipos de LANalyzer: uno que puede monitorear todos los segmentos de un servidor y el otro que sólo puede analizar un segmento. LANalyzer soporta todos los agentes de SNMP y los nueve grupos del estándar RMON, estableciendo así una interoperatibilidad con varias consolas de administración. Las características de LANalyzer son:

- El LANalyzer aporta las siguientes estadísticas: utilización de segmentos de red, tasa de *byte*, tasa de paquete, tasa de *Broadcast/multicast* y errores.

- Estadísticas por estación: de segmentos utilizados, errores, *brodcast/ multicast*, tiempo de primera y ultima transmisión, etc.

- Estadísticas por conversación.

- Alarmas.

Sus requerimientos son los siguientes:

- Servidor NetWare 3.11 o 4.0.

- Tarjeta de interface *Ethernet* o *Token Ring*.

- 1MB de espacio mínimo en disco duro.

- 2 MB de RAM.

6.2.4 OpenView de Hewlett Packard

Hewlett Packard es una de las empresas que han apoyado más a las normas establecidas por OSI, ya que desde el surgimiento de éste, HP ha acatado estas disposiciones en sus desarrollos. El sistema desarrollado por HP se denomina OpenView y está formado por cuatro áreas:

- Intérprete de comandos de red: En esta área se brinda soporte a los comandos, se da apoyo a la ejecución de comandos remotos, al administrador de recursos remotos y al análisis de rendimiento de los objetos administrados, teniendo un número de seguridad para soportar operaciones.

- *Status* y monitor de diagnóstico: Esta área es muy similar a la de OSI, pues mantiene una bitácora de control de actividad, rendimiento, información de *status*, haciendo uso de esta información para diagnósticos.

- Medición de la dificultad para transmitir: Esta función tiene como finalidad monitorear y realizar mediciones de problemas y dificultades a través de los módulos de *software* de HP.

- Administrador de Puentes: La función de este punto se refiere al análisis y diagnóstico a problemas que ocurren en puentes.

HP tiene una basta gama de productos, los cuales se venden como modulos independientes que se pueden conbinar.

6.2.5 AG Group (para Apple)

AG Group ofrece los analizadores de protocolos EtherPeek y LocalPeek basados en software, para redes LocalTalk y Ethernet. Corriendo sobre una computadora Macintosh proveen una vista completa de todos los datos transmitidos sobre una red.

Estas son herramientas valiosas para reducir los problemas que existan en una red. Como cualquier administrador de red conoce.

Los dos programas operan en medios físicos diferentes, sin embargo, son virtualmente idénticos, comparten la misma interface de usuario y capacidad de análisis de protocolo. Como comparten el mismo diseño de programa e interface con el usuario, las descripciones que se harán a continuación se aplican a los dos programas excepto donde se mencione lo contrario.

LocalPeek 1.0 corre sobre cualquier Macintosh plus o mayor (incluyendo la portable) corriendo el sistema 6.0 o mayor. Se puede utilizar para resolver los problemas de

cableado en una red LocalTalk de Apple, red telefónica de Farallon Computing Inc., y otras redes compatibles con LocalTalk. LocalPeek decodifica todos los protocolos de AppleTalk.

EtherPeek 1.4 necesita una Mac II o Mac SE/30 con sistema 6.0 o mayor y una tarjeta Ethernet. La lista de tarjetas de red soportadas es muy extensa, ellas pueden venir de Apple, 3Com, Farallon y Shiva Corp. El programa decodifica un amplio rango de protocolos para muchas redes Ethernet, haciéndolo muy útil sobre todo en ambientes multifabricante que incluyan a segmentos de LAN de TCP/IP, DECnet, NetWare y EtherTalk.

Para hacer un diagnóstico de ambos productos, se instalaron en una Mac IIci con un disco duro de 80 Mb, sistema 6.0.7. y 16 Mb de RAM. Para implementar las conexiones con Ethernet, se utilizó le tarjeta Etherport II de Excelan conectedo mediante cable par trenzado a un concentrador SynOptics serie 3000. Como el software analizador cabe en un diskette de 800 Kb, es fácil monitorear una red desde cualquier parte. Si se tiene una red grande o si es frecuente que se impriman archivos gráficos, se recomienda incrementar la memoria RAM a 2 MB, especialmente en el caso de EtherPeek. Si se elije el comando "*Get Info*" en el menú de archivos, aparecerá un cuadro de diálogo que muestra todas las particularidades del programa. En la parte inferior derecha se listan los tamaños de memoria sugeridos para le aplicación. Se puede ajustar el tamaño de memoria cambiándolo a un valor actual (en kilobytes). Para la evaluación de los productos AG, se destinaron 4 Mb para uso exclusivo de dichos productos, de los 16 Mb en RAM totales.

La instalación resultó sencilla, después de copiar todos los archivos a un directorio se inicio inmediatamente con la operación del programa. En el caso de EtherPeek, se permite seleccionar la tarjeta Ethernet en la cual se va a muestrear los paquetes, con un menú del tipo *pulling down*. Una vez que se elige la tarjete, EtherPeek está listo para trabajar.

Una característica especial de EtherPeek es que permite elegir una de las dos tarjetas de red que se pueden instalar en una Mac. Una tarjeta captura paquetes internemente mientras que la otra tiene las funciones normales de una tarjeta de red, sin embargo, no es recomendable ya que decae la velocidad de proceso considereblemente.

La primera tarea es obtener una lista de todos los números de nodo de la red y traducirlos a nombres entendibles que pueden ser de usuario o de localización. Esto es importante ya que los paquetes de datos capturados se referencian y se despliegan por medio de direcciones de red.

Una vez que se tengan todas estas listas de números y nombres, se pueden meter en una tabla de nombres, dicha tabla puede ser creada en un procesador de textos ó en el mismo programa. Simplemente se crea una lista que asigna un nombre simbólico a la dirección de nodo de red. Entonces, seleccionando un nombre en el menú *"Node Display Format"*, todos los paquetes capturados aparecerán con sus direcciones simbólicas, por ejemplo la dirección 08:00:75:A6:22:13 puede ser referenciado como Isabel IIci.

El AG Group incluye un programa llamado "*GetTheirAddress*", el cual requiere un "*responder*" de Apple (incluido en el *software* del sistema de la Mac) sobre cada estación de trabajo. *GetTheirAddress* regresa la dirección de la tarjeta de red y su correspondiente nombre. Se puede salvar esta información en un archivo de texto para así, facilitar la creación de la tabla de nombres.

Se puede comenzar a capturar datos casi inmediatamente aunque no se esté muy familiarizado con el filtrado de paquetes, selección de protocolo, tamaño de *buffer* o criterio de disparo. Ambos productos hacen un buen uso de la interface con el usuario que da la Macintosh, por lo cual se pueden considerar como analizadores de protocolos de fácil utilización.

En el menú *"Capture"*, el primer parámetro que necesita ser checado es el filtrado del protocolo. La selección lista inicialmente los cinco protocolos de Ethernet más comunes. La lista puede incrementarse o decrementarse dependiendo de los protocolos que se estén buscando mediante la selección de todos los elementos de la lista, solamente los elegidos o todos menos los elegidos.

El filtrado de direcciones es el siguiente paso en la selección de lo que sólo se desea visualizar para el caso en que se necesite inspeccionar una estación en particular o algún otro dispositivo sobre la red. La dirección fuente y destino se encuentran en el *header* del paquete de 14 bytes, por lo tanto EtherPeek busca direcciones físicas Ethernet. Los criterios de dirección se presentan por pares independientes. Por ejemplo, supongamos que A y B forman un par, de esta manera se pueden ver todos los paquetes que van hacia A y salen de ella, los que van y salen de B y los que salen de A y van a B, o los que salen de B y van hacia A. Esto es importante en el caso de que se quiera reducir la búsqueda de 1 a 4 direcciones destino y fuente.

Este *software* también le permite al administrador buscar patrones de bits específicos en los paquetes, se permiten patrones de 4 bytes con la consideración de que del byte 20 al 22 de cualquier paquete contiene la dirección lógica de la estación de trabajo.

Recordemos que los recursos mínimos en una Macintosh para que puedan correr estos productos es de 2MB en RAM (1 MB en el caso de LocalPeek). Cuando se tenga este monto de memoria, se estará limitado en el número de paquetes que se puedan capturar; sin embargo, la opción de "*Triggering*" dentro del menú "*Capture*", provocará que el programa comience o detenga la captura de paquetes dependiendo de un criterio seleccionado. Las condiciones para que se active un disparo pueden ser: tiempo, dirección, protocolo, eventos, o errores en los paquetes. Todos estos parámetros son ajustables especialmente el método de notificación.

Los programas también tiene implementado un sistema de alermas para alertar al usuario sobre cualquier condición de disparo que se haya puesto. Se puede elegir un cuadro de notificación con un mensaje que el usuario determine o seleccionar alguno de los sonidos estándar que maneja la Mac, incluso dichos sonidos pueden formar una voz con un mensaje hablado. Por ejemplo se puede poner una condición de disparo para alertar de cualquier paquete que mande o reciba la computadora de nuestro jefe,

de esta manera cuando el jefe se conecte a la red una voz nos dirá "disparo" y así sabremos que el jefe ya llegó.

Las opciones de captura de *buffer* proveen alternativas para el almacenamiento de datos. Se puede adecuar al programa para que se detenga la captura cuando el *buffer* esté lleno. Otras opciones son: después de llenar el *buffer*, borrar todo lo capturado y comenzar de nuevo, salvar todos los paquetes a un archivo y comenzar de nuevo o detener segmentos de paquetes. Esta detención de segmentos de paquetes es un método para capturar el mínimo monto de datos de un paquete (en EtherPeek es de 25 bytes y en Localpeek es de 16). Esto permitirá que se capturen más paquetes cuando se esté interesado en la información de la fuente y el destino, en lugar de los datos contenidos en el paquete.

Teniendo en cuenta estos parámetros, se puede comenzar entonces con la operación de captura y empezar a revisar la ventana principal para visualizar los resultados. En la evaluación se pudo colocar el número de líneas de datos del paquete que se quería ver obteniéndose una vista en tiempo real de los datos que se comenzaron a recibir y filtrar.

Hay una buena flexibilidad en la presentación de los datos de la ventana, se pueden desplegar direcciones Ethernet lógicas, direcciones físicas o el alias que se haya asignado antes. Es muy recomendable que se tenga un monitor a color ya que permite la mejor diferenciación de los datos y eventos.

La ventana principal tiene una lista desplazable de la actividad de los paquetes. Se puede ver al flujo de los datos en líneas que se van recorriendo hacia arriba y en las cuales se listan las direcciones fuente y destino, el tipo de paquete, su tamaño y el tiempo del sistema. Una barra indica el espacio que le queda al *buffer* y se vuelve roja cuando hay poca memoria. Un doble *click* sobre cualquier paquete que esté visible en ese momento da una vista de pantalla completa, así mismo, una combinación de *shif-click* sobre paquetes alternados nos dará la vista de pantalla completa de los mismos. Por otra parte se pueden asignar colores a muchas de las condiciones y direcciones.

El LocalPeek puede decodificar completamente el *stack* de protocolos de AppleTalk incluyendo LLAP, DDP, NBP, ATP, AEP, PAP, ADSP y ZIP los cuales fueron descritos en el capítulo dos. EtherPeek además puede decodificar TCP/IP, IPX de NetWare, DECnet fase IV y XNS.

Otra ventaja del *software* de AG es de componer su propio paquete y mandarlo a la red tantas veces como se desee; esta característica se usa principalmente para probar un repetidor intermitente o un ruteador sobrecargado que olvide sus tablas de ruteo.

Por otra parte el menú de estadísticas muestra graficas de barra horizontales, estadística de transmisión o recepción por nodo, la actividad del nodo por paquetes mandados y recibidos y la actividad del protocolo por su tipo (ver Figura 6.2.c).

Por ejemplo, si se ve en la grafica de barra que existe mucha actividad en cierto nodo, se puede ir al menú "*Especial*" para poner alarmas. La opción de "*Search*" -en el menu de "*Edit*"- buscará texto en ASCII o bytes en hexadecimal en cualquier datagrama que se haya capturado.

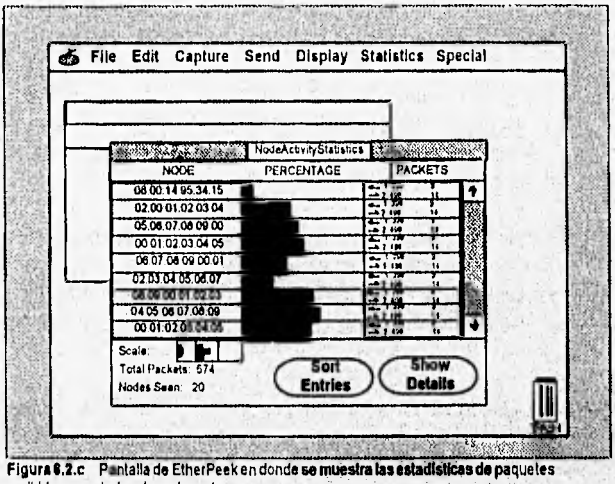


Figura 6.2.c Pantalla de EtherPeek en donde se muestra las estadísticas de paquetes recibidos y enviados de cada nodo

Finalmente y dada una evaluación global de estos dos productos, podría ser adecuada la administración de una red Ethernet o Appletalk con este *software*, ya que cubre algunas de las caracterísiticas de analizadores complejos como Sniffer o LANalyzer, sin cubrir los altos costos que demandan éstos.

6.3 LA RED DE LA INSTITUCION POLITICA

La Institución Política, para el cumplimiento cabal de sus funciones, ha incorporado dentro de sus procesos internos el uso de equipo de cómputo de diversos fabricantes, el cual, por consecuencia, cuenta con diferentes características. Dicho equipo se ha visto en la necesidad de conectarse entre sí, para que de esta manera sea factible aprovechar al máximo los recursos y permitir compartir la información bajo un marco de estricta seguridad, agilizando de esta manera sus funciones.

Como se ha mencionado esta organización cuenta con diversos equipos interconectados, formando de esta manera tres redes locales independientes, que geograficamente se encuentran en tres edificios pertenecientes a un mismo campus, cabe recalcar que en este momento las tres redes se encuentran incomunicadas entre sí, y que una de estas redes mantiene contacto vía modem con otras redes locales en el interior de la república.

Las redes están configuradas de la siguiente manera:

• Red de Edificio 1

- Servidor : Un servidor de archivos Pentium de 60 MHz, 32 MB en RAM y un disco duro de 2 GB. Sistema operativo Novell Netware 3.12 (versión 50 usuarios).

- Estaciones de trabajo: Veinte estaciones de trabajo en su mayoría tienen microprocesador 386 de 33 MHZ, 4 MB en RAM y disco duro de 170 MB.

- Impresoras: LaserJet 4si y LaserJetIII.

- Concentrador: Se utilizan un concentradores (*Hubs*) de 24 puertos cada uno. De marca cabletron con soporte a SNMP.

- Modem : 3 modem tipo hayes a 14,400 bps.

- Cable: Cable Par Trenzado (*Unshield Twisted Pair*, UTP) de 8 hilos y conectores RJ45.

- Topología: Estrella (física) y *Bus* (lógica).

- Tarjetas de red: SMC de 16 bits con soporte a SNMP Ethernet 802.3 interface BNC y RJ45.

- Protocolo de comunicación: IPX (en el nivel de red).

• Red de Edificio 2.

- Servidor: Un servidor de archivos 486 de 50 MHz, 16 MB en RAM y un disco duro de 1 GB. Sistema operativo de red Novell Netware 3.11 (50 usuarios).

- Estaciones de trabajo: 30 estaciones de trabajo en su mayoría tienen microprocesador 386 de 33 MHZ, 4 MB en RAM y disco duro de 170 MB.

- Impresora: LaserJet 4.

- Tarjetas de red: SMC de 16 bits con soporte e SNMP Ethernet 802.3 interface BNC y RJ45.

- Cable: Coaxial RG-58 *ohms*.

- Topología: *Bus*.

- Protocolo de comunicación: IPX (en el nivel de red).

• Red de Edificio 3.

- Multiusuario: HP 9000 32 MB en RAM, disco duro de 6 GB y Sistema Operativo UNIX SVR4.

- 6Terminales: VT 320 no gráficas.

- 20 Microcomputadoras: En su mayoría son 386, 25 MHz, 4MB en RAM y 170 MB en disco duro.

- Impresora: Matriz de punto de alta velocidad (1,500 lpm).

- concentrador: concentrador HP 12 puertos con soporte a SNMP

Impresora: LaserJet III:

- Tarjetas de red: SMC de 16 bits con soporte a SNMP Ethernet 802.3 interface BNC y RJ45.

- Cable Coaxial RG-58 ohms.

Las tres redes realizan procesos diferentes pero que se relacionan e interactúan entre si, con lo cual surge la necesidad de compartir los recursos de cada una entre las otras.

En la figura 6.3a se muestra la distribución actual de las redes y se puede entender de mejor manera que por estar separadas se requiere una administración propia en cada red, lo cual resulta poco práctico y no recomendable para un sólo administrador o supervisor de red, ya que si se sucita un problema en alguna de estas, el supervisor tiene que trasladarse hasta el punto en conflicto para remediarlo. Además de esta manera no puede mantener la misma atención y control en las tres redes a la vez.

Por las razones anteriores se sugiere la unificación de las tres redes y el establecimiento de una administración global e integral que comprenda las necesidades de cada red, pero a la vez que mantenga el orden y seguridad entre las mismas, permitiendo de esta manera obtener los siguientes beneficios.

- Agilizar la transferencia de información al eliminar el traslado de archivos por medio de *diskettes* y/o cintas.

- Eliminar la duplicidad de la información que existe actualmente.

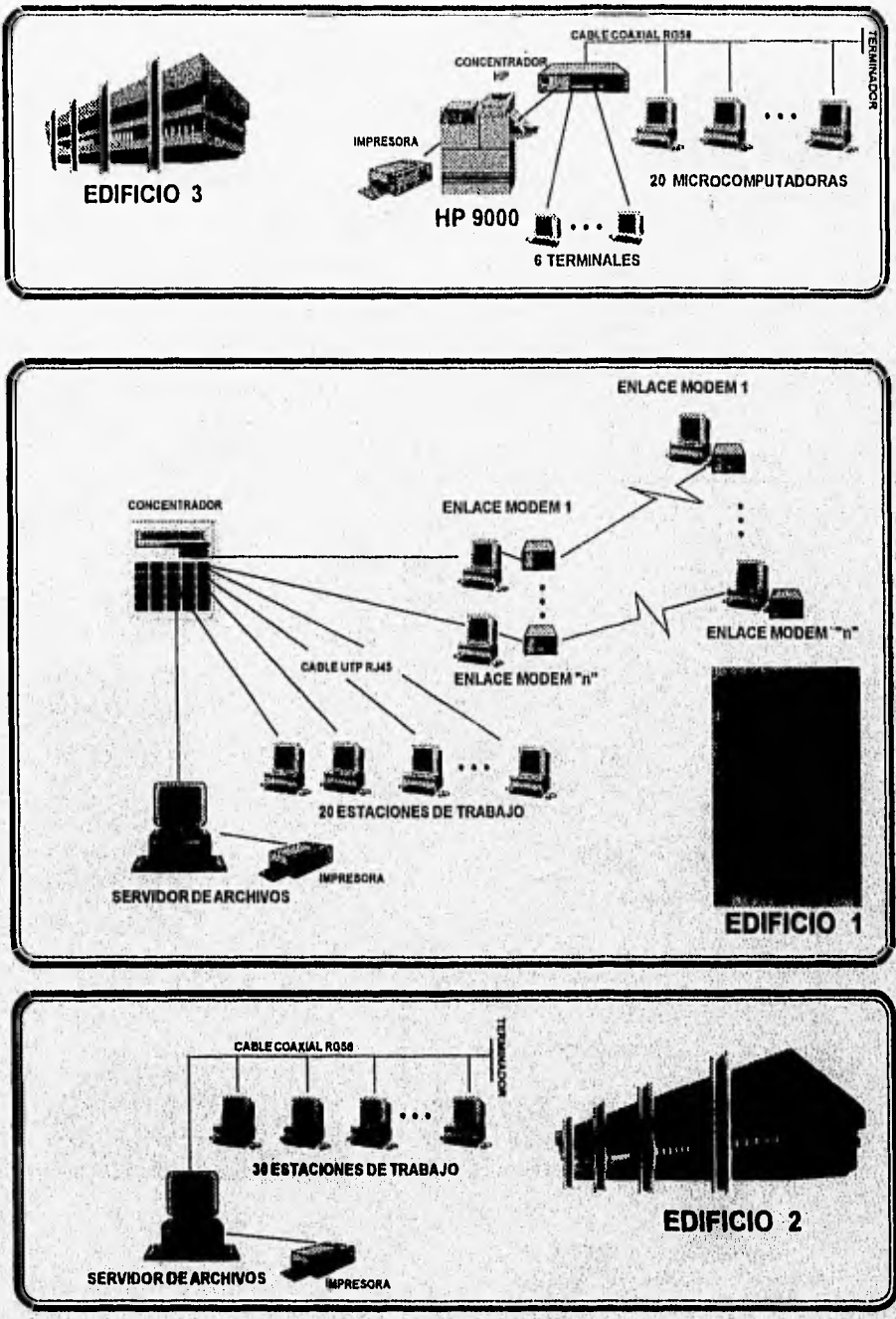


Figura 6.3 La red de la Institución Política actualmente.

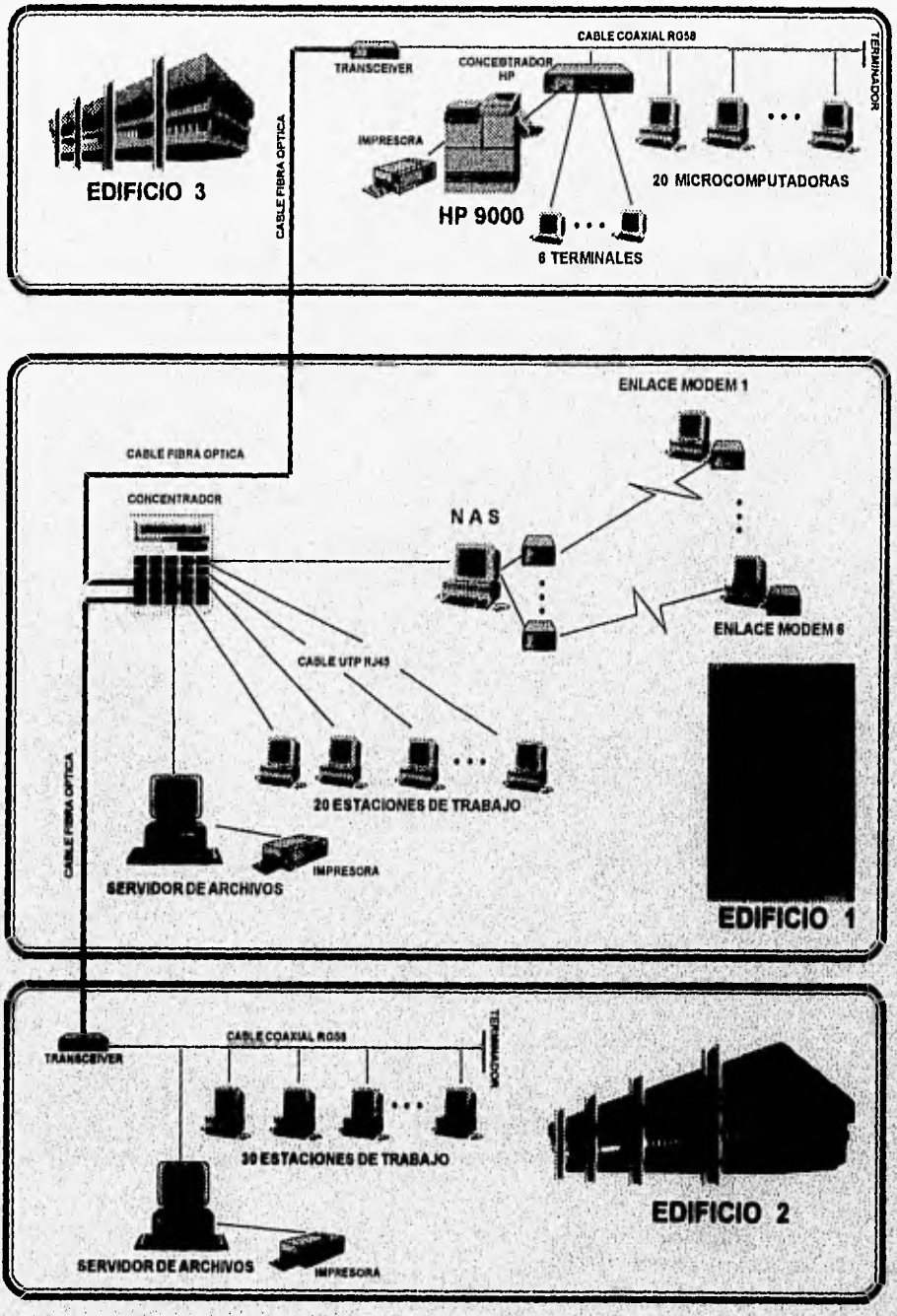


Figura 6.4 La red de la Institución Política Integrada.

- Accesar los servicios y recursos diversos desde cualquier punto de las dos redes: impresoras, discos, etc.

Considerando las características de las tres redes, y las necesidades de la Institución, se ha proyectado la interconexión de los tres edificios mediante fibra óptica, dos *transeivers* de fibra óptica a cable coaxial, además de dos puertos de fibra óptica en el concentrador de la red del edificio 1. Esto permitira en conjunto hacer las tres redes de los edificios más eficientes y brindar la infraestructura suficiente para ejercer una administración integral y centralizada.

Los enlaces se efecturán de la siguiente manera, los troncales entre edificios serán de fibra óptica acompañados de *transeivers*, uno en el edificio 2 y otro en el tres, así como la instalación de dos puertos de fibra óptica en el concentrador del edificio 1. Además, en la red de este edificio se instalará un servidor de comunicaciones (*NetWare Service Acces*,NAS), el cual puede atender seis sesiones a la vez vía modem , permitiendo con esto la reducción de la cantidad de computadoras conectadas a una línea y a un modem para realizar la misma función , además de que con NAS el enlace es transparente para el usuario final, la red unificada se ilustra en la figura 6.3b.

6.4 ELECCION DEL ANALIZADOR Y MONITOR MAS FACTIBLE PARA LA RED DE LA INTITUCION POLITICA.

Como ya se ha tratado ampliamente hasta este momento, la administración de una red no es fácil y mucho menos si se trata de tres como es el caso, donde cada red tiene características específicas y problemáticas diferentes y/o parecidas . Es aquí donde el administrador de red o supervisor requiere de un apoyo sólido y eficiente, que le permita a su vez resolver de manera oportuna y acertada las dificultades que en su trabajo se presenten, todo esto obviamente con la finalidad de evitar la degradación de la red a su cargo y mantener así la confiabilidad de la misma.

En el caso de la red de la Institución Política, la carga de trabajo es constante y en ocasiones, bastante grande. El trabajo que se lleva a cabo en las redes de los edificios 1 y 2 son muy similares. El control de la información se encuentra distribuida en la siguiente estructura de subdirectorios, lo cual se aplica a los servidores novell y UNIX:

- Subdirectorio basura: En él se almacena los archivos resultantes de las depuraciones de los discos, para permitir su pronta recuperación en caso de que alguien lo demande.

- Subdirectorio *Software*: En este *drive* se almacena todo lel *Software* de uso público, el cual trabaja en su mayoria bajo el ambiente gráfico de Windows® , por lo tanto, normalmente se realizan bastantes accesos a los discos duros de la red y el tráfico de la misma se incrementa debido al gran tamaño de los archivos gráficos generados por paquetes como Page Maker™, Excel™ etc. Las estaciones de los tres edificios tendrían acceso a dichos paquetes.

- Subdirectorio consulta: En este subdirectorio se almacenan la información de consulta, común para todas las áreas de la institución. Los archivos que forman dicha información actualmente ocupan un volumen aproximado al 40% de la capacidad total, sin embargo, esta es una información que crece constantemente.

- Subdirectorio usuario\grupo: En este directorio la información que se almacena corresponde a la información confidencial de usuarios de varios departamentos.

Como se puede apreciar la información almacenada en las redes son de injerencia general o privada; debido a esta última, la seguridad en la red es punto de gran importancia.

Cabe aclarar que, el grueso del tráfico no se da sólo con consultas y el uso de paquetes gráficos, sino con la captura que en ciertas épocas del año llega a realizarse a lo largo de 24 horas del día, .

Debido a su capacidad de almacenamiento, la HP 9000 (bajo UNIX) se utiliza para procesar un gran volumen de información. Cuando exista la comunicación entre las redes, dicha información podrá combinarse con la existente en los edificios 1 y 2, evitando los tiempos perdidos de segmentación, compactación, respaldo a cintas o a *diskettes*, etc. Evidentemente, esto incrementará notablemente el tráfico en la red, ya que habrá transferencias de archivos de cientos de *megabytes*.

Además, no se deben descartar las fallas debidas a factores externos como: fallas de tarjetas, repetidores, en el suministro de energía, en servidores o concentradores, etc.

Por lo anteriormente expuesto, la red de la Institución Política antes descrita no es la excepción y requiere de una administración confiable, pues como se ha podido ver, los procesos que se llevan a cabo no toleran demoras ya que no existe la posibilidad de repetirlos, debido a que son muy largos y complejos, además de que el factor tiempo es importante. Con la inclusión de un sistema de administración de red se tendrán las siguientes ventajas:

- Permitirá al administrador tener un control confiable de la red.

- Le ayudará a detectar y corregir problemas en la red con mucho mayor facilidad, oportunidad y precisión.

- Le proporcionará la capacidad de conocer realmente a su red, mediante estadísticas de utilización y rendimiento, lo cual proveerá al administrador de la información necesaria para definir y/o adecuar las políticas de crecemiento.

- Le brindará herramientas para mantener la seguridad de la red, proporcionándole un monitoreo de los accesos a la misma.

La característica importante de los analizadores y monitores es la capacidad de proporcionar los apoyos anteriormente descritos desde un solo punto de la red, esto es, que el administrador de la red es capaz de analizar y monitorear el funcionamiento de todos y cada uno de los elementos de la red global, desde su propia consola. Sin embargo, es importante hacer notar que algunos analizadores permiten el análisis distribuido, lo cual significa que el monitoreo se puede realizar desde distintos puntos de la red (sólo aquellos nodos designados para tener esta función), compartiendo la información de administración entre ellos, lo cual sería útil para aquellas redes LAN o WAN .

Debido a los intereses inherentes de la organización, como en muchas otras empresas, tales como bancos, casas de bolsa, etc., la seguridad de su información es vital. Por ello es importante que se cuente con un sistema de seguridad consistente, con un control de accesos a la red, monitorear y/o evitar intrusos que puedan "colgarse" con una computadora ajena a la red.

Es pues, bastante clara la necesidad del apoyo de un sistema de administración de red. Antes de poder elegir algún producto que nos proporcione dicho apoyo, es importante seleccionar el modelo de administración de red a seguir.

Como se trató en el capítulo 3, el modelo de administración de redes OSI tiene como característica el ser muy completo, opera en los últimos tres niveles de OSI (aplicación, presentación y sesión), mientras que, los estándares de Internet e IEEE hacen uso de algunas de las convenciones del nivel de presentación de OSI. A pesar de que desde 1979, compañías como AT&T, Digital y British Telecom , han hecho grandes contribuciones al estándar de administración OSI, las funciones de administración de OSI están aún en inicios, pues el estándar aún no se estabiliza y consecuentemente existen pocos productos que soportan este estándar. Las grandes ventajas de la administración OSI se encuentran en la participación de organizaciones como las antes mencionadas, que poseen una amplia base de experiencia práctica en el área de administración de redes, además de que se han organizado alrededor de las técnicas orientadas a objetos. Sin embargo, tienen un fundamento muy teórico y conceptual, por lo que el modelo de administración OSI ha sido difícil de implementar y por lo tanto podemos considerarlo como un estándar que será más utilizado en el futuro cuando éste se haga más tangible.

Por otro lado, debemos considerar que la red de la Institución Política cuenta con una HP 9000, en la cual existe UNIX que a su vez podría mantener una administración de red basada en su modelo director , que se vale de CMISE para administrar los componentes propios de DEC, así como los puntos de enlace con otros ambientes. Por ser una HP 9000 de DEC podría pensarse en la opción de administración de OSI, pero como tenemos ambiente UNIX y la comunicación se realiza mediante TCP/IP, ambos cuentan con soporte a SNMP, por lo que resulta viable una admisistración SNMP.

Con respecto al modelo de administración de redes de la IEEE, éste se encuentra en vías de desarrollo, pues la mayoría de sus partes no se han definido completamente y algunas de ellas no se han estabilizado. Por todo ello, no existen productos en el mercado que lo soporten y no existirán mientras éste no se consolide. Por lo anterior, el modelo de la IEEE queda descartado de posible consideración para nuestra selección del modelo de administración de redes más adecuado.

Respecto al modelo de administración de Internet, con su protocolo SNMP, ha sido adoptado por la mayoría de compañías desarrolladoras de *software* de administración, debido a su simplicidad. SNMP cuenta con pocos comandos (*Get request*, *Get-Next request*, *Get response* , *Set request* y *Trap*) que cubren gran parte de las operaciones de administración. A pesar de que SNMP se deriva del ambiente TCP/IP, sus comandos requieren solamente de servicios de transporte básicos (UDP), lo que hace al protocolo independiente. Esto significa que la información en SNMP se puede intercambiar con casi cualquier protocolo de red local.

En función de las consideraciones anteriores podemos concluir que el modelo de administración de red más viable es el de la Internet. Con esto procederemos a realizar una breve revisión y selección de los productos que soporten SNMP y que satisfagan las necesidades y características de la red de la Institución.

Hasta el momento de la realización de este trabajo, los productos aquí tratados se consideraron como los más representativos. Sin embargo, podemos asegurar que en un futuro existirá una gama mayor de los mismos, tal vez con características superiores a las de los actuales.

A continuación se hará la selección del producto, en base a las consideraciones hechas anteriormente , a las características, necesidades y a los recursos con los que cuenta la Institución:

- *Sofware* de administración de redes propietario: Implementados por British Telecom, AT&T, DEC e IBM. Los sistemas propietarios son excelentes productos que pueden llegar a ser estándares de facto, que servirán perfectamente a los usuarios de sus equipos. Sin embargo, estos sistemas tratan de ser implementaciónes del modelo de administración de redes OSI, pues no se apegan totalmante , ya que contienen variantes segun el punto de vista de cada una de astas empresas, lo cual lo hace un poco inestable en el sentido de total compatibilidad con otros productos que no sean propietarios, por estas razones y por las expuestas en párrafos anteriores no lo consideramos un modelo viable para éste caso.

- Sniffer de Network General: En primer lugar, Network General soporta estaciones SNMP.

Network General permite tener un control de varias redes interconectadas, ofrece el análisis en redes Ethernet, además brinda el análisis y/o monitoreo de redes de cualquier extensión geográfica. Los recursos que requiere son factibles dentro de la Institución, además de que permite tener la seguridad requerida por la misma. Realiza traducción de diversos protocolos, entre los que se cuentan TCP/IP e IPX. Con todo lo anterior podemos concluir que el Sistema *Sniffer* Distribuido de Network General es un producto viable.

- SynOptics: El sistema de administración de SynOptics soporta consola central y agentes SNMP en un ambiente multiredes. Sin embargo, requiere de concentradores de SynOptics que sirvan como agentes, lo que resulta ser un gran inconveniente y por lo mismo no recomendable.

- Novell: Como ya vimos en secciones anteriores, Novell cuenta con cuatro productos fundamentales para la administración de redes, ajustándose tres de ellos a las necesidades de la Institución, que son: el NMS, NMA y LANalyzer. Como se recordará estos productos soportan SNMP en su totalidad y además son congruentes con los recursos ya existentes dentro de la red, por lo que se les puede considerar como una buena opción. Cabe señalar que las perspectivas de crecimiento de la red de la Institución son tendientes siempre hacia Novell.

- OpenView de Hewlett-Packard: Aunque la filosofía de Hewlett-Packard es la de seguir los lineamientos propuestos por OSI, su sistema de administración OpenView soporta completamente SNMP, pero requiere para correr de equipos poderosos como son HP9000, Sun SPARC*station* o estación IBM RS/6000, lo que representa una fuerte limitante.

- AG Group: Como vimos en su correspondiente apartado, AG cuenta con dos productos, el *LocalPeek* y el *EtherPeek*, el primero de ellos opera exclusivamente sobre una red *LocalTalk* con MACs conectadas a ella, por lo cual de entrada queda descartada esta opción ya que no existe tal equipo en la Institución. Con respecto al segundo producto, aunque puede decodificar protocolos tales como IPX o DECnet, requiere de correr en una MAC la cual significaría un costo extra. Ambos productos no soportan SNMP por lo que quedan definitivamente descartados.

De las descripciones anteriores concluimos que tanto los productos de Novell como de Netware General son analizadores factibles. De ellos se mencionarán características importantes que nos permitirán finalmente elegir alguno de ellos. Los criterios a considerar serán:

- Elementos, facilidades y características importantes en la administración

- Disponibilidad, seguridad y mantenimiento.
- Costo.

En las siguientes tablas se resumen los elementos, características y facilidades importantes de cada uno de los productos.

ELEMENTOS	NOVELL	NETWORK GENERAL
ESTADISTICAS:		
• Global por red		
- Utilización de la red.	●	●
- No. de tramas o paquetes.	●	●
- No. de bytes	●	●
- Tamaño promedio de la trama o paquete	●	●
- No. de Errores.	●	●
- Utilización por protocolo.	●	●
- Utilización por tamaño de trama o paquete.	●	●
• Por estación		
- Utilización de la red.	●	●
- Transmisión y recepción a nivel paquete y a nivel byte.	●	●
- No. de errores.	●	●
- Utilización de protocolos.	●	
- *Broadcast / Multicast.*	●	
- Tamaño promedio de trama.	●	
ALARMAS:		
• Global por red		
- Paquetes.	●	
- Utilización (alta, baja).	●	●
- Tasa de *broadcast* y *multicast*.	●	●
- Duplicaciones de IP	●	
- Errores CRC.	●	●
- Paquetes fuera de tamaño.	●	●
- Estación ociosa.		●
- Violación de seguridad (intrusos o estaciones desconocidas).		
- Paquetes extra (sin motivo).	●	●
• Por estación		
- Errores.	●	●
- Tiempo de respuesta.	●	●
- Uso relativo.	●	●
- Configuración personalizada	●	●
REPORTES:		
- Numéricos.	●	●
- Gráficos.	●	●
- Configuración personalizada	●	●
- Generación automática a intervalos de tiempo.	●	●
- Envío de reportes a impresora, pantalla y/o disco.	●	●
OTRAS FACILIDADES:		
- Bitácoras.	●	●
- Solución a problemas rápidamente.	●	●
- Diagnóstico de fallas.		●
- Análisis de protocolos en los 7 niveles del OSI.	●	●
- Tiempo de respuesta de las aplicaciones.	●	●
- Utilización de *buffers*.	●	●
- Configuración de la estación de trabajo y del servidor.	●	
- Prueba de cableado y de estaciones de trabajo.		
- Análisis distribuido y centralizado.	●	●
- Soporte a administraciones de red NetView y EMA/DEC	●	●

Tabla 6.a Elementos y facilidades de los productos de Novell y Network General.

OTRAS CARACTERISTICAS	NOVELL	NETWORK GENERAL
• No. de Estaciones soportadas.	500 estaciones.	1024 estaciones.
• Decodificación de protocolos.	NetWare, AppleTalk, TCP/IP y NFS. DECnet, AppleTalk, X.25 e ISO.	SNA, NetWare, TCP/IP, Bynes, XNS,
• Soporte de SNMP.	Total.	Casi total.
• Ambiente.	Windows.	DOS.
• Representación Gráfica de la red.	SI.	No.
• Soporte de protocolos WAN.	TCP/IP	SDLC, X.25, X.3, X.28, X.29, SNA
• Soportes a futuro.	FDDI	FDDI, WANs T1 y T3, ISDN y análisis de protocolos de sistemas expertos.

Tabla 6.b Características de los productos de Novell y Network General.

REQUERIMIENTOS	NOVELL	NETWORK GENERAL
• En la consola central.	- Microprocesador 80386 o mayor. - Monitor VGA/SVGA - 12 MB en RAM. - 40-80 MB libres en DD, dependiendo el tamaño de la red. - MS-Windows 3.1 DOS 5.0 o mayor - Tarjeta de red (ODI-compatible) - Drive 31/2 pulg. HD.	- (*SniffMaster*) 80386 o mayor. - Monitor VGA. - 4 MB en RAM. - 60 MB libres en DD - DOS 4.01 o mayor - Tarjeta de red (ISA o EISA). - Drive 31/2 pulg. HD.
• En el agente.	- Servidor con NetWare 3.11 o mayor. y para versión ejecutable en una 80386. - 2 MB en RAM en la version NLM 8 MB en RAM en la versión EXE. - 1 MB libre en DD versión NLM 80 MB libres en DD.	- (ServidorSniffer) En 80386 o mayor - 5 MB en RAM - 40 MB en DD.

Tabla 6.c Requerimientos de los productos de Novell y Network General.

COSTOS	NOVELL	NETWORK GENERAL
• En la consola central. • En el agente. • En el LanAlizer multiple.	- US $ 2,005.07 (NMS) - US $ 1,448.00 (NMA) - US $ 2,048.00 (LANalyzer)	- US $ 3,997.50 (*SniffMaster*) - US $ 8,995.00 (Servidores *Sniff*)
TOTAL	- US $ 5,501.07	- US $ 12,992.50

Tabla 6.d Costos de los productos de Novell y Network General.

De las tablas anteriores podemos apreciar que Network General tiene gran ventaja sobre Novell en relación a la capacidad de manejo de múltiples protocolos y proporciona el soporte para un posible crecimiento de la red hacia otras plataformas, así como de

un crecimiento del número de nodos. Estas características se explican fácilmente en base a que el producto de Network General es un producto orientado netamente a redes de cobertura amplia (a nivel nacional e internacional) y convivencia de múltiples plataformas y protocolos, ya que su instalación no requiere de la presencia de algún sistema operativo de red en particular, siendo su única limitante la necesidad de DOS. Y por otro lado el producto de Novell es un producto, hasta cierto punto, propietario pues se instala única y exclusivamente en redes con sistema operativo NetWare. Sin embargo, ello lo convierte en un excelente producto para redes con dicho sistema operativo. Considerando que las perspectivas de crecimiento de la red de la Institución siempre son hacia Novell, los productos NMS, NMA y LANalyzer de Netware constituyen una excelente opción.

En cuanto a los recursos requeridos, podemos considerar que Network General consume menos recursos, al ejecutarse bajo un ambiente DOS, pero requiere de una estación dedicada como agente, lo cual eliminaría una posible estación de usuario. Por otro lado Novell, como se mencionó anteriormente, nos proporciona una compatibilidad total con el tipo de red que se maneja en la Institución, al trabajar bajo Netware y con ello proporcionar un análisis más específico de la red , además de que ya se cuenta con el *hardware* necesario, debido a que las instalaciones de los NMA se realizan sobre los servidores existentes.

Como se pudo apreciar en la Tabla 6.a., tanto el producto de Network General como los de Novell posee facilidades importantes y necesarias, sin embargo, Network General aventaja a Novell en aquella que se refiere al analisis de protocolos.

Finalmente, haciendo un comparativo de costos, podemos apreciar que los productos de Novell son los más accesibles, pues proporcionan un ahorro monetario de 57.65%.

CONCLUSIONES

Haciendo un análisis final de costo/beneficio, podemos concluir que los productos NMS, NMA y LANalyzer *Agent* Múltiple de Novell son los productos que nos proporcionarán la mejor relación de costo/beneficio, pues su costo es sensiblemente bajo y el beneficio que ofrece es equiparable al que proporcionaría el Sistema *Sniffer* Distribuido de Network General. Sin embargo cabe señalar que el producto de Network General es un producto completo y mejor que los de Novell (en el analisis de protocolos), debido a los elementos considerados en los párrafos anteriores, y que su alto costo se justifica en función de su capacidad de analizar y monitorear redes de tamaño, complejidad y heterogeneidad considerables. Sin embargo, en función de las políticas de crecimiento, necesidades y recursos existentes de la Institución Política, podemos afirmar que el Sistema *Sniffer* Distribuido de Network General estaría "sobrado" para la red de la Institución, además de que no soporta en su totalidad el estándar SNMP, mientras que el NMS, NMA y el LANalyzer, son una opción razonable y adecuada.
Finalmente, consideremos que la red de la Institución Política quedaría configurada de la siguiente manera:

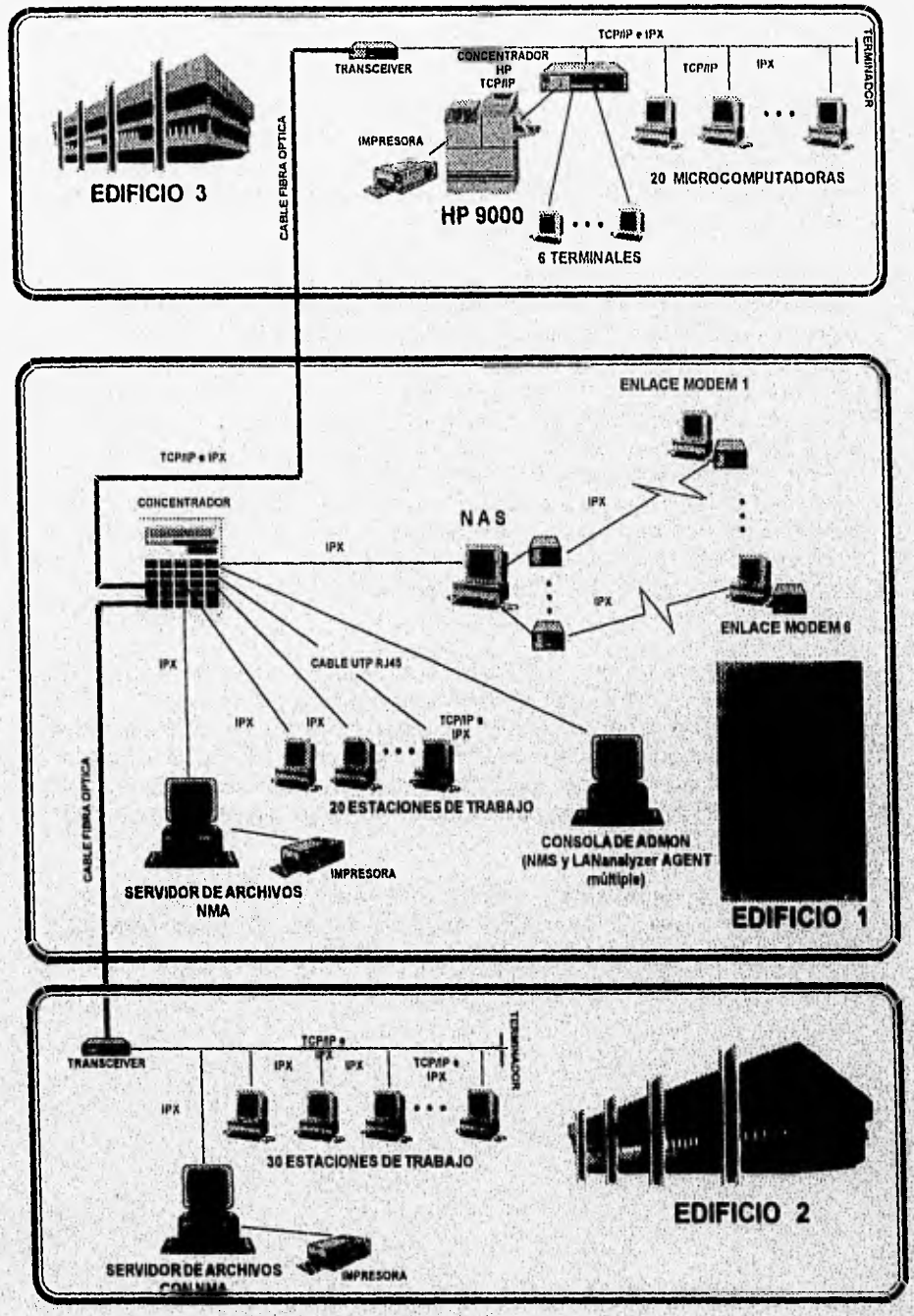


Figura 6.6 Propuesta de configuración y administración de la red de la Institución Política.

De tal forma que la propuesta de configuración y administración de la red de la Institución Política es instalar el *NetWare Management System*™ (NMS) y el LANalyzer *Multiple*™ en una de las microcomputadoras ubicadas en el Edificio 1. Además, en cada uno de los servidores Novell se pondrán Agentes de Administración de NetWare (NMA), tal como se muestra en la figura anterior.

Es importante tener en cuenta que en el medio de comunicación (cable coaxial RG-58, fibra óptica y par trenzado) se encuentran circulando TCP/IP e IPX generados por la comunicación con la HP 9000 y los servidores Novell respectivamente, dicha convivencia de protocolos no afecta la comunicación entre los distintos ambientes. Por ejemplo, en el caso de que una estación de trabajo de la red del Edificio 2 desee accesar un recurso en la HP 9000, deberá conectarse mediante TCP/IP y quedará bajo el ambiente UNIX. De la misma manera, una de las microcomputadoras del Edificio 3 puede llegar a ser una estación de trabajo de alguno de los servidores de los otros dos edificios, con tan solo cargar el IPX. El caso más general, es aquel en el que una estación de trabajo de cualquiera de los tres edificios pueda accesar los recursos de los servidores de Novell y los de la HP 9000 al mismo tiempo; esto se lograría al cargar el IPX y un *packet driver*, el cual homologa al IPX y al IP permitiendo a la misma tarjeta Ethernet funcionar con ambos protocolos.

Por otra parte, como se apreció en la tabla 6.4.c, LANalyzer decodifica tanto IPX como TCP/IP, lo cual nos asegura que éste pueda llevar a cabo sus funciones de agente. Además, con la instalación del LANalyzer Múltiple en la Consola de Administración del Edificio 1, reportará continuamente al NMS información recolectada en los tres segmentos, ya que la característica distintiva de un agente mútiple, es que puede monitorear todos los segmentos de red que se encuentren conectados al concentrador y que soporten SNMP.

De tal suerte, se propone instalar la consola NMS en alguna microcomputadora de la red del edificio 1, con el fin de proveer al administrador de la información generada a lo largo de los tres segmentos que componen a la red.

CONCLUSIONES

FINALES

CONCLUSIONES

Como se sabe, en todo proceso productivo es importante mantener un control de los procesos involucrados en éste, además de poder contar con la capacidad de solucionar cualquier tipo de contingencia que se presente, para con esto evitar que dicho proceso se detenga. De esto podemos concluir que mantener un nivel de productividad adecuado es un fin deseable y que por lo tal, para conseguirlo, es importante desarrollar procedimientos y sistemas con este fin.

Todo lo anterior es aplicable al mundo de las redes de computadoras, pues en este caso lo que se maneja es información (la mayor de las veces crítica para una empresa), la cual deberá llegar al destino correcto y sin corrupción alguna, de tal manera que el cuidar de la administración de una red se convierte en una actividad de suma importancia y de extrema dificultad, sobre todo cuando la red está formada por diversos equipos dando como resultado una red heterogénea. Atendiendo a esto es evidente la necesidad de elegir un sistema de administración de red que facilite esta tarea.

La red de la Institución Política no es la excepción, pues su administrador requiere del apoyo de un sistema de administración de red debido a las características de la misma.

Desde finales de los ochentas, varias empresas han trabajado en la creación de un estándar de administración de redes que permita proporcinar herramientas de administración de red, tales como estadísticas, seguridad, utilización, prevención y detección de fallas y errores.

Empresas como DEC, AT&T y British Telecom, apoyados en el modelo de referencia OSI, han trabajado en el desarrollo del estándar de OSI, el cual se basa en el uso de agentes distribuidos en la red, los cuales almacenan información importante para la administración en una Base de Información de Administración, la cual será accesada por el administrador a través de una consola conectada a la red. Este estándar de administración de red es muy completo, pues considera el uso de hasta 20 servicios repartidos en CMISE, CMIP, ACSE y ROSE para el manejo de la información, además aprovecha los beneficios del diseño orientado a objetos. Sin embargo, ha sido muy teórico y difícil de implementar, por lo que hasta la fecha no existen suficientes productos que den soporte a este estándar.

En cambio, el grupo de Internet retomó las bases del estándar de OSI, y creó su propio estándar más simple y mucho menos teórico. El estándar toma la idea básica de tener agentes que generan información y la almacenan en una Base de Información de Administración que a diferencia de la de OSI, sólo permite 5 servicios y no es dinámica; además de la consola que proporciona la visualización de dicha información al administrador de la red. El estándar de Internet descansa en su protocolo SNMP, que como sus siglas lo indican, es un protocio simple.

Debido a la caracterítica de simplicidad, el estándar de Internet ha tenido gran aceptación entre los fabricantes de elementos de comunicación y sistemas de administración, pues actualmente existe una gran variedad de productos que soportan SNMP.

Por otro lado la IEEE ha tratado de establecer su estándar con una arquitectura muy similar a la de Internet, sin embargo aún se encuentra inconcluso en muchas de sus partes, y algunas de ellas aún no se estabilizan. Consecuentemente, no existen implementaciones del estándar de IEEE en el mercado.

La tendencia en el mercado es pues, fabricar productos que soporten SNMP, aunque cabe señalar que algunos ya empiezan a ofrecer soporte a CMIP, sin contar las implementaciones hechas por DECnet, AT&T e IBM del estándar OSI de una manera propietaria.

Sin lugar a dudas SNMP tendrá mercado por varios años, y mientras CMIP no sea un estándar sencillo y costeable de implementar para los fabricantes de elementos de comunicación y *software* de administración, los productos de administración de red que soporten SNMP serán una excelente opción.

De aquí que, finalmente, en este trabajo se haya llevado a cabo una selección entre los productos de Novell y los de Network General, como sistemas que son factibles a proporcionar un apoyo al administrador de la red de la Institución Política.

Es importante señalar que a lo largo de esta tésis se fue logrando un conocimiento de las estructuras y bases utilizadas por los distintos organismos involucrados en la creación de estándares de administración de red, para finalmente tener la visión suficiente que nos permita llevar a cabo la selección del *software* de administración que nos proporcionará las herramientas indispensables en una red específica.

APENDICE

A

NORMAS DEL ESTANDAR DE ADMINISTRACION DE REDES DE OSI

APENDICE A

NORMAS DEL ESTANDAR DE ADMINISTRACION DE REDES DE OSI

Administrador de Objetos 10164-1
En esta norma se definen las reglas para la creación, borrado, renombramiento y listado de objetos manejados, así como el borrado y modificaciones en los atributos de los objetos.

Administrador de Estado 10164-2
En este estándar se pueden distinguir dos modelos de estado para los objetos manejados: administrativo y operacional.El estado operacional puede quedar definido como ocupado, activo, habilitado o deshabilitado con respecto al estado administrativo puede presentar tres subestados: *shutting down*, bloqueado o desbloqueado.

Administrador de Conexión 10164-3
Aquí se define la conexión del administrador de objetos y se definen los siguientes tipos:

-Directa: Una porción de información asociada con un objeto administrado indicando identificadores de otros objetos administrados.

-Indirectos:Una conexión se deduce entre dos objetos manejados cuando existe una concatenación de dos o más objetos conectados directamente.

-Simetría: Las reglas que interaccionan entre dos objetos deben de ser las mismas.

-Asimetría: Las reglas que interaccionan entre dos objetos administrados son diferentes.

Reporteo de Alarmas 10164-4
Aquí se definen cinco categorías básicas de error y son las siguientes:

-Comunicaciones
-Calidad de Servicios
-Procesado

Reporteo de Eventos 10164-5
En esta norma se definen los componentes que soportan el reporteo de eventos remotos y el procesamiento de los eventos locales. El estándar se basa en el concepto de conjunto de discriminadores. Por ejemplo, el discriminador de envío de eventos es responsable del filtrado de eventos, el cual se basa en un número de criterios de selección, así como de decidir si el evento está siendo reportado. Para los eventos de filtrado, el discriminador establece el umbral y otros criterios que deben satisfacerse para que el evento sea enviado.

Función de Control de Bitácora 10164-6
En esta norma se define la operación de control de bitácora para un sistema administrador de red, en ella se plantea como se conserva la información alrededor eventos y de los objetos administrados. El mecanismo de bitácora especifica un control de tiempos para la ocurrencia, reanudamientos y suspensiones de accesos a la red (*loggings*). Se definen operaciones para la recuperación y borrado de los registros de *log*, así como la modificación de los criterios utilizados en la creación de los registros de *logging*.

Funciones de Reporteo de Alarmas de Seguridad 10164-7
En este rubro se definen cinco tipos de alarmas para la seguridad en la administración de redes, las cuales se muestran en las siguientes líneas:

- Integridad: Cuida de violaciones que interfieran o interrumpan el flujo de datos de la red. Alerta de traslapamientos, de eliminaciones e incidentes no permitidos.

- Operacional: Se refiere a la respuesta inadecuada de un servicio, debido a un mal funcionamiento o a que el servicio ha sido mal invocado.

- Físico: Indica cuando se detecta alguna falla en el recurso que se maneja.

- Servicio de Seguridad: Vigila y reporta si alguno de los dispositivos que guardan la seguridad de la red ha detectado un ataque a ésta.

- Tiempo de Permanencia: Se encarga de vigilar que algún evento no ocurra fuera del tiempo asignado.

Función de Rastreo e Inspección de Seguridad 10164-8
Esta función se parece mucho a la función de control de bitácora pero con la variante de tener un rastreo e inspección de la bitácora que lleva un historial de los eventos acaecidos en la red, en esta norma se hace una inspección de la información concerniente al conteo, seguridad, desconexiones, conexiones y otras operaciones de administración.

Control de Acceso para Objetos y Atributos 10164-9
El objetivo de este documento es prevenir al administrador de la red de accesos no autorizados a un número de ciertos objetos de la administración. Aquí se definen las reglas y los mecanismos que darán acceso a los objetos administrados.

Función de Medición y Conteo 10164-10
En este estándar se define como se identificarán y registrarán las cargas, costos y nivel de uso de la red, para así mantener una buena administración de la misma.

Función de Monitoreo de la Carga de Trabajo 10164-11
En este documento se especifica la manera en que se hará el monitoreo de la carga de trabajo que presente la red para la adecuada administración de los recursos de la misma. Se establecen procedimientos para el establecimiento de umbrales de advertencias de peligro, así como niveles de carga. Se establecen los valores de los parámetros anteriormente enunciados bajo los cuales la red se aproxima a sobrecargas. Define cómo medir la utilización de los recursos y cómo inicializar las diversas condiciones relacionadas con la administración de los objetos.

Funciones de la serie 10164-X
Hasta el momento no existe la información concreta acerca de lo que especificamente se definirá en cada una de éstas.

Se establecen procedimientos para el establecimiento de rangos de advertencias de peligro, así como, niveles de carga, en otras palabras se establecen los valores de los parámetros anteriormente enunciados bajo los cuales la red se aproxima a sobrecargas.

Funciones de la serie 10164_X
De estas normas hasta el momento no existe todavía la información concreta de lo que se definirá en cada una de éstas dentro de estas funciones se encuentran todas las que aparecen en la figura. 3.2 c del capítulo tres.

APENDICE

B

REVISION DE LA NOTACION DE SINTAXIS ABSTRACTA (ASN.1)

APENDICE B

REVISION DE LA NOTACION DE SINTAXIS ABSTRACTA (*ABSTRACT SINTAXIS NOTATION*, ASN.1)

Cada pieza de información que se intercambia entre los usuarios de OSI tiene un tipo y un valor. El tipo es una clase de información, tal como entero, booleano, octeto, etc. Un tipo puede utilizarse para describir una colección o grupo de valores. Por ejemplo, el tipo entero describe a todos los valores numéricos que no tienen una parte decimal.

El valor es una instancia del tipo, tal como un número o un texto. Por ejemplo, si decimos "P de tipo entero" y "P=9", esto significa que la instancia de P tiene un valor de 9. En un mensaje de OSI, los campos pueden estar definidos de un tipo entero o cadena de caracteres con una cierta instancia o valor.

Para que las máquinas puedan interpretar datos, deberán conocer primero el tipo de los datos (valores) para que sean procesados. Los estándares definen varios tipos predefinidos los cuales se resumen en la Tabla B.a.

BOOLEAN	Identifica datos lógicos (condiciones cierto o falso).
ITEGER	Números con signo y sin parte decimal (cardinales).
BIT STRING	Datos binarios (secuencias de unos y ceros).
OCTET STRING	Texto o datos que pueden ser descritos por una secuencias de octetos (*bytes*)
NULL	Podría ser un valor en el cual hay varias alternativas pero ninguna es aplicable.
SEQUENCE	Es un tipo estructurado que está definido por la referencia de una lista ordenada de tipos.
SEQUENCE OF	Es un tipo estructurado que está definido por la referencia de un solo tipo. Cada valor en el tipo es una lista ordenada (si existe).
SET	Es un tipo estructurado similar a *SEQUENCE*, excepto que está definido por la referencia de una lista no ordenada de tipos y permite que los datos sean mandados en cualquier orden.
SET OF	Es un tipo estructurado similar a *SEQUENCE*, excepto que está definido por la referencia de un solo tipo y cada valor en el tipo es una lista no ordenada (si existe la lista).
CHOICE	Modela un tipo de datos seleccionado de una colección de tipos alternativos y permite una estructura de datos para almacenar más de un tipo.
SELECTION	Modela una variable cuyo tipo proviene de una elección previa.
TAGGED	Modela un nuevo tipo de uno ya existente pero con un identificador distinto.
ANY	Modela datos cuyo tipo no tiene restricciones. Puede utilizarse con cualquier tipo válido.
OBJECT IDENTIFIER	Es un valor asociado a un objeto o un grupo de ellos.
CHARACTER STRING	Modela cadenas de caracteres para algún conjunto de ellos.
ENUMERATED	Es un tipo sencillo, sus valores son dados por distintos identificadores como parte del tipo de notación.
REAL	Modela valores reales, por ejemplo $M \times B^e$.
ENCRYPTED	Un tipo cuyo valor es el resultado de la encriptación de otro tipo.

Tabla B.a Tipos de datos en ASN.1.

Otra característica importante de estos estándares es el uso de banderas. Para distinguir a los diferentes tipos, una estructura de valores (por ejemplo, un registro de una base de datos) o un solo elemento (un campo dentro del registro de base de datos), puede tener una etiqueta asociada que identifica el tipo. Para ilustrar esto, supongamos que una etiqueta del mensaje de alarma de la administración de red OSI se llame *PRIVATE* 22, éste es usado para identificar al registro e informar al receptor acerca de la naturaleza de su contenido, por lo cual ASN.1 provee una etiqueta para cada tipo.

ASN.1 define cuatro clases de tipo. Cada etiqueta se identifica por su clase y su número (como en el ejemplo de *PRIVATE* 22). Las clases de tipo se definen como:

- Universal: Tipos independientes de la aplicación.

- Aplicación amplia: Son específicos de la aplicación pero que se emplean en otros estándares, tales como OSI, X.400, Sistema de Manejo de Mensajes (*Message Handling System*, MHS), Manejo de Acceso y Transferencia de Archivos (*File Transfer and Access Management*, FTAM), etc.

- Contexto específico: Son específicos a la aplicación y que están limitados a un conjunto dentro de la misma aplicación.

- Uso privado: Reservados para uso privado y no están definidos en los estándares.

Varias etiquetas se emplean para la asignación universal. Recuerde que una etiqueta se usa para identificar a una clase y que consta de dos partes: un identificador de clase y un número. En la **Tabla B.b** se muestran las etiquetas para la clase universal:

UNIVERSAL 1	*BOOLEAN*
UNIVERSAL 2	*ITEGER*
UNIVERSAL 3	*BITSTRING*
UNIVERSAL 4	*OCTETSTRING*
UNIVERSAL 5	*NULL*
UNIVERSAL 6	*OBJECTIDENTIFIER*
UNIVERSAL 7	Descripción de objeto
UNIVERSAL 8	*EXTERNAL*
UNIVERSAL 9	*REAL*
UNIVERSAL 10	*ENUMERATED*
UNIVERSAL 11	*ENCRYPTED*
UNIVERSAL 12-15	Reservados para uso futuro
UNIVERSAL 16	*SEQUENCE y SEQUENCE OF*
UNIVERSAL 17	*SET y SET OF*
UNIVERSAL 18	*NumericString*
UNIVERSAL 19	*PrintableString*
UNIVERSAL 20	*TeletexString*
UNIVERSAL 21	*VideotexString*
UNIVERSAL 22	*IA5String*
UNIVERSAL 23	*UTCTime*
UNIVERSAL 24	*GeneralizedTime*
UNIVERSAL 25	*GraphicString*
UNIVERSAL 26	*VisibleString*
UNIVERSAL 27	*GeneralString*
UNIVERSAL 28	*CharacterString*
UNIVERSAL 29+	Reservados para adiciones

Tabla B.b Etiquetas para la clase UNIVERSAL.

Reglas de ASN.1

ASN.1 impone una serie de reglas al programador, dichas reglas son sencillas pero muy importantes. A continuación se presenta un resumen de ellas:

- Varios tipos predefinidos se incluyen en el estándar.

- Los nombres de todos los tipos deberán comenzar con una letra mayúscula.

- Las palabras reservadas se deberán escribir todas sus letras con mayúsculas y tienen un significado especial dentro del estándar.

- Ciertos nombres deberán comenzar con una letra minúscula. Se insertan estos nombres para asistir a las personas en la lectura de un código en ASN.1. Estos nombres no tienen efecto sobre ningún código posterior.

Símbolos en la notación ASN.1

Un par de paréntesis triangulares (< >) encierran a un elemento tal como un nombre de tipo, un nombre de módulo, una definición de tipo o a un cuerpo de módulo. El ::= significa "definido como".

La siguiente notación describe las reglas para la notación de un módulo:

< nombre de módulo > *DEFINITIONS* ::= *BEGIN*
< cuerpo de módulo > *END*

El nombre de módulo identifica de manera única a un módulo y es un identificador ASN.1. La palabra *DEFINITIONS* indica que el módulo es definido con las definiciones ASN.1 que se encuentren entre las palabras *BEGIN* y *END*. En otras palabras, un módulo contiene otras definiciones propias de ASN.1.

Dentro de los módulos existen las definiciones de tipos. Ellos tienen la siguiente forma:

< nombre de tipo > ::= < definición de tipo >

Esta notación es un ejemplo de un tipo "sencillo". Se llama así por que especifica directamente el conjunto de sus propios valores. El nombre de tipo es el identificador del tipo. La definición de tipo describe la clase y varios otros atributos de los cuales se hará sólo un resumen.

El siguiente ejemplo muestra como tres definiciones de tipo son codificados dentro de un módulo:

< nombre de módulo > *DEFINITIONS* ::= *BEGIN*
< nombre de tipo > ::= < definición de tipo >
< nombre de tipo > ::= < definición de tipo >
< nombre de tipo > ::= < definición de tipo >
END

Al ejemplo anterior lo llamaremos "tipo estructurado". Este contiene una referencia a uno o más tipos diferentes. Los tipos dentro del tipo estructurado son llamados tipos de componente.

Por otra parte, en la Figura B.a resume los mayores apectos de ASN.1. Las banderas ASN.1 se categorizan en cuatro clases, a cada clase se le asigna un número. Las cuatro clases pueden codificarse con una estructura sencilla (una codificación en la cual el tipo no incluye a otros tipos) o una más compleja estructura (una codificación que incluye otros tipos). La categoría "otros" no está definida en el estándar.

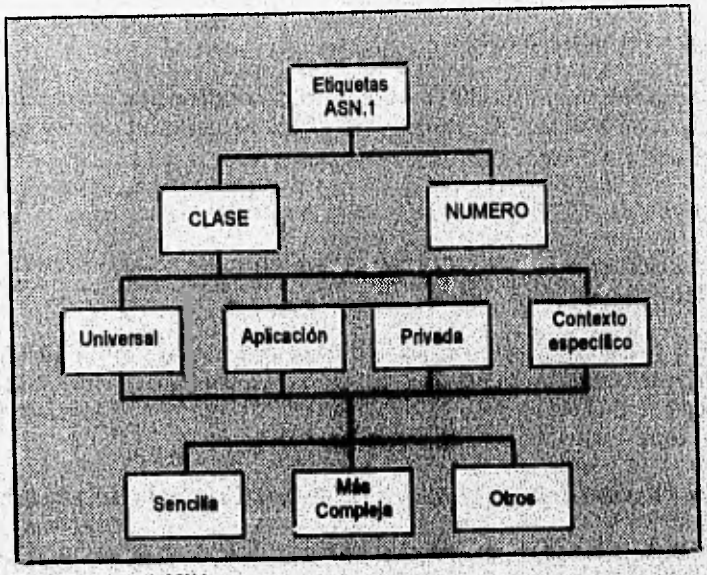


Figura B.a Clases de ASN.1.

Ejemplos de codificación ASN.1 en la administración de red OSI

Nos ayudaremos de algunos ejemplos sencillos para juntar las piezas que componen al ASN.1. Los ejemplos emplean a CMIP el cual se explica en el capítulo 3 de este trabajo, sin embargo, dichos ejemplos no requieren que el lector conozca este protocolo.

Considere el siguiente código que define a un objeto manejado base:

```
ObjetoManejadoBase ::= SEQUENCE {
        claseObjetoManejadoBase         ClaseObjeto,
        instanciaObjetoManejadoBase     InstanciaObjeto }

ClaseObjeto ::= CHOICE {
        sinFormaGlobal          [0] IMPLICIT OBJECT IDENTIFIER
        sinFormaEspecífica      [1] IMPLICIT INTEGER }

InstanciaObjeto ::= CHOICE {
        nombreDistintivo        [2] IMPLICIT NombreDistintivo
        sinFormaEspecífica      [3] IMPLICIT OCTET STRING
        formaNumerada           [4] IMPLICIT INTEGER }
```

En el primer bloque de código se define al ObjetoManejadoBase como de tipo *SEQUENCE*, el cual significa que es una lista ordenada de tipos. Estos tipos están definidos dentro de los paréntesis llaves como ClaseObjeto e InstanciaObjeto. Note que la ClaseObjeto e InstanciaObjeto se asignan a los nombres claseObjetoManejadoBase e instanciaObjetoManejadoBase respectivamente. Estos nombres han sido designados para conveniencia del lector. Ellos no tienen efecto sobre la codificación del valor del tipo y deberán comenzar con una letra minúscula.

El nombre del tipo ClaseObjeto se define en el segundo segmento del código. Dicho tipo se define como *CHOICE*. Esta notación significa que el valor de tipo *CHOICE* deberá ser sólo una de las alternativas que están dentro de los paréntesis llaves. En este ejemplo, el ClaseObjeto deberá tomar alguno de los valores definidos con los nombres opcionales formaGlobal o formaNoEspecífica. También, en esta parte del código se indica que *CHOICE* no tiene etiqueta. De esta manera, todos los elementos en un módulo de elección deberán ser etiquetados.

Podemos observar los nombres opcionales en el segundo segmento del código; sin ellos, las notaciones [0] *IMPLICIT OBJECT IDENTIFIER* y [1] *IMPLICIT INTEGER* serían poco comprensibles.

El resto de esta parte del código significa que los [0] y [1] asignan nuevos números de etiquetas a las dos opciones de ClaseObjeto. A partir de aquí, en lugar de que estos tipos comiencen identificados como *OBJECT IDENTIFIER* e *INTEGER* con los valores de 6 y 2 (ver Tabla B.2), ellos son ahora definidos como 0 y 1. Es posible que la codificación de estos tipos de etiqueta pudieron haber sido [*APPLICATION* 4], [*PRIVATE* 6], y así sucesivamente. Si palabras reservadas como *APPLICATION* o *PRIVATE* se incluyen en paréntesis cuadrados, los tipos son definidos como de aplicación-específica o privada respectivamente. Sin embargo, como la notación no contiene estos símbolos, la clase de etiquetas es de contexto-específico. Este término

denota que el significado del valor puede derivarse sin mayor definición y codificación, después de todo, el valor final será etiquetado con 0 o 1. De esta manera, se conoce de que contexto es la etiqueta del tipo.

La notación *IMPLICIT* es opcional pero muy útil. Si está presente, el valor original de la etiqueta no aparecerá en la codificación. Si no está presente, ambas etiquetas aperecerán en dicha codificación, por supuesto que se requiere de *bits* extras para representar esta información redundante.

El tipo *OBJECT IDENTIFIER* es un valor distinguible de todos los demás valores y se usa para identificar sin ambigüedades cualquier objeto. Típicamente, se usa para asignar un identificador único a un elemento tal como un PDU, un archivo, un módulo de biblioteca de funciones, etc. Esto se emplea frecuentemente en ISO e Internet en sus convenciones para nombrar elementos. La etiqueta [0] sirve para determinar las especificaciones del identidicador del objeto.

El tipo InstanciaObjeto se define en el tercer segmento del código. La única notación nueva que aparece en esta parte es NombreDistintivo. Recordemos que la administración de red OSI emplea a un nombre distintivo para la identificación plena de un objeto en un árbol de información de administración.

La Figura B.b es un ejemplo de código CMIP que define al nombre distintivo. Note que el código utiliza la palabra reservada *IMPORTS*. Esta característica de ASN.1 se usa para listar los tipos que están definidos en cualquier parte haciendo referencia a esta definición. Se emplea por que obvia la doble codificación del tipo.

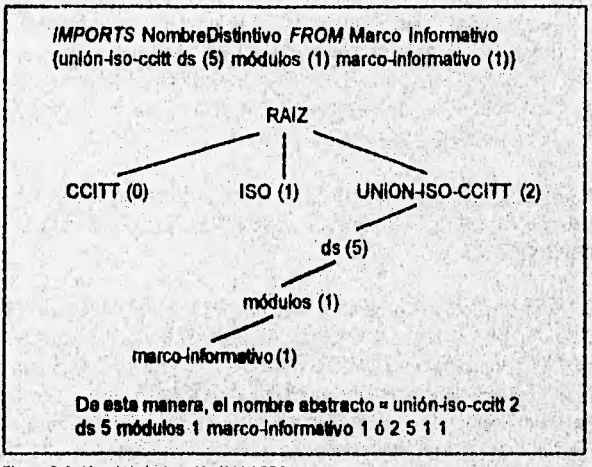


Figura B.b Uso de la instrucción *IMPORTS*.

El código del cuadro superior utiliza la forma de asignar nombres de CCITT e ISO para identificar el objeto como unión-iso-ccitt ds (5) modulos (1) marco-información (1). Esta notación puede ser seguida hacia abajo del árbol por una de las ramas para responder con el valor del nombre distintivo de 2511.

La figura B.c provee un ejemplo de un árbol de directorio y cómo se usa al ASN.1 para describir los elementos de directorio. Se aprovecha la figura para abundar más sobre la explicación sobre el nombre distintivo. El nombre distintivo es una secuencia de nombres distintivos relativos, el cual es un *SET OF* AfirmacionValorAtributo. El *SET OF* es similar a *SEQUENCE* excepto que los elementos en el primero no tienen que ser ordenados como deben de serlo en *SEQUENCE*.

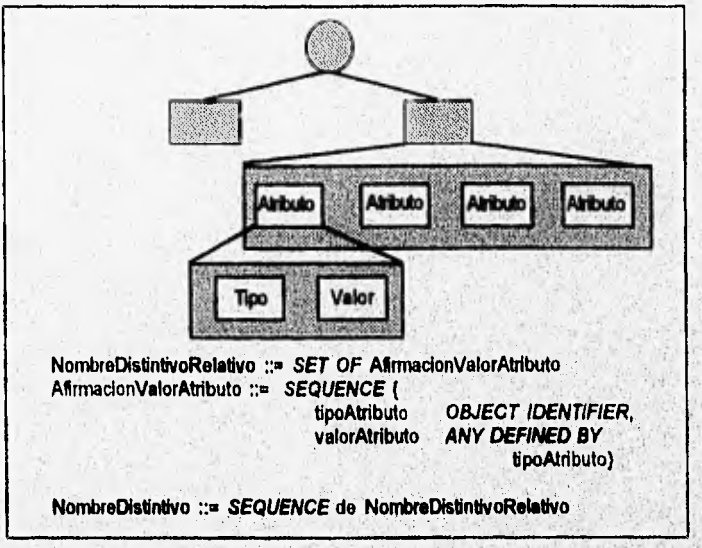


Figura B.c Identificación de elementos de directorio valiéndose de ASN.1.

Finalmente, se definen a los atributos tipo y valor como de tipos *OBJET IDENTIFIER* y *ANY DEFINED BY*. El segundo tipo también es codificado simplemente como *ANY* y se usa cuando el tipo no tiene que ser especificado; en otras palabras, cualquier tipo ASN.1 definido puede ser utilizado. El tipo *ANY* es muy general y no se emplea en algunos estándares, sin embargo la administración de red OSI hace un uso extenso de él.

La generalidad de *ANY* molesta a algún sector de usuarios, por lo que el estándar ASN.1 fue mejorado para permitir un *ANY DEFINED BY*. Esta notación utiliza <nombre> para identificar un elemento solicitado de el *SEQUENCE* y para dar información acerca de qué tipo podría asumir *ANY*. En la figura, *ANY DEFINED BY* realmente significa *OBJECT IDENTIFIER* ya que el <nombre> se remplaza por tipoAtributo, el cual es un *OBJECT IDENTIFIER*. Esta es una forma segura para definir un tipo.

BIBLIOGRAFIA

LIBROS

ALABAU, A.

Teleinformática y Redes de Computadoras. 2a ed., México, Ed. Publicaciones Marcombo,1987. 351 p. (serie: Mundo Electrónico)

BLACK, Uyless D.

Network Managment Standards / the OSI, SNMP and CMOL protocols. 1a ed., New York, Mc Graw Hill, 1992. 336 p. (Uyless Black series on computer communications)

_____

Redes de Computadoras / Protocolos, Normas e Interfaces. México, 1a ed., Ed. Macrobit / Ra-Ma, 1990. 421p.

_____

TCP/IP and Related Protocols. New York, 1a,Mc Graw Hill, 1992, 336 p. (Uyless Black series on computer comunications)

KOSIUR, Dave y Nancy E. H. Jones

MacWorld / Networking / Hand book. 1 ed., San Mateo, CA.,Ed. IDG books World Wide, 1992. 540 p.

MALAMUT, Karl.

Analysing Novell Networks. 1 ed., Nueva York. Ed. VamsNostramp Reinhold, 1992, 343 p.]

NOVELL.

Netware System Interface. Technical Overview. 1 ed., Massachusetts, U.S.A. Ed. Addison Wesley, 1990, 346 p.

WHITE, Gene.

Internetworking and addressing. 1 ed., Nueva York. Ed. McGraw Hill, 1992. 209 p. Uyless Black series on computer communications.

WHITE, John A.

Técnicas de Análisis Económico en Ingeniería. Tr. Vicent Agut Armer, México, Ed. LIMUSA, 1981, 577 p.

MANUALES

Digital Equipment Corporation. **DECnet / OSI for VMS**. 1 ed., U.S.A. 1990, 300 p.

Bosch García, Carlos. **La Técnica de Investigación Documental**. Facultad de Ciencias Políticas y Sociales. UNAM. México, 1978. 69 p.

Garza mercado, Ario. **Manual de Técnicas de Investigación**. El Colegio de México. México, 1981. 3 ed. 287 p.

Novell. **Netware v 3.11. Installation NetWare**. Utha U.S.A., 1991. 200 p. Novell

Novell. **Netware v 3.11. TCP/IP Transport**. Utha U.S.A., 1991. 150 p. Novell

Novell. **Netware v 3.11. System Administration**. Utha U.S.A., 1991. 501 p. Novell

Universidad Pedagógica Nacional. **Redacción e Investigación Documental I**. México, 1982. 233 p.

FOLLETOS

Network General Corparation™. **Company Overview**. U.S.A., 1991. 172 p.

Novell. **NetWare® Distributed Management Services**. *White Paper*. U.S.A. Octubre, 1993. 16 p.

Novell. **NetWare® Hub Services**. *Integrated hub management providing reliable and affordable connectivity*. U.S.A. Octubre, 1993. 2p.

Novell. **NetWare® LANalyzer Agent 1.0**. *Distributed network monitoring and troubleshooting for the NetWare Management System*. U.S.A. Octubre, 1993. 2p.

Novell. **Netware® Management Agent 1.5**. *Open, interoperable software to enable central management of NetWare 3.x and 4.x servers.* U.S.A. Octubre, 1993. 2p.

Novell. **NetWare® Management System**. *A family of products for enterprise wide management NetWare networks.* U.S.A. Octubre, 1993. 4 p.

APUNTES Y CURSOS

Arrieta Márquez, Norberto. **Introducción a la Red CECAFI**. Secretaría General. Centro de Cálculo. Facultad de Ingeniería. UNAM. Marzo, 1992. 139 p.

Introducción a Redes (LAN) de Micros (Parte I). División de Educación Continua. Facultad de Ingeniería. UNAM. Septiembre, 1991. Apuntes 298 p.

La Seguridad en Redes (LAN) y el Supervisor. Taller de redes LAN de Microcomputadoras. División de Educación Continua. Facultad de Ingeniería. UNAM. Diciembre, 1991. Apuntes 150p.

Redes Digitales. *Actualidades y Perspectivas*. División de Educación Continua. Facultad de Ingeniería. UNAM. Agosto, 1993. Apuntes 940 p.

Redes (LAN) de Micros (Parte II). División de Educación Continua. Facultad de Ingeniería. UNAM. Noviembre, 1991. Apuntes 305 p.

Seminario de Conectividad Avanzada. Intersys. Noviembre, 1993. Apuntes 68 p.

Unix TCP/IP. Dirección General de Servicios de Cómputo Académicos. UNAM. Agosto, 1993. Apuntes 200 p.

REVISTAS

Raman, Lakshmi. **CMISE functions and Services**. *IEEE Communications Magazine*. n. 5, v. 31, Piscataway, N.J. U.S.A. Mayo, 1993. 46-50.

Treece, Terry. **Control from the Console**. *LAN Technology*. n. v. , Redwood City, CA. U.S.A. Octubre, 1992. 67-78.

Cevallos de Rosillo, Guadalupe. **El director de la orquesta. ¿Cómo debe ser un administrador de red?** *RED*. n. 22, año III, México, Junio, 1992. 16-17.

Adler, Joseph. **Importancia de la administración de redes**. *RED*. n. v. , México. Agosto, 1991. 17-23.

Hanes, Charles F. **Network General's Distributed Approach to Network Troubleshooting**. *LAN Technology*. n. 10, v. 7, Redwood City, CA. U.S.A. Octubre, 1991. 71-78.

Tie, Liao y Dominique Seret. **Network Management: Interoperatibility and Information Model**. *Computer Communications*. n. 10, v. 14, U.S.A. Diciembre, 1991. 588-597.

Editores de LAN Magazine. **OpenView. Hewlett Packard**. *LAN The Network Solutions Magazine*. n. 2. Vol. 9., San Francisco, CA. U.S.A. Febrero, 1994. 68-68.

H. Pyle, Raymond. **OSI Network Management Systems**. *IEEE Communications Magazine*. n. 5, v. 31, Piscataway, N.J. U.S.A. Mayo, 1993. 18-19.

T. Beerman, Richard. **Peeking at Packets: AG's EtherPeek and LocalPeek**. *LAN Technology*. n. 10, v. 7, Redwood City, CA. U.S.A. Octubre, 1991. 81-86.

Kaiten Quintanilla, Guillermo. **Productos de Novell para la administración de redes**. *RED*. n. 22, año III, México, Junio, 1992. 26-32.

Hurwics, Mike. **RMON amplia las posibilidades de SNMP**. *RED*. n. 22, año III, México, Junio, 1992. 18-25.

Kaiten Quintanilla, Guillermo. **SNMP abarca ahora aplicaciones y sistemas**. *RED*. n. 22, año III, México. Junio, 1992. 35-35.

Klerer, S. Mark. **System Management Information Modeling**. *IEEE Communications Magazine*. n. 5, v. 31, Piscataway, N.J. U.S.A. Mayo, 1993. 38-44.

Yemini, Yechiam. **The OSI Network Management Model**. *IEEE Communications Magazine*. n. 5, v. 31, Piscataway, N.J. U.S.A. Mayo, 1993. 20-29.