



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

REESTRUCTURACION, ACTUALIZACION Y MONITOREO DEL SERVICIO DE RESOLUCION DE NOMBRES DE UNA RED PUBLICA DE DATOS.

T E S I S

QUE PARA OBTENER EL TITULO DE: INGENIERO EN TELECOMUNICACIONES

P R E S E N T A N :

SANDRA DANIELA SANCHEZ MORAN GONZALO ORTIZ NICOLI

Y E L T I T U L O D E

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A :

MANUEL ANDRES JIMENEZ VELAZQUEZ

ASESOR: M.I. LAURO SANTIAGO CRUZ



CIUDAD UNIVERSITARIA

2002

TESIS CON
FALSA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A mi familia :

Agustín, Martha, Georgina, Osbe y Carlos.

Por su amor

A:

Miriam Sandoval, Magdalena Sánchez, Alejandro Pardo, Sergio Peredo, Mario Alberto Macedo, Benjamín Riquelme, Julieta Hernández, Miriam Vázquez, Octavio Cruz, Manuel González, Gerardo De la cruz, Pedro(Piedrita), Ricardo Flores, Irma Girón, Mauricio Jacobo, Ricardo Ibarra y Gonzalo García.

Por su amistad, cariño y comprensión.

En especial a Gonzalo Ortiz por su apoyo y esfuerzo en este trabajo.

A mi asesor:

M.I. Lauro Santiago Cruz.

Por su ayuda

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería.

Por mis estudios y por haber enriquecido mi vida.

Gracias

Daniela Sánchez Morán

A mi madre:

No tengo palabras suficientes para agradecerte todo el amor, el esfuerzo y la dedicación que me has dado a lo largo de mi vida, gracias por haberme ayudado a llegar a este momento y por esforzarte en hacer de mi una mejor persona.

A mi hermana:

Quien ha estado conmigo apoyándome en los momentos más importantes de mi vida. más que una hermana eres una amiga. Gracias por compartir este momento conmigo.

A Norma:

Por haber enriquecido mi vida con tu cariño, entusiasmo y energía, me siento muy afortunado tenerte a mi lado y me alegra poder compartir este momento contigo. "Le monde exist parce que deux etres s'aiment".

A mis compañeros de generación:

Rita, Julia, Kesia, Francisco, Rodolfo, Victor, Javier, Eduardo, Samuel y Luis que juntos tuvimos la suerte de formar parte de la primera generación de Ingenieros en Telecomunicaciones. Gracias por su amistad, entusiasmo y apoyo durante todos estos años. "Somos de Primera".

A Daniela Sánchez:

Por el privilegio de trabajar contigo, ya que sin tu empeño, entusiasmo, dedicación y paciencia no hubiera sido posible la realización de este trabajo.

Al Ing. Jesús Reyes:

Por todo el apoyo y la paciencia que tuvo conmigo y con la primera generación de Ingenieros en Telecomunicaciones, le agradezco su amistad y le reitero su calidad profesional y humana.

A la UNAM y a la Facultad de Ingeniería:

Por aceptarme en sus aulas para darme una sólida formación académica y humana.

Por encontrar en ustedes una parte muy importante de mi vida profesional y personal, les dedico nuestro esfuerzo. Mil gracias.

Gonzalo Ortiz Nicoli

A Rosario
Por el apoyo en todo momento para la culminación de mi carrera.

A mi madre Margarita
Por su abnegación y cariño sin interés.

A mis hermanas
Por indicarme el camino a seguir en mis estudios.

A mi tío Gustavo
Por compartir momentos difíciles.

A mis maestros y compañeros de Tesis.

Gracias

Manuel Andrés Jiménez Velázquez

ÍNDICE

	Página	
INTRODUCCIÓN	1	
CAPÍTULO I	CONCEPTOS BÁSICOS	5
1.1. Historia de Internet	5	
1.2. TCP/IP	6	
1.2.1. Características	7	
1.2.2. Estándares de TCP/IP	7	
1.2.3. Arquitectura de protocolo	7	
1.2.4. Capa de acceso	11	
1.2.5. Capa de Internet	12	
1.2.6. Capa de transporte	15	
1.2.7. Capa de aplicación	16	
1.2.8. Direccionamiento	16	
1.3. Evolución de los servicios de resolución de nombres	19	
1.3.1. Nombres y direcciones	20	
1.3.2. Tabla de <i>hosts</i>	21	
1.3.3. Servicio de información de red	23	
1.3.4. Servicio de nombres de dominio	24	
1.4. Estructura del DNS de Internet	24	
1.4.1. Espacio de nombres de dominio	25	
1.4.2. Delegación	29	
1.4.3. Servidores y zonas	30	
1.4.4. Aplicaciones de resolución de nombres	31	
1.4.5. Métodos de resolución	32	
1.4.6. Cache	34	
1.4.7. Herramientas de diagnóstico	36	
1.5. BIND	37	
1.5.1. Archivos de BIND	37	
CAPÍTULO II	ANÁLISIS DE LA PROPUESTA	42
2.1. Análisis de la Propuesta	42	
2.1.1. Situación Actual	42	
2.1.2. Causas de falla en el servicio de DNS	44	
2.2. Desempeño de los enlaces y servidores	44	
2.2.1. Ocupación de los enlaces	45	
2.2.2. Desempeño de memoria y CPU en los servidores	61	
2.2.3. Tráfico de DNS	69	
2.3. Distribución de DNS en los servidores de acceso	77	
2.4. Inventarios	78	
2.5. Análisis de tipos y versiones de <i>software</i> para los servidores	82	

CAPÍTULO III	DISEÑO DEL SERVICIO DE DNS	84
3.1.	Diseño de la arquitectura del servicio de DNS	84
3.1.1.	Distribución física	84
3.1.2.	Distribución lógica	86
3.1.3.	Distribución de dominios	87
3.1.4.	Asignación de servidores	88
3.2.	Diseño del sistema de administración	89
3.3.	Diseño del sistema de monitoreo	94
3.3.1.	Monitoreo del servicio	95
3.3.2.	Monitoreo del desempeño de los servidores	95
3.3.3.	Monitoreo de los enlaces	96
3.4.	Requerimientos del servicio	96
3.4.1.	Ubicación física	96
3.4.2.	Definición del equipamiento	96
3.4.3.	Versiones de BIND	97
3.4.4.	Seguridad de servidores	98
3.4.5.	Dimensionamiento de enlaces	99
CAPÍTULO IV	IMPLANTACIÓN Y PUESTA EN OPERACIÓN	101
4.1.	Implantación	101
4.1.1.	Instalación de los servidores	101
4.1.2.	Instalación y configuración de QIP	106
4.1.3.	Pruebas del funcionamiento de QIP	111
4.1.4.	Configuración de BIND	113
4.1.5.	Pruebas de resolución	117
4.1.6.	Pruebas de transferencia	119
4.2.	Puesta en operación	122
4.2.1.	Adecuación de sitios	122
4.2.2.	Estrategia de migración	122
4.2.3.	Pruebas de operación	123
4.2.4.	Monitoreo de los servidores de DNS	126
CAPÍTULO V	RESULTADOS Y CONCLUSIONES	137
5.1.	Resultados	137
5.1.1.	Desempeño de la CPU	141
5.1.2.	Desempeño de la memoria RAM	141
5.1.3.	Desempeño de la memoria SWAP	142
5.1.4.	Sistema de Administración	142
5.1.5.	Enlaces	143
5.1.6.	Distribución de cargas	144
5.1.7.	Versiones de BIND	145
5.1.8.	Sistema de monitoreo	145
5.2.	Conclusiones	145

BIBLIOGRAFÍA

147

ANEXO A Vital Net

ANEXO B Especificaciones del Equipo Netra T1120

ANEXO C QIP Enterprise

ANEXO D Firehunter

ANEXO E Sun Management Center 3.0

ANEXO F Programas de Verificación de Resolución de Nombres

ÍNDICE DE FIGURAS

CAPÍTULO I

Figura 1.1. Modelo de referencia OSI	8
Figura 1.2. Capas de la arquitectura TCP/IP	10
Figura 1.3. Encapsulamiento de datos	10
Figura 1.4. Estructura de datos	11
Figura 1.5. Formato de un datagrama IP	14
Figura 1.6. Ruteo a través de Gateway's	14
Figura 1.7. Formato de archivo host.txt	23
Figura 1.8. Espacio de nombres de dominio	25
Figura 1.9. Restricciones de duplicidad	26
Figura 1.10. Definición de dominio	27
Figura 1.11. Definición de subdominio	27
Figura 1.12. Diferencias entre zona y dominio	31
Figura 1.13. Proceso de resolución de nombres	33
Figura 1.14. Estructura para resoluciones inversas	35

CAPÍTULO II

Figura 2.1. Topología de la red de servidores de DNS de la Ciudad de México	45
Figura 2.2. Utilización del enlace uno de la Ciudad de México	47
Figura 2.3. Utilización del enlace dos de la Ciudad de México	48
Figura 2.4. Utilización del enlace tres de la Ciudad de México	49
Figura 2.5. Utilización del enlace cuatro de la Ciudad de México	50
Figura 2.6. Topología de la red de servidores de DNS de la Ciudad de Guadalajara	52
Figura 2.7. Utilización del enlace uno de la Ciudad de Guadalajara	53
Figura 2.8. Utilización del enlace dos de la Ciudad de Guadalajara	54
Figura 2.9. Utilización del enlace tres de la Ciudad de Guadalajara	55
Figura 2.10. Topología de la red de servidores de DNS de la Ciudad de Monterrey	57
Figura 2.11. Utilización del enlace uno de la Ciudad de Monterrey	58
Figura 2.12. Utilización del enlace dos de la Ciudad de Monterrey	59
Figura 2.13. Utilización del enlace tres de la Ciudad de Monterrey	60
Figura 2.14. Desempeño del servidor dns	63
Figura 2.15. Desempeño del servidor nsmex1	64
Figura 2.16. Desempeño del servidor nsmex2	65
Figura 2.17. Desempeño del servidor nsmex3	66
Figura 2.18. Desempeño del servidor nsgdl1	67
Figura 2.19. Desempeño del servidor nsmt1	68
Figura 2.20. Consultas por segundo al servidor dns.	71
Figura 2.21. Consultas por segundo al servidor nsmex1	72
Figura 2.22. Consultas por segundo al servidor nsmex2	73
Figura 2.23. Consultas por segundo al servidor nsmex3	74
Figura 2.24. Consultas por segundo al servidor nsgdl1	75
Figura 2.25. Consultas por segundo al servidor nsmt1	76

CAPÍTULO III

Figura 3.1. Distribución regional de servidores	85
Figura 3.2. Ejemplo de distribución en servidores de DNS	89
Figura 3.3. Arquitectura del sistema de administración	93

CAPÍTULO IV

Figura 4.1. Carga de servidores	108
Figura 4.2. Parámetros de desempeño del servidor dnsadm-interno	127
Figura 4.3. Parámetros de desempeño del servidor nsmex1	128
Figura 4.4. Parámetros de desempeño del servidor nsmex2	129
Figura 4.5. Parámetros de desempeño del servidor nsmex3	130
Figura 4.6. Parámetros de desempeño del servidor nsmex 4	131
Figura 4.7. Parámetros de desempeño del servidor nsmdl1	132
Figura 4.8. Parámetros de desempeño del servidor nsmdl2	133
Figura 4.9. Parámetros de desempeño del servidor nsmt1	134
Figura 4.10. Parámetros de desempeño del servidor nsmt2	135

ÍNDICE DE TABLAS

CAPÍTULO I

Tabla 1.1. Clases de direcciones IP	18
Tabla 1.2. Ejemplos de ccTDLs	29

CAPÍTULO II

Tabla 2.1. Servidores de DNS	43
Tabla 2.2. Resumen de enlaces de la Ciudad de México	51
Tabla 2.3. Resumen de enlaces de la Ciudad de Guadalajara	56
Tabla 2.4. Resumen de enlaces de la Ciudad de Monterrey	61
Tabla 2.5. Características de memoria en los servidores	62
Tabla 2.6. Distribución de servidores de acceso	77
Tabla 2.7. Estado general del servicio por zonas	78
Tabla 2.8. Resolución por zonas por servidor DNS	80
Tabla 2.9. Distribución de redes	81
Tabla 2.10. Relación de versiones de Bind y de sistema operativo	82
Tabla 2.11. Vulnerabilidad de Bind 8.1.2	83

CAPÍTULO III

Tabla 3.1. Distribución de servidores por región	86
Tabla 3.2. Distribución de dominios por servidor	88
Tabla 3.3. Asignación de servidores	88
Tabla 3.4. Comparación de sistemas de administración de DNS	91
Tabla 3.5. Lista de bugs corregidos en Bind 8.2.5	97
Tabla 3.6. Lista de acceso para enrutadores	99

CAPÍTULO IV

Tabla 4.1. Distribución del disco	102
Tabla 4.2. Configuración de la tarjeta de red	103
Tabla 4.3. Perfiles de usuarios	103
Tabla 4.4. Fechas de puesta en operación	122
Tabla 4.5. Resumen de mediciones de Firehunter	126
Tabla 4.6. Promedio de parámetros de servidores	136

CAPÍTULO V

Tabla 5.1. Porcentajes de CPU	141
Tabla 5.2. Porcentajes de RAM	142
Tabla 5.3. Porcentajes de SWAP	142
Tabla 5.4. Métricas de administración	143
Tabla 5.5. Porcentajes promedio de utilización	144

INTRODUCCIÓN

Uninet es una empresa mexicana que da servicios de Internet, la cual ha alcanzado una alta participación y posicionamiento desde hace seis años en el mercado de las telecomunicaciones de nuestro país.

Dentro de los servicios que ofrece **Uninet** para Internet se encuentran los siguientes:

- Internet Residencial
- Internet Corporativo
- Correo electrónico
- Hospedaje de páginas web
- Servicio de resolución de nombres por DNS

Internet Residencial

Este servicio está enfocado a los clientes residenciales para que puedan acceder a Internet desde sus hogares, con cobertura en más de 1,100 poblaciones en toda la República Mexicana. Existen dos modalidades de conexión, por línea telefónica analógica y digital.

Internet Corporativo

A través de este servicio se da conexión a Internet a empresas que requieren de uno o más enlaces dedicados para poder tener acceso a Internet.

Correo Electrónico

Mediante este servicio el cliente tiene la posibilidad de enviar y recibir mensajes a través de Internet, pudiendo almacenar hasta 10 Mb de información.

Hospedaje de Páginas

Por este servicio el cliente puede almacenar en un máximo de 10 Mb sus páginas web de manera que estén siempre disponibles en Internet.

Servicio de Resolución de Nombres por DNS

El servicio de resolución de nombres permite a los clientes encontrar las direcciones de Internet de los servidores de web y de correo, facilitándole aprender nombres en vez de direcciones de Internet. Esta función la realiza un programa llamado DNS (*Domain Name Service*, Servicio de Nombres de Dominio) que es el más utilizado.

La resolución de nombres por DNS es ocupada por todos los clientes. En el caso de los clientes corporativos existe una mayor interacción, ya que ellos requieren tener su nombre asociado a las direcciones de Internet de sus servidores, por razones de imagen e independencia. Además, si los clientes corporativos lo requieren, **Uninet** provee el servicio de administración de DNS, evitándoles adquirir al menos un servidor y un administrador. En el caso de los clientes residenciales el servicio es algo que pasa desapercibido, por no intervenir directamente sobre él.

Pareciera este servicio, no estar dirigido a todos los clientes, pero es importante mencionar que sin él no se podría tener acceso a ninguna página web ni hacer uso del correo electrónico. Para **Uninet** la resolución de nombres es un servicio crítico y de vital interés para soportar los demás servicios que presta. Sin embargo se han presentado diversos problemas que han mermado la calidad del mismo.

Problemática de resolución de nombres por DNS

Desde los inicios de **Uninet**, se puso en operación el servicio de DNS con seis servidores. En ese entonces estos equipos eran suficientes para brindar el servicio que requería la empresa. Con el paso del tiempo la demanda creció y los equipos empezaron a ser insuficientes; cabe comentar que estos equipos fueron administrados por diferentes personas, lo que a la postre significó que se perdiera con ello el control de los servidores, y por darle mayor atención a las nuevas tecnologías de transporte de datos se descuidó este servicio.

Los servidores presentan problemas serios de desempeño, que han degradado la calidad del servicio, reflejándose en lentitud para resolver nombres, y en ocasiones hasta en indisponibilidad del servicio. Muchos de estos problemas se relacionan con la poca memoria de los equipos y con el crecimiento exponencial de Internet. No existe ningún tipo de monitoreo sobre este sistema, y cuando ocurre una falla, ésta se detecta hasta que los usuarios se quejan por no poder navegar.

En el servicio de Internet residencial se tienen problemas de balanceo de cargas, por lo que algunos servidores son más consultados que otros, saturándolos y dejando ociosos a otros.

El DNS trabaja básicamente con tablas de nombres y direcciones, estas tablas están en archivos ASCII (*American Standard Code for Information Interchange*, Código Estándar Americano para Intercambio de Información) y su sintaxis es muy propensa a errores. Las tablas en **Uninet** son muy grandes, resultado del número de direcciones administradas (alrededor de 500,000) y del número de clientes manejados (alrededor de 600 clientes corporativos y 900,000 residenciales), en promedio las tablas por servidor tienen 20,000 líneas, las cuales presentan errores de sintaxis muy severos. El factor principal de estos errores es que la configuración se hace manualmente, siendo muy susceptible a errores humanos en la captura de la información.

Debido a los crecientes ataques sobre los servidores de DNS los fabricantes de software y de hardware han hecho mejoras en la seguridad del DNS y en los servidores. En el caso de **Uninet** no se han implantado muchas de estas mejoras y se corre el riesgo de afectar el servicio. Existe mucha información sobre qué es y cómo funciona el servicio de DNS, pero casi no existe información sobre la administración del mismo, lo que dificulta que **Uninet** mejore el servicio.

Objetivo y estructura de la tesis

El objetivo del presente trabajo de tesis es analizar, proponer e implantar una solución a la problemática del servicio de resolución de nombres en **Uninet**, la cual permita contar con una plataforma robusta y un sistema confiable para brindarles a los usuarios un mejor servicio, que satisfaga la demanda actual y futura.

La tesis se estructuró en los siguientes capítulos:

Capítulo I. Conceptos básicos. Provee el fundamento teórico de este trabajo de tesis, como son: la historia de Internet y las características de protocolo TCP/IP que lo sustenta; se presenta el desarrollo y estructura del DNS, además de la descripción del programa BIND.

Capítulo II. Análisis de la Propuesta. En este se presenta un análisis de todos los aspectos que afectan al servicio de DNS, entre ellos: el porcentaje de utilización

de los enlaces para cada ciudad donde se ubican los servidores, el desempeño de la memoria y la CPU, además de la cuantificación de peticiones hechas a los servidores en una semana típica.

Capítulo III. Diseño del servicio de DNS. Se describe la solución dada a la problemática detectada, a través de la definición de funciones, distribución de la carga de trabajo, un sistema de administración y uno de monitoreo.

Capítulo IV. Implantación y puesta en operación. Cubre la instalación de todos los equipos y el *software* utilizado, pruebas de funcionalidad y la puesta en operación.

Capítulo V. Resultados y Conclusiones. En este capítulo se muestran los resultados del diseño del servicio de DNS implementado y puesto en operación, comparando en tablas las diferencias antes y después de las modificaciones. Además se exponen las conclusiones de la presente tesis.

Finalmente se presenta la bibliografía consultada y los apéndices generados.

CAPÍTULO I

CONCEPTOS BÁSICOS

En este capítulo se presenta la evolución de Internet, así como las características que definen a la suite protocolos TCP/IP. También se describe la arquitectura de operación de TCP/IP junto con su modelo de comunicación. Para terminar, con la explicación a detalle de cada una de las capas de su estructura jerárquica y la forma en que está organizado el direccionamiento. Así pues, la estructura del protocolo y su funcionamiento en Internet proporcionan las bases para entender posteriormente el DNS y su aplicación en redes públicas de datos.

1.1. Historia de Internet

En 1969 en EE.UU. la ARPA (*Advance Research Projects Agency*, Agencia de Programas de Investigación Avanzada) del Departamento de Defensa; promovió la investigación y el desarrollo de una red experimental de conmutación de datos a través de paquetes. Esta red, llamada ARPANET, fue diseñada con el fin de estudiar técnicas que proporcionarían sistemas de comunicaciones de datos seguros y confiables.

A raíz de esto, el IP (*Internet Protocol*, Protocolo de Internet) y el TCP (*Transmission Control Protocol*, Protocolo de Control de Transferencia) fueron desarrollados inicialmente en 1973, por el informático estadounidense Vinton

Cerf, como parte de un proyecto dirigido por el ingeniero estadounidense Robert Kahn y patrocinado por la ARPA.

En 1975 la ARPANET se convirtió de una red experimental a una red operacional, y la responsabilidad de administrarla fue otorgada al Departamento de Defensa de EE.UU. Sin embargo, el desarrollo de la ARPANET no se detiene solo por el hecho de haber dejado de ser una red experimental. Los protocolos TCP/IP fueron adoptados como estándares Militares en EE.UU. en 1983, y a todas las computadoras conectadas a la red, fue necesario adaptarles los nuevos protocolos. Al mismo tiempo la palabra *Internet* se volvía un termino de uso común para referirse a la ARPANET.

En 1985 la NSF (*National Science Foundation*, Fundación Nacional de Ciencias) crea la NSFNet y se adhiere a Internet. Evento significativo en la historia de Internet porque la NSF trajo consigo una nueva visión del uso de esta. Poco después el *World Wide Web* (WWW) se desarrolla en 1989 por el informático británico Timothy Berners-Lee para el Consejo Europeo de Investigación Nuclear (CERN, siglas en francés).¹

En 1990, la ARPANET formalmente deja de existir, la NSFNet cesa su rol y se forma el principal nodo² de Internet en 1995. Cada año Internet duplica su tamaño como red desde 1983; para el año de 1998 se conformaba por cerca de 95,000 redes en todo el mundo.

Finalmente con todos estos cambios, algo ha permanecido constante: "Internet sigue construido bajo los protocolos TCP/IP".

Internet es la colección de redes interconectadas en el ámbito mundial, que ocupa el Protocolo de Internet (IP) para enlazar varias redes físicas dentro de una red lógica.

1.2. TCP/IP

TCP/IP es la suite de protocolos común utilizado por todas las redes de computadoras conectadas a Internet, de forma que éstas puedan comunicarse entre sí. Tomando en cuenta que en Internet existen computadoras conectadas con sistemas operativos diferentes e incompatibles en ocasiones y con distintos tipos de *hardware*, además de toda la diversidad de medios de conexión. Conjuntamente, TCP/IP tiene ventajas significativas respecto a otros protocolos. Por ejemplo, consume pocos recursos de red. Igualmente puede ser implementado a un costo mucho menor que otras opciones disponibles actualmente.

¹ "Internet." *Enciclopedia® Microsoft® Encarta 2001*. © 1993-2000 Microsoft Corporation. Reservados todos los derechos.

² Nodo, en informática y Redes, se refiere a una ubicación que puede tener varios enlaces hacia uno o más destinos.

1.2.1. Características

TCP/IP es el conjunto de protocolos estándar en Internet. Estos protocolos cubren las distintas capas o niveles del modelo OSI (*Open System Interconnection*, Interconexión de Sistemas Abiertos), que define funciones de protocolos para el intercambio de información. Por estas razones se enuncian sus características más importantes:

- Estándares de Protocolo Abierto, disponibles de manera fácil y desarrollados de manera independiente de cualquier tipo de *hardware* o sistema operativo. Por lo mismo de su amplio soporte, TCP/IP es ideal para unificar diferentes tipos de *hardware* y *software*, inclusive sin comunicarse a través de Internet.
- Independencia de cualquier tipo de red física, esto facilita la integración de diferentes tipos de red. TCP/IP puede aplicarse sobre una red *Ethernet*, una *Token Ring*, una línea de Dial-up y virtualmente en cualquier medio de transmisión física.
- Un direccionamiento común que soporta cualquier dispositivo TCP/IP con una única dirección que ningún otro dispositivo tiene en toda la red, así sea una red mundial como Internet.

1.2.2. Estándares de TCP/IP

Las redes homogéneas, contienen el protocolo de comunicación diseñado exclusivamente para operar con la arquitectura de *hardware* de la red y su sistema operativo. Por lo tanto es difícil adherir otra red homogénea por su protocolo diseñado con otra arquitectura. TCP/IP pretende crear redes heterogéneas a través de protocolos abiertos que son independientes de cualquier diferencia de arquitectura o de sistema operativo. Los cambios o el desarrollo de los protocolos TCP/IP se llevan a cabo por consenso y no por decisión arbitraria de una sola empresa. Lo que hace posible el desarrollo de productos que puedan aprovechar la característica de ser protocolos abiertos.

TCP/IP no es un único protocolo, sino que es en realidad lo que se conoce como un conjunto de protocolos que cubren distintos niveles o capas. Obviamente los dos protocolos más importantes son TCP e IP que son los que dan el nombre al conjunto. Existen tres diferentes publicaciones de estándares para protocolos diseñados bajo TCP/IP: un número importante bajo estándares militares (MIL STD, *Military Standards*), otros están publicados como IEN (*Internet Engineering Notes*), y la mayor parte en el RFC (*Requests For Comments*).

1.2.3. Arquitectura de protocolo

Para discutir sobre redes de computadoras, es necesario utilizar términos que tienen un significado especial. Tener una referencia común siempre es necesaria para entender las comunicaciones de datos.

Un modelo de arquitectura desarrollado por ISO (*International Standards Organization*, Organización Internacional de Estándares) es frecuentemente ocupado para describir la estructura y funcionamiento de las comunicaciones de datos. Este modelo de arquitectura que es nombrado OSI (*Open System Interconnection*, Modelo de Interconexión de Sistemas Abiertos) proporciona una referencia común para discutir sobre comunicaciones. El modelo de referencia OSI contiene siete *capas*, los términos definidos por este modelo describen las funciones que deben entregar los protocolos con el objeto de intercambiar información entre diferentes sistemas. Cada capa del modelo OSI representa una función ejecutada cuando los datos son transferidos entre aplicaciones de una red.

Aparte, cada capa depende de las que están debajo de ella, y a su vez proporciona alguna funcionalidad a las superiores. La figura 1.1 identifica cada capa por nombre y aporta una corta descripción. Al observar esta figura, las capas parecen estar apiladas unas sobre otras. Por esta apariencia, la estructura del modelo OSI es igualmente llamada protocolo de Pila.

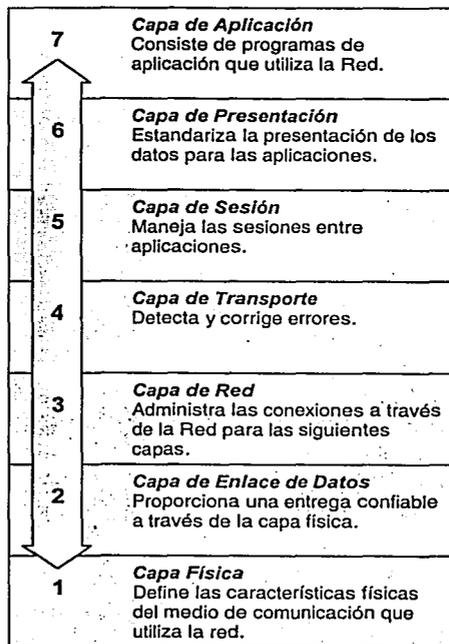


Figura 1.1. Modelo de referencia OSI.

Una capa no define un simple protocolo, ésta sólo define funciones para la comunicación de datos que pueden ser ejecutados por cualquier número de protocolos. Dicho esto, cada capa tiene múltiples protocolos, cada uno realiza una función acorde con ella. Cada protocolo se comunica con su contraparte. La contraparte es una implementación del mismo protocolo en la equivalente capa del sistema remoto.

Por lo tanto, existen reglas de convenio para pasar los datos entre capas en una computadora, porque cada capa está comprometida con enviar los datos desde una aplicación local a su equivalente aplicación remota. Las capas siguientes confían en las anteriores para transferirles los datos a través de la red.

Los datos pasan de forma descendente por las capas antes de ser transmitidos sobre la red por los protocolos de capa física. En el lado remoto los datos suben a través de las capas para ser recibidos por la aplicación. Las capas por si mismas no necesitan saber la función entre las capas siguientes o anteriores, solo deben conocer como pasarse los datos ente ellas.

Aislando las funciones de comunicación de red, de las diferentes capas del modelo, se minimiza el impacto de los cambios tecnológicos sobre el protocolo. Nuevas aplicaciones pueden ser incluidas sin cambiar físicamente la red y, nuevos aditamentos físicos pueden ser adheridos sin modificar las aplicaciones de *software*.

En contraste, cabe mencionar que existen inconveniencias en el modelo a pesar de tener un valor didáctico práctico. Algunas están señaladas sobre las dimensiones de las capas de aplicación y sesión que se encuentran vacías.

La terminología del modelo de referencia OSI, sin embargo, nos ayuda a describir al Modelo TCP/IP, pero para completar la idea de esto, es necesario usar un modelo de arquitectura más cercano a las características del TCP/IP.

Generalmente TCP/IP es percibido como un modelo compuesto por un número menor de capas que el modelo de referencia OSI. El modelo de cuatro niveles Ilustrado en la figura 1.2 esta basado en cuatro capas.

Este modelo representa las capas dentro de la Jerarquía del protocolo TCP/IP. Como en el modelo OSI, los datos pasan por la pila de capas hacia abajo cuando serán enviados a la red, y hacia arriba por las capas cuando son recibidos de la red. En la estructura de cuatro capas del TCP/IP se ve el sentido en que los datos son tomados por las capas desde la de aplicación hasta la capa física.

Cada capa en la pila agrega información de control para asegurar su apropiada entrega. Esta información de control es llamada *encabezado*, por que es colocado enfrente de los datos que serán transmitidos. Las capas tratan a toda la

información recibida de la capa anterior como datos y coloca su propio encabezado enfrente de la información.

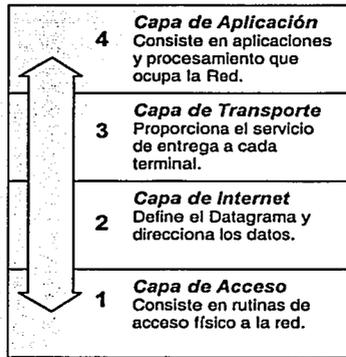


Figura 1.2. Capas en la arquitectura de TCP/IP.

La información de entrega agregada en cada capa se le conoce como *encapsulado*. (Observar la figura 1.3).

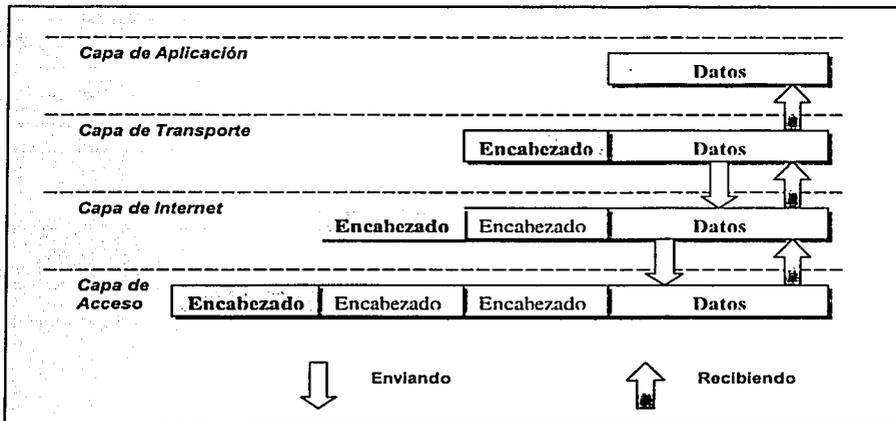


Figura 1.3. Encapsulamiento de datos.

Cuando los datos se reciben ocurre lo contrario. Cada capa extrae su encabezado antes de pasar los datos a la siguiente capa. Como la información recibida fluye de las capas inferiores, toda información que es recibida es interpretada como encabezado y datos. Aún así, cada capa tiene su propia estructura de datos y su propia terminología para describir su estructura.

Cada capa tiene su propia estructura de datos independiente. Conceptualmente cada capa se desentiende de la estructura de las demás capas. En realidad la estructura de datos en cada una de ellas se diseña en compatibilidad con las otras capas, para hacer eficiente la comunicación entre sí.

Para transmitir información a través de TCP/IP, ésta debe estar dividida en unidades de menor tamaño. Esto representa ventajas en el manejo de los datos que son transferidos y, por lo tanto es común en cualquier protocolo de comunicación.

La figura 1.4 muestra los términos utilizados por las diferentes capas del TCP/IP para referirse a los datos transmitidos. La capa de aplicación llama a los datos *streams*, en la capa de transporte los llama *segmentos*, la capa de Internet los nombra *datagramas*, y finalmente la de acceso *tramas*. Así, TCP/IP utiliza diferentes tipos de capa de Red, cada una puede tener una terminología específica para nombrar a los datos.

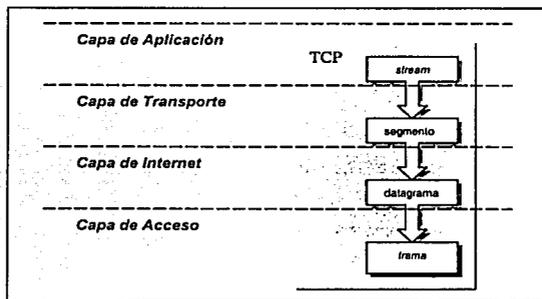


Figura 1.4. Estructura de datos.

1.2.4. Capa de acceso

La capa de acceso es la capa más baja dentro de la jerarquía de TCP/IP. Los protocolos pertenecientes a esta capa proporcionan el medio a la red para entregar los datos a los demás dispositivos. Particularmente, esta capa define cómo usar la red para transferir los *datagramas* IP. A diferencia de los protocolos

de alto nivel³, los protocolos de capa de acceso deben saber todos los detalles que constituyen la transferencia de datos (la estructura de los paquetes de datos, el direccionamiento, etc.) para dar formato de forma correcta a los datos que serán transmitidos de acuerdo con el tipo de red. La capa de acceso en TCP/IP comprende las funciones de las dos capas más bajas del modelo de referencia OSI (capa de Enlace de Datos, y capa física).

Cada ocasión que emerge nueva tecnología, nuevos protocolos de acceso son desarrollados, para que redes TCP/IP puedan hacer uso de ella. Consecuentemente existen muchos protocolos de acceso, uno para cada estándar de red física.

Las funciones que operan en este nivel incluyen encapsulamiento de *datagramas* IP dentro de tramas transmitidas por la red, mapeo de direcciones IP a las direcciones físicas utilizadas por la red. La dirección IP debe convertirse en una dirección que sea apropiada para la red física sobre la cual el *datagrama* se transmite.

Aprovechar un enfoque de una sola red (dentro de un conjunto de redes interconectadas) significa ampliar el esquema de estratificación por capas de protocolos para adicionar la función de conexión entre computadoras.

Finalmente, resumiendo las funciones de la capa de acceso:

- Determinación de la entrega de *datagramas* desde el origen hasta el destino sobre distintos tipos de red física.
- Retransmisión de tramas de bits que se hayan detectado incompletas.
- Control de congestión, contabilidad y problemas con la heterogeneidad entre las redes.

1.2.5. Capa de Internet

La capa superior a la capa de Acceso en la jerarquía del protocolo TCP/IP es la capa de *Internet*. El protocolo de Internet (IP), es el corazón del TCP/IP y el más importante protocolo en la capa de Internet. IP proporciona el servicio de entrega de paquetes de datos sobre el que están constituidas las redes TCP/IP. Todos los protocolos, en las capas superiores e inferiores de IP, ocupan el protocolo de Internet para entrega de datos. Todos los datos fluyen a través de IP, ingresando o saliendo a pesar del destino final.

El protocolo de Internet incluye funciones como:

- Definir el *Datagrama*, que es la unidad básica de transmisión en el Internet.
- Define el esquema de direccionamiento.

³ Son protocolos de alto nivel : IP, TCP, UDP, etc. Por su estructura de datos y su distribución de funciones sobre una red.

- Traslada los datos entre la capa de Acceso y la de Transporte.
- Enruta los *datagramas* a las terminales remotas (computadoras).
- Realiza la fragmentación y re-ensambla a los *datagramas*.

Algunas características más de la capa de Internet: Primero, IP es un protocolo orientado a *no-conexión*, es decir que no necesita intercambiar información de control (llamada "*handshake*") para establecer una conexión de computadora a computadora antes de transmitir un dato. En contraste, un *protocolo orientado a conexión* intercambia información de control con el sistema remoto para verificar que está listo para recibir datos antes de que se envíen. Cuando el *handshake* tiene éxito, está diciendo que el sistema tiene establecida la conexión. El protocolo de Internet cuenta con los protocolos de otras capas para que establezcan dicha conexión.

Inclusive IP confía en otros protocolos para que hagan la labor de detectar y corregir errores durante la recepción y transmisión de datos. Por esta razón en ocasiones es llamado el IP como un protocolo poco confiable por que no contiene corrección de errores y recuperación de código. Esto no quiere decir que el protocolo sea equivocado, por el contrario, la función confiable de él es únicamente la entrega de datos. IP no supervisa si llegan correctamente a su destino. Diferentes protocolos se encargan de realizar la revisión cuando es requerida.

El TCP/IP fue diseñado para transmitir datos sobre la ARPANET, que fue una red de conmutación de paquetes. Un paquete es un bloque de datos que acarrea con él la información necesaria para ser entregado. Una red de conmutación de paquetes utiliza la dirección destino contenida en la información del paquete para pasar los datos de una red física⁴ a otra, trasladando estos hasta su destino final. Cada paquete viaja en la red independientemente de los demás paquetes.

El *datagrama* es el formato de paquete definido por IP. La figura 1.5 es una representación grafica de un *datagrama* en el protocolo de Internet. Las primeras cinco o seis palabras⁵ de 32 bits del *datagrama* es información de control llamada encabezado. El valor típico de la longitud de encabezado son cinco palabras, la sexta palabra es opcional. Debido a que la longitud del encabezado es variable, existe un campo llamado *IHL (Internet Header Length, longitud de encabezado)* que indica la longitud del encabezado en unidades de palabra. El encabezado contiene toda la información necesaria para entregar el paquete de datos.

El protocolo de Internet entrega el *datagrama* tomando en cuenta la dirección destino localizada en la quinta palabra del encabezado. La dirección destino es un estándar de 32 bits que identifica el destino de red. Si la dirección destino es una dirección de una computadora, los paquetes serán entregados directamente a su

⁴ Se entiende por "red física" al conjunto de computadoras y medios de transmisión física capaces de interconectarse entre si para establecer una comunicación.

⁵ Una palabra puede estar compuesta de 4 bits, 16, 32 o más; en el caso del TCP/IP las palabras están compuestas de 32 bits.

destinatario. Si la dirección destino no pertenece a la red local, el paquete es entregado a un *Gateway*⁶ para la entrega. Los *Gateways* son dispositivos que conmutan paquetes entre diferentes redes físicas.

0					10					20					31																
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Versión		IHL		Tipo de servicio		Longitud total																									
Identificación					Banderas		Desplazamiento de fragmento																								
TTL			Protocolo			Checksum																									
Dirección IP origen																															
Dirección IP destino																															
Opciones IP (si las hay)												Relleno																			
Datos																															

Figura 1.5. Formato de un *datagrama* IP.

Al decidir qué *Gateway* utilizar se le conoce como enrutar. IP tiene la decisión de enrutar cada paquete de forma individual.

Comúnmente los *Gateways* son referidos como "Enrutadores IP" porque ocupan el protocolo de Internet para enrutar los paquetes de datos entre redes.

La figura 1.6 muestra el uso de los *Gateways* para transportar paquetes. La terminal o computadora destino procesa los paquetes por las cuatro capas del protocolo, mientras los *Gateways* procesan los paquetes solo dentro de la capa de Internet, donde también se toman las decisiones de ruteo.

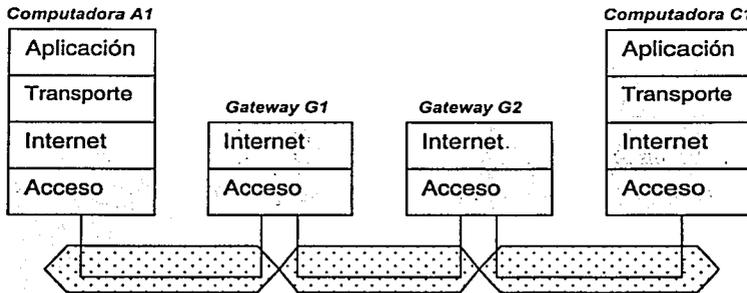


Figura 1.6. Ruteo a través de *Gateway*'s.

Los sistemas de redes locales solo pueden entregar paquetes de datos entre sus dispositivos de red. Paquetes desde A1 destinados a C1 son transportados a

⁶ Un *Gateway* IP tradicional es un Enrutador.

través de G1 y G2. La computadora A1 primero entrega el paquete al G1, con el cual comparte la red A. El Gateway G1 entrega el paquete a G2 sobre la red B. El Gateway G2 entonces entrega el paquete a C1, porque comparte la red C. A1 confía en que G1 pueda enrutar de la mejor forma los paquetes para que lleguen a su destino.

Cada *datagrama* es enrutado por diferentes redes y es necesario para el Protocolo de Internet dividir los *datagramas* en pequeñas partes. Un *datagrama* recibido desde cualquier red puede ser tan largo como para ser transmitido en paquetes de menor tamaño sobre distintas redes. Esto ocurre sólo cuando los *Gateways* interconectan redes diferentes. Cada tipo de red tiene de referencia una "Unidad Máxima de Transmisión" (MTU, *Maximum Transfer Unit*), que es la longitud en bits del paquete más grande que puede ser transferido sobre ella. Si el *datagrama* recibido de una red es más grande que el MTU de quien recibe, es necesario dividir en pequeños fragmentos para transmitirlos. Este proceso es conocido como fragmentación.

El formato de cada fragmento es el mismo que el de cualquier *datagrama*. El encabezado guarda en la segunda palabra la información que identifica al fragmento del *datagrama* y la forma de re-ensamblarlo de nueva cuenta.

1.2.6. Capa de transporte

La capa de transporte se sitúa por encima de la capa de Internet y su función principal es la comunicación punto a punto final entre dos equipos sin importarle la trayectoria por la que viaje la información, ya que este trabajo se le deja a la capa inferior. Los dos protocolos más importantes de esta capa son:

- *TCP* (*Transport Control Protocol*, Protocolo de Control de Transporte)
- *UDP* (*User Datagram Protocol*, Protocolo de *datagramas* de Usuario)

El protocolo *UDP* le permite a los programas acceso directo al servicio de entrega de *datagramas*, la ventaja es que les permite intercambiar mensajes sobre la red con un mínimo de encabezados en el protocolo. Con esto la cantidad de tráfico que se transmite es menor, optimizando el ancho de banda y el tiempo de respuesta. *UDP* es un protocolo poco confiable ya que está orientado a no conexión, es decir, no tiene ningún método para verificar que los datos enviados lleguen a su destino correctamente. *UDP* utiliza 16 bits para asignar el puerto⁷ origen y el puerto destino.

Las aplicaciones que requieren de transmitir información de forma confiable utilizan *TCP*, ya que es un protocolo confiable orientado a conexión. Para garantizar la información se utiliza un protocolo llamado *PAR* (*Positive Acknowledgment with Re-Transmission*, Confirmación Positiva con Re-

⁷ Puerto es un número que permite diferenciar los servicios en los protocolos *TCP* y *UDP*. Por ejemplo, *Telnet* utiliza el puerto 23 de *TCP*.

Transmisiones), de forma normal un sistema manda un paquete usando *PAR* y espera determinado tiempo para recibir una confirmación; si ésta llega, manda el siguiente paquete de datos hasta terminar la transmisión; en caso de que no llegue una confirmación se vuelve a enviar la información, a esta acción se le conoce como retransmisión. Existe un número máximo de retransmisiones antes de dar por fallida una comunicación. La unidad de datos intercambiados se llama segmento, cada segmento contiene un campo para detección de errores que utiliza el receptor para determinar si se presentaron errores durante la transmisión, si el segmento llega sin errores se manda una confirmación, si el segmento está dañado, el receptor lo descarta, después de un período de tiempo el emisor retransmite la información.

1.2.7. Capa de aplicación

En la cima del protocolo TCP/IP se encuentra la capa de aplicación, ésta incluye todos los procesos que usa la capa de transporte para entregar la información. En esta capa hay muchos protocolos de aplicaciones, la mayoría provee servicios de usuario y frecuentemente se agregan más. Los protocolos más comúnmente implementados son:

- TELNET: Protocolo de terminal de red que provee conexiones remotas a un equipo a través de la red.
- FTP: (*File Transfer Protocol*, Protocolo para Transferencia de archivos): éste protocolo es usado para intercambiar archivos de forma interactiva con otro equipo.
- SMTP: (*Simple Mail Transfer Protocol*, Protocolo de Transferencia Simple de Correo): con éste protocolo se brinda el servicio de correo electrónico en Internet.

Mientras que FTP, SMTP y TELNET son los protocolos más comúnmente implementados en TCP/IP, hay muchos otros que son más usados por los administradores de sistemas, entre ellos tenemos:

- DNS: (*Domain Name Service*, Servicio de Nombre de Dominio).
- RIP: (*Routing Information Protocol*, Protocolo de Enrutamiento de Información).
- NFS: (*Network File System*, Sistema de Archivos en Red).

1.2.8. Direccionamiento

Las direcciones IP son números globales únicos de 32 bits, asignados por el NIC (*Network Information Center*, Centro de Información de Red). Las direcciones globales únicas permiten que las redes IP en el mundo se comuniquen una con otra. Por simplicidad y claridad, estos bits se representan normalmente mediante cuatro octetos (8 bits=1 octeto). Cada octeto se representa como un número

decimal entre 0 y 255, y después se separa a cada octeto por un punto, esta notación se conoce como punto-decimal.

Por ejemplo, una dirección IP representada en bits podría ser la siguiente:

10101100000100000011001000001010

Para representar esta dirección en formato punto-decimal, dividimos la dirección anterior en 4 octetos:

10101100	00010000	00110010	00001010
----------	----------	----------	----------

y convertimos cada uno de ellos en un número decimal:

172	16	50	10
-----	----	----	----

La dirección entonces se escribe como 172.16.50.10.

Las direcciones IP se analizan en dos secciones, un identificador de la red y un identificador del *host*^a, las cuales varían haciendo muy flexible el número de redes y *hosts* que se pueden asignar.

Para clasificar las redes por tamaño, las direcciones IP se dividen en clases como se muestra en la tabla 1.1. Esta división se definió en el *RFC 791*, y consigue hacer un balanceo entre el número de redes y *hosts* en cada clase.

Ésta clasificación sigue algunas reglas básicas:

- Cada clase utiliza subsecuentemente menos bits en la sección del *host* y subsecuentemente más bits en el prefijo de la red.
- El límite entre la red y las secciones de la identificación del *host* está fijado en cada clase.
- Cada clase utiliza los bits más significativos (el primer dígito de izquierda a derecha) de la dirección para identificar donde está el límite.

^a *Host* se define como un equipo conectado a Internet en cualquier punto, por lo que tiene asignado al menos una dirección IP, puede ser una computadora, un servidor, una impresora, un enrutador, etc.

Clase	Identificador de Clase	Prefijo de Red	Sección del host	Número de redes posibles	Número de <i>hosts</i> posibles por red	Total de direcciones de host por Clase
A	Primer octeto dentro del rango de 1-126	Primer octeto de 1.xxx.xxx.xxx a 126.xxx.xxx.xxx	Los restantes tres octetos de xxx.0.0.0 a xxx.255.255.255	126	16,777,214	2,113,928,964
B	Primer octeto dentro del rango de 128-191	Primeros dos octetos de 128.0.xxx.xxx a 191.255.xxx.xxx	Los restantes dos octetos de xxx.xxx.0.0 a xxx.xxx.255.255	16,384	65,534	1,073,709,056
C	Primer octeto dentro del rango 192-223*	Primeros tres octetos de 192.0.0.xxx a 223.255.255.xxx	El octeto restante de xxx.xxx.xxx.0 a xxx.xxx.xxx.255	2,097,152	254	532,676,608
D	Primer octeto dentro del rango 224-239	Esta clase está reservada exclusivamente para identificar grupos de mensajes dirigidos				
E	Primer octeto dentro del rango 240-254	Esta clase también está reservada para usos futuros				

Tabla 1.1. Clases de direcciones IP.

1.3. Evolución de los servicios de resolución de nombres

En 1969 cuando el NIC de la ARPANET es creado por Doug Engelbart, no se tenía una estructura de los nombres de los usuarios que hacían uso de la red.

La ARPAnet era tan pequeña, que se conocían todos los usuarios y servidores entre sí. En este esquema, la interacción entre los usuarios no tenía ningún problema cuando hacían uso del servicio de la red. Después de algunos años, la red creció en tamaño, por lo que se tenía que reestructurar el directorio global, pero por falta de infraestructura, nunca se concluyó.

En 1971, Peggy Karp concibe los *host mnemonics* (mnemónicos de *host*), o más simple los nombres de internet. Ella creó una tabla que relacionaba todas las resoluciones de redes con sus *hosts* en un archivo de texto, llamado *hosts.txt*, la tabla contenía todos los nombres de los *hosts* y su dirección IP. Los operadores de la red, tenían que instalar éste archivo en sus servidores locales, para que la computadora pudiera realizar la resolución de nombres sin saturar la red. Sin embargo, cuando un operador adicionaba una nueva máquina a la red, tenía que enviar un correo electrónico con toda la información requerida hacia la gente del SRI (*Stanford Research Institute*, Instituto de Investigaciones de *Stanford*), quienes a su vez, los recopilaban para incluirlos en la siguiente versión del *host.txt* y tenerlos disponibles en un servidor FTP.

Como la red creció en popularidad y nuevos *hosts* fueron sumándose, el tamaño del archivo *hosts.txt* aumentó sin proporción y no existía control de los registros. Si el operador de red no actualizaba su registro *hosts.txt*, provocaba en primer lugar una coalición de nombres y toda clase de confusiones. La coalición de nombres, ocurre cuando la red piensa que más de un *host* comparte el mismo dominio. Esto fue suficiente para que la ARPANET tuviera fallas en el servicio. Para la solución de este problema, los ingenieros concluyeron que se tenía que reemplazar el método del archivo *hosts.txt*.

En la década de los 80's nació el DNS y fue creado por Paul Mockapetris en colaboración con Jon Postel de la Universidad del Sur de California y posteriormente, Paul Vixie. Juntos desarrollaron lo que hasta ahora conocemos como el DNS y el BIND.

Originalmente, el uso del DNS involucró solamente instituciones académicas, de investigación y por supuesto, la milicia de los EEUU. Eran los tiempos en que las universidades empezaban a realizar su conexión a las múltiples redes. Por este crecimiento era importante establecer un orden en cuanto a los equipos que ingresaban a la red.

Se crearon entonces los nombres de dominio genéricos de primer nivel (*gTLD*, *generic Top-level Domain*, Dominio de nivel superior genérico), que originalmente fueron administrados por el SRI-NIC de la universidad de Stanford en Menlo Park,

California, pero pronto cambiaría a InterNIC (Centro de información de redes de Internet).

En 1992, la NSF, quien administraba el *back bone* de Internet decide licitar la operación del InterNIC y en 1993, a través de un convenio de cooperación, le otorga esta función a la empresa NSI (*Network Solutions Inc.*, Asociación para la Soluciones de la Red).

Desde mediados de 1996, la IANA (*Internet Assigned Numbers Authority*, Autoridad para la Asignación de Números de Internet) es el organismo administrador de las direcciones de IP y nombres de dominio, se mantiene de manera operativa en el Internet. A escala continental, la IANA delega grandes bloques de direcciones IP a los denominados registros regionales, de los que, de momento, existen tres en el mundo:

RIPE NCC (*Réseaux IP Européens Network Coordination Center*, Centro Coordinador de Redes para el registro de IP Europeo) es el registro delegado de Internet a nivel europeo y se encarga, entre otras cosas, de la asignación de bloques de direcciones IP a los proveedores de servicios Internet en Europa y su área de influencia.

AP-NIC (*Asia Pacific Network Information Center*, Centro de información de la Red para Asia Pacífico) lleva a cabo la tarea de asignación de bloques de direcciones IP a los proveedores de la región de Asia Pacífico.

ARIN (*American Registry For Internet Numbers*, Registro Americano para Números de Internet) se encarga de la asignación de bloques de direcciones IP a los proveedores de Internet en América del Norte, América del Sur, el Caribe y África Sub-Sahariana.

1.3.1. Nombres y direcciones

Cada una de las interfaces en la red trabaja con el protocolo TCP/IP y es identificada por una dirección única de 32 bits. Un nombre de *host* puede ser asignado a un equipo que tiene una dirección IP. Los nombres son asignados a los equipos porque a comparación con las direcciones, son más fáciles de recordar y de escribir en un teclado correctamente. El software de la red local, no requiere de nombres, pero con el nombre asignado hace más fácil a los humanos el uso de la red.

En muchos casos, el nombre y la dirección numérica pueden ser usados alternadamente. Si un usuario de la red local desea conectarse, con TELNET, a su estación de trabajo, de dirección IP 128.66.12.2, puede entrar de las siguientes dos maneras:

```
1era % telnet 128.66.12.2
```

o usando el nombre del *host*, asociado con la dirección IP y entrar con el comando equivalente:

```
2da. % telnet peanut.nuts.com
```

Si un comando entra con una dirección o un nombre de *host*, la conexión a Internet siempre toma la dirección de IP. El sistema convierte el nombre de *host* a una dirección antes de hacer la conexión a la red. El administrador de la red local es responsable de asignar nombres y direcciones, y guardarlos en la base de datos usada para las conversiones de nombres a direcciones.

Trasladar los nombres dentro de sus direcciones, no es simplemente una tarea local. El comando *telnet peanut.nuts.com* está preparado para trabajar correctamente con todos los *hosts* que están conectados a la red, es decir, son datos seguros y confiables. Si *peanut.nuts.com* es conectado a Internet, todos sus *hosts* en el mundo deben de estar disponibles para trasladar el Nombre *peanut.nuts.com* dentro de su propia dirección. Por tanto debe de existir alguna facilidad para diseminar la información de los nombres de *hosts* para todos los que están conectados a la red.

Para esto existen dos métodos para trasladar nombres a direcciones. El más viejo y simple, es buscar en una tabla llamada *host table* (Tabla de *host*) el Nombre del *host* y encontrar su dirección IP. La otra es una técnica nueva llamada DNS para trasladar nombres a direcciones y viceversa.

1.3.2. Tabla de *hosts*

La tabla de *hosts* es un simple archivo de texto que asocia las direcciones de IP con el nombre del *host*. La tabla está en el archivo */etc/hosts* y contiene las direcciones de IP, separadas por un espacio en blanco de una lista de nombres asociadas con sus direcciones IP, los comentarios están identificados con el signo "#".

Por ejemplo, la tabla del *host peanut*, contiene las siguientes inscripciones:

```
#
# Tabla de IP           direcciones y Nombres de hosts
#
128.66.12.2            peanut.nuts.com        peanut
127.0.0.1              Localhost
128.66.12.1            almond.nuts.com        almond          loghost
128.66.12.4            walnut.nuts.com        walnut
128.66.12.3            pecan.nuts.com         pecan
128.66.1.2             filbart.nuts.com       filbart
128.66.6.4             salt.plant.nuts.com    salt.plant      salt
```

La primera inscripción en la tabla anterior, después de los comentarios, es para *peanut*. La dirección de IP 128.66.12.2 es asociada con el nombre de *host peanut.nuts.com* y el nombre alterno del *host* (o alias) *peanut*. El nombre de *host* y

todos sus alias resuelven para la misma dirección de IP, en este caso para 128.66.12.2.

Los alias proveen cambios en el nombre, cambios en la ortografía y nombres abreviados en los nombres de *hosts*, es decir, dan flexibilidad para probar direcciones, así como para desahogar rutas en una misma dirección, a los administradores de la red.

De la tercera a la séptima inscripción, son los alias con que cuenta la dirección de *peanut*. Aunque el sistema de la tabla de *hosts* ha sido superada por el DNS, ésta sigue siendo usada por las siguientes razones:

- Muchos sistemas tienen una tabla de *host* pequeña conteniendo información importante de nombres y direcciones de su red local. Estas tablas pequeñas son usadas cuando el DNS no corre, es decir en el arranque del sistema. Si invariablemente se usa el DNS, se tiene que crear un pequeño archivo llamado */etc/host*, que contenga inscripciones de los *hosts* para el *host* local, para las entradas y servidores de la red local.
- Hay sitios que usan el NIS (*Network Information Service*, Servicio de Información de la Red) para entrar a su base de datos de la tabla de sus *hosts*, éste se puede usar con el DNS, siempre y cuando se usen juntos. Los NIS también tiene inscripciones importantes sobre sus *hosts*.
- Existen sitios muy pequeños que utilizan tablas de *hosts*, y que no requieren de hacer cambios importantes y no necesitan comunicarse vía TCP/IP con sitios remotos, siendo así una desventaja para usar el DNS.
- Algunos sitios tienen software muy viejo que no pueden usar el DNS y no se pueden actualizar, para ellos sólo funciona un archivo de tabla de *hosts*.

Todos los *hosts* que están conectados a Internet deben de usar el DNS. A pesar de esto, muchos sitios de Internet siguen usando tablas de *hosts*. Para entender como se construyen estas tablas, se necesita trasladarse al sistema de Unix⁹ que provee comandos que automáticamente construyen los archivos */etc/hosts* y */etc/networks* con datos disponibles del NIC. El archivo */etc/networks* se usa para trasladar las direcciones de la red a los nombres de la misma.

El NIC tiene una tabla de *hosts* muy grande en Internet llamada *NIC host table* y es almacenada en la dirección *nic.ddn.mtl*, en el archivo *netinfo/host.txt*, en el cual incluyen todos los nombres y direcciones de los *hosts* para todos los sitios en Internet. Los nuevos *hosts* son adicionados bajo circunstancias especiales que impone el NIC, ya que la información del *host* nuevo muchas veces es limitada, por otro lado, la tabla es tan grande, que es un camino muy ineficiente para convertir nombres a direcciones IP, además que sería imposible darle servicio a todos los *hosts* de Internet al mismo tiempo.

⁹ Sistema operativo multiusuarios, multitareas y multiprocesos, muy popular para manejar aplicaciones muy grandes que demandan muchos recursos de cómputo.

1.3.3. Servicio de información de red

El NIC provee en su archivo *host.txt* (tabla de *hosts*), tres tipos de inscripciones: *Network records* (registro de redes), *gateway records* (registros de entrada) y *host record* (registro de huéspedes). Cada registro viene con su clave que identifica el tipo de registro, seguido por la dirección IP y uno o más nombres asociados con la dirección (alias). La dirección IP y el nombre del *host* son extraídos para construir el archivo */etc/host*. Los datos del registro de redes son usados para construir el archivo */etc/networks*. En la figura 1.7 se muestra el formato del archivo *host.txt*.

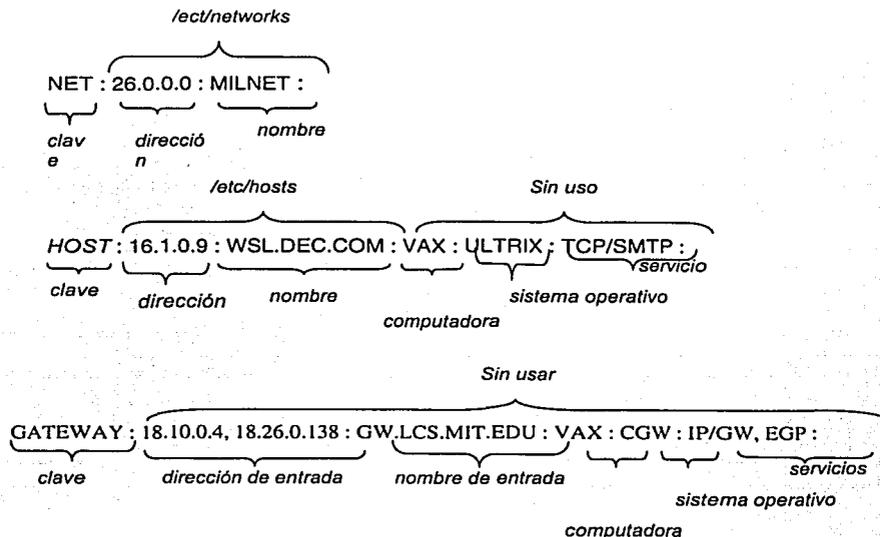


Figura. 1.7. Formato de archivo *host.txt*.

El archivo de *hosts.txt*, contiene una cantidad de información que Unix no usa, en el *Gateway records* no se usan todos ni tampoco el cuarto, quinto y sexto lugar del *host records*, ya que los datos que describen estos lugares (*hardware* de la computadora, sistema operativo y el servicio) no son confiables.

En el sistema operativo de Unix, existe un comando llamado *gettable*, cuya función es obtener el archivo *hosts.txt* y de ahí construir los archivos */etc/hosts* y */etc/networks*. La sintaxis es simple: *gettable host*, donde *host* es el nombre del sitio remoto del cual se quiere recobrar el archivo *hosts.txt*. Para obtener el archivo *hosts.txt* del *nic.ddn.mil* se debe de teclear como sigue:

```
% gettable nic.ddn.mil
```

Una vez que se obtiene el archivo *hosts.txt* se procede a construir los otros dos archivos */etc/hosts* y */etc/networks*, usando el comando *htable* con la siguiente sintaxis:

```
htable hosts.txt
```

que a su vez crea otros tres archivos de salida, que son los siguientes: *hosts*, *networks* y *gateway*, los cuales son copiados exactamente del sitio remoto tratado y de éstos se obtiene la información para construir los archivos */etc/hosts* y */etc/networks* en la terminal solicitante.

Mucha de la información en el archivo *hosts.txt* no es usada, como el *gateway records* es ignorado y el *host records* no es necesario porque ahora el DNS provee la información de los nombres de *hosts*. Solamente la información del servicio de red que viene en los *network records* es totalmente usada. El archivo */etc/networks* que es creado de los *network records*, se sigue usando para mapear las direcciones de la red con los nombres, porque muchos nombres no son incluidos en la base de datos del DNS.

1.3.4. Servicio de nombres de dominio

En el servicio de DNS la función primordial es hacer el mapeo entre los nombres de las computadoras y las direcciones de Internet, dicho de otra manera es básicamente una base de datos con información de *hosts*. El DNS es usado por todo el software entre redes, lo utilizamos siempre que enviamos un correo electrónico, cuando navegamos en el Internet o cuando accedemos a una terminal remota. Otra importante característica del DNS, es mantener la información de la computadora disponible todo el tiempo en Internet.

1.4. Estructura del DNS de Internet

El DNS es una base de datos distribuida con información de nombres y direcciones IP que trabaja en un esquema cliente-servidor. Esta compuesto de tres partes principales:

- **Cientes DNS:** También conocidos como *resolvers*, pueden ser cualquier equipo de cómputo conectado a Internet, los clientes DNS envían las peticiones de resolución de nombres a un servidor DNS. Las consultas que se hacen son para saber qué dirección IP está asociada a un nombre de dominio en particular o viceversa.
- **Servidores de Nombres:** Conocidos también por su nombre en inglés *name servers*. Los servidores DNS contestan a las peticiones de los clientes consultando su base de datos. Si no disponen de la dirección solicitada pueden reenviar la petición a otro servidor.

- Espacio de nombres de dominio: Su nombre en inglés es *domain name space*. Se trata de una base de datos distribuida entre distintos servidores.

1.4.1. Espacio de nombres de dominio

El DNS se indexa por nombres de dominio, cada dominio es esencialmente una ruta en una estructura de árbol llamada espacio de nombres de dominio, ésta estructura se ilustra en la figura 1.8 y es similar a la estructura de archivos de un sistema Unix. El árbol tiene una raíz única que se encuentra en la cima de la estructura y es conocida por su nombre en inglés *root*, este árbol puede ramificarse muchas veces en cada intersección, cada una de las intersecciones representa un nodo. La profundidad del árbol está limitada a 127 niveles.

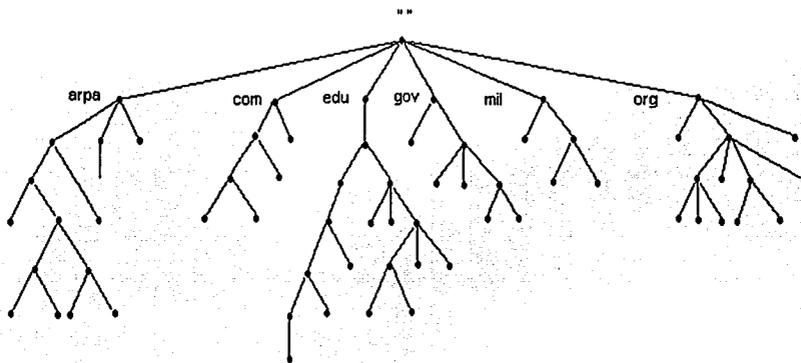


Figura 1.8. Espacio de nombres de dominio.

Cada nodo en el árbol tiene una etiqueta que puede tener una longitud de hasta 63 caracteres. La etiqueta nula, es decir, que tiene cero caracteres está reservada para *root*. El nombre de dominio completo de cualquier nodo en el árbol es la secuencia de etiquetas de la ruta existente desde el nodo hasta *root*. Los nombres de dominio son siempre leídos desde el nodo hasta *root*, es decir, subiendo por el árbol y en cada nodo el nombre es separado de los otros por un punto.

El DNS requiere que los nodos hijos de un mismo padre tengan etiquetas diferentes, ésta restricción garantiza que cada nombre de dominio identifica a un nodo único dentro del árbol. Esto se ilustra en la figura 1.9. La restricción no es una limitante ya que las etiquetas deben ser diferentes entre los nodos hijo de un mismo padre no entre todos los nodos del árbol.

Un dominio es simplemente un subárbol del espacio de nombres de dominio. El nombre de dominio de un dominio es el mismo que tiene el nodo padre y todos los nodos hijos pertenecen a ese dominio.

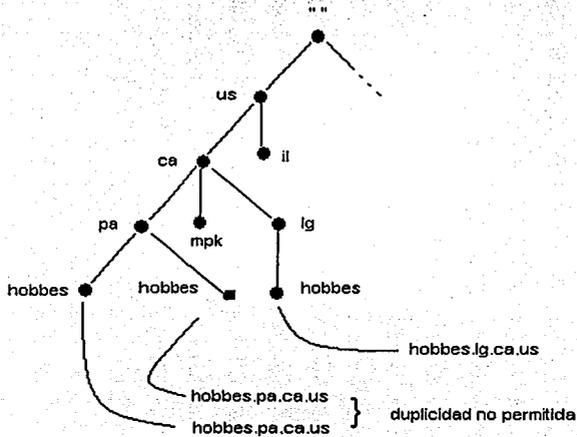


Figura 1.9. Restricciones de duplicidad.

Por ejemplo, el dominio *purdue.edu* es el que tiene el nodo llamado *purdue.edu* y todos los nodos que cuelgan de él pertenecen al dominio *purdue.edu*, tal como se muestra en la figura 1.10.

Cualquier nombre de dominio en el subárbol es considerado parte del dominio. Debido a que un nombre de dominio puede estar en muchos subárboles, un nombre de dominio puede pertenecer a varios dominios. Por ejemplo, el dominio *pa.ca.us* es parte del dominio *ca.us* y también parte del dominio *us*, como se muestra en la figura 1.11.

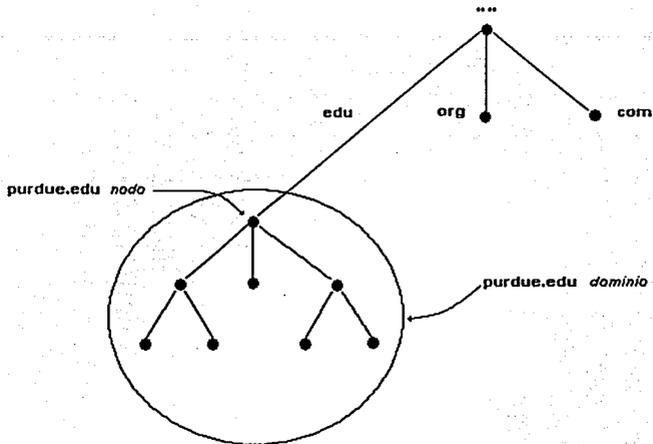


Figura 1.10. Definición de dominio.

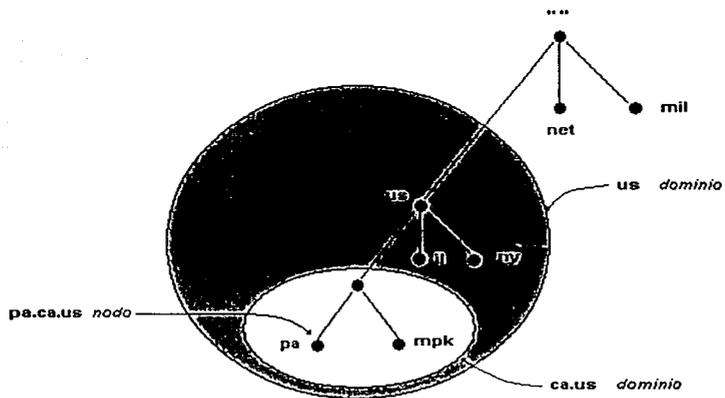


Figura 1.11. Definición de subdominio.

Podemos concluir que un dominio es sólo un subárbol del espacio de dominios, es importante tomar en cuenta que los *hosts* están representados por nombres de dominios. Hay que recordar que los nombres de dominios son sólo índices dentro de la base de datos del DNS. Los *hosts* están relacionados lógicamente, algunas veces geográficamente o por afiliaciones organizacionales, y no necesariamente por redes, direcciones o tipo de *hardware*.

El tipo de información que es recibida de una consulta depende del contexto en que se esté usando. Por ejemplo, mandar un correo electrónico a alguien en un dominio regresa información de enrutamiento de correo, mientras que hacer una conexión remota o TELNET al mismo, puede dar información del *host*.

Los dominios están clasificados en dos grandes grupos dentro del espacio de dominios.

Dominios de Primer Nivel: Mejor conocidos por su nombre en inglés *Top-Level Domain* (TLD), son todos aquellos dominios cuyo nodo padre es *root*.

Dominios de Segundo Nivel: Son todos aquellos dominios cuyo nodo padre es un dominio de primer nivel y de ahí hacia abajo.

El DNS no impone muchas reglas en las etiquetas de los nombres de dominio, no requiere que tengan algún significado en especial. Cuando se administra una parte de un dominio, se pueden decidir los nombres de forma totalmente independiente, si el administrador decide nombrar a sus máquinas o subdominios de la A a la Z puede hacerlo y ninguna autoridad en Internet puede forzarlo a cambiarlos.

En el espacio de nombre de dominio de Internet, existe una estructura ya definida para dominios de niveles superiores, estos siguen ciertas tradiciones (no reglas).

Los TLDs están divididos en siete grupos:

com Asignado para organizaciones comerciales y empresas.

edu Dedicado para las organizaciones educativas, principalmente las universidades.

gov Asignado para las instituciones gubernamentales de E.U.A.

mil Se utiliza para las organizaciones militares.

net Utilizado para todas las organizaciones que intervienen en el desarrollo y en los servicios de Internet.

org Se utiliza para nombrar a las organizaciones no comerciales o sin fines de lucro.

int Asignada para organizaciones internacionales.

Existe otro TLD llamado *arpa*, que originalmente fue usado en la red ARPA durante la transición de la resolución por *hosts* a la resolución por tablas de DNS. Este dominio se explicará más adelante. En un inicio no se había contemplado el desarrollo y la penetración de Internet a nivel mundial, por lo que los TLDs fueron pensados sólo para uso dentro de los E.E.U.U., actualmente estos dominios son conocidos como gTLDs .

Para ajustarse a la internacionalización de Internet, se decidió agregar dominios que representen asignaciones geográficas, además de las afiliaciones organizacionales ya mencionadas. Por lo que se reservaron nuevos TLDs, los cuales corresponden a cada país del mundo, sumando un total de 243 países y se les conoce como ccTLD (*Country Code Top Level Domain*, Dominios de Primer Nivel por Código de País). Se dice que se reservaron por que no todos están en uso y dependen de las políticas de cada país. Las reglas utilizadas para cada nombre están descritas en el estándar de la ISO 3166, donde entre otras cosas se marca que se utilizarán dos letras para cada país. Algunos de ellos se muestran en la tabla 1.2 y la lista completa se puede revisar en la siguiente página web: <http://www.iana.org/cctld/cctld-whois.htm>.

PAÍS	NOMBRE DE DOMINIO
Australia	au
Colombia	co
España	es
Francia	fr
México	mx
Nigeria	ne
...	...

Tabla 1.2. Ejemplos de ccTDLs.

1.4.2. Delegación

Una de las características más importantes del DNS es su administración descentralizada, la cual se logra con la delegación de responsabilidades; con lo que cada organización es responsable de mantener y actualizar sus dominios, dándole total libertad de cambiar los datos que le pertenecen e inclusive de subdividir su dominio en uno o más subdominios, pudiendo delegar alguno de ellos a otra entidad. El dominio padre sólo tendrá apuntadores o referencias a las fuentes de los subdominios.

Un ejemplo muy claro de delegación es el que se aplica a los ccTLDs, ya que cada país es libre de utilizar los subdominios que considere pertinentes dentro del ccTLD asignado, por ejemplo en el caso de México, el NIC de nuestro país definió que bajo el dominio *mx* sólo se pueden registrar dominios bajo *com.mx*, *edu.mx*, *gob.mx* y *net.mx*. España es un caso más simple por que no tiene subdominios definidos (*com*, *edu*, etc.), es decir cada nuevo nombre estará bajo el dominio es.

1.4.3. Servidores y zonas

Los programas que almacenan datos acerca del espacio de nombres de dominio se llaman servidores de nombres. Los servidores de nombres generalmente tiene información completa sobre una parte del espacio de nombres de dominio, llamada zona, la cual leen de un archivo o de otro servidor de nombres. Al servidor de nombres que tiene asignada la administración de nombres del dominio o de una zona, se le llama autoridad para la zona; un servidor de nombres puede ser autoritativo para múltiples zonas.

La diferencia entre una zona y un dominio es importante pero sutil. Todos los TLDs tienen muchos dominios de segundo nivel e inferiores que son partidos en pequeñas unidades que son manejables para su delegación. Estas unidades son llamadas zonas. Por ejemplo en la figura 1.12 vemos el dominio *edu*, el cual está dividido en las zonas *berkeley.edu*, *purdue.edu* y *nwu.edu*. Es más fácil que se delegue la zona *berkeley.edu* a Berkeley y *purdue.edu* a Purdue.

Una zona contiene los nombres de dominio que pertenecen al dominio con el mismo nombre, excepto por los nombres delegados en subdominios. Si el subdominio de la zona no está delegado, la zona contiene los nombres de dominio y los datos del subdominio.

Para resumir podemos decir que un dominio puede contener más información de la que un servidor de nombres pudiera necesitar. Un dominio podría contener datos delegados a otro servidor de nombres, mientras que una zona está limitada por la delegación y nunca incluirá datos delegados.

Los servidores de DNS están divididos en dos tipos: primario o *masters* (maestros) y secundarios o *slaves* (esclavos). Los servidores primarios leen la información de una zona desde un archivo en el propio servidor. Un servidor secundario para una zona lee la información de un servidor primario, cuando un servidor secundario inicia su operación contacta a su servidor primario y si es necesario toma la zona, a este proceso se le llama transferencia de zona.

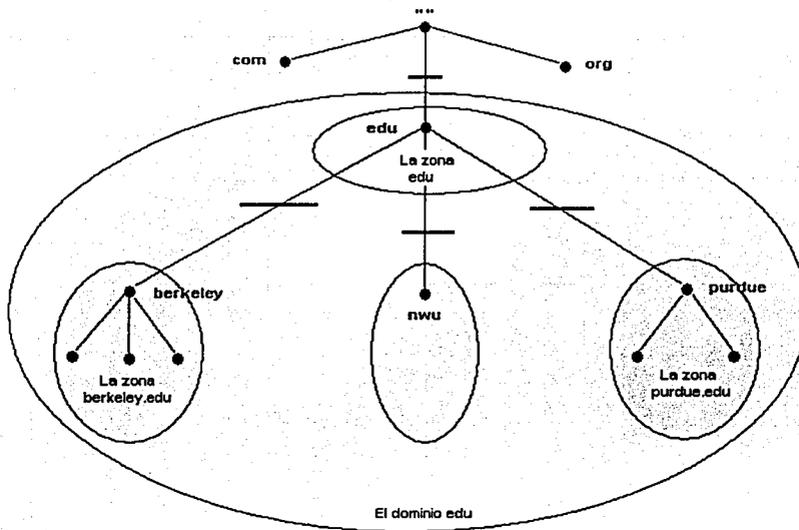


Figura 1.12. Diferencias entre zona y dominio.

Tanto el servidor primario como el secundario son autoritativos para la zona que administran. El DNS define estos dos tipos de servidores para hacer más fácil la administración, por que una vez que se actualiza información de una zona en particular, se transfiere información al o a los servidores secundarios sin necesidad de ir a actualizar la información en cada uno de ellos de forma manual.

Los servidores secundarios son importantes porque permiten tener más servidores que resuelvan información de la zona permitiendo balancear cargas y tener redundancia en el servicio.

1.4.4. Aplicaciones de resolución de nombres

Existen diversas herramientas que permiten dar el servicio de resolución de nombres, la mayoría están asociadas directamente con el sistema operativo del servidor de nombres. Entre ellas se encuentra DNS para Windows, que corre sobre servidores NT¹⁰, Lucent DNS que puede correr tanto en ambientes Unix como en ambientes NT, pero el más popular de todos es BIND que corre

¹⁰ NT es un sistema operativo para trabajo en grupo desarrollado por Microsoft y que corre sobre plataformas Intel.

únicamente sobre sistemas Unix y es el usado por los organismos internaciones que regulan y administran el DNS, como es el INTERNIC o el NIC de cada uno de los países.

BIND ha tenido una gran evolución desde que fue creada, con cada nueva familia de versiones se van introduciendo nuevas funcionalidades que mejoran la calidad del servicio, por ejemplo de la familia 4.9.X a la familia 8.X se introdujeron funcionalidades de seguridad, para cuidar la integridad de la información administrada. Actualmente la versión soportada y recomendada por el INTERNIC es la versión 8.2.5. La última familia de BIND es la 9.X y actualmente sólo cuenta con versiones de prueba o también llamadas beta, la funcionalidad más importante que se incluye en esta nueva familia es soportar IPV6¹¹.

Para fines del presente trabajo utilizaremos como aplicación a BIND en su familia de versiones 8.X.

1.4.5. Métodos de resolución

Los servidores de nombres son expertos en obtener datos del espacio de nombres de dominio. Ellos no sólo pueden dar información sobre los datos de la zona que son autoritativos, si no que también pueden buscar dentro del espacio de nombres de dominio para encontrar datos de los que ellos no son autoritativos. Este proceso se conoce como resolución de nombres o simplemente resolución.

Debido a que el espacio de nombres de dominio está estructurado como un árbol invertido, un servidor de nombres necesita saber sólo un dato para encontrar el camino a cualquier punto del árbol: la dirección IP de los servidores de root. Un servidor de nombres puede consultar a un servidor de root sobre cualquier dato dentro del espacio de nombres de dominio y el servidor de root indicará que camino debe seguir.

Los servidores de root conocen cuales son los servidores de nombres autoritativos para los TLD's. Si se le hace una consulta a un servidor de root, al menos provee el nombre y la dirección IP del servidor de dominio que es autoritativo para los TLD's. Y los servidores de nombres de los TLD's tienen la lista de los servidores de segundo nivel del dominio en que estén. En cada consulta del servidor de nombres se obtiene la mejor referencia de cómo acercarse a resolver la información buscada.

Los servidores de root son una pieza clave para la resolución de cualquier consulta hecha por los servidores de nombres. En el mundo existen trece servidores de root, que dada su importancia están distribuidos estratégicamente en varias partes del mundo, para tener la suficiente redundancia y diversidad de acceso para soportar las consultas de todo el mundo.

¹¹ IPV6 es un nuevo esquema de direccionamiento en donde se crecen de 4 a 8 octetos las direcciones IP, esto con el fin de tener suficientes direcciones para soportar el crecimiento mundial de equipos conectados al Internet.

La figura 1.13 muestra el proceso de resolución real para una dirección de un *host* en un dominio cualquiera, incluyendo el proceso de recorrer el árbol del espacio de nombres de dominio.

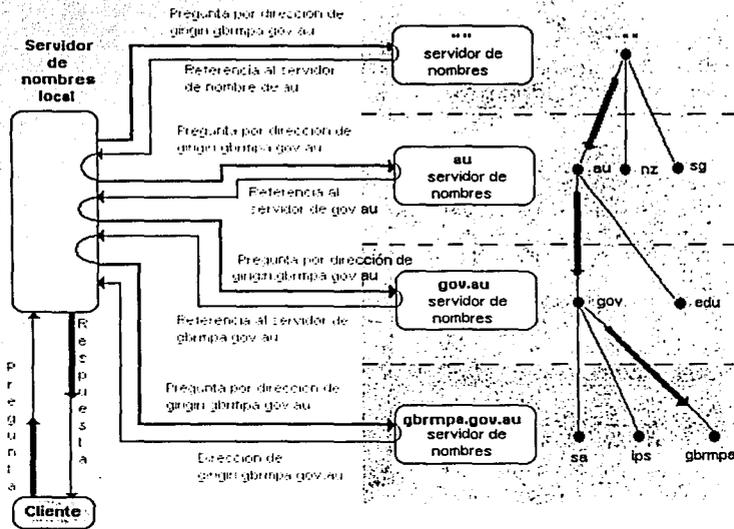


Figura 1.13. Proceso de resolución de nombres.

El servidor de nombres local consulta al servidor de root por una dirección de *girigiri.gbrmpa.gov.au* y éste es referenciado al servidor de nombres de *au*. El servidor de nombres local hace la misma pregunta pero ahora en el servidor de nombres de *au* y es referido al servidor de nombres de *gov.au*. El servidor de nombres de *gov.au* referirá al servidor de nombres local al servidor de *gbrmpa.gov.au*. Finalmente el servidor de nombres local pregunta a *gbrmpa.gov.au* por la dirección IP buscada obteniendo la información solicitada.

Dependiendo de la forma en que estén configurados los servidores de nombres pueden resolver las consultas, existen dos tipos de resoluciones las recursivas o iterativas y las no recursivas.

En una resolución recursiva el servidor de DNS es responsable de resolver el nombre sin importar a cuantos servidores pregunte en el proceso de resolución, por cada servidor al que le pregunte se obtiene la mejor referencia del nombre

buscado, para después preguntarle a esa referencia; este proceso se repite hasta resolver el nombre solicitado. Este método de resolución es el más utilizado y el que más consume recursos del servidor, pero sin él no se podría resolver ningún nombre.

En el caso de una resolución no iterativa el servidor de nombres está configurado para dar únicamente la mejor referencia que él conoce, es decir, no repite el proceso de búsqueda consultando a otros servidores. Esto es muy usado en los servidores de root y de dominios de segundo nivel, que no tienen más que referencias a otros dominios. Este proceso simplifica la carga de trabajo de los servidores ya que no tienen que descender a través del árbol del espacio de nombres de dominio.

Hasta este momento sólo se han tratado los casos en que un cliente desea conocer la dirección IP asociada a un nombre de dominio. Pero existen casos en que es necesario hacer la operación inversa, es decir, conocer el nombre de dominio que está asociado a una dirección IP específica. Este tipo de consultas se utilizan principalmente en archivos de bitácoras, en el correo electrónico y en validaciones básicas de Unix para acceder a ciertas aplicaciones del sistema. Cuando se usa resolución por archivos de *hosts* este proceso es transparente para la resolución de nombres, pero cuando se está trabajando con DNS tiene un tratamiento diferente.

Para las resoluciones inversas, fue utilizando el mismo principio de asociar etiquetas a los dominios en el espacio de nombres de dominio, pero con la diferencia de que para las resoluciones inversas las etiquetas contienen direcciones IP. En Internet la porción del espacio de nombres de dominio es el dominio *in-addr.arpa*.

Los nodos del dominio *in-addr.arpa* están etiquetados con el dominio *in-addr.arpa*, seguido de la dirección IP donde cada octeto se separa por un punto. El dominio *in-addr.arpa* tiene una división muy particular en la que los nodos hijos están etiquetados con un número entre 0 y 255, que corresponden a todos los posibles números del primer octeto de una dirección IP, a su vez cada uno de estos nodos tiene 256 nodos hijos igualmente numerados del 0 al 255 y así sucesivamente hasta cubrir los cuatro octetos que componen a una dirección IP. Con base a este esquema podemos decir que el árbol de las resoluciones inversas, en el espacio de nombres de dominio bajo el dominio de *in-addr.arpa*, tiene cuatro niveles hacia abajo, cada uno con 256 nodos, que se puede apreciar en la figura 1.14.

1.4.6. Cache

El proceso de resolución podría parecer muy complejo para los usuarios que solo han trabajado con archivos de *hosts* para resolver nombres, incluso podría parecer que el tiempo de respuesta del servicio se ve mermado por las constantes búsquedas que se requieren hacer para llegar al servidor de nombres indicado.

Pero existe una funcionalidad dentro de DNS llamada *cache*¹² que incrementa la velocidad de respuesta de un servidor de nombres.

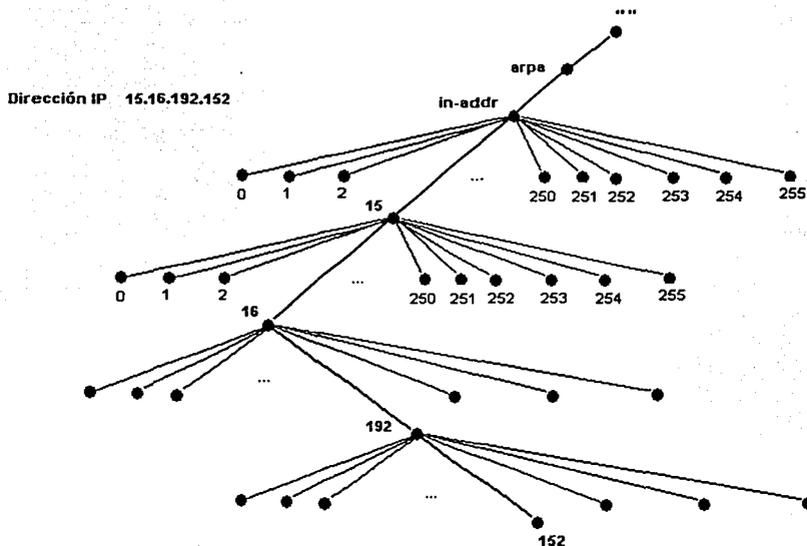


Figura 1.14. Estructura para resoluciones inversas.

Un servidor de nombres que resuelve de forma recursiva tiene que hacer varias consultas antes de encontrar al servidor indicado. Durante este proceso de búsqueda el servidor de nombres aprende mucha información sobre las zonas en el espacio de dominios, al final cuando resuelve la consulta puede almacenar la información aprendida para hacer uso de ella nuevamente en futuras consultas, de manera que al recibir una nueva consulta sobre el mismo nombre de dominio simplemente resuelve de la información almacenada en lugar de ir a buscar nuevamente en el espacio de nombres de dominio. Este proceso de almacenar datos se conoce como *cache*.

¹² Cache es una palabra en inglés que significa escondite, pero en el ambiente de DNS se utiliza para denotar almacenamiento temporal y no se utiliza alguna traducción para denotar esta funcionalidad.

Esta funcionalidad optimiza los recursos de los servidores de DNS y del ancho de banda de la red. Pero se corre el riesgo de mantener información almacenada que ya no sea vigente.

Para resolver el problema de la desactualización de información se introduce el concepto de TTL (Time To Live, o tiempo de vida). Este es un parámetro que se configura por zona en el cual se define el tiempo medido en segundos que cualquier servidor no autoritativo va a mantener en cache un dato de su zona, una vez que ese tiempo expira el dato se borra del cache y el servidor forzosamente tendrá que iniciar el proceso de búsqueda al recibir una nueva consulta sobre el mismo dato. El TTL es configurado por el administrador de cada zona y el valor asignado depende mucho de la frecuencia con que se cambien los datos, por ejemplo si se hacen pocos cambios en una zona en un mes convendría tener un TTL equivalente a 30 días.

1.4.7. Herramientas de diagnóstico

Como el DNS es sólo un programa que resuelve nombres, no interactúa directamente con el usuario; son las aplicaciones como el correo, un TELNET, FTP o un navegador quienes interactúan con el DNS. Es por eso que en caso de falla en el DNS solo recibimos un mensaje de error de la aplicación que estemos usando, que nos indica que no fue posible encontrar el nombre de dominio buscado, haciendo difícil el proceso de resolución de problemas.

Para facilitar el diagnóstico de problemas en el DNS existen algunos comandos de UNIX, que incluso se han implementado en NT, con los que se puede obtener la suficiente información para detectar donde hay problemas de resolución de nombres, y poder tomar acciones para corregir la falla en cuestión.

El primer comando es *host*, con el cual se pueden hacer consultas simples en línea a un servidor de nombres sobre un recurso específico, pero tiene la desventaja de no regresar mucha información.

Uno de los comandos más utilizados es *nslookup* (*Name Sever Look Up*, búsqueda en servidores de nombres). Este comando permite hacer consultas tal como las haría un servidor de nombres o un cliente normal, y puede emular cualquier opción de las manejadas en BIND, dependiendo del problema que se quiera resolver. Este comando permite hacer consultas a múltiples servidores de nombres pudiendo cambiar en cualquier momento de servidor, se le pueden configurar tiempos límite de resolución para detectar lentitud en las respuestas, puede hacer búsquedas de dominio completos o por subdominios, puede simular ser un servidor secundario para probar la transferencia de una zona desde un servidor primario, es capaz de resolver consultas usando NIS y */etc/hosts*; y tiene opción para correr interactivamente con el usuario.

Otro comando que ayuda a detectar problemas es *dig*, el cual hace prácticamente lo mismo que el comando *nslookup* pero presenta la información de forma diferente y no permite interactuar directamente con el cliente.

1.5. BIND

BIND es el software más usado para la configuración y administración de los servidores de DNS. Este programa funciona solamente en ambientes Unix.

Los componentes de BIND se pueden clasificar en dos grupos: los programas ejecutables y los archivos de configuración. Los programas ejecutables son básicamente dos, el primero llamado *named* que es responsable de contestar a todas las consultas que recibe el servidor y el segundo programa es *named-xfer* que se encarga de hacer todas las transferencias de zona. Los archivos de configuración se detallan a continuación.

1.5.1. Archivos de BIND

Para el correcto funcionamiento de BIND deben existir al menos tres archivos de configuración:

- *named.conf*
- *db.127.0.0*
- *db.cache*

El archivo *named.conf* es la base de configuración de BIND, ya que en él se especifican las características de todos los dominios administrados por el servidor, tanto primarios (*master*) como secundarios (*slave*), además de las opciones de configuración del servidor. Por ejemplo, en qué directorio se encuentran los archivos de configuración. A continuación se presenta un archivo típico de *named.conf*.

```
/*
 * BIND 8.X Name server configuration file
 */

options {
    directory "/qipsw/named.data";
};

zone "." in {
    type hint;
    file "db.cache";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "db.127.0.0";
};

zone "13.in-addr.arpa" in {
```

```

type master;
file "db.13";
check-names warn;
allow-update { any; };
allow-query { any; };
allow-transfer { any; };
notify no;
};

zone "mex" in {
type master;
file "db.mex";
check-names warn;
allow-update { any; };
allow-query { any; };
allow-transfer { 200.33.150.103;13.141.4.71; };
notify no;
};

zone "compusat.net.mx" in {
type slave;
file "db.compusat.net.mx";
masters { 200.38.206.66; };
check-names warn;
allow-update { none; };
allow-query { any; };
allow-transfer { none; };
notify no;
};

```

En este archivo los dominios de la red 13 y el dominio *mex* son primarios para este servidor, por lo tanto tiene un tipo *master*; mientras que el dominio *compusat.net.mx* es un dominio secundario, y por lo tanto su tipo es *slave* e indica la dirección de su servidor primario desde donde se realizará la transferencia de zona.

El archivo *db.127.0.0* es un archivo necesario para el servidor para cubrir la dirección especial 127.0.0.1 que utilizan los *hosts* para direccionar su propio tráfico. Este archivo se utiliza para relacionar la dirección 127.0.0.1 con el nombre del servidor. A continuación se muestra un ejemplo de este archivo.

```

;=====
; Local server zone information (db.127.0.0)
;=====
@      IN      SOA      hunics02.uninet.net.mx.
root.hunics02.uninet.net.mx. (
                                1          ; Serial No.
                                21600       ; Refresh
                                3600        ; Retry
                                6666666    ; Expire
                                86400 )    ; Minimum
                                IN         NS      hunics02.uninet.net.mx.
1      IN      PTR     localhost.

```

Los parámetros *Serial No.*, *Refresh*, *Retry*, *Expire* y *Minimum* se explicarán más adelante.

El archivo *db.cache* es otro archivo necesario para el servidor, por que contiene las relaciones de los nombres de los servidores de *root* con sus direcciones IP, y se utiliza para realizar peticiones a estos servidores. A continuación se muestra un ejemplo de este archivo.

```

;=====
; Hints file that points to the root servers (db.cache)
;=====
; *****
; Db.cache extension
; *****
;
;      @(#)root.cache 1.15      (Berkeley)      89/09/18
;
; Initial cache data for root domain servers.
;
;
;          99999999          IN          NS          NS.NIC.MX.
;          99999999          IN          NS          NS.INTERNIC.NET.
;          99999999          IN          NS          NS.NIC.DDN.MIL.
;          99999999          IN          NS
KAVA.NISC.SRI.COM.
;          99999999          IN          NS          NIC.NORDU.NET.
;          99999999          IN          NS          NS.NASA.GOV.
;          99999999          IN          NS          TERP.UMD.EDU.
;          99999999          IN          NS          A.ISI.EDU.
;          99999999          IN          NS          AOS.BRL.MIL.
;          99999999          IN          NS          GUNTER-
ADAM.AF.MIL.
;          99999999          IN          NS          C.NYSER.NET.

```

Además de estos tres archivos, existe uno por cada zona que administré el servidor. Dependiendo de sus funciones, pueden relacionar los nombres de *hosts* con las direcciones IP (resolución normal) o relacionan las direcciones IP con los nombres de *hosts* (resolución inversa).

En los archivos de relación *hosts-direcciones IP*, existe uno por cada zona administrado en el servidor. Es importante mencionar que estos archivos son los que se transfieren entre los servidores primarios y secundarios durante el proceso de transferencias de zona. A continuación se muestra un ejemplo de archivo de zona para una resolución normal.

```

;=====
; Addresses and other host information for zone: mex
;=====
$TTL 3600
@           IN          SOA          hunics02.uninet.net.mx. root.hunics01.mex. (
;          38           ; Serial No.
;          21600        ; Refresh
;          3600         ; Retry

```

```
604800 ; Expire
86400 ) ; Minimum
```

```
IN      NS      hunics02.uninet.net.mx.
IN      NS      hmexmc01.mex.mex.
IN      NS      hunics03.uninet.net.mx.
```

```
;*****
; A records
;*****
```

```
nsmtly2      IN      A      13.52.152.152
nsmtly1      IN      A      13.52.152.149
nsmex4       IN      A      13.52.152.67
nsmex3       IN      A      13.52.152.137
nsmex2       IN      A      13.52.152.87
nsmex1       IN      A      13.52.152.79
nsgd12       IN      A      13.52.152.83
nsgd11       IN      A      13.52.152.19
dnsuni2      IN      A      13.52.152.119
dnsuni1      IN      A      13.52.152.55
```

En el caso de los archivos de relación *direcciones IP-hosts*, existe un archivo por red y por su subred donde se agrupan las direcciones IP de cada una para relacionarlas con sus nombres, estos archivos también se transfieren entre los servidores primarios y secundarios. A continuación se muestra un ejemplo típico de estos archivos.

```
=====; Reverse
Addresses (PTR Records) for zone: 13.in-addr.arpa
=====@      IN
$TTL 3600
SOA      hunics02.uninet.net.mx. root.hunics02.uninet.net.mx. (
        38      ; Serial No.
        21600   ; Refresh
        3600    ; Retry
        604800 ; Expire
        86400 ) ; Minimum
```

```
IN      NS      hunics02.uninet.net.mx.
IN      NS      hmexmc01.mex.mex.
IN      NS      hunics03.uninet.net.mx.
```

```
;*****
; PTR records
;*****
```

```
19.152.52   IN      PTR      nsmtly2.mex.
55.152.52   IN      PTR      nsmtly1.mex.
67.152.52   IN      PTR      nsmex4.mex.
79.152.52   IN      PTR      nsmex1.mex.
83.152.52   IN      PTR      nsgd12.mex.
87.152.52   IN      PTR      nsmex2.mex.
119.152.52  IN      PTR      dnsuni2.mex.
137.152.52  IN      PTR      nsmex3.mex.
149.152.52  IN      PTR      nsmtly1.mex.
152.152.52  IN      PTR      nsmtly2.mex.
```

Cada uno de estos archivos cuenta con dos secciones: la primera es el registro SOA (*Start Of Authority*, Inicio de autoridad), el cual indica autoridad para la zona; La segunda sección es para los tipos de registros que se emplean en los archivos de zonas para relacionar direcciones IP con *hosts* y viceversa.

A continuación se mencionan los parámetros de configuración para el registro SOA:

- *TTL*. Intervalo de tiempo asociado a cada uno de los registros de los archivos por omisión. Cuando un servidor responde una petición, envía en la respuesta el *TTL* para indicarle al servidor que pregunta, por cuanto tiempo debe guardar en *cache* este registro.
- *Serial*. Número de versión del archivo. Debe ser incrementado cada vez que se hace un cambio al archivo, de lo contrario el servidor secundario (*slave*) no se actualizará.
- *Refresh*. Intervalo de tiempo, contado desde la última vez que se hizo la actualización de las tablas del secundario. Al final del cual el servidor secundario debe copiar los archivos del servidor de DNS primario.
- *Retry*. Tiempo que el servidor secundario debe esperar para reiniciar la actualización de tablas del servidor primario en caso de que falle la conexión al hacer el *refresh*.
- *Expire*. Tiempo después del cual, sino se ha logrado hacer el *refresh*, se desecha el archivo; el servidor secundario deja de responder peticiones sobre el nombre del dominio al que se refiere el archivo.

La segunda sección contiene los siguientes tipos de registro:

- NS. Define un servidor de DNS para un dominio.
- A. Define una dirección para un nombre de un *host*.
- CNAME. Define un alias para un *host*.
- HINFO. Define el tipo de CPU y sistema operativo de un *hosts*.
- TXT. Define información de un *hosts*.
- MX, Define un servidor de correo electrónico para un dominio y un nivel de preferencia.

CAPÍTULO II

Análisis de la propuesta

El análisis que se presenta en este capítulo muestra el estado de los servidores de DNS, los problemas de operación y sus causas. Los puntos que se desarrollaran son: determinación de requerimientos, cuantificación de tráfico, desempeño de los equipos, distribución de cargas para los servidores de acceso y de DNS, inventarios de zonas y análisis de tipos y versiones de software y hardware.

2.1. Determinación de Requerimientos

Se revisará la situación actual, las causas de las fallas encontradas y las actividades a realizar para mejorar el desempeño del servicio de DNS.

2.1.1. Situación actual

El servicio de DNS para Uninet está operando con equipos que tienen cuatro años de antigüedad, aunado a problemas de administración, configuración y operación que causan un deficiente servicio.

Para ubicar la dimensión de los conflictos que guarda el servicio de DNS en Uninet, es importante conocer primero los nombres de los servidores en operación y la ciudad donde se encuentran. Esto se representa en la tabla 2.1.

SERVIDOR	UBICACIÓN
dns	Cd. México
nsmex1	Cd. México
nsmex2	Cd. México
nsmex3	Cd. México
nsgdl1	Cd. Guadalajara
nsmtly1	Cd. Monterrey

Tabla 2.1. Servidores de DNS.

Estos seis servidores han presentado problemas, por falta de una correcta administración y mantenimiento. Debido a esto, se tienen errores de resolución por no seguir la sintaxis y el orden correcto del archivo */etc/named.conf*, además de saturación en las capacidades para resolver peticiones de clientes, provocando que no se resuelvan los dominios que se encuentran configurados.

Es necesario decir, que el servidor dns es el que tiene más carga de trabajo, ya que éste es el servidor primario de los tres dominios más importantes: para las resoluciones inversas de todas las clases (las cuales se utilizan para la asignación dinámica de direcciones IP) y como secundario para los clientes corporativos.

Hoy en día el servidor dns tiene registrados archivos de más de 300 dominios, de los cuales algunos ya no están operando, lo que provoca que se estén utilizando recursos de los servidores en procesos innecesarios. Por otra parte dns no está llevando a cabo adecuadamente las transferencias de zona a los servidores secundarios.

En dicha operación de transferencia, se configuran dos o más dominios por día, además de configuraciones de resoluciones inversas para clientes; por este motivo se reinicia el proceso de BIND para que se actualicen los cambios realizados, pero esto implica que durante el tiempo que no esté disponible este proceso los usuarios de Internet no pueden navegar y el servicio de correo electrónico se retrasa.

Existen problemas similares con los otros servidores que tienen funciones de servidores secundarios de los dominios principales y de las resoluciones inversas que se detallan adelante.

2.1.2. Causas de falla en el servicio de DNS

A continuación se mencionan puntos de falla que se han detectado en administración, configuración y operación en el servicio de DNS.

- Mala administración. Debido a que los diferentes administradores no tenían control de los servidores y no hicieron una planificación del crecimiento de dominios, esto ocasionó que los inventarios no fueran confiables y se tuvieran dominios obsoletos de clientes que ya no tienen contrato.
- Errores de sintaxis en la configuración. En el archivo *named.conf* y en los otros archivos de configuración de resoluciones, la sintaxis es muy sensible a las diferencias entre minúsculas y mayúsculas, así como de los espacios de los tabuladores; por lo que si BIND detecta errores, no carga los datos y no resuelve adecuadamente.
- La capacidad insuficiente de los servidores. Los servidores ya no tienen espacio suficiente en disco duro para almacenar los datos de DNS y las bitácoras del sistema, además de tener insuficiente memoria y sobrecarga de CPU.
- Falta de monitoreo. Debido a la versión del sistema operativo y a las características de los servidores no se puede monitorear el servicio de DNS ni el servidor, ya que las herramientas de monitoreo necesitan al menos la versión 2.7 del sistema operativo Solaris y los servidores tienen la versión 2.5 y no pueden ser migrados de versión por la capacidad del hardware.

2.2. Desempeño de los enlaces y servidores

El objetivo de este análisis, es detectar los puntos críticos a nivel de tráfico y de capacidad en los enlaces y servidores que afectan directamente al servicio de DNS de Uninet.

Cuando se menciona el tráfico de usuarios y el desempeño de los servidores, hablamos de cantidades importantes de usuarios navegando en Internet, enviando y recibiendo correo electrónico (especialmente largas listas de correo) y programas que realizan diferentes peticiones e intercambios entre *hosts*, que consecuentemente exigen mayores recursos de memoria. Por esta razón, la capacidad de utilización de los enlaces, la proporción de memoria ocupada y el desempeño de la CPU (*Central Process Unit*, Unidad Central de Procesos) son probablemente las estadísticas más importantes en el monitoreo de operación de un servidor de DNS.

De forma particular y como anteriormente se menciona, existen 6 servidores de DNS en Uninet en tres ciudades diferentes; cuya problemática se desglosará: primero, en relación a la ocupación de sus enlaces, posteriormente en su desempeño de memoria y CPU, para finalizar analizando el número de peticiones recibidas (*Queries*) por servidor en un periodo determinado. Todo esto con el fin de encontrar una solución precisa a la problemática del servicio de DNS.

2.2.1. Ocupación de los enlaces

a) Ciudad de México

En la revisión de la conectividad de los servidores a la red, se encontró que los cuatro ubicados en la Ciudad de México (dns, nsmex1, nsmex2 y nsmex3) están compartiendo cuatro enlaces E1¹ con otros 25 servidores de aplicaciones ajenas a DNS, situación que los obliga a competir por el ancho de banda, ocasionando serios conflictos al tiempo de respuesta. La figura 2.1 muestra la topología bajo la cual se encuentran conectados los servidores.

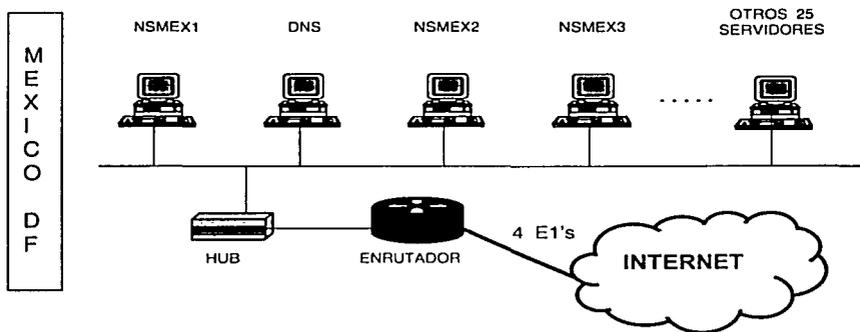


Figura 2.1. Topología de la red de servidores de DNS de la Ciudad de México.

Para poder analizar el comportamiento de los enlaces de la Ciudad de México, se utilizó un programa de *Lucent Technologies*, conocido como *Vital Suite* (Ver Anexo A), el cual hace mediciones de desempeño de elementos de red, como son los enlaces, a los cuales monitorea cada 3 minutos para representarlos gráficamente. Dichas mediciones se hacen a través de poleos SNMP (*Simple Network*

¹ Enlace de comunicación cuya Ancho de Banda es de 2.048 Mbs.

Management Protocol, Protocolo Simple de Administración de Red), utilizando las MIB's (*Management Information Base*, Base de Información de Administración) de los enrutadores, con las que se obtiene el porcentaje de utilización de cada enlace, su picos de utilización, porcentaje de errores y de descartes.

Las mediciones (figuras 2.2, 2.3, 2.4 y 2.5) se realizaron durante la semana del 6 al 12 de enero de 2002, y representan una semana típica del tráfico generado por los 29 servidores.

Después de analizar las figuras, se encontraron los siguientes puntos críticos de los enlaces:

Enlace uno

- Presenta periodos pico de ocupación mayores al 90%, con duración de 12 hrs.
- Se Identifican niveles menores al 50% de ocupación en el fin de semana.
- Existe un pico de descartes de 0.5% por una hora.

Enlace dos

- Tiene el promedio de utilización más alto de los cuatro enlaces en días hábiles (por encima del 70%).
- Se observan picos de utilización superiores al 90% en periodos de 16 hrs.
- Destaca un periodo de descartes de paquetes de hasta el 40% del tráfico de usuarios.

Enlace tres

- Como en los anteriores enlaces, existen periodos pico de utilización altos (arriba del 60%) en periodos de hasta 24 hrs., aunque cuenta con un reducido porcentaje de descartes (aprox. 0.005%).

Enlace cuatro

- En este enlace existe un periodo de utilización de más de 48 hrs. al comienzo de la semana, con un pico y promedio de utilización del 100%.
- Existen periodos de descartes de 24 hrs. superiores al 9%.

E1 tk mce-reduno-1(S0/0/1) - vallejo-5(S6/1/0)

Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

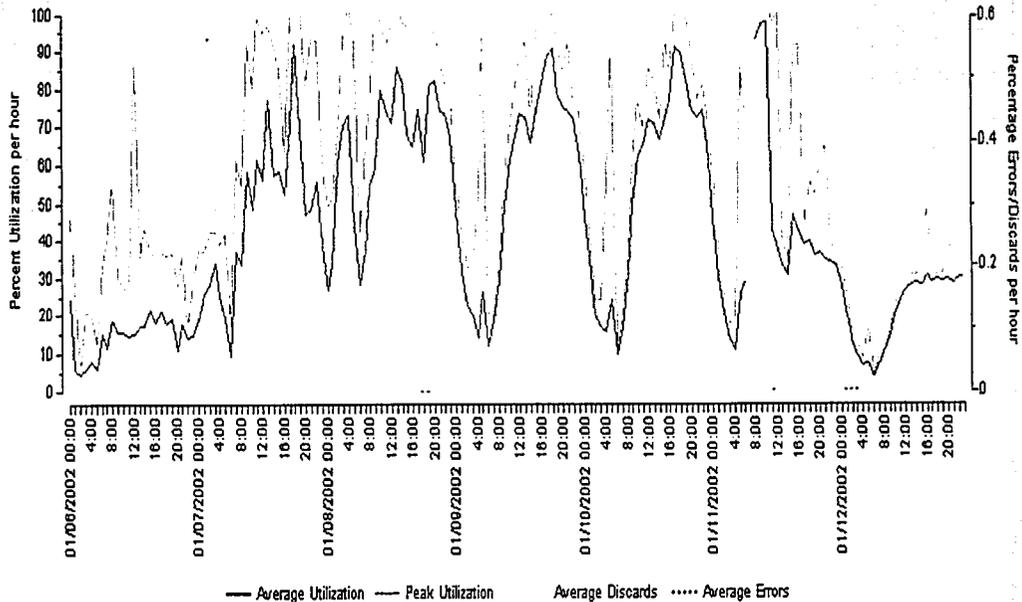


Figura 2.2. Utilización del enlace uno de la Ciudad de México.

E1 tk mce-reduno-1(S1/0/0) - nextengo-2(S0/4)
Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

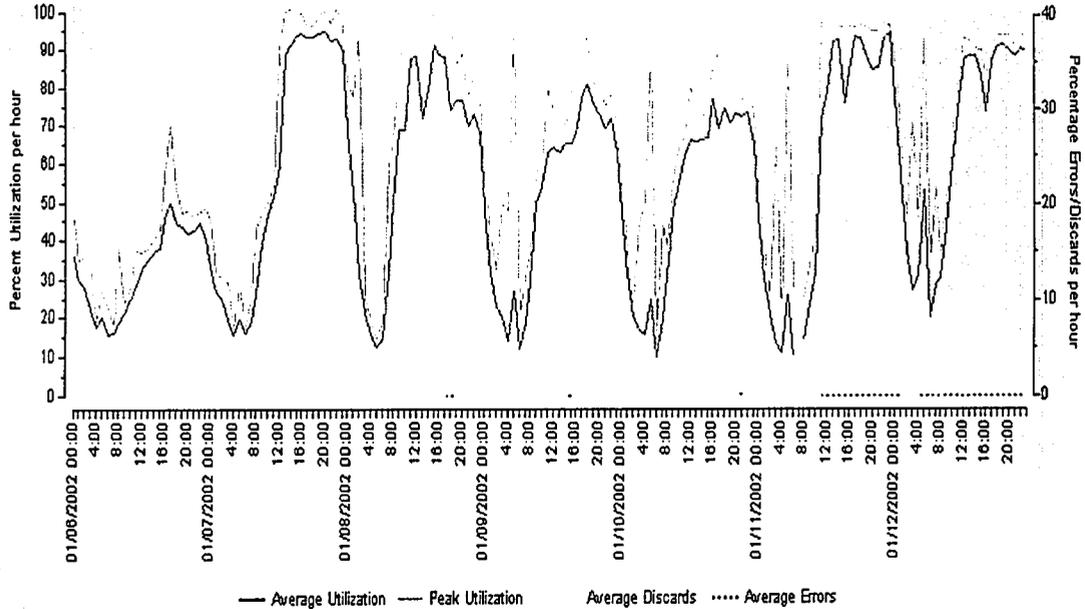


Figura 2.3. Utilización del enlace dos de la Ciudad de México.

E1 tk mce-reduno-1(S1/0/2) - TECKM_PB_IDP (S2/0:0)

Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

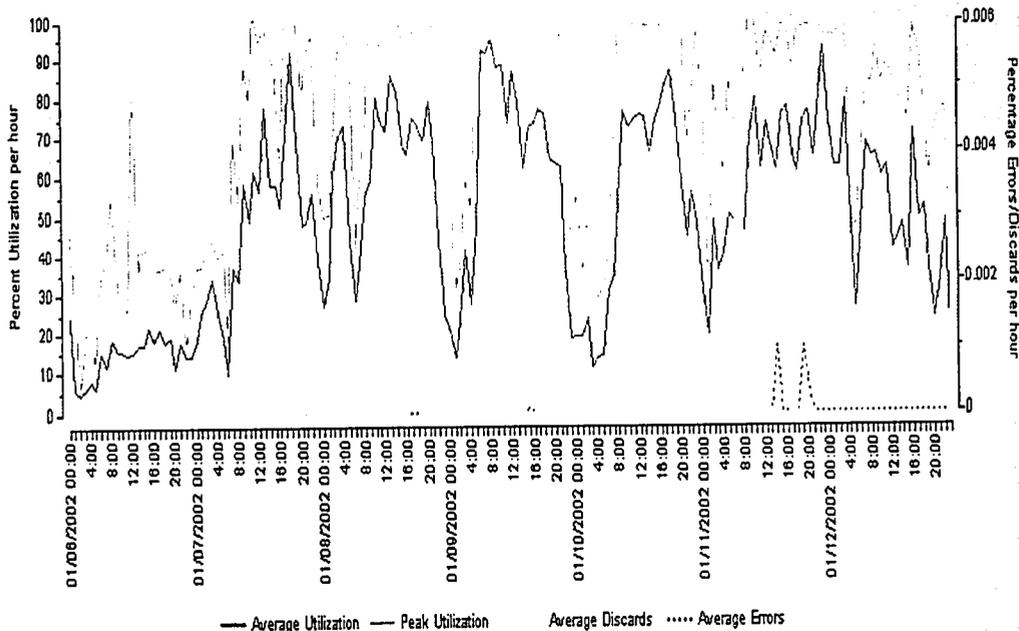


Figura 2.4. Utilización del enlace tres de la Ciudad de México.

mce-mex-reduno-2_S1000_6_ENLACE NEXTENGO3 0002

Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

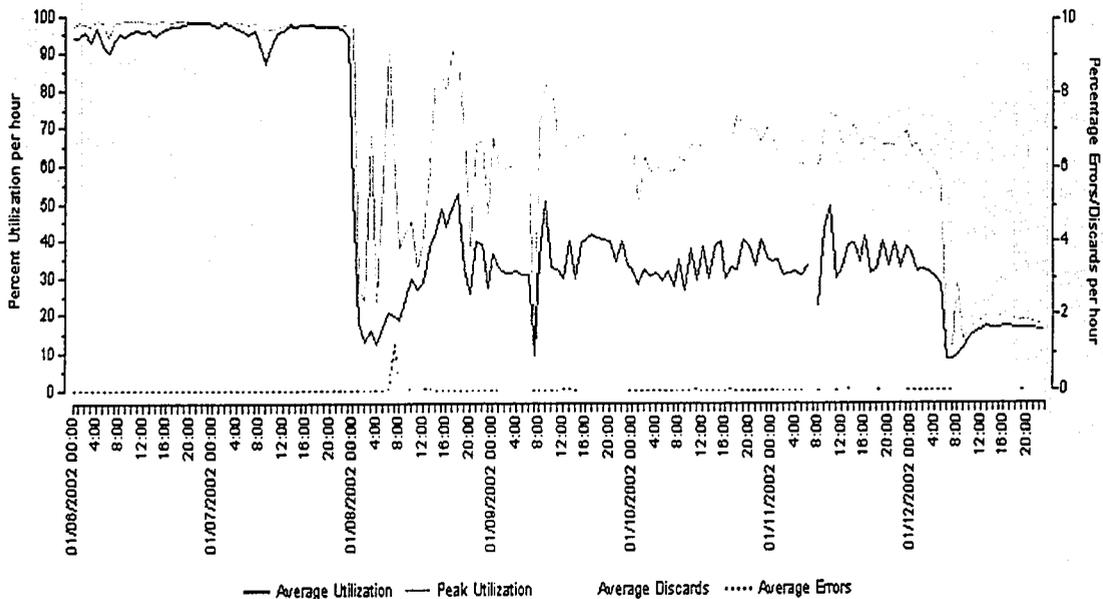


Figura 2.5. Utilización del enlace cuatro de la Ciudad de México.

Adicional a las graficas, se muestra la tabla 2.2 que tiene como objetivo resumir el comportamiento de los cuatro enlaces durante la semana del 6 al 12 de enero. Esto indica que tres de los enlaces tienen promedios de utilización muy altos y en caso de falla en alguno, los otros enlaces no soportarían el tráfico demandado afectando el servicio. También es importante resaltar el promedio de descartes, por la pérdida de información que este genera, degradando el servicio.

Enlace	Promedio de utilización	Pico máximo de utilización	Promedio de errores	Promedio de descartes
Uno	51.9%	100%	≈0%	0%
Dos	71.3%	100%	≈0%	3%
Tres	62.2%	100%	≈0%	0%
Cuatro	35.1%	98%	≈0%	0.8%

Tabla 2.2. Resumen de enlaces de la Ciudad de México.

**TEJIS CON
FALLA DE ORIGEN**

b) Ciudad de Guadalajara

Para el caso del servidor de la Ciudad de Guadalajara, se tienen tres enlaces E1 dedicados al servicio de DNS. En la figura 2.6 se representa su topología de conexión.

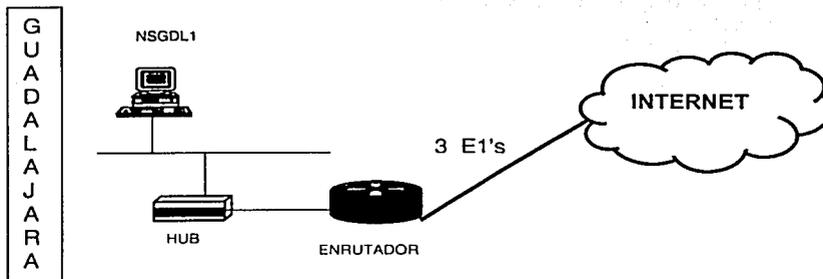


Figura 2.6. Topología de la red de servidores de DNS de la Ciudad de Guadalajara.

Las gráficas de ocupación por enlace se muestran en las figuras 2.7, 2.8 y 2.9 y la descripción de su comportamiento se da a continuación.

Enlace uno

- La ocupación es constante durante toda la semana.
- No hay picos superiores al 50% en ningún día.
- Los descartes y errores son casi nulos, se presentó un pico de 0.007%.

Enlace dos

- La ocupación es constante durante toda la semana.
- No hay picos superiores al 70% en ningún día.
- Los descartes y errores son casi nulos, pico máximo de 0.035%.

Enlace tres

- La ocupación es constante.
- Los picos de utilización no rebasan el 80%.
- No hay descartes y errores.

STM1 tk bup-jal-ctg-1(P4/0) - bup-tlaquepaque-1(P4/0)
 Detail, WAN MIB II Statistics

01/05/2002 00:00-01/12/2002 23:00

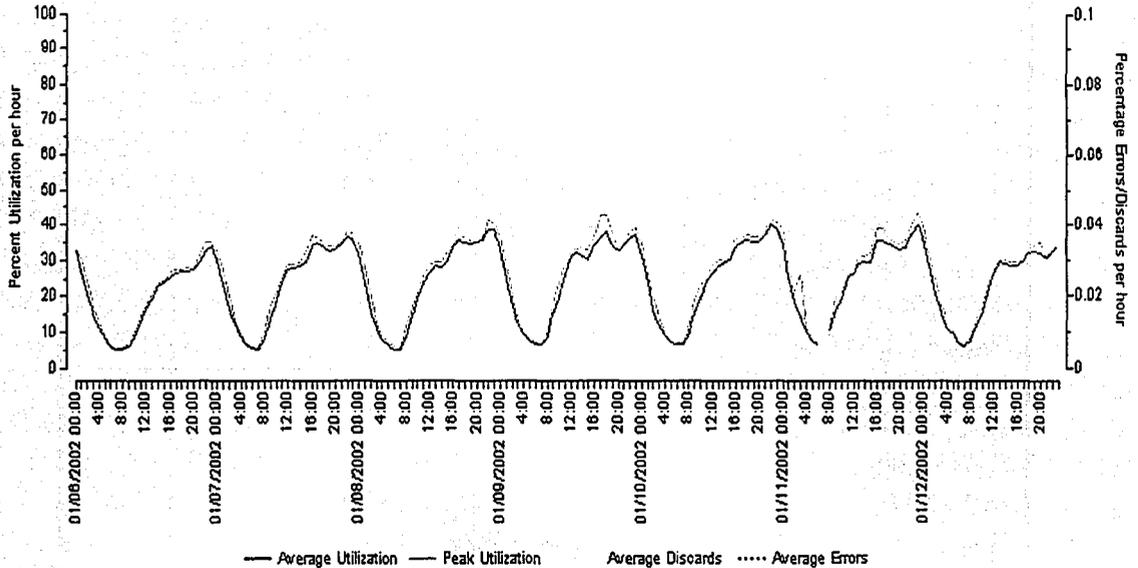


Figura 2.7. Utilización del enlace uno de la Ciudad de Guadalajara.

STM1 tk bb-tlaquepaque-1(P4/0) - bb-revolucion-1(P4/1)

Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

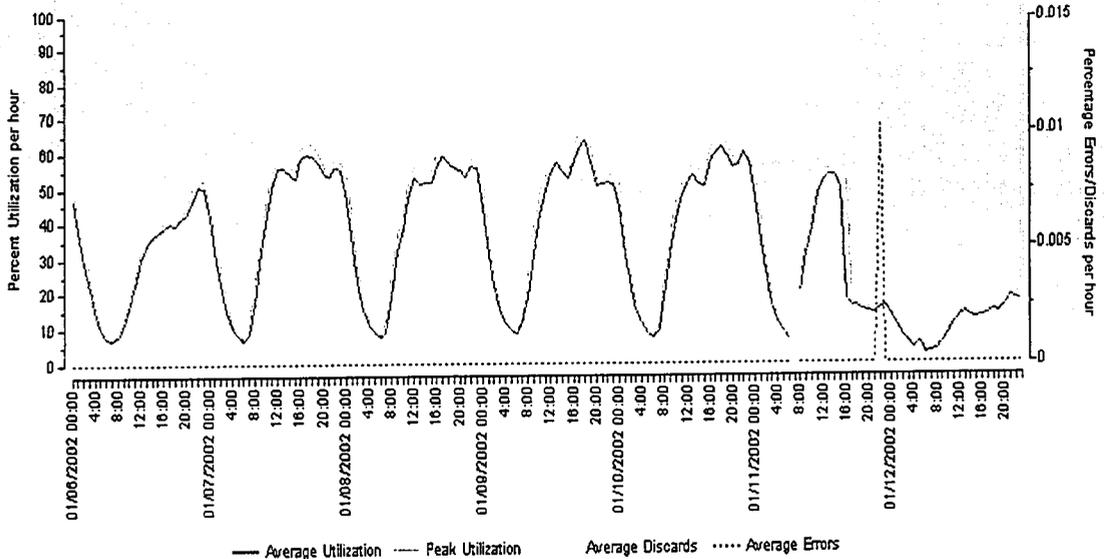


Figura 2.8. Utilización del enlace dos de la Ciudad de Guadalajara.

STM1 tk bb-tlaquepaque-1(P4/1) - bb-vallejo-1(P4/1)
 Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

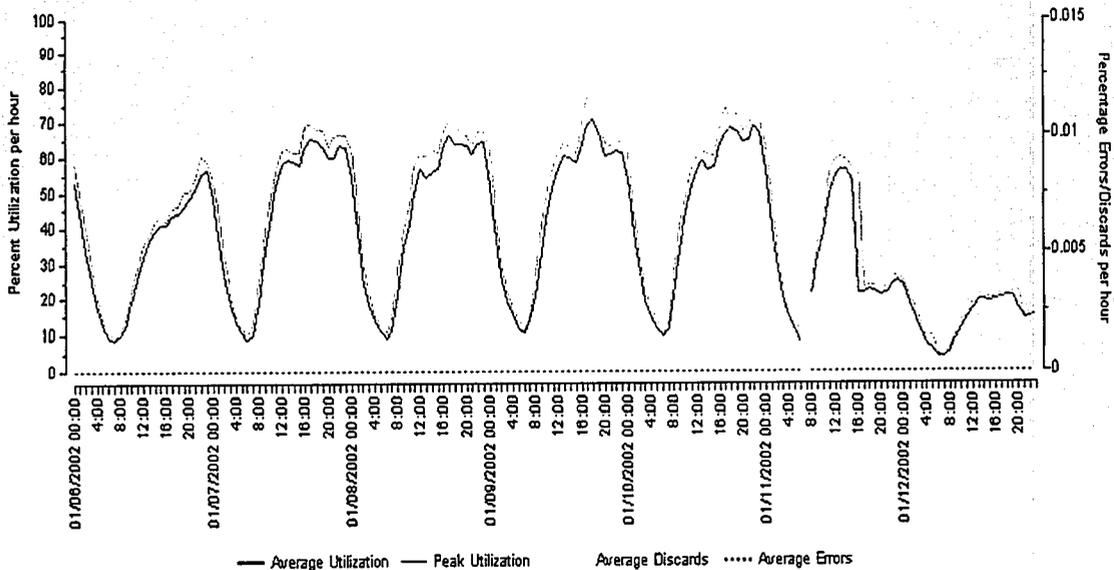


Figura 2.9. Utilización del enlace tres de la Ciudad de Guadalajara

Para complementar la información de las gráficas anteriores se presenta la tabla 2.3 que resume el comportamiento de los tres enlaces durante la semana del 6 al 12 de enero en la Ciudad de Guadalajara. En la tabla se puede observar que el promedio de descartes y errores es nulo, también es importante mencionar que los promedios de utilización y los picos máximos no afectan al servicio.

Enlace	Promedio de utilización	Pico máximo de utilización	Promedio de errores	Promedio de descartes
Uno	23%	47.3%	0%	0%
Dos	46.4%	67.5%	0%	0%
Tres	51.7%	77.2%	0%	0%

Tabla 2.3. Resumen de los enlaces de la Ciudad de Guadalajara.

En términos generales, los tres enlaces se encuentran en buen estado. Cabe mencionar que a diferencia de los 4 enlaces compartidos para los 29 servidores de la Ciudad de México (entre ellos 4 de DNS), los 3 enlaces de Guadalajara son para uso exclusivo del servicio de DNS. Por lo anterior, no se observan problemas de saturación, ni de descartes y errores.

Además, en caso de falla en alguno de los enlaces, su tráfico puede ser absorbido por los otros dos, sin afectar el servicio. Este tipo de redundancia garantiza que los enlaces balanceen su capacidad y mantengan sus porcentajes de operación en límites de correcto funcionamiento.

c) Ciudad de Monterrey

Para el servidor de Monterrey también existen tres enlaces dedicados para el servicio de DNS, como se muestra en la figura 2.10.

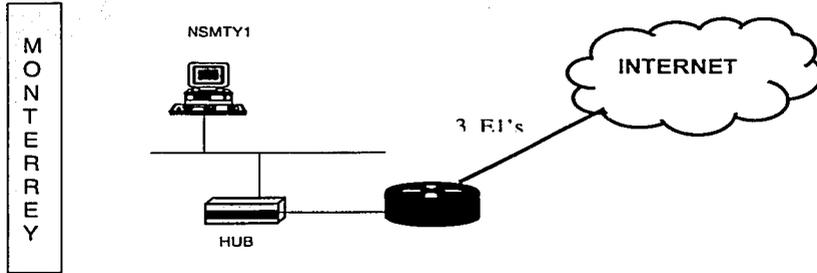


Figura 2.10. Topología de la red de servidores de DNS de la Ciudad de Monterrey.

Las gráficas de ocupación por enlace se muestran en las figuras 2.11, 2.12 y 2.13 y la descripción de su comportamiento se da a continuación.

Enlace uno

- La ocupación es constante durante toda la semana.
- No hay picos superiores al 35% en ningún día.
- Los descartes y errores son casi nulos, pico mínimo de 0.007%.

Enlace dos

- La ocupación es constante durante toda la semana.
- No hay picos superiores al 90% en ningún día.
- Los descartes y errores son casi nulos, pico máximo de 0.035%.

Enlace tres

- La ocupación es constante.
- Los picos de utilización no rebasan el 80%.
- No hay descartes y errores.

inet-nvl-mayo-3_S400_10_CONEXION A inet-dgo-zarco-1-PTO-6000 D34-0008-0027 JI

Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

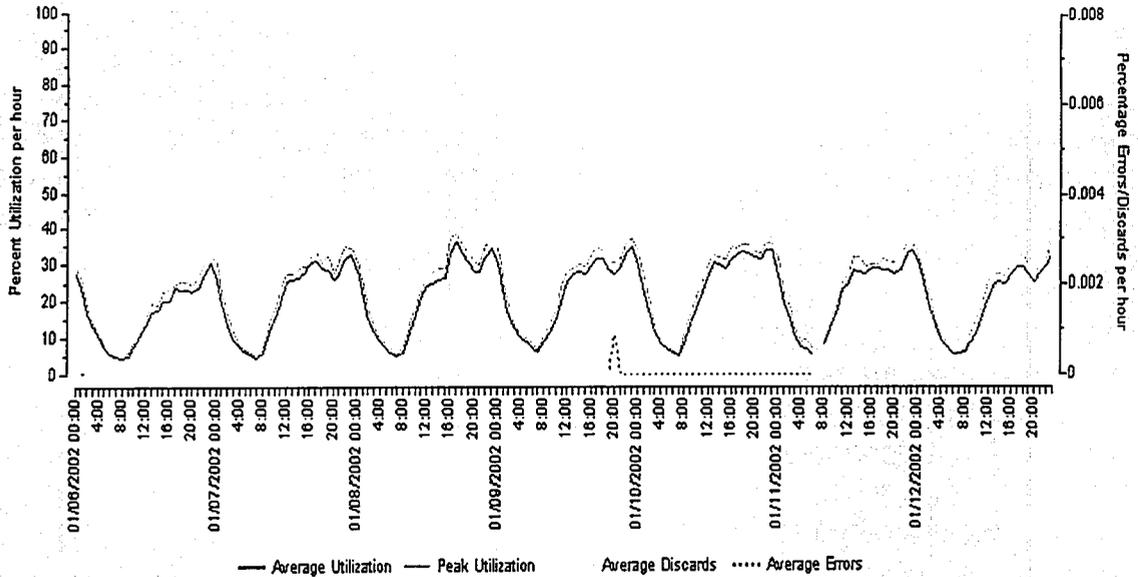


Figura 2.11. Utilización del enlace uno de la Ciudad de Monterrey.

inet-nvl-mayo-3_S4/0/1_11_ENLACE E3 A inet-coa-ayuntamiento-1 S6/0/0 D34-0010-00
Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

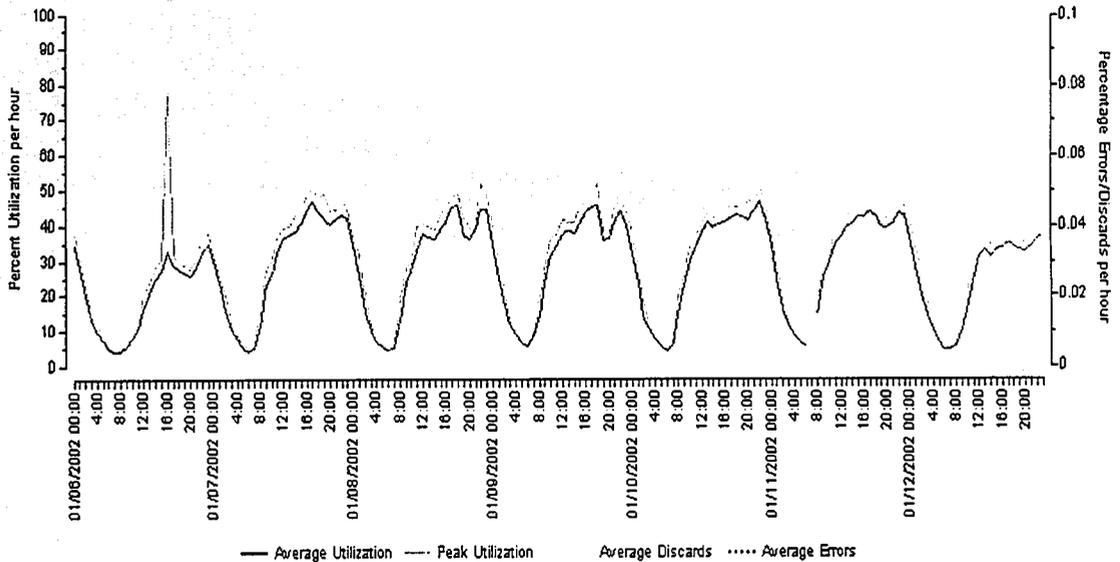


Figura 2.12. Utilización del enlace dos de la Ciudad de Monterrey.

inet-nvl-mayo-3_S4/1/0_12_ENLACE E3 A inet-chi-catedral-1 S6/0/0 D34-0011-0023,
 Detail, WAN MIB II Statistics

01/06/2002 00:00-01/12/2002 23:00

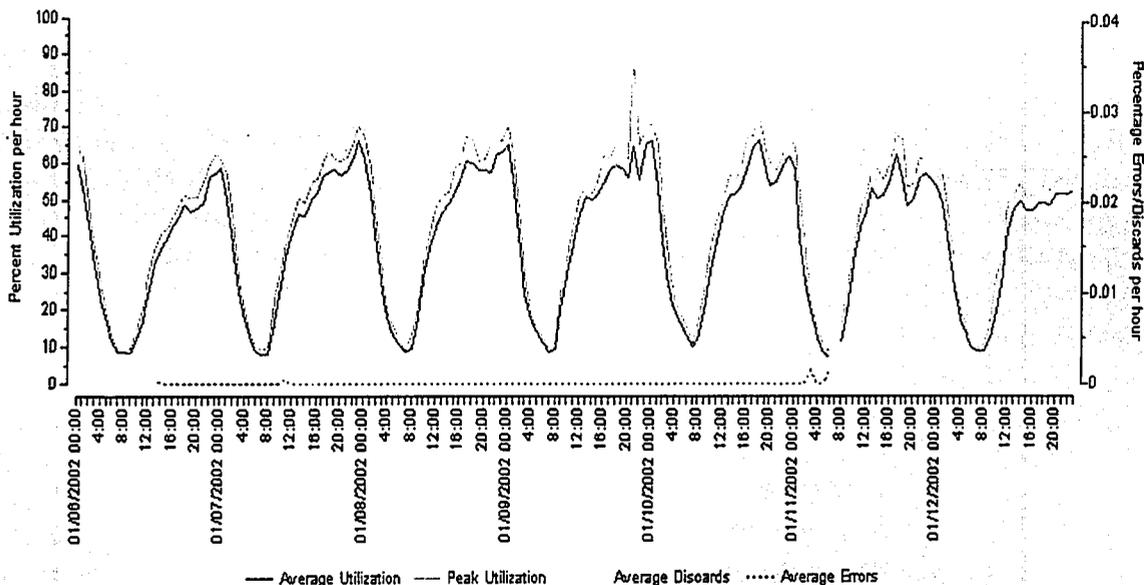


Figura 2.13. Utilización del enlace tres de la Ciudad de Monterrey.

De las gráficas anteriores se presenta la tabla 2.4 que resume el comportamiento de los tres enlaces durante la semana del 6 al 12 de enero en la Ciudad de Monterrey. En esta tabla se observa que el promedio de descartes y errores es cero, también es importante mencionar que los promedios de utilización y los picos máximos no afectan al servicio.

Enlace	Promedio de utilización	Pico máximo de utilización	Promedio de errores	Promedio de descartes
Uno	23%	47.3%	0%	0%
Dos	46.4%	67.5%	0%	0%
Tres	51.7%	77.2%	0%	0%

Tabla 2.4. Resumen de los enlaces de la Ciudad de Monterrey.

Los enlaces del servidor de la Ciudad de Monterrey, como se puede observar, se encuentran dentro de los parámetros operacionales. Las capacidades de cada uno de los tres, no exceden el 55% de utilización.

Al igual que en la ciudad de Guadalajara, existe redundancia entre ellos para evitar que al fallar cualquiera, se caiga el enlace.

2.2.2. Desempeño de memoria y CPU en los servidores

Como parte fundamental de análisis del servicio de DNS, es muy importante conocer el desempeño de la memoria y de la CPU de cada uno de los servidores, ya que de ellos depende directamente la correcta operación del servidor, por esto es necesario obtener a través de programas (que se describen en el anexo A) información que pudiese ser graficada para observar su comportamiento en una semana típica de servicio.

En el caso de la memoria, se toman como referencia la RAM y la SWAP², que representan en conjunto, la capacidad real de memoria por servidor. Y en la CPU sólo es necesario tomar el porcentaje de utilización para determinar su

² Conocida también como memoria virtual. Es definida en disco duro y cumple la función de almacenar ciertas instrucciones del microprocesador y sus periféricos.

grado de desempeño. Para dar una idea de las capacidades máximas de memoria en cada servidor, se muestra la tabla 2.5 que señala el modelo (igual para todos los servidores) y sus unidades de memoria.

Modelo de servidor	Enterprise 1
Capacidad de disco duro	2 GB
Memoria RAM	128 MB
Memoria SWAP	360 MB

Tabla 2.5. Características de memoria en los servidores.

Tomando como referencia la tabla anterior y el valor máximo de memoria RAM y SWAP como el 100% de utilización, podemos analizar las figuras 2.14, 2.15, 2.16, 2.17, 2.18, y 2.19, tomadas en periodos de 15 minutos durante una semana.

Después de observar las gráficas, es claro que los seis servidores tienen una muy alta ocupación de memoria RAM y de la CPU, en estos dos parámetros existen pequeños periodos de tiempo en los que su ocupación baja considerablemente, pero esto se debió a que se reinician los procesos de BIND dos veces al día, ya que de lo contrario los servidores alcanzarían su ocupación máxima, dejando de resolver consultas de cualquier tipo. Esta condición es muy crítica, porque un servidor de DNS por naturaleza debe de estar operando varios meses sin necesidad de reiniciarlo y en Uninet, este proceso se hace varias veces al día. En cuanto a la memoria SWAP, se detecta que la ocupación crece conforme se va acabando la memoria RAM y lo ideal sería que los servidores tuvieran suficiente RAM para hacer un uso mínimo de SWAP. También se observa que los servidores nsmex1 y nsmex3 a pesar de tener picos de ocupación casi del 100% no llegan a tener la saturación tan grande que presenta dns, la razón de estas diferencias se puede deber a una mala distribución del número de usuarios que los consultan ó al número de zonas que cada uno administra.

Cabe mencionar que no existe una regla absoluta de qué porcentaje de memoria y CPU en ocupación debe tener un servidor de DNS. Del análisis anterior concluimos que, en el desempeño de los servidores no existió atención a la demanda de usuarios, que requiere un servidor de DNS, y a la escalabilidad del sistema, pero sobre todo, a la necesidad de monitorear estos dos aspectos.

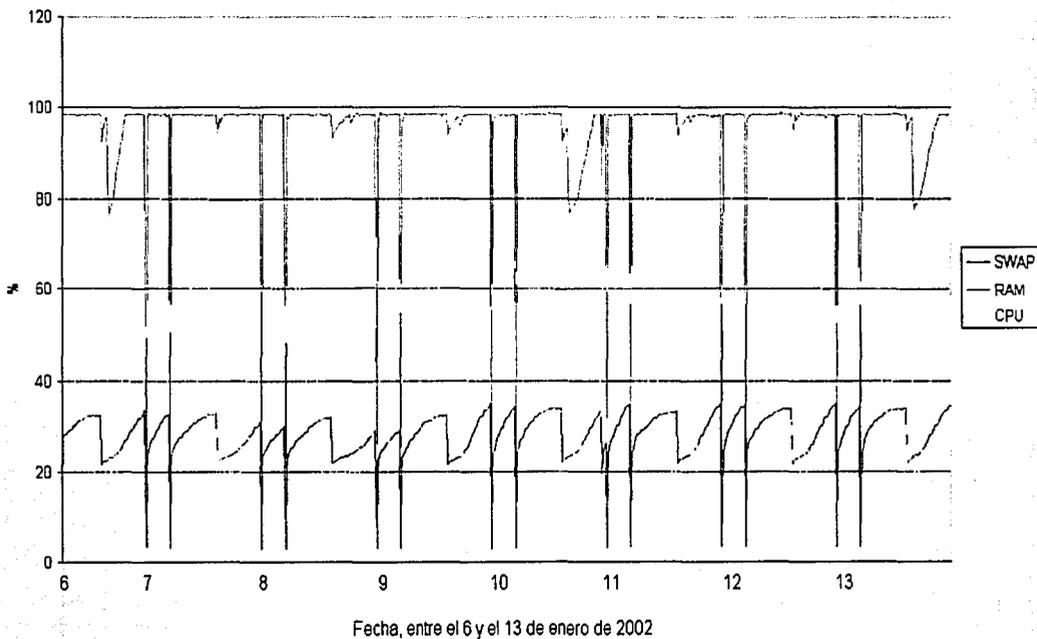


Figura 2.14. Desempeño del servidor dns.

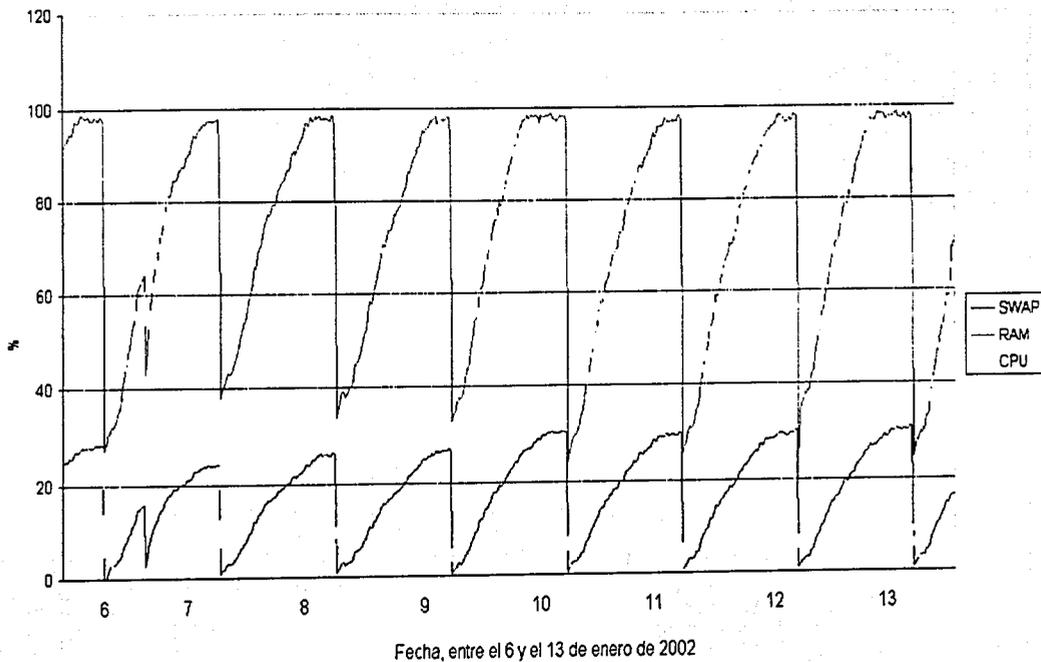


Figura 2.15. Desempeño del servido nsmex1.

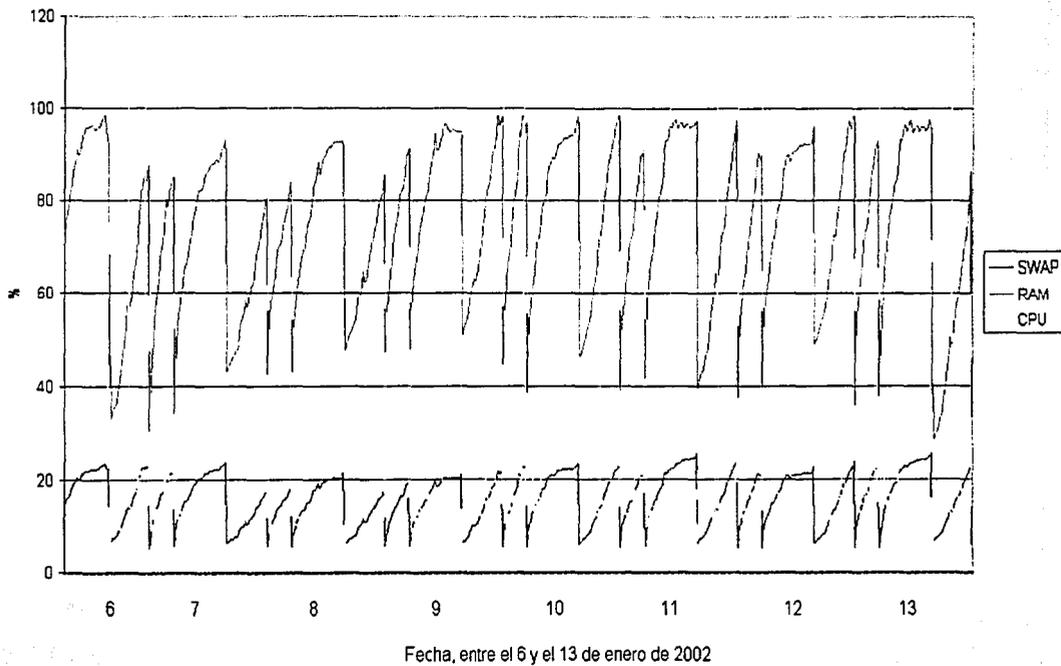


Figura 2.16. Desempeño del servidor nsmex2.

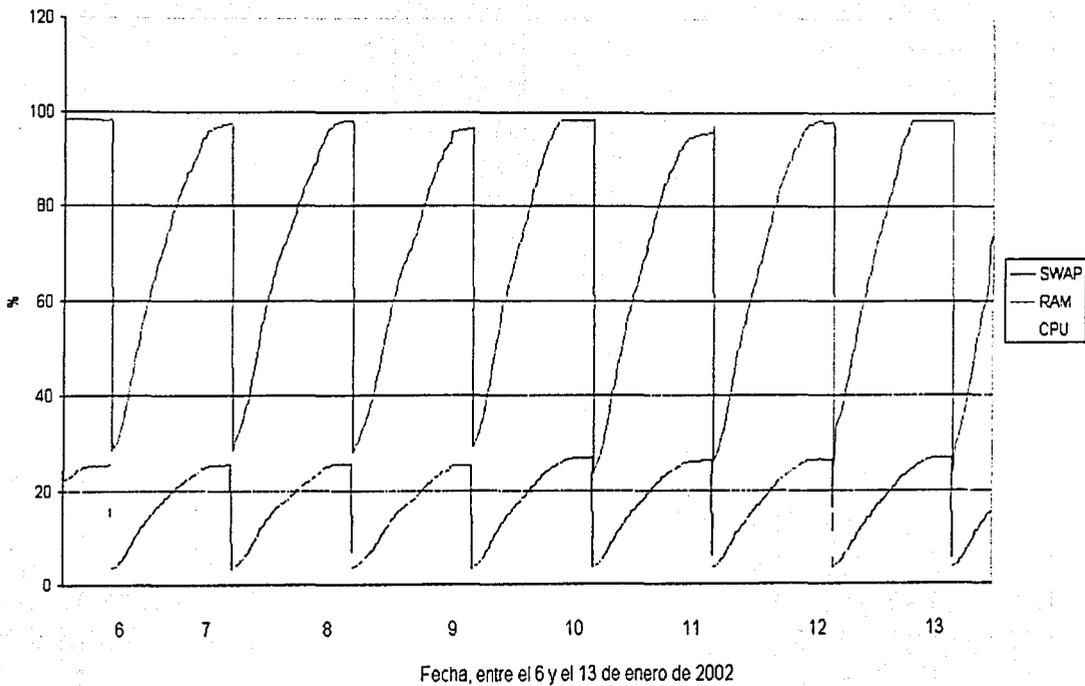


Figura 2.17. Desempeño del servidor nsmex3.

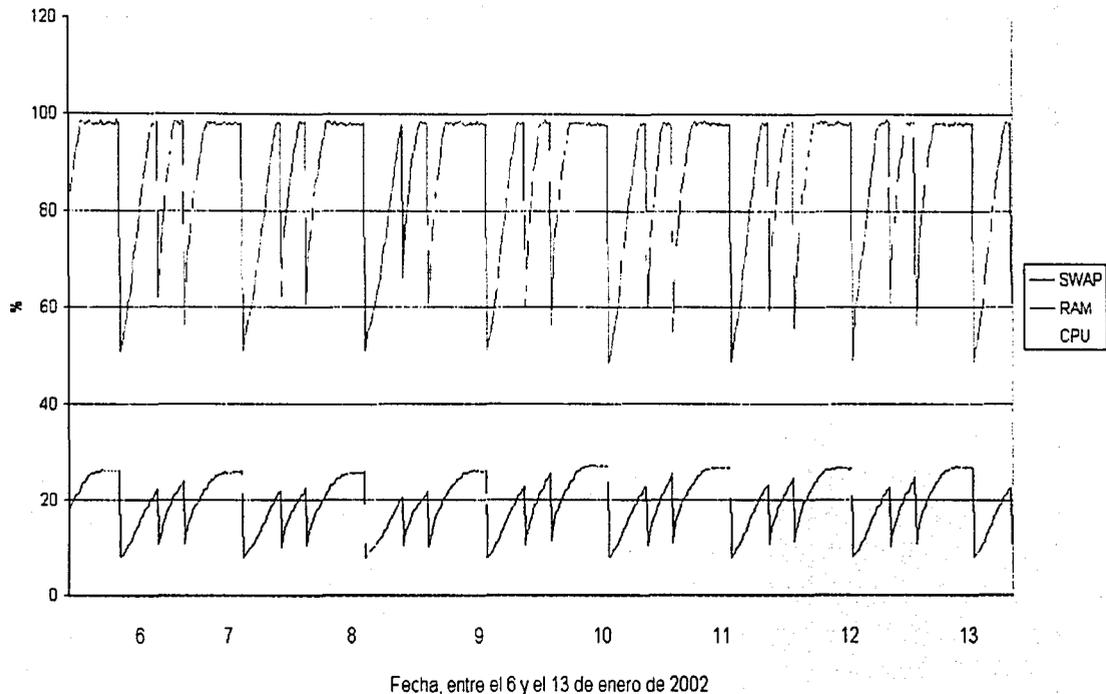


Figura 2.18. Desempeño del servidor nsgdl.

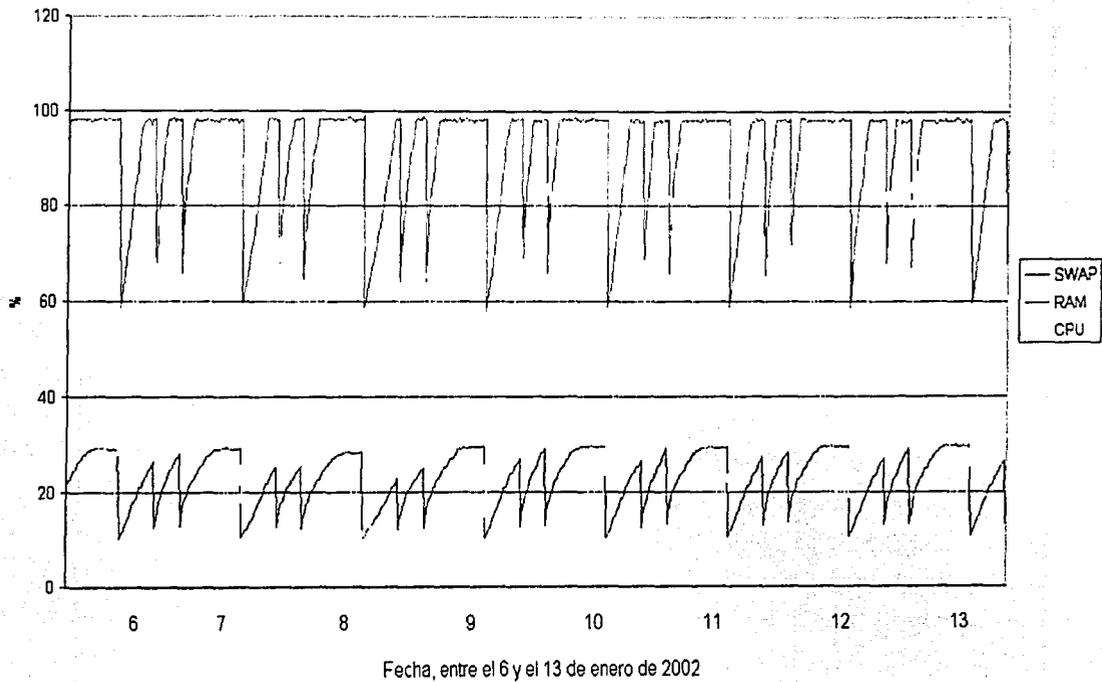


Figura 2.19. Desempeño del servidor nsmtv.

2.2.3. Tráfico de DNS

Otra estadística relevante es el número de consultas por segundo recibidas en el servidor de DNS. Medición que forma parte del cálculo del tráfico de DNS en una red. La medición se obtiene a través de las utilerías de BIND, las cuales depositan el número de peticiones en un archivo de texto y se procesan para ser graficadas.

Por ejemplo, la secuencia siguiente constituye un mensaje de BIND para especificar, entre otras cosas, el número de peticiones recibidas y contestadas por el servidor. Donde SAns, es el número de respuestas del servidor y RQ, el de peticiones.

```
Aug 1 11:00:49t terminator named(103): NSTATS 965152849
959476930 A=8
SOA=356966 PTR=2 TXT=32
SAns =391191 RQ=458031
```

Se puede tener una estimación del volumen del tráfico en una red local multiplicando el número de peticiones (RQ) más el de respuestas (SAns) en una hora por 800 bits (100 es el promedio de bytes por mensaje de DNS entre *hosts* y servidor), y dividido entre 3600 (segundos por hora). Esto puede dar idea de cuanto ancho de banda consume el servicio de DNS en una red local.

Para el caso anterior:

$((391191 + 458031) 800 \text{ bits}) / 3600 \text{ seg} = 188,716 \text{ bits/seg} = 188.716 \text{ kb/seg}$

Como se puede apreciar es muy baja la tasa bits, en comparación con las capacidades de los enlaces E1 (2.048 Mb) para este ejemplo. Sin embargo, en realidad esta cantidad llega a rebasar la cantidad soportada por un enlace E1.

Por tal motivo, para completar el análisis de tráfico se requiere saber cuantas peticiones por unidad de tiempo se hacen a un servidor y determinar el comportamiento de la carga de peticiones para cada uno. Es decir, existe la posibilidad de que un servidor esté recibiendo la mayor parte de las peticiones y en consecuencia afectando su tiempo de respuesta, mientras otro recibe la minoría. Situación que desbalancea la carga de trabajo.

Se pueden observar las estadísticas de peticiones realizadas a los servidores de DNS por segundo en una semana, mostradas en las figuras 2.20, 2.21, 2.22, 2.23, 2.24 y 2.25.

Cotejando todas ellas, se observa un comportamiento relevante, los días lunes por la mañana hay incrementos notables en el número de peticiones en los

servidores de la Ciudad de México. Por otra parte, es notorio que los servidores dns, nsmex1, nsgdl1 y nsmt1 reciben un mayor número de consultas, consumiendo mayores recursos de memoria RAM y CPU. Estos resultados coinciden con el análisis del desempeño de los servidores, ya que los servidores que reciben mayor número de consultas son los que tiene mayores problemas de ocupación. Por el contrario los servidores nsmex2 y nsmex3 reciben el menor número de consultas.

En estudios del comportamiento de tráfico de redes y DNS, es común darse cuenta que los días lunes (comienzo de semana laboral) la mayor parte de las personas que ingresan a Internet suelen revisar su correo electrónico y empezar a navegar. El problema es que gran cantidad de ellas lo hace al mismo tiempo, lo que requiere de mayor capacidad en los equipos para soportar la demanda. Aunque esto no es una regla, pues depende tanto de la ubicación del servidor, como de la cantidad de usuarios que lo consulten.

Por otra parte, se observa que el servidor dns recibe el mayor número de peticiones, tal como se muestra en la gráfica de la figura 2.20. Este parámetro máximo de peticiones se puede tomar como referencia en el cálculo de ciertas características de los servidores, como se verá más adelante.

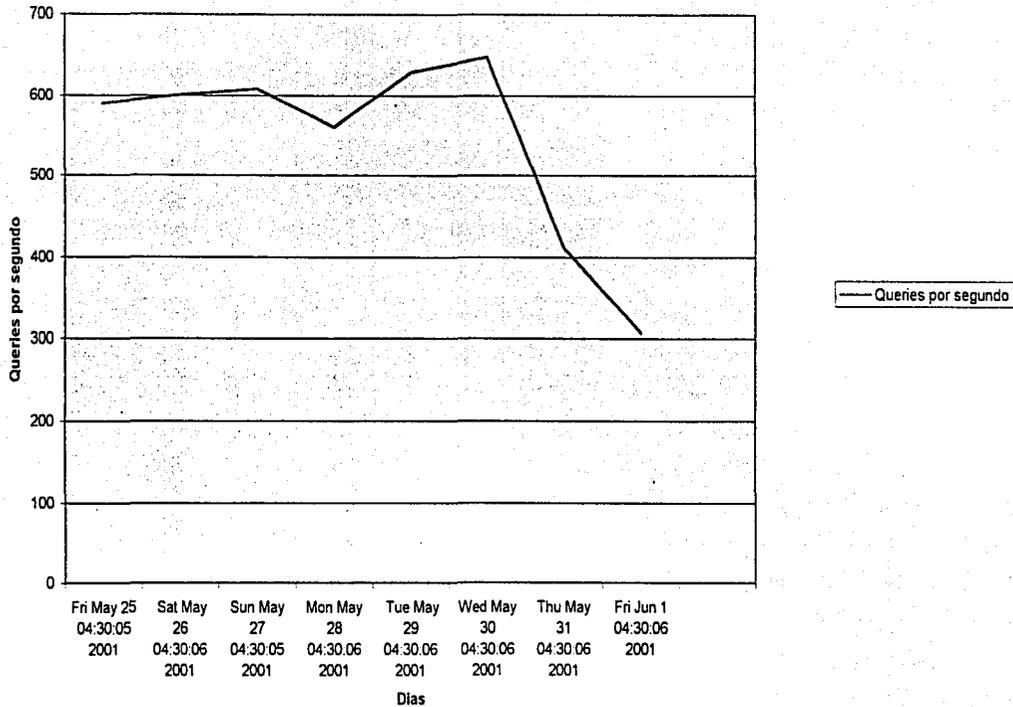


Figura 2.20. Consultas por segundo al servidor dns.

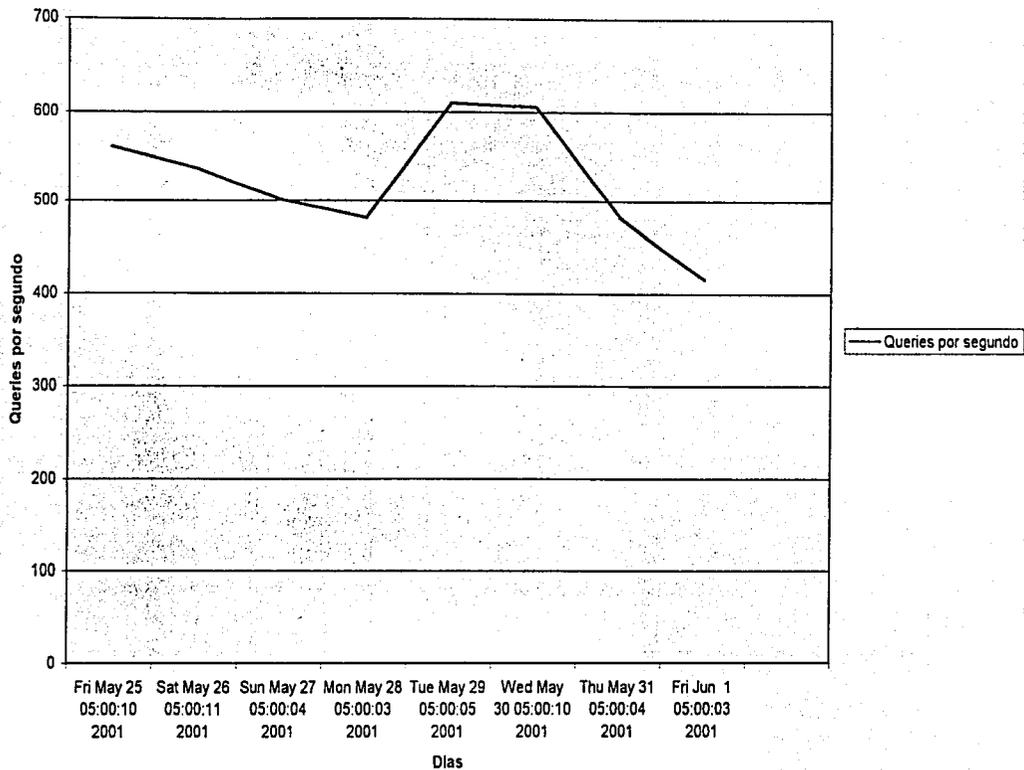


Figura 2.21. Consultas por segundo del servidor nsmex1.

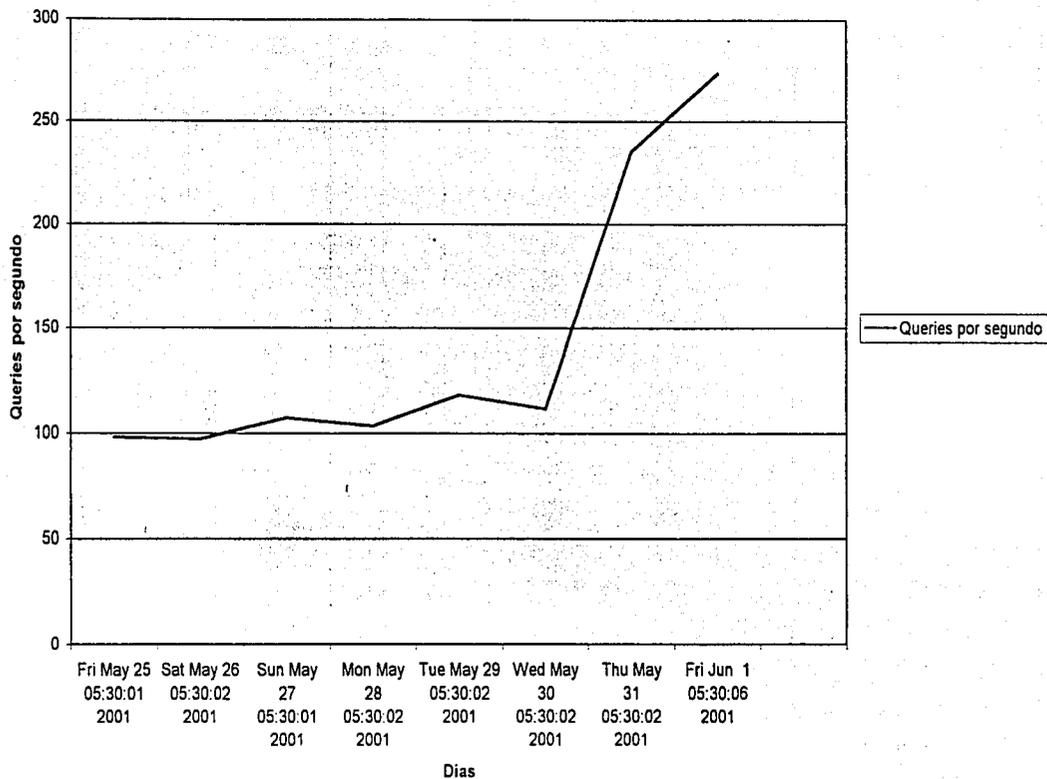


Figura 2.22. Consultas por segundo al servidor nsmex2.

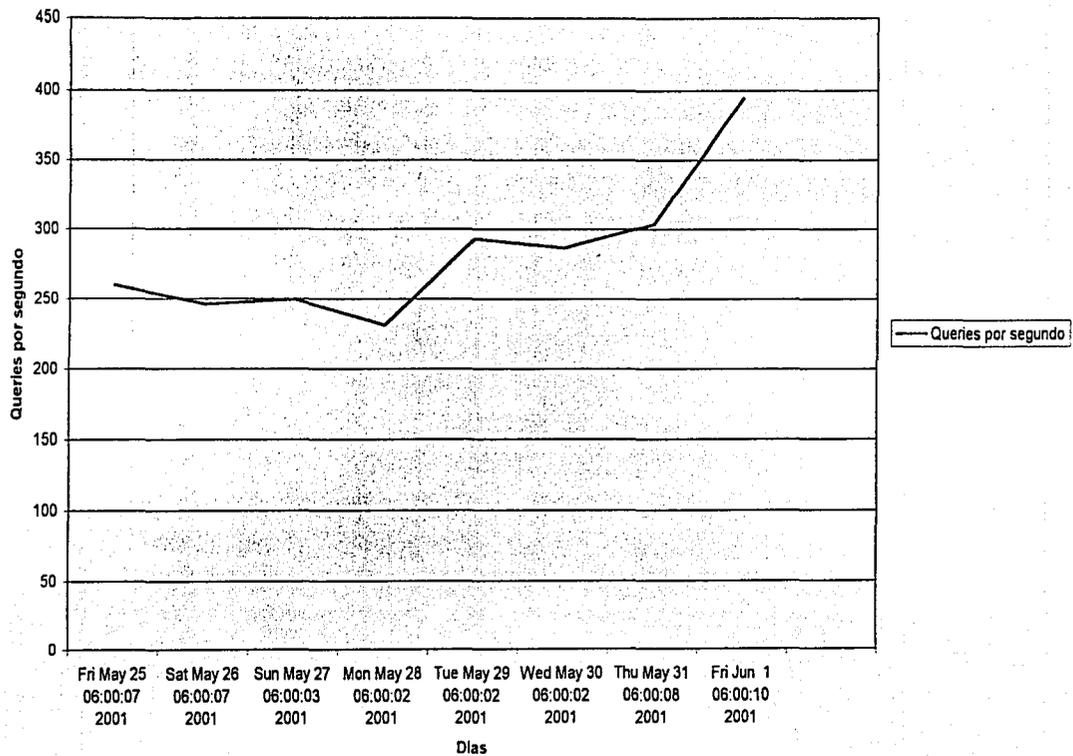


Figura 2.23. Consultas por segundo al servidor nsmex3.

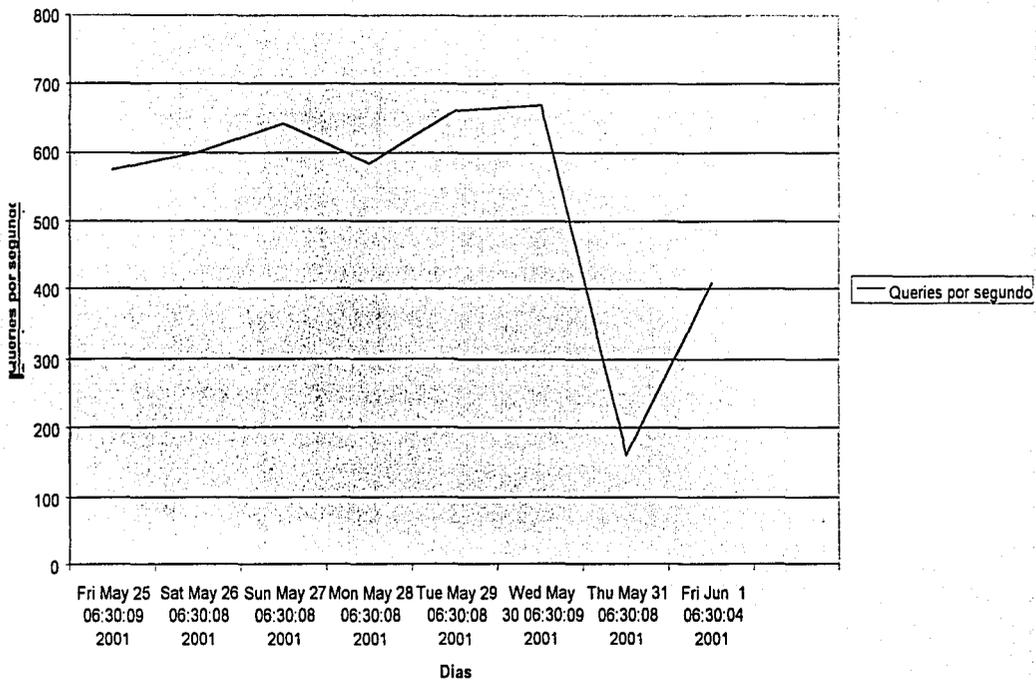


Figura 2.24. Consultas por segundo al servidor nsgdl.

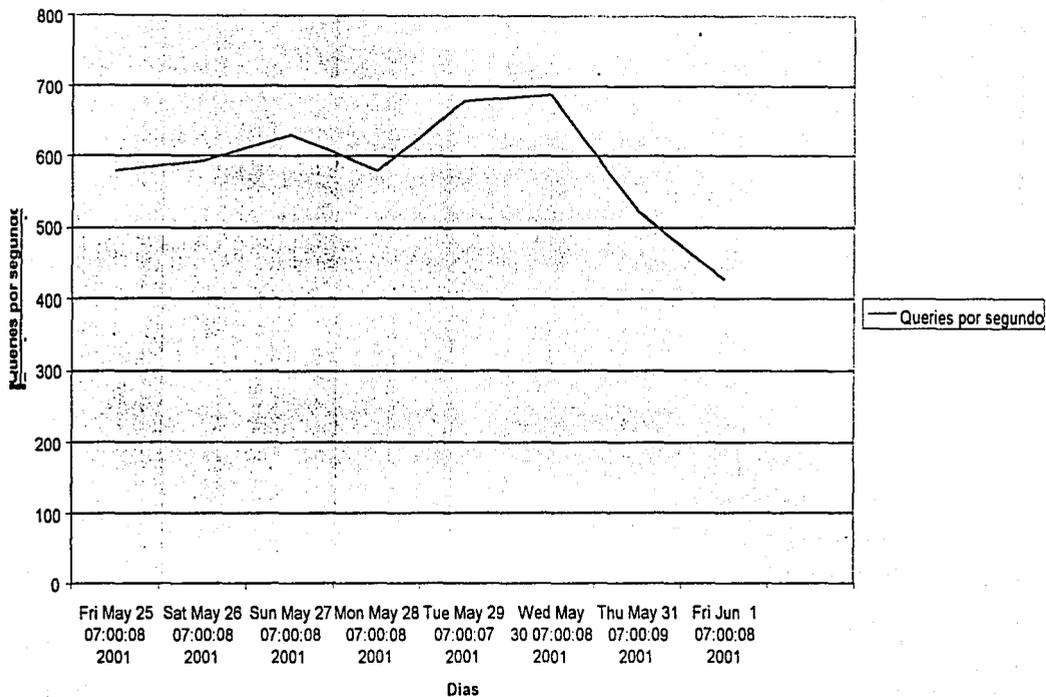


Figura 2.25. Consultas por segundo al servidor nsmtv.

2.3. Distribución de DNS en los servidores de acceso

Para el servicio de Internet residencial, el servidor de acceso a la red (NAS, *Network Access Server*) da la posibilidad de conectarse a través de la red telefónica a Internet, por lo que puede ser definido como un dispositivo de enlace entre la red telefónica y la red de datos.

Una de las funciones más importantes de un servidor de acceso es asignar a una computadora de manera temporal una dirección IP, un gateway, y dos servidores de DNS para poder resolver nombres, esta última es muy importante ya que Uninet tiene aproximadamente 900,000 usuarios, y si la asignación de los servidores de DNS no está debidamente planificada, la carga de trabajo de cada servidor se verá seriamente afectada.

Dentro de la infraestructura de Uninet existen 700 servidores de acceso distribuidos en todo el territorio nacional, la distribución está basada en las siguientes regiones:

Región Centro: La cual abarca el Estado de México, Distrito Federal, Guanajuato, Hidalgo, Michoacán, Morelos y Querétaro.

Región Sureste: La cual abarca los estados de Campeche, Chiapas, Guerrero, Oaxaca, Puebla, Quintana Roo, Tabasco, Tlaxcala, Veracruz y Yucatán.

Región Occidente: La cual abarca los estados de Aguascalientes, Colima, Jalisco, Nayarit, San Luis Potosí y Zacatecas.

Región Norte: La cual abarca los estados de Baja California Norte, Baja California Sur, Durango, Chihuahua, Coahuila, Nuevo León, Sinaloa, Sonora y Tamaulipas.

Para determinar como se estaban asignando los servidores de DNS en los servidores de acceso, se entró a cada uno de ellos para revisar su configuración y el resultado se muestra en la tabla 2.6.

Servidor de DNS	Región	Primer Servidor	Segundo Servidor
Dns	Centro	15	102
Nsmex1	Centro	32	11
Nsmex2	Centro	355	155
Nsmex3	Sureste	81	367
nsgdl1	Occidente	126	37
Nsmt1	Norte	91	28

Tabla 2.6. Distribución de servidores de acceso.

Como se puede observar, los servidores nsmex2 y nsmex3 son asignados con mayor frecuencia, y nsmex1 es de los menos utilizados. Con lo que se comprobó que hay servidores que tienen mayor número de asignaciones.

Otra problemática detectada durante este análisis, fue la falta de procedimientos para la atención de fallas, ya que al presentarse un problema en alguno de los servidores de DNS, los operadores modificaban manualmente la configuración de cada uno de los servidores de acceso, una vez solucionada la falla no se regresaba la configuración, provocando un continuo movimiento en la asignación de servidores de DNS.

Con esto se concluye que existe un total desbalanceo en cuanto a la asignación de servidores de acceso para cada región, afectando directamente el desempeño de cada uno de los servidores de acceso y por el servicio de resolución de nombres de Uninet.

2.4. Inventarios

En una infraestructura tan grande como lo es la de Uninet, es necesario contar con un inventario confiable para llevar un control de todas las zonas, de las direcciones y de los equipos. Esta actividad se realizó a través de la revisión del archivo named.conf, de las tablas de BIND y de la bitácora del sistema, de cada uno de los servidores.

Se obtuvo un listado de todas las zonas de los servidores, revisando contra la base de datos del NIC México qué servidores deben ser primarios o secundarios para cada zona y se realizó un programa para probar como estaban resolviendo las zonas. El resultado se muestra en la tabla 2.7.

ZONAS	SITUACIÓN ACTUAL	ESTADO OPERATIVO	OBSERVACIONES
prodigy.net.mx uninet.net.mx telmex.net.mx	Primario ante NIC. dns Secundario ante NIC. nsgd1 nsmt1	Las zonas funciona de forma irregular, ya que no resuelve de forma autoritativa en nsgd1 y nsmt1.	A pesar de resolver las zonas no es de la forma en que están configurados.

Tabla 2.7. Estado general del servicio por zonas. (Continúa)

Resolución Inversa Corporativos	<p>Para una clase B se utilizan: dns nsmex1</p> <p>Para otra B se utilizan: nsmex2 nsmex3.</p> <p>Para otra B se utilizan: dns nsgdl1 nsmty1</p>	Los inversos no funcionan de acuerdo a su configuración.	<p>Se registraron ante el NIC servidores que no están asignados para estas zonas.</p> <p>No se resuelven en su totalidad.</p>
Resolución Inversa Dial-Up	<p>Para una clase se tienen como servidores a: dns nsmty1 nsgdl1</p> <p>Para otra clase se tienen como servidores de nombres a: dns nsgdl1</p>	Los inversos no funcionan de acuerdo a su configuración.	<p>Falta configurar zonas.</p> <p>Se registraron ante el NIC servidores que no están asignados para esta zona.</p> <p>No resuelven en su totalidad.</p>
Resolución Inversa Redes Internas	Se tienen configurados a los servidores: dns nsmex1 nsmex2 nsmex3 nsmty1 nsgdl1	Los inversos no funcionan de acuerdo a su configuración.	<p>Falta configurar zonas.</p> <p>Se registraron ante el NIC servidores que no están asignados para esta zona.</p> <p>No resuelven en su totalidad.</p>
Zonas Clientes (Secundarios)	Se tienen configurados a los servidores: dns nsmex1 nsmex2 nsmex3 nsmty1 nsgdl1	Los inversos no funcionan de acuerdo a su configuración.	<p>No funcionan las transferencias de zona.</p> <p>No resuelven en su totalidad.</p>

Tabla 2.7. Estado general del servicio por zonas.

En la tabla 2.8 se presentan con mayor detalle las resoluciones de zonas configuradas en los servidores DNS.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

SERVIDOR DNS	ZONAS INVERSOS CONFIGURADOS	RESPUESTA CORRECTA	RESPUESTA NO AUTORITATIVA	NO RESUELVE	FALLA
Dns	Primario de 616 zonas de Corporativo	605	6	5	
Dns	Primario de 1077 zonas de Dial-Up	1066	2	9	
Dns	Primario de 5 zonas internas	5			
Dns	Primario de 17 zonas delegadas	10	5	2	
nsmex1	Primario de 9 zonas de corporativo	9			
nsmex1	Primario de 4 zonas internas	4			
nsmex1	Primario de 159 zonas de corporativo	139		29	
nsmex2	Primario de 7 zonas internas	6		1	
nsmex3	Primario de 22 zonas de corporativo	22			
nsmex3	Primario de 2 zonas de corporativo	2			
nsmex3	Primario de 5 zonas de UniNet	5			
nsgld1	Secundario de 244 zonas de corporativo	235	2	7	
nsgld1	Secundario de 598 zonas de Dial-Up	498	6	174	
nsmtly1	Secundario de 207 zonas de corporativo	175	5	26	1
nsmtly1	Secundario de 494 zonas de Dial-Up	473	8	13	
nsmtly1	Secundario de 1 zona interna	1			
Dns	Primarios Clientes	3			
nsmex1	2 Primarios de Clientes	2			
nsmex1	65 Secundarios de Clientes	33	22	10	
nsgld1	2 Secundarios de Clientes			2	

Tabla 2.8. Resoluciones de zonas por servidor DNS.

El inventario de distribución de redes en los servidores se muestra en la tabla 2.9.

EQUIPO	SITUACIÓN ACTUAL	DIRECCIONAMIENTO
dns	Está registrado ante el NIC Tiene la resolución de zonas de clientes como secundario (656) y primario (3). Existen clientes corporativos cuyo resolver apunta a esta máquina.	200.33.150.193
nsmex1	Tiene configurado 8 zonas de inversos para redes internas. Está configurado para 6 zonas de clientes (primarios y secundarios).	200.33.146.193
nsmex2	Esta configurado como servidor secundario para 38 zonas de inversos de corporativos y 135 primarios de corporativos, una red interna y 4 redes que no deberían estar aquí.	200.33.146.201
nsmex3	Se tienen registradas las resoluciones inversas como primario de 23 clases "C", 22 de corporativo y 1 red interna. Se tiene configurada la resolución inversa de 5 clases "C" de UniNet.	200.33.146.209
nsgdl1	Tiene configuradas dos zonas de clientes: hiopernet.com.mx itslp.edu.mx Esta registrado ante el NIC como secundario para las zonas: uninet.net.mx telmex.net.mx prodigy.net.mx	200.23.242.193
nsmtly1	Esta registrado ante el NIC como secundario para las zonas: uninet.net.mx telmex.net.mx prodigy.net.mx	200.33.148.193

Tabla 2.9. Distribución de redes.

Como se muestra en las tablas la distribución de zonas por servidor no es equitativa, lo que causa mayor utilización de memoria, CPU y SWAP en algunos servidores que afecta al desempeño del servidor. Por otra parte los operadores configuran las zonas sin tener una definición de las funciones de los servidores, lo que provoca errores, desorden y una mala distribución en la configuración.

Además se tienen errores de sintaxis en los archivos de BIND que provocan que algunas zonas no resuelvan o no realicen transferencias de las zonas que se tienen asignadas como secundarias.

2.5. Análisis de tipos y versiones de software para los servidores

Cada uno de los servidores de DNS tiene una versión de BIND instalada, que idealmente debe ser la misma para todos. Como parte del análisis se hizo una revisión de dichas versiones, así como también del sistema operativo que tiene los servidores. Los resultados de este análisis se muestran en la tabla 2.10.

Servidor de DNS	Versión BIND	Versión Sistema Operativo
dns.uninet.net.mx	8.1.2	Solaris 2.5
hermes.uninet.net.mx	8.1.2	Solaris 2.5
nsmex2.uninet.net.mx	4.9.3	Solaris 2.5
nsmex3.uninet.net.mx	8.1.2	Solaris 2.5
nsmtly1.uninet.net.mx	8.1.2	Solaris 2.5
nsgdl1.uninet.net.mx	8.1.2	Solaris 2.5

Tabla 2.10. Relación de versiones de BIND y de sistema operativo.

Podemos observar que no todos los servidores cuentan con la misma versión de BIND, y las consecuencias de estas diferencias se han reflejado en la operación del servicio de DNS de la siguiente manera:

- La sintaxis entre la familia de versiones 4.X y la 8.X es totalmente diferente y los operadores sólo están familiarizados con la sintaxis de la 8.X, por lo que ya no se le da mantenimiento al servidor nsmex2.
- La carga de datos en BIND se dejó de realizar en el servidor nsmex2 y se distribuyó entre los demás servidores, provocando un desbalanceo en el volumen de información que cada uno tiene.
- Problemas en las transferencias de zonas por una posible incompatibilidad de versiones, provocando que el servidor nsmex2 no cuente con información actualizada.

Es importante mencionar que la versión 8.1.2 terminó su ciclo de vida en noviembre de 1998, por lo tanto es una versión no soportada por el fabricante. En las diferentes versiones que se han desarrollado después de la 8.1.2 se han corregido muchos *bugs*³, principalmente de seguridad, y también se les ha integrado nuevas funcionalidades. Si no se hace una actualización de la versión de BIND se corren riesgos fuertes que podrían afectar a la operación del servicio de DNS. En la tabla 2.11 se resumen las vulnerabilidades de la versión 8.1.2.

³ Error de programación en un sistema de cómputo, que provoca un comportamiento anormal, este puede ser tan simple que no se perciba o tan grave que tire completamente una aplicación.

NOMBRE DEL BUG	AFECTACION	DESCRIPCIÓN
Naptr	Negación del Servicio	Provoca validación errónea de los registros PTR cuando la tabla de la zona es leída por el servidor, en ocasiones llega a tirar por completo un servidor de DNS.
Maxdname	Negación del Servicio	El uso de sprin ⁴ tf() con datos tomados de la red puede provocar una sobre carga de memoria, cuyo resultado se manifiesta en un comportamiento inesperado de la aplicación.
Solinger	Negación del Servicio	Es posible provocar remotamente pausas a BIND por intervalos superiores a los 120 segundos, utilizando sesiones anormales de TCP.
Infoleak	Acceso a Información	Es posible hacer una consulta de resolución inversa que permita leer remotamente información del sistema.

Tabla 2.11. Vulnerabilidades de BIND 8.1.2.

En lo que respecta al sistema operativo de los servidores, todos cuentan con la versión 2.5 de Solaris, la cual dejó de ser soportada por Sun Microsystems a inicios del 2001, por lo que cualquier falla dentro del sistema operativo ya no podrá ser resuelta, con base en esto, se hace la recomendación de migrar a una versión más reciente.

⁴ Rutina de Unix que se utiliza para desplegar información.

CAPÍTULO III

Diseño del servicio de DNS

En el presente capítulo se desarrolla el diseño del servicio de DNS, el sistema de administración y el sistema de monitoreo, los cuales deben tomar en cuenta todos los requerimientos del servicio y dar solución a la problemática detectada en el análisis realizado.

3.1. Diseño de la arquitectura del servicio de DNS

La arquitectura del servicio del DNS es un punto clave, ya que de su correcto diseño y configuración depende en gran medida el buen desempeño que se le ofrezca a los usuarios. En el diseño se define el número de servidores y su distribución física para posteriormente especificar las funciones lógicas de cada uno de ellos. Acorde con estas funciones se propone la distribución de cargas mediante la asignación de los servidores a los usuarios.

3.1.1. Distribución física

Para la distribución física se tomó en cuenta la división regional hecha por Uninet; en cuatro regiones. En cada una de ellas se consideró el contar con dos servidores de DNS, de manera que uno sea el respaldo del otro, brindando una

mayor redundancia al servicio. Con esta definición también evitamos que una región curse innecesariamente tráfico de DNS entre ellas. La distribución propuesta se muestra gráficamente en la figura 3.1. Esto implica crecer de seis a ocho el número de servidores dedicados a resolver consultas de los usuarios.

Por cuestiones de administración es necesario contar con un nuevo servidor, además de los ocho de las regiones, el cual tendrá funciones de configuración y distribución de información hacia los servidores de cada región. Este equipo por facilidad de manejo se recomienda que se ubique en las mismas instalaciones donde se encuentren los administradores del servicio, es decir, en el corporativo de Uninet en la Ciudad de México.

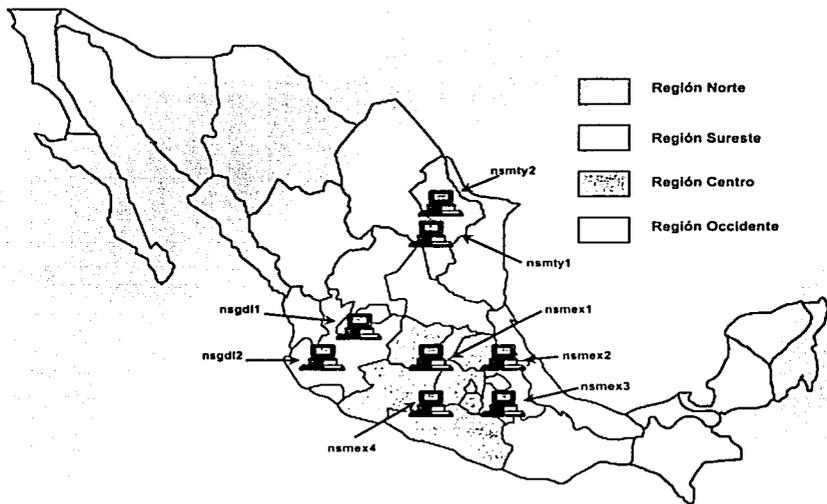


Figura 3.1. Distribución regional de servidores.

Los equipos que ya existían se quedarán con sus mismos nombres y direcciones, a los nuevos se les asignarán nombres y direcciones siguiendo la misma nomenclatura. La distribución de los nueve equipos, sus nombres y direcciones IP se muestran en la tabla 3.1.

REGIÓN	SERVIDOR DNS	DIRECCIÓN IP	CIUDAD
CENTRO	nsmex1	200.33.146.193	Ciudad de México
	nsmex2	200.33.146.201	Ciudad de México
SURESTE	nsmex3	200.33.146.209	Puebla
	nsmex4	200.33.146.217	Puebla
OCCIDENTE	nsgdl1	200.23.242.193	Guadalajara
	nsgdl2	200.23.242.201	Guadalajara
NORTE	nsmtly1	200.33.148.193	Monterrey
	nsmtly2	200.33.148.201	Monterrey
	dnsadm-interno	200.33.150.193	Ciudad de México

Tabla 3.1. Distribución de servidores por región.

3.1.2. Distribución lógica

Para realizar el diseño lógico se tomó como premisa el contar con un punto único de administración del servicio, para que se distribuya homogéneamente la información de las zonas a todos los servidores.

El servidor administrativo deberá configurarse como primario para todas las zonas administradas por Uninet, y los servidores de cada región serán configurados como secundarios, de esta manera cada modificación se hará únicamente en el servidor administrativo y automáticamente se realizarán las actualizaciones en las regiones.

Los valores definidos para las transferencias de zonas se eligieron con base a la experiencia del servicio en Uninet, debido a que no existen reglas establecidas para dichos parámetros. En el servidor primario se deberán establecer los siguientes parámetros:

Refresh: 14400 segundos, con este valor se determina que como máximo cada cuatro horas los servidores secundarios actualizarán las zonas que hayan sufrido alguna modificación en el servidor primario. Este valor se fijó con base al tiempo de respuesta máximo comprometido con los clientes para cambiar información en los servidores.

Retry: 3600 segundos, la definición de este parámetro se hizo con base al tiempo promedio de respuesta para resolver fallas de comunicación de un nodo en la red, asegurando que como máximo transcurrirá una hora para reestablecer la comunicación entre el servidor primario y sus servidores secundarios.

Expire: 604800 segundos. Este parámetro se definió con base al tiempo máximo para solucionar una falla crítica en un servidor. Con este valor se evita resolver consultas con información desactualizada por más de una semana.

3.1.3. Distribución de dominios

Tomando en cuenta el inventario de dominios, se ve la necesidad de hacer una distribución de los dominios que cada servidor deberá resolver.

El servidor primario por definición no aceptará consultas de ningún equipo, y su única función será administrar las zonas a los demás servidores. Esto dará la flexibilidad a los administradores de modificar o incluso suspender la operación del servidor sin afectar el servicio.

Por definición de los servicios de Uninet, las resoluciones se dividen en:

- Resolución de nombres para usuario final Uninet.
- Resolución de nombres de dominio internos Uninet.
- Resolución de nombres de dominio asignados a clientes corporativos Uninet.
- Resolución de dominios inversos para las direcciones IP asignadas a Uninet.

Cada uno de estos tipos de resoluciones deben darse de alta en organismos internacionales (NIC y ARIN), para que puedan ser consultados desde Internet, y también, cabe mencionar, que estos organismos dentro de su proceso de registro sólo dan de alta a dos servidores. Con base a esto se hace la distribución que se muestra en la tabla 3.2.

SERVIDOR DNS	FUNCIÓN
dnsadm-interno	DNS Interno Primario para los nombres de dominio: uninet.net.mx, prodigy.net.mx, telmex.net.mx Dominios inversos del total de direcciones IP Uninet.
nsmex1	Primario ante el ARIN para los dominios inversos de las direcciones IP internas Uninet. Secundario ante el NIC para los nombres de dominio: uninet.net.mx, prodigy.net.mx, telmex.net.mx
nsmex2	Primario ante el ARIN para los dominios inversos de las direcciones IP asignadas al servicio de Dial Up.
nsmex3	Primario ante el ARIN para los dominios inversos de las direcciones IP asignadas a Internet Corporativo.
nsmex4	Secundario ante el NIC para los nombres de dominio asignados a los clientes de Internet Corporativo. Secundario ante el ARIN para los dominios inversos de las direcciones IP asignadas a Internet Corporativo.
nsgdl1	Secundario ante el ARIN para los dominios inversos de las direcciones IP internas Uninet. Secundario ante el NIC para los nombres de dominio: uninet.net.mx, prodigy.net.mx, telmex.net.mx

Tabla 3.2. Distribución de dominios por servidor. (Continúa)

nsgdl2	Secundario ante el ARIN para los dominios inversos de las direcciones IP asignadas al servicio de DialUp.
nsmy1	Secundario ante el ARIN para los dominios inversos de las direcciones IP internas Uninet. Primario ante el NIC para los nombres de dominio: uninet.net.mx, prodigy.net.mx, telmex.net.mx
nsmy2	Secundario ante el ARIN para los dominios inversos de las direcciones IP asignadas al servicio de DialUp.

Tabla 3.2. Distribución de dominios por servidor.

3.1.4. Asignación de servidores

Con el fin de balancear el número de consultas hechas por los usuarios del servicio residencial a cada uno de los servidores de DNS, se realizará una distribución de los servidores de DNS que se le asignan a cada usuario por los servidores de acceso.

Para la resolución de nombres a los usuarios finales les serán asignados dos servidores de DNS, de acuerdo con la región y con el estado al que pertenecen. En la tabla 3.3 se muestra la relación por región de los servidores de DNS que deberán configurarse para su asignación a los usuarios del servicio residencial en los servidores de acceso.

REGIÓN	ESTADOS	OPCIÓN DNS 1	OPCIÓN DNS 2
Norte	Baja California Norte, Baja California Sur, Durango, Chihuahua, Sonora y Sinaloa	nsmy2	nsmy1
	Coahuila, Nuevo León y Tamaulipas	nsmy1	nsmy2
Occidente	Aguascalientes, San Luis Potosí y Zacatecas	nsgdl1	nsgdl2
	Colima, Jalisco y Nayarit	nsgdl2	nsgdl1
Sureste	Guerrero, Oaxaca, Puebla y Tlaxcala	nsmex3	nsmex4
	Campeche, Chiapas, Quintana Roo, Tabasco, Veracruz y Yucatán	nsmex4	nsmex3
Centro	Estado de México y Distrito Federal	nsmex2	nsmex1
	Guanajuato, Hidalgo, Michoacán, Morelos y Querétaro	nsmex1	nsmex2

Tabla 3.3. Asignación de servidores.

Con esta asignación cada servidor de acceso consultará al servidor de DNS más cercano y cada región soportará su carga de resoluciones. Esta asignación también facilita la configuración de los servidores de acceso de cada región.

Para ejemplificar esta asignación, en la figura 3.2 se observan algunos de los estados que pertenecen a la región Sureste, se dibujaron dos servidores de

acceso uno en Oaxaca y otro en Campeche. De acuerdo a la tabla 3.3 cada uno asignará los servidores nsmex3 y nsmex4 en diferente orden, para que a los usuarios que se conecten les sean asignados estos servidores en distinto orden para que uno consulte a nsmex3 y otros a nsmex4 y en caso de falla le pregunten a otro servidor.

3.2. Diseño del sistema de administración

Uno de los resultados del análisis señala que existen problemas en el desempeño del servicio por falta de control en el inventario y en la configuración de los servidores. Por estas razones es necesario buscar una solución que tenga un punto único de control para hacer modificaciones a las zonas del DNS y de ahí distribuir la información a todos los servidores, que se encargue de generar los archivos de configuración de forma automática, sin que los operadores tengan necesidad de entrar a cada servidor a configurar BIND de forma manual, que permita llevar de forma sencilla el inventario de zonas, direcciones y contactos, y por último que sea amigable para los operadores, facilitando su labor de administración.



Figura 3.2. Ejemplo de distribución en servidores de DNS.

Existen varias alternativas para dar solución a todas las necesidades de administración, una de ellas es desarrollar un sistema personalizado, pero esta solución no es viable debido a que en Uninet no hay gente capacitada para hacer un desarrollo de este tipo y también que el tiempo de implementación sería muy grande. Por estos motivos se tomó la decisión de buscar un producto existente cuyas funcionalidades cumplan con las necesidades del servicio.

Los productos evaluados fueron:

- *DNS ONE de Infoblox*
- *NetID de Nortel Networks*
- *Meta IP de Check Point Software Technologies*
- *Shadow IP de Network Systems*
- *QIP Enterprise de Lucent Technologies*
- *Nixu Name Surfer de Nixu*

De cada uno de ellos se evaluó su funcionalidad, compatibilidad con la plataforma de Uninet, facilidad de administración, escalabilidad, que tengan soporte en México y el precio. El producto que cumple con todas los requerimientos de Uninet es *QIP Enterprise* (Ver Anexo C). El resumen de la evaluación se muestra en la tabla 3.4. A continuación se describen las características más importantes de QIP.

- Sistema cliente servidor, con una base de datos centralizada donde se lleva el control del direccionamiento y de los dominios administrados.
- Soporta los sistemas operativos más comunes, como son Solaris, HP-UX y Windows.
- Puede trabajar con BIND en todas sus versiones, con el DNS de Windows, tiene una versión propia de Lucent compatible con BIND.
- Es muy flexible en la administración.
- Tiene un control muy estricto de todos los movimientos que realizan los usuarios.
- Genera reportes de ocupación de redes y subredes.
- Genera de forma automática las tablas de BIND y las distribuye a los servidores de DNS.
- Soporta la asignación dinámica de direcciones IP.

La última versión liberada de QIP es la versión 5.0, por el volumen manejado por Uninet requiere instalarse en un servidor Unix, QIP soporta Oracle y Sybase pero por facilidad de administración se montará sobre una base de datos Oracle en su versión 7.3.4. Cada servidor de DNS deberá tener instalado un agente de QIP que permita hacer la actualización de tablas. Todos los usuarios que requieran hacer consultas o modificaciones al DNS deberán de entrar a través de los clientes de QIP instalados en las computadoras personales de los operadores.

PRODUCTO	SISTEMA OPERATIVO	BASE DE DATOS SOPORTADA	FACILIDAD DE ADMINISTRACIÓN	ESCALABILIDAD	SOPORTE DEL PROVEEDOR	PRECIO	COMENTARIOS
DNS One	Propietario, no tiene compatibilidad con Solaris ni HP-UX.	Propietaria	Descubre de forma automática todas las direcciones IP y genera las tablas de DNS. El acceso al sistema es vía WEB en cada servidor de DNS.	Por cada servidor DNS se requiere tener un equipo dedicado, no es un sistema centralizado.	Infobloc no tiene soporte ni distribuidor en México	Por no tener distribuidor en México no se entregó propuesta comercial	No se continuó la evaluación por que Uninet requiere que el proveedor tenga presencia en México
Net ID	Soporta Solaris y Windows, no hay compatibilidad con HP-UX	Oracle 7 3 Sybase 11.0	Sistema amigable en ambiente gráfico. Es compatible 100% con BIND 8 X. Tiene un módulo para manejar alarmas de DNS.	Puede crecer conforme se crezca el hardware. Requiere un servidor Sun muy robusto con 4 procesadores.	Nortel si tiene soporte en México	\$200,000 USD para direcciones ilimitadas	El sistema tiene muchas ventajas pero no es compatible con HP-UX, además es muy pesado y lento y para compensarlo requiere equipos muy robustos.
Meta IP	Soporta Windows y Solaris, no hay compatibilidad con HP-UX	Propietaria	Sistema amigable, con clientes windows. Es compatible 100% con BIND 8.X. Puede hacer auditorías del uso del direccionamiento. Soporta conexiones seguras con los servidores de DNS.	El número de direcciones IP administradas depende del licenciamiento. No hay restricción por el número de servidores administrados. Puede crecer ilimitadamente siempre y cuando haya recursos de hardware.	Check Point tiene soporte en México pero por experiencia en el uso de sus productos es muy caro el contrato de mantenimiento	\$530,000 USD para direcciones ilimitadas	El sistema es muy completo cumple con las funcionalidades requeridas, pero no es compatible con HP-UX y es un producto muy caro.
Shadow IP	Propietario, no tiene compatibilidad con Solaris ni HP-UX.	Propietaria	Descubre las direcciones IP y genera las tablas de DNS. El acceso al sistema es vía WEB en cada servidor de DNS.	Por cada servidor DNS se requiere tener un equipo dedicado, no hay un servidor central dedicado	Network Sistemas si tiene soporte en México	Por no tener compatibilidad con la plataforma Uninet no pidio propuesta comercial.	No se continuó con la evaluación de este producto por no ser compatible con nuestra plataforma.

Tabla 3.4. Comparación de sistemas de administración de DNS. (Continúa)

QIP Enterprise	Soporta Solaris, HP-UX y Windows	Oracle 7.3.4 y 8.1.2 Sybase 11.0	Ambiente de trabajo amigable con clientes Unix, Windows y WEB. Es compatible 100% con BIND 8 X. Permite asignar privilegios muy específicos por operador. Tiene reportes predefinidos de quién y cuándo realizó movimientos en el sistema. Cuenta con muchas herramientas administrativas para importar y exportar datos, controlar todos los DNS.	El número de direcciones IP administradas depende del licenciamiento. No hay restricción por el número de servidores administrados. Puede crecer ilimitadamente siempre y cuando hayan recursos de hardware.	Lucent cuenta con soporte en México, y el soporte de este producto se puede agregar al contrato ya existente de Uninet con Lucent de otros sistemas.	\$210,000 USD para direcciones IP ilimitadas	Cumple con los requerimientos de Uninet
Nixu Name Surfer	Soporta Solaris, HP-UX y Windows	Propietaria	Ambiente de trabajo amigable con clientes Unix, Windows y WEB. Es compatible 100% con BIND 8 X. Permite asignar privilegios muy específicos por operador. Tiene reportes predefinidos de quién y cuándo realizó movimientos en el sistema. Cuenta con muchas herramientas administrativas para importar y exportar datos, controlar todos los DNS.	El número de direcciones IP administradas depende del licenciamiento. No hay restricción por el número de servidores administrados. Puede crecer ilimitadamente siempre y cuando hayan recursos de hardware.	Nixu sí tiene soporte en México	\$3970 USD por cada 1024 direcciones IP, Uninet administra 500,000 por lo que el precio sería \$1,895,000 USD	Cumple con las funcionalidades de Uninet pero el precio es muy alto.

Tabla 3.4. Comparación de sistemas de administración de DNS.

Los servidores de DNS ya están definidos, pero no el servidor de administración donde se instalará QIP, el fabricante hace recomendaciones de las características necesarias, las cuales dependen en gran medida del número de direcciones IP que serán administradas (500,000) y del número de servidores de DNS(9). Con base en estos datos se recomienda un servidor que tenga 500 MB en RAM, un procesador a 400 MHz o superior y 6 GB en disco duro. Uninet, tomando como referencia esto datos, asignará el siguiente equipo:

Servidor HP C3000
1 GB en RAM
1 procesador a 450 MHz
8 GB de disco duro

El sistema será administrado a través de la interfase gráfica de QIP, la cual tiene filtros que evitan introducir caracteres inválidos. Esta herramienta será la encargada de generar automáticamente los archivos de configuración de BIND y distribuirlos en los servidores de DNS, esto se puede hacer en horarios determinados por el administrador.

Se contará con la ventaja de tener diferentes niveles de administración, ya sea para configuración o consulta de información, además de reportes de las modificaciones hechas por los usuarios.

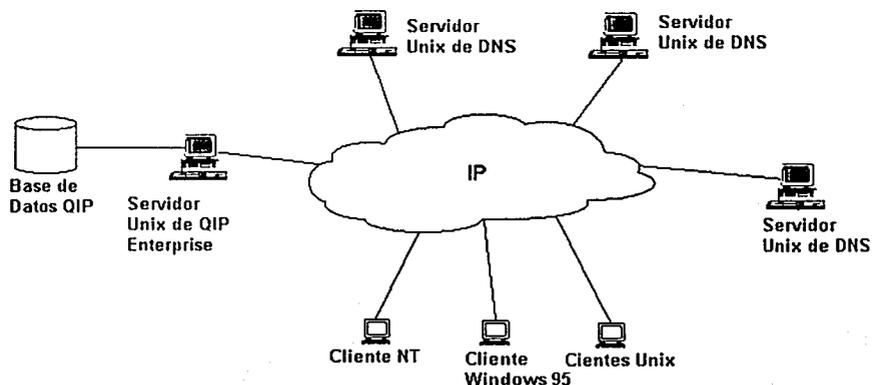


Figura 3.3. Arquitectura del sistema de administración.

Con base en la información presentada, la arquitectura del sistema de administración que se propone consta de una base de datos Oracle, que se alojará en un servidor de administración; agentes instalados en los servidores de DNS, que permitirán la actualización de las tablas de BIND de forma automática; y por último clientes instalados en plataformas Windows, NT y Unix, a través de los cuales los operadores tendrán acceso a la información. Estos componentes se muestran en la figura 3.3.

3.3. Diseño del sistema de monitoreo

Una vez que se tiene definido el diseño de la arquitectura y de la administración del servicio, es necesario contar con un sistema que permita monitorear el desempeño del servicio para mantenerlo dentro de sus rangos operativos, este debe cubrir el monitoreo del servicio, el desempeño de los servidores y el desempeño de los enlaces.

3.3.1. Monitoreo del servicio

Este monitoreo está enfocado al servicio de resolución de nombres, sin importar el estado de los servidores ni el de los enlaces; a través de él se medirá la disponibilidad del servicio, los tiempos de respuesta de los servidores y en caso de falla notificará a los operadores.

Existen varios programas que pueden cumplir con las necesidades de Uninet, pero no se hizo ninguna evaluación dado que ya se cuenta con *Firehunter* (Ver Anexo D), software de *Agilent Technologies*, dedicado a medir el desempeño de los servicios de Internet, entre ellos el de DNS. Este software no se había podido utilizar por que requiere que los servidores tengan como sistema operativo Solaris 2.6 o una versión superior.

El programa funciona de la siguiente manera: para cada uno de los servicios que mide, simula ser un cliente que hace consultas a cada uno de ellos, estas consultas se realizan en intervalos de 3 minutos, que es el mínimo tiempo configurable, en caso de que alguno de los servidores no responda automáticamente mandará un mensaje de error a los operadores para que tomen las acciones pertinentes y restablezcan el servicio.

Una de las desventajas de este sistema es que sólo se enfoca en el servicio y no toma en cuenta el estado de la red ni de los servidores, y únicamente notificará cuando el servicio no esté disponible sin importar la causa. Por eso es necesario contar con otros sistemas que midan lo que *Firehunter* no puede.

3.3.2. Monitoreo del desempeño de los servidores

Para tener un mejor control de los servidores, es necesario monitorear sus recursos y medir su desempeño, lo que facilitará mantenerlos dentro de los

parámetros de operación y de ser necesario en un futuro, hacer algún crecimiento de los mismos antes de que se presenten problemas de capacidad.

Uninet recientemente adquirió el producto SMC (*Sun Management Center 3.0*, Centro de Administración Sun) (Ver Anexo E) para monitorear todos sus servidores Sun y dado que la plataforma seleccionada para los servidores de DNS también es Sun, se tomó la decisión de incluirlos en el mismo sistema.

SMC se instala en un servidor Unix que realizará poleos SNMP (*Simple Network Management Protocol*, Protocolo Simple de Administración de Red), a través del cual se medirá cada minuto la ocupación del procesador, de la memoria RAM y SWAP, y del disco duro, así como también verificará que los procesos de BIND estén funcionando. Si los procesos de BIND dejaran de funcionar por algún motivo, automáticamente SMC intentará levantarlos, de no poder hacerlo mandará una notificación a los operadores para que tomen las acciones pertinentes. En caso de que alguno de los siguientes umbrales se rebasara, también mandaría un mensaje de alarma:

- CPU 60% de ocupación
- Memoria RAM 75% de ocupación
- Memoria SWAP 20% de ocupación
- Disco Duro 90% por partición

3.3.3. Monitoreo de los enlaces

El último aspecto que es necesario medir de forma periódica, es la utilización de los enlaces asignados a cada servidor, para que en caso de ser necesario se hagan los trámites para aumentar el ancho de banda de los mismos. Para realizar estas mediciones se utilizará el software Vital Suite, el mismo software que se utilizó para hacer el análisis de tráfico.

Este programa también realiza poleos SNMP para obtener la ocupación de los enlaces, pero los hace sobre los enrutadores a los que están conectados los servidores. La frecuencia de poleo también será configurada a 3 minutos por ser el menor tiempo que maneja la aplicación. Este programa almacena los datos recolectados y en base a ellos genera pronósticos de tráfico a uno, tres y doce meses, para saber si es necesario aumentar el ancho de banda de los enlaces. Con esto podemos asegurarnos que por volumen de tráfico el servicio no se verá degradado.

3.4. Requerimientos del servicio

Una vez definida la arquitectura del servicio, es necesario determinar las características que van a soportarlo. Para ello se tomó en cuenta la ubicación física, después el modelo del equipo que cumpliera con las características de la

ubicación, la versión de BIND y finalmente el número de enlaces necesarios para el tráfico requerido.

3.4.1. Ubicación Física

Los equipos de datos de Uninet se encuentran instalados en centrales telefónicas de Telmex, por ser puntos estratégicos para la interconexión de Uninet con la red de Telmex. Además, estos edificios cuentan con energía eléctrica ininterrumpida, aire acondicionado, resistencia estructural contra sismos y control de acceso.

En cada región de Uninet existen dos nodos principales, cada uno ubicado en una central diferente. Para aprovechar la infraestructura ya instalada de los equipos de datos, los servidores de DNS deberán estar en los nodos principales de Uninet. Con esto se da mayor redundancia al servicio, ya que en caso de falla de alguno de ellos existe un respaldo en la misma región.

3.4.2. Definición de equipamiento

Para la selección de la plataforma de los servidores de DNS se tomaron en cuenta las más recientes versiones de BIND, que sólo son compatibles con el sistema operativo Solaris. Por cuestiones de soporte y actualización se decidió utilizar la última versión de Solaris, que es la 8.

El modelo del equipo se seleccionó de acuerdo a las características físicas de las centrales telefónicas, en donde la alimentación es de corriente directa y todos los equipos deben de estar en bastidores. Los servidores Sun que cumplen con estos requerimientos pertenecen a la familia Netra (Ver Anexo B). Para dar la mayor redundancia posible se eligió el modelo T1120 que tiene doble fuente de alimentación y discos duros en espejo¹⁷. El espacio mínimo requerido en disco duro es de 1.8 Gb, con lo cual se puede almacenar la información de los dominios y las bitácoras del sistema. De fábrica los discos duros actualmente tienen una capacidad mínima de 18 Gb, lo cual cumple con el requerimiento de almacenamiento.

La memoria RAM es uno de los parámetros más importantes en la selección de un servidor, pese a esto no existe una regla exacta que permita dimensionar la cantidad de memoria que debe de tener un servidor de DNS. Para mejorar el desempeño de los nuevos equipos es necesario que la RAM sea mayor a la ocupación máxima de la memoria RAM y SWAP de los antiguos servidores, por que el tiempo de respuesta de un servidor es menor cuando toda la información se encuentra en la RAM y no tiene que acceder al disco para leer la SWAP. Observando las gráficas de desempeño de los antiguos servidores vemos que la ocupación máxima de RAM es de 128Mb y la de SWAP de 122 Mb, por lo que la RAM requerida es de 250 Mb. De fábrica los equipos Netra T1120 tienen como

¹⁷ Configuración en la cual el servidor escribe información en ambos discos y en caso de falla en uno, el otro sigue trabajando sin afectar la operación del equipo.

mínimo 1 Gb, por lo cual se cumple el requerimiento mínimo de memoria. Es importante mencionar que estos equipos son escalables hasta 2 Gb.

3.4.3. Versiones de BIND

Para la definición de la versión de BIND se tomaron en cuenta dos puntos la recomendación del NIC y los reportes de seguridad de BIND.

BIND como producto ha pasado por muchas versiones y en cada una de ellas se corrigen errores de las versiones anteriores, y se incluyen nuevas funcionalidades. Cada administrador de DNS tiene la libertad de escoger la que más le convenga, sin embargo Uninet tomó la recomendación del NIC México, el cual sugirió la versión 8.2.5, por ser la versión más estable que ellos han probado y es la que actualmente tienen en operación.

En cuanto a la parte de seguridad la versión 8.2.5 corrige los huecos de seguridad que se muestran en la tabla 3.5.

NOMBRE DEL BUG	AFECTACIÓN	DESCRIPCIÓN
Naptr	Negación del Servicio	Provoca validación errónea de los registros PTR cuando la tabla de la zona es leída por el servidor, en ocasiones llega a tirar por completo un servidor de DNS.
Maxdname	Negación del Servicio	El uso de sprintf() ¹⁸ con datos tomados de la red puede provocar una sobrecarga de memoria, cuyo resultado se manifiesta en un comportamiento inesperado de la aplicación.
Solinger	Negación del Servicio	Es posible provocar remotamente pausas a BIND por intervalos superiores a los 120 segundos, utilizando para ello sesiones anormales de TCP.
Infoleak	Acceso a Información	Es posible hacer una consulta de resolución inversa que permita leer remotamente información del sistema.
Tsig	Acceso a sistema	Es posible provocar un desbordamiento de los buffers y como consecuencia se podría entrar al sistema.
Complain	Corrupción de Información	A través de consultas se puede desbordar los buffers y provocar corrupción de tablas.

Tabla 3.5. Lista de bugs corregidos en BIND 8.2.5. (Continúa)

¹⁸ Función para direccionar la salida de información a diferentes dispositivos.

Zxfr	Negación del Servicio	Por un error en el código del programa, la compresión de los archivos de las zonas puede provocar la caída del sistema.
Sigdiv0	Negación del Servicio	Por argumentos improprios durante la verificación de información, se puede generar una división entre 0, la cual provoca la caída del sistema.
Srv	Negación del Servicio	Por un bug en el manejo de la compresión de tablas de inversos, se puede crear una secuencia infinita.
Nxt	Acceso al sistema	Por un bug en el procesamiento de los registros NXT, se puede conseguir acceso al sistema.
Sig	Negación del Servicio	Por una validación inadecuada de los registros SIG, se puede provocar la caída del sistema.

Tabla 3.5. Lista de bugs corregidos en BIND 8.2.5.

3.4.4. Seguridad de servidores

Debido a que los servidores de DNS asignados al servicio de resolución de nombres en Uninet deben de localizarse dentro de Internet, es necesario considerar medidas de seguridad adicionales a las utilizadas tradicionalmente, como la combinación de user-password para evitar accesos no autorizados a los equipos. Estas medidas son las siguientes:

Listas de acceso: En las interfaces de los enrutadores que conectan a los servidores con el resto de Internet, deberán de aplicarse listas de acceso que limiten el paso de los paquetes IP's hacia los servidores, a sólo aquellos relacionados con el servicio de DNS y transferencias de zonas para los usuarios de Internet, y accesos del tipo TELNET y FTP desde las máquinas de administración exclusivamente, analizando los paquetes por dirección origen y destino y puerto destino. Los puertos autorizados por este tipo de listas de acceso serán las mostradas en la tabla 3.6.

Los servicios de capa cuatro de TCP/IP no utilizados en los servidores deberán permanecer deshabilitados, mediante comentarios en los archivos de configuración */etc/services* y */etc/inetd.conf*, de forma tal que los servidores estén imposibilitados para escuchar paquetes IP y solicitudes de puertos distintos a los utilizados por el servicio de DNS.

SERVICIO	PROTOCOLO	PUERTO	ORIGEN	DESTINO
Resolución de Nombres	TCP	Domain	Cualquiera	Todos los servidores
	UDP	Domain	Cualquiera	Todos los servidores
	UDP	Mayor a 1023	Cualquiera	Todos los servidores
Transferencia de Zona	IP	No aplica	200.33.150.193	Todos los servidores
Telnet	TCP	Telnet	200.33.150.69	Todos los servidores
FTP	TCP	FTP	200.33.150.69	Todos los servidores
	TCP	FTP data	200.33.150.69	Todos los servidores

Tabla 3.6. Lista de acceso para enrutadores.

En cada uno de los servidores deberá configurarse un *password* de *Dial-Up*, necesario para establecer una sesión de trabajo con el servidor. Este es un *password* de acceso adicional cuya característica principal radica en que no tiene un usuario correspondiente en la tabla */etc/passwd* del servidor. Es importante mencionar que la configuración de los mismos no acepta ninguna conexión *Dial-up* sobre los equipos.

Por cada servidor deberá instalarse el programa para protección *TCP/Wrapper*, y en consecuencia substituirse los programas tradicionales de TELNET y FTP por su equivalentes seguros. Adicionalmente deberá configurarse la máquina para aceptar sesiones de TELNET y FTP desde las direcciones IP de las máquinas de administración.

Por último, en el archivo *named.conf* de BIND se definirán listas que sólo permitan la transferencia de zonas entre los servidores de Uninet, no dejando que algún otro servidor pueda transferir tablas de configuración.

3.4.5. Dimensionamiento de enlaces

Una vez determinada la ubicación física de los servidores, es necesario dimensionar los enlaces que les darán conexión a la red. Para ello se determina el tráfico actual con la ecuación 3.1.¹⁹ El valor *Q* se tomó de las gráficas de peticiones analizadas en el capítulo II y para el cual el valor máximo de consultas por segundo promedio recibidas en un servidor fue de 665.

¹⁹ Ecuación tomada del libro DNS and BIND, Paul Albitz & Cricket Liu Editorial O'Reilly, pag. 200.

$$\begin{aligned} \text{Tráfico} &= Q \times 2 \times 100 \times 8 && \dots\dots\dots(3.1.) \\ \text{Tráfico} &= 665 \times 2 \times 100 \times 8 \\ \text{Tráfico} &= 1,064,000 \text{ bps} \end{aligned}$$

Donde:

- Q: Es el número de consultas promedio por segundo.
- 2: Es el factor utilizado para ajustar la consulta y la respuesta.
- 100: Es el tamaño promedio en bytes de una consulta de DNS.
- 8: Factor para convertir de bytes a bits.

Con base a pronósticos de las áreas comerciales de Uninet se espera que el tráfico actual crezca un 100% en los próximos dos años. Tomando este dato como referencia se espera un tráfico de 2.1 Mbps. Para satisfacer esta demanda de tráfico se requiere que cada servidor tenga dos enlaces E1 dedicados. Con estos enlaces se tendrá mayor redundancia en el servicio, ya que en caso de falla de alguno de ellos el otro soportará la operación sin afectar el servicio. Los enlaces E1 asignados a Monterrey y Guadalajara, se disminuyeron de tres a dos, pues la carga de trabajo en cada uno se balanceó al agregar otros servidores.

Con esto se concluye el diseño de la arquitectura del servicio, dando la solución técnica más eficiente para satisfacer los requerimientos del servicio. Por ser un proyecto de misión crítica para Uninet, el área de finanzas autorizó la compra de todos los recursos planteados en el diseño y que son necesarios para el éxito del proyecto.

CAPÍTULO IV

Implantación y puesta en operación

Este capítulo describe el proceso de instalación, pruebas y puesta en operación del servicio de DNS propuesto. La instalación es parte fundamental de la implantación, en la cual se necesitan realizar pruebas para verificar el correcto funcionamiento del servicio, garantizando con ello una puesta en operación sin errores.

4.1. Implantación

El proceso de implantación del servicio de DNS comprende: la instalación del software requerido para su operación, la configuración de los equipos como servidores de DNS, las restricciones de seguridad, las pruebas de funcionalidad que garanticen su correcto funcionamiento y finalmente la instalación de agentes para el monitoreo y la administración de los servidores.

4.1.1. Instalación de los servidores

La instalación inicia con el sistema operativo Solaris 8 en los nueve servidores, así como parches de *kernel*¹ esenciales para el buen funcionamiento de estos, después se instala el software BIND versión 8.2.5 compatible con Solaris 8, además de los agentes de *Firehunter* y SMC para poder realizar el monitoreo.

¹ El *kernel* es el programa que controla el *hardware* en un equipo Unix.

Finalmente para la seguridad se instala el programa *TCPWrappers* a fin de restringir la entrada a los sistemas, además de configurar una contraseña adicional para los usuarios que se conecten vía remota, llamada *dial-password*.

En la instalación del sistema operativo Solaris 8, es necesario definir diferentes parámetros y los más importantes para el correcto funcionamiento de los servidores se mencionan en los siguientes incisos.

a) Asignación de espacio

La distribución del espacio en disco para todos los servidores se configuró como se muestra en la tabla 4.1.

NOMBRE	ESPACIO EN (MB)	FUNCIÓN
/	3126	Partición de root
<i>Swap</i>	2001	Espacio reservado para memoria swap
<i>Overlap</i>	18000	Indica espacio total del disco
<i>/var</i>	2991	Partición para bitácoras
<i>/opt</i>	3832	Partición para la instalación de software
<i>/export/home</i>	2100	Partición para directorios de usuarios
<i>/var/named</i>	3202	Partición para archivos de BIND
<i>/unused</i>	10	Partición para realizar el disco en espejo

Tabla 4.1. Distribución del disco.

b) Configuración de espejo del disco duro

Para la configuración de discos duros en espejos se utilizó el programa *Disk Suite*, con el cual se señala que el segundo disco será una copia fiel del primero, este procedimiento es muy sencillo, sólo se instala *Disk Suite* que viene incluido con los discos de sistema operativo y mediante su interfaz gráfica se indica que disco va a ser el espejo del otro y su distribución.

c) Configuración de Red

En la configuración de red se definen la dirección IP, la máscara de red y el *gateway* a utilizar por cada equipo, las cuales se muestran en la tabla 4.2. Estos nombres, direcciones, máscaras y *gateway's* son definitivos y como algunos de ellos son los mismos que tienen los servidores en operación, todas las pruebas se realizan en una red aislada para evitar duplicidad de direcciones y nombres.

NOMBRE	DIRECCIÓN	MÁSCARA	GATEWAY
dnsadm-interno	200.33.150.193	255.255.255.192	200.33.150.198
nsgdl1	200.23.242.193	255.255.255.192	200.23.242.198
nsgdl2	200.23.242.201	255.255.255.192	200.23.242.206
nsmex1	200.33.146.193	255.255.255.192	200.33.146.198
nsmex2	200.33.146.201	255.255.255.192	200.33.146.206
nsmex3	200.33.146.209	255.255.255.192	200.33.146.214
nsmex4	200.33.146.217	255.255.255.192	200.33.146.219
nsmtly1	200.33.148.193	255.255.255.192	200.33.148.198
nsmtly2	200.33.148.201	255.255.255.192	200.33.148.206

Tabla 4.2. Configuración de la tarjeta de red.

d) Cuentas de usuarios

Para la creación de cuentas de usuarios se definieron dos perfiles dependiendo de su función, los cuales se muestran en la tabla 4.3. Estos dos perfiles nos ayudan a llevar un control de las actividades de cada usuario y a mantener la integridad de la información.

PERFIL	PERMISOS	No. CUENTAS
operador	Sólo podrán tener acceso a los archivos de configuración de BIND, como sólo lectura, y no tendrán acceso a los archivos de sistema operativo.	10
administrador	Se tendrá acceso a los archivos de configuración de BIND y sistema operativo con permisos de lectura y escritura	2

Tabla 4.3. Perfiles de usuarios.

e) Configuración de servicios

Como se definió en el diseño del servicio, se deshabilitarán todos los puertos que no se relacionan con el DNS ni con la administración de los servidores. Al instalar el sistema operativo se crea el archivo */etc/services* con los puertos básicos de comunicación, el formato del archivo maneja tres columnas: nombre del puerto, número del puerto/protocolo y comentario. Para deshabilitar los puertos sólo se inserta el signo # al inicio del renglón. Por lo que el archivo */etc/services* debe tener la siguiente información, dejando habilitados los puertos de *ftp*, *ftp-data*, *telnet*, *name* y *domain*:

```
#tcpmux      1/tcp
#echo        7/tcp
#echo        7/udp
#discard     9/tcp          sink null
```

```

#discard          9/udp          sink null
#sysstat         11/tcp          users
#daytime         13/tcp
#daytime         13/udp
#netstat         15/tcp
#chargen         19/udp          ttytst source
#chargen         19/udp          ttytst source
ftp-data         20/tcp
ftp              21/tcp
telnet           23/tcp
#smtp            25/tcp          mail
#time            37/udp          timserver
#time            37/udp          timserver
name             42/udp          nameserver
#whois           43/tcp          nicname           # usually to sri-nic
domain           53/udp
#bootps          67/udp          # BOOTP/DHCP server
#bootpc          68/udp          # BOOTP/DHCP client
#hostnames       101/tcp         # usually to sri-nic
#pop2            109/tcp        # Post Office Protocol-V2
#pop3            110/tcp        # Post Office Protocol-V3
#sunrpc          111/udp
#sunrpc          111/tcp
#imap            143/tcp
#ldap            389/tcp
#ldap            389/udp
#submission      587/tcp
#submission      587/udp
#ldaps           636/tcp
#ldaps           636/udp
#
# Host specific functions
#
#tftp            69/udp
#rje             77/tcp
#finger          79/tcp
#link            87/tcp          ttylink
#supdup          95/tcp
#iso-tsap        102/tcp
#x400            103/tcp          # ISO Mail
#x400-snd        104/tcp
#csnet-ns        105/tcp
#pop-2           109/tcp          # Post Office
#uucp-path       117/tcp
#nntp            119/tcp          usenet
#ntp             123/tcp
#ntp             123/udp
#netbios-ns      137/tcp
#netbios-ns      137/udp
#netbios-dgm     138/tcp
#netbios-dgm     138/udp
#netbios-dgm     139/tcp
#netbios-ssn     139/udp
#netbios-ssn     139/udp
#NEWS            144/tcp          news

```

```

#slp                427/tcp          slp                # Service Location P-V2
#slp                427/udp          slp                # Service Location P-V2
#mobile-ip          434/udp          mobile-ip          # Mobile-IP
#cvc_hostd          442/tcp          # Network Console
#
# UNIX specific services
#
# these are NOT officially assigned
#
#exec               512/tcp
#login              513/tcp
#shell              514/tcp          cmd                # no passwords used
#printer            515/tcp          spooler            # line printer spooler
#courier            530/tcp          rpc                # experimental
#uucp               540/tcp          uucpd              # uucp daemon
#biff               512/udp          comsat
#who                513/udp          whod
#syslog             514/udp
#talk               517/udp
#route              520/udp          router routed
#ripng              521/udp
#klogin             543/tcp          # Kerberos authenticated
#kshell             544/tcp          cmd                # Kerberos authenticated
#new-rwho           550/udp          new-who            # experimental
#rmonitor           560/udp          rmonitord          # experimental
#monitor            561/udp          # experimental
#pcserver           600/tcp          # ECD Integrated PC board
#sun-dr             665/tcp          # Remote Dynamic Reconfig
#kerberos-adm       749/tcp          # Kerberos V5 Admin
#kerberos-adm       749/udp          # Kerberos V5 Admin
#kerberos           750/udp          kdc                # Kerberos key server
#kerberos           750/tcp          kdc                # Kerberos key server
#krb5_prop          754/tcp          # Kerberos V5 KDC
#ufsd               1008/tcp          ufsd               # UFS-aware server
#ufsd               1008/udp          ufsd
#cvc                1495/tcp          # Network Console
#ingreslock         1524/tcp
#www-ldap-gw        1760/tcp          # HTTP to LDAP gateway
#www-ldap-gw        1760/udp          # HTTP to LDAP gateway
#listen             1766/tcp          # System V listener port
#nfsd               2049/udp          nfs                # NFS server daemon
#nfsd               2049/tcp          nfs                # NFS server daemon
#eklogin            2105/tcp          # Kerberos encrypted
#lockd              4045/udp          # NFS lock daemon/manager
#lockd              4045/tcp
#dtspd              6112/tcp          # CDE subprocess control
#fs                 7100/tcp          # Font server
qip_ibm             1093/tcp          #VitalQIP IBM DHCP Monitor daemon
qip_sec             1095/tcp          #VitalQIP Secondary management
qip_ctl             1096/tcp          #VitalQIP Control daemon
qip_msgd            2468/udp          #VitalQIP DHCP Remote Message
qip_msgd            2468/tcp          #VitalQIP DHCP Remote Message
qip_dns             3119/tcp          #VitalQIP DNS Service
qip_qdhcp           2490/tcp          #VitalQIP DHCP Lease Update
qdhcp_failover     647/udp          #VitalQIP DHCP Failover Service
qip_login           2366/tcp          #VitalQIP Login Service
qip_audup           2765/tcp          #VitalQIP Audit Update Service

```

```
qip_dhcpext      5678/tcp          #VitalAccess DHCP Extension
qip_acmupdate    3120/tcp          #VitalAccess DHCP Update Service
```

f) Instalación y configuración *TCPWrappers*

Para reforzar la seguridad del sistema se instala *TCPWrappers*. Este proceso es relativamente sencillo, se obtiene de Internet un archivo ejecutable llamado *tcpd* que controlará todas las conexiones TELNET y FTP mediante listas de direcciones permitidas. Este archivo se instala en el directorio */usr/sbin* y para que tome efecto hay que modificar el archivo */etc/inetd.conf* con las siguientes líneas:

```
# Configuration file for inetd(1M).  See inetd.conf(4).
#
#
# Ftp and telnet are standard Internet services.
#

ftp  stream      tcp  nowait root /usr/local/sbin/tcpd  in.ftpd
telnet stream      tcp  nowait root /usr/local/sbin/tcpd  in.telnetd
```

Las listas de acceso permitirán que solo los administradores y operadores responsables del servicio y de los servidores se puedan conectar por TELNET y FTP, para ello hay que crear dos archivos: */etc/hosts.allow* y */etc/hosts.deny*, a continuación se presenta el contenido de */etc/hosts.allow*:

```
in.telnetd,in.ftpd :
200.98.135.24,200.98.135.28,200.98.135.29,200.98.135.30,
200.98.135.34,200.98.135.36
```

El archivo */etc/hosts.deny* debe llevar la siguiente línea para restringir todos los puertos que no se listan en él */etc/hosts.allow*:

```
ALL:ALL
```

Con estas últimas modificaciones se considera que el sistema es más seguro, minimizando la posibilidad de recibir un ataque.

4.1.2. Instalación y configuración de QIP

El programa QIP se instala sobre una base de datos Oracle. Para garantizar su buen funcionamiento tiene que dimensionarse conforme al número de direcciones IP's, subredes y dominios que se manejarán. El proveedor recomienda utilizar la fórmula 4.1 para calcular el tamaño más adecuado de tres archivos (*qipdata*, *qipindex* y *qiptemp*), necesarios para la operación de Oracle:

$$\underline{x} = (\# \text{ Direcciones IP} \times 2500) + (\text{Subredes} \times 500) + (\# \text{ Dominios} \times 1300) \dots (4.1.)$$

$$y = (\underline{x} / (1024 \times 1024))$$

$qipdata \text{ (Mb)} = y \times 4$
 $qipindex \text{ (Mb)} = qipdata / 3$

$qiptemp \text{ (Mb)} = qipdata / 4$

Direcciones IP = 500000, Subredes = 12789, Dominios = 700

$x = 1257304500$ $y = 1199.06$

$qipdata = 4796.23 \text{ Mb}$, $qipindex = 1598.74 \text{ Mb}$, $qiptemp = 1199.05 \text{ Mb}$

Después de realizar el cálculo y redondear el número (dado que los *bytes* son números enteros), obtenemos los tamaños que cada archivo requiere.

$qipdata = 4797 \text{ Mb}$
 $qipindex = 1599 \text{ Mb}$
 $qiptemp = 1200 \text{ Mb}$

Cada uno de estos archivos tiene funciones especiales dentro de la estructura interna de Oracle; *qipdata* se utiliza para guardar todos los registros, *qipindex* para realizar acciones propias de la base de datos y *qiptemp* para guardar datos temporalmente mientras se realizan procesos con la base de datos.

El siguiente paso consiste en la instalación de QIP, que solamente requiere de la ejecución de un comando desde el CD de instalación. Una vez instalado QIP, se debe iniciar la carga de datos, esta labor requiere tener un inventario confiable de las redes, subredes y dominios que son responsabilidad de Uninet.

El fabricante de QIP recomienda hacer uso de algunas utilerías diseñadas para importar información masivamente. Cada una de ellas lee los datos de un archivo de texto en donde la información de cada registro está separada por comas, es importante mencionar que no todos los campos son obligatorios. Está es la secuencia recomendada para la carga de datos:

- 1) Servidores de DNS
- 2) Dominios
- 3) Redes
- 4) Subredes
- 5) Direcciones IP con sus nombres

La carga de los servidores se hace vía interface gráfica, puesto que sólo son 9 servidores y los pasos son muy sencillos; como se muestra en la figura 4.1. Primero seleccionamos del menú *Infrastructure* la opción *Server* en la que se elige *Add New Server(s)* y se llenan los campos de los parámetros de configuración.

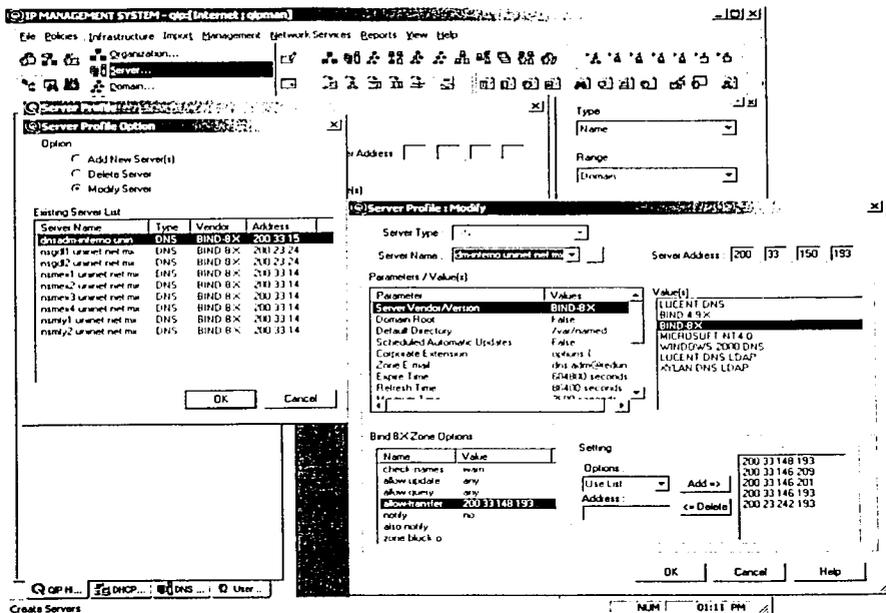


Figura 4.1. Carga de servidores.

Para importar redes se ejecuta el siguiente comando en la línea de comando en el servidor de QIP.

```
# internetwork -f archivo -s servidor -u usuario -p contraseña
```

y el formato del archivo de donde se van a obtener los datos es:

```
172.17.0.0,16,root@nsroot.qip.com,QIPNET-1,,10,2,nsroot.qip.com
172.18.0.0,16,root@nsroot.qip.com,QIPNET-2,,10,2,nsroot.qip.com
172.19.0.0,16,root@nsroot.qip.com,QIPNET-3,,10,2,nsroot.qip.com
192.168.0.0,16,root@nsroot.qip.com,QIP-CIDR-NET-4,Y,10,2,nsroot.qip.com
10.0.0.0,8,root@nsroot.qip.com,QIPNET-5,,,,nsroot.qip.com
```

Donde el orden de los campos es:

1. Dirección de la red
2. Longitud de la máscara de red
3. Dirección de correo electrónico
4. Nombre de la red

5. CIDR
6. Porcentaje de ocupación
7. Tipo de advertencias que se van a recibir
8. Nombre del servidor DNS primario

Para importar subredes se ejecuta el siguiente comando:

```
# entersubnet -f archivo -s servidor -u usuario -p contraseña
```

y el formato del archivo es:

```
10.1.1.0,255.255.255.0,10.0.0.0,Subnet-1,,,,,,,,,qip.com
192.168.0.0,255.255.192.0,192.168.0.0,Subnet-
2,,,,,,,,,qip.com,north.qip.com
192.168.64.0,255.255.192.0,192.168.0.0,Subnet-
3,,,,,,,,,qip.com,south.qip.com
192.168.128.0,255.255.192.0,192.168.0.0,Subnet-
4,,,,,,,,,qip.com,east.qip.com
192.168.192.0,255.255.192.0,192.168.0.0,Subnet-
5,,,,,,,,,qip.com,west.qip.com
172.17.1.0,255.255.255.224,172.17.0.0,Subnet-
6,,,,,,,,,qip.com,north.qip.com,
south.qip.com,east.qip.com,west.qip.com
172.17.1.32,255.255.255.224,172.17.0.0,Subnet-7,,,,,,,,,qip.com
172.17.1.64,255.255.255.224,172.17.0.0,Subnet-8,,,,,,,,,qip.com
172.17.1.96,255.255.255.224,172.17.0.0,Subnet-9,,,,,,,,,qip.com
172.17.1.128,255.255.255.224,172.17.0.0,Subnet-10,,,,,,,,,qip.com
172.17.1.160,255.255.255.224,172.17.0.0,Subnet-11,,,,,,,,,qip.com
172.17.1.192,255.255.255.224,172.17.0.0,Subnet-12,,,,,,,,,qip.com
172.17.1.224,255.255.255.224,172.17.0.0,Subnet-13,,,,,,,,,qip.com
```

Donde el orden de los campos es:

1. Dirección de la subred
2. Máscara de la subred
3. Dirección de la red
4. Nombre de la subred
5. Nombre de la aplicación primaria
6. Apellido del contacto
7. Nombre del contacto
8. Calle donde se localiza 1
9. Calle donde se localiza 2
10. Ciudad
11. Estado
12. Código postal
13. País
14. Tipo de *hardware*
15. Si es servidor de TFTP
16. Localización de facturación
17. Grupo usuarios del servicio de facturación

18. Dominio

Para importar objetos se ejecuta el siguiente comando:

```
# entersimpleobj -f archivo -s servidor -u usuario -p contraseña
```

y el formato de archivo es:

```
198.200.234.2,tst2,198.200.234.0,,usa.world.com,Workstation,,,,,3,,,,wo  
rk-ny,station2
```

Donde el orden de los campos es:

1. Dirección IP del objeto
2. Nombre del objeto
3. Dirección de la subred
4. Servicio de DHCP
5. Nombre de Dominio
6. Descripción de tipo de objeto
7. Comentario
8. Dirección MAC
9. Apellido del contacto
10. Nombre del contacto
11. No. Telefónico del contacto
12. Configuración dinámica
13. Tipo de *hardware*
14. Manufactura
15. Tipo de modelo
16. Tiempo de vida
17. Alias
18. Lista de usuarios
19. Clase de usuarios
20. Servicio de facturación
21. Nombre del servicio de facturación
22. Clase de objetos de facturación

Una vez terminada la carga de los datos, se compara la información de QIP con la contenida en los servidores en operación hasta que no falte ningún dato y los registros estén libres de errores. Teniendo todos los datos correctos y configurados en QIP, se generan los archivos de configuración de BIND. Estos archivos están libres de errores de sintaxis y las actualizaciones de información son automáticas y definidas por el operador. Para realizar cambios o agregar nuevos registros es necesario contar con acceso a esta herramienta.

Una vez listos los servidores, se inicia la fase de pruebas para asegurar la adecuada operación de QIP.

4.1.3. Pruebas del funcionamiento de QIP

Las funciones para las que fue configurado QIP son para mantener la información íntegra del direccionamiento de Uninet y la generación de archivos de configuración de BIND, por lo que antes de realizar la puesta en operación necesitamos corroborar el buen funcionamiento de estas tareas, para lo cual se aplicaron las siguientes pruebas:

La primera prueba consistió en generar los archivos de configuración de BIND, revisando que no faltara ninguno y verificando la bitácora del sistema operativo `/var/adm/messages`, la finalidad fue ver si existían mensajes de error en la sintaxis de los archivos obteniendo 200 líneas como las siguientes:

```
May 12 16:28:30 dnsadm-interno named: [daemon error] master zone
"223.223.148.in.addr.arpa" (IN) rejectd due to errors
```

```
May 12 16:28:30 dnsadm-interno named: [daemon warning] host name
"administración-dslam-uninet.net.mx" (owner "125.247.223.148.in-
addr.arpa") IN (primary) invalid -proceeding
```

Estos mensajes indican que la sintaxis no es correcta, como por ejemplo terminar con un "-" el nombre del *host*, por lo que se corrigió la sintaxis y se generaron los archivos nuevamente para revisar la bitácora, que en esta ocasión no presentó ninguno de estos mensajes, con lo que se da por concluida la prueba.

La segunda prueba fue para verificar que se incrementaran los números de serie de las tablas al ser modificadas en su contenido. Para lo cual se modificó el contenido de diez tablas tres veces, generándose los archivos de configuración; en cada uno de ellos se revisaron los números de serie para verificar que se hubieran incrementado. A continuación se muestra un ejemplo de una de estas tablas:

SOA de una tabla generada la primera vez, con el número de serie 1.

```
;*****
; Prefix zone extension
;*****
;
$TTL 3600
;=====
; Reverse Addresses (PTR Records) for zone: 150.33.200.in-addr.arpa
;=====
@           IN           SOA           dnsadm-interno.uninet.net.mx. adm-
dns.reduno.com.mx. (
1           ; Serial No.
14400      ; Refresh
3600       ; Retry
604800     ; Expire
3600 )     ; Minimum
```

SOA de la tabla generada la segunda vez con el número de serie 2.

```
;*****
; Prefix zone extension
;*****
;
$TTL 3600
;=====
; Reverse Addresses (PTR Records) for zone: 150.33.200.in-addr.arpa
;=====
@      IN      SOA      dns-interno.uninet.net.mx. adm-dns.reduno.com.mx.
(
                2      ; Serial No.
                14400 ; Refresh
                3600  ; Retry
                604800 ; Expire
                3600  ) ; Minimum
;
                IN     NS     dns.uninet.net.mx.
                IN     NS     nsmex4.uninet.net.mx.
                IN     NS     nsmex1.uninet.net.mx.
                IN     NS     nsgd11.uninet.net.mx.
                IN     NS     nsmt1.uninet.net.mx.
```

SOA de la tabla generada la tercera vez con el número de serie 3.

```
;*****
; Prefix zone extension
;*****
;
$TTL 3600
;=====
; Reverse Addresses (PTR Records) for zone: 150.33.200.in-addr.arpa
;=====
@      IN      SOA      dns-interno.uninet.net.mx. adm-dns.reduno.com.mx.
(
                3      ; Serial No.
                14400 ; Refresh
                3600  ; Retry
                604800 ; Expire
                3600  ) ; Minimum
;
                IN     NS     dns.uninet.net.mx.
                IN     NS     nsmex4.uninet.net.mx.
                IN     NS     nsmex1.uninet.net.mx.
                IN     NS     nsgd11.uninet.net.mx.
                IN     NS     nsmt1.uninet.net.mx.
```

Podemos observar que se realiza correctamente el incremento de los números de serie de los archivos, esto es fundamental para las actualizaciones de la información.

La tercera prueba también está relacionada con la actualización de la información, pero esta vez mediante la generación automática de los archivos en horarios establecidos por el administrador. En QIP se configuró la generación diaria de los

archivos en el servidor `dnsadm-interno` en el directorio `/var/named` a las 14:00 y 19:00 horas. Para verificar esta funcionalidad se revisó que en este directorio estuvieran los archivos con las fechas correspondientes a cada actualización, las siguientes líneas son parte del listado de los archivos del directorio y con ellas se pudo verificar el buen funcionamiento de la generación automática de tablas.

Primera generación:

```
-rw-r--r-- 1 root other 13816 May 12 14:01 db.207.248.158
-rw-r--r-- 1 root other 13693531 May 12 14:01 db.prodigy.net.mx
-rw-r--r-- 1 root other 533135 May 12 14:01 db.uninet.net.mx
-rw-r--r-- 1 root other 13986 May 12 14:01 db.200.33.250
```

Segunda generación:

```
-rw-r--r-- 1 root other 13816 May 12 19:01 db.207.248.158
-rw-r--r-- 1 root other 13693531 May 12 19:01 db.prodigy.net.mx
-rw-r--r-- 1 root other 533135 May 12 19:01 db.uninet.net.mx
-rw-r--r-- 1 root other 13986 May 12 19:01 db.200.33.250
```

Estas tres pruebas fueron suficientes para verificar el correcto funcionamiento de QIP.

Una vez efectuadas estas pruebas, el siguiente paso es configurar BIND en todos los servidores con sus parámetros.

4.1.4. Configuración de BIND

Con los archivos generados por QIP en cada uno de los servidores, se necesita completar la configuración de estos equipos como servidores de DNS, configurando los siguientes archivos: `/etc/hosts`, `/etc/resolv.conf` y `/etc/nsswitch.conf`.

El archivo `/etc/hosts` contiene el nombre y la dirección IP del propio equipo para que el sistema operativo pueda reconocer su nombre. Esto facilita la operación de los servidores de DNS para no depender de BIND y darle la libertad al administrador de detener la resolución de nombres. También se pueden agregar nombres y direcciones de otros equipos que se necesiten reconocer por nombre sin que estén en BIND. A continuación se presentan los archivos de los nueve servidores:

`dnsadm-interno`

```
#
# Internet host table
#
127.0.0.1      localhost
200.33.150.193 dnsadm-interno dnsadm-interno.uninet.net.mx  loghost
200.33.150.69  hmexms03       hmexms03.uninet.net.mx
200.33.150.28  smc.uninet.net.mx
148.235.172.132 customer-148-235-172-132.uninet.net.mx
```

```
148.235.172.135 customer-148-235-172-135.uninet.net.mx
148.235.172.142 customer-148-235-172-142.uninet.net.mx
200.33.150.30 jvalenzu.uninet.net.mx
```

nsmex1

```
#
# Internet host table
#
127.0.0.1 localhost
200.33.146.193 nsmex1 nsmex1.uninet.net.mx loghost
200.33.150.69 hmexms03 hmexms03.uninet.net.mx
200.33.150.28 smc.uninet.net.mx
148.235.172.132 customer-148-235-172-132.uninet.net.mx
148.235.172.135 customer-148-235-172-135.uninet.net.mx
148.235.172.142 customer-148-235-172-142.uninet.net.mx
200.33.150.31 jvalenzu.uninet.net.mx
```

nsmex2

```
#
# Internet host table
#
127.0.0.1 localhost
200.33.146.201 nsmex2 nsmex2.uninet.net.mx loghost
200.33.150.69 hmexms03 hmexms03.uninet.net.mx
200.33.150.28 smc.uninet.net.mx
148.235.172.132 customer-148-235-172-132.uninet.net.mx
148.235.172.135 customer-148-235-172-135.uninet.net.mx
148.235.172.142 customer-148-235-172-142.uninet.net.mx
200.33.150.31 jvalenzu.uninet.net.mx
```

nsmex3

```
#
# Internet host table
#
127.0.0.1 localhost
200.33.146.209 nsmex3 loghost nsmex3.uninet.net.mx
200.33.150.69 hmexms03 hmexms03.uninet.net.mx
200.33.150.28 smc.uninet.net.mx
148.235.172.132 customer-148-235-172-132.uninet.net.mx
148.235.172.135 customer-148-235-172-135.uninet.net.mx
148.235.172.142 customer-148-235-172-142.uninet.net.mx
200.33.150.31 jvalenzu.uninet.net.mx
```

nsmex4

```
#
# Internet host table
#
127.0.0.1 localhost
200.33.146.217 nsmex4 nsmex4.uninet.net.mx loghost
200.33.150.69 hmexms03 hmexms03.uninet.net.mx
200.33.150.28 smc.uninet.net.mx
148.235.172.132 customer-148-235-172-132.uninet.net.mx
148.235.172.135 customer-148-235-172-135.uninet.net.mx
148.235.172.142 customer-148-235-172-142.uninet.net.mx
200.33.150.31 jvalenzu.uninet.net.mx
```

nsgdl1

```
#  
# Internet host table  
#  
127.0.0.1      localhost  
200.23.242.193 nsgdl1 nsgdl1.uninet.net.mx  loghost  
200.33.150.69  hmexms03 hmexms03.uninet.net.mx  
200.33.150.28  smc.uninet.net.mx  
148.235.172.132 customer-148-235-172-132.uninet.net.mx  
148.235.172.135 customer-148-235-172-135.uninet.net.mx  
148.235.172.142 customer-148-235-172-142.uninet.net.mx  
200.33.150.31  jvalenzu.uninet.net.mx
```

nsgdl2

```
#  
# Internet host table  
#  
127.0.0.1      localhost  
200.23.242.201 nsgdl2 nsgdl2.uninet.net.mx  loghost  
200.33.150.69  hmexms03 hmexms03.uninet.net.mx  
200.33.150.28  smc.uninet.net.mx  
148.235.172.132 customer-148-235-172-132.uninet.net.mx  
148.235.172.135 customer-148-235-172-135.uninet.net.mx  
148.235.172.142 customer-148-235-172-142.uninet.net.mx  
200.33.150.31  jvalenzu.uninet.net.mx
```

nsmt1

```
#  
# Internet host table  
#  
127.0.0.1      localhost  
200.33.148.193 nsmt1 nsmt1.uninet.net.mx  loghost  
200.33.150.69  hmexms03 hmexms03.uninet.net.mx  
200.33.150.28  smc.uninet.net.mx  
148.235.172.132 customer-148-235-172-132.uninet.net.mx  
148.235.172.135 customer-148-235-172-135.uninet.net.mx  
148.235.172.142 customer-148-235-172-142.uninet.net.mx  
200.33.150.31  jvalenzu.uninet.net.mx
```

nsmt2

```
#  
# Internet host table  
#  
127.0.0.1      localhost  
200.33.148.201 nsmt2 nsmt2.uninet.net.mx  loghost  
200.33.150.69  hmexms03 hmexms03.uninet.net.mx  
200.33.150.28  smc.uninet.net.mx  
148.235.172.132 customer-148-235-172-132.uninet.net.mx  
148.235.172.135 customer-148-235-172-135.uninet.net.mx  
148.235.172.142 customer-148-235-172-142.uninet.net.mx  
200.33.150.31  jvalenzu.uninet.net.mx
```

En el archivo `/etc/resolv.conf` se define el dominio al que pertenece el equipo y el servidor de DNS al que debe preguntar cuando requiera hacer una petición, se

recomienda poner por lo menos dos servidores para tener redundancia. A continuación se presentan los archivos de cada uno de los servidores:

dnsadm-interno

```
domain          uninet.net.mx
nameserver      200.33.150.193
nameserver      200.33.146.193
```

nsmex1

```
domain          uninet.net.mx
nameserver      200.33.146.193
nameserver      200.33.146.201
```

nsmex2

```
domain          uninet.net.mx
nameserver      200.33.146.201
nameserver      200.33.146.209
```

nsmex3

```
domain          uninet.net.mx
nameserver      200.33.146.209
nameserver      200.33.146.217
```

nsmex4

```
domain          uninet.net.mx
nameserver      200.33.146.217
nameserver      200.23.242.193
```

nsgdl1

```
domain          uninet.net.mx
nameserver      200.23.242.193
nameserver      200.23.242.201
```

nsgdl2

```
domain          uninet.net.mx
nameserver      200.23.242.201
nameserver      200.33.148.193
```

nsmtly1

```
domain          uninet.net.mx
nameserver      200.33.148.193
nameserver      200.33.148.201
```

TELIS CON
FALLA DE ORIGEN

nsmtty2

```
domain          uninet.net.mx
nameserver      200.33.148.201
nameserver      200.23.242.193
```

El archivo `/etc/nsswitch.conf` contiene la configuración del orden de búsqueda de nombres en un sistema Unix, es decir, si se busca primero una respuesta en el servidor de DNS y no la encuentra, busca en el archivo `/etc/hosts`, en caso de no tenerla, el servidor manda un mensaje de error. Todos los servidores deberán tener el archivo con la siguiente información:

```
#
# /etc/nsswitch.files:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

passwd:        files
group:         files
hosts:         dns [NOTFOUND=continue]      files [NOTFOUND=return]
ipnodes:       files
networks:      files
protocols:    files
rpc:           files
ethers:        files
netmasks:     files
bootparams:    files
publickey:    files
# At present there isn't a 'files' backend for netgroup; the system will
# figure it out pretty quickly, and won't use netgroups at all.
netgroup:      files
automount:     files
aliases:       files
services:      files
sendmailvars:  files
printers:      user files

auth_attr:     files
prof_attr:     files
project:       files
```

A partir de este momento se pueden iniciar las pruebas de resolución para asegurar la confiabilidad de la información del sistema.

4.1.5. Pruebas de resolución

Al contar con los archivos de configuración de QIP probados y tener los servidores con todo su software instalado, se inician las dos pruebas de resolución.

La primera prueba consiste en ejecutar el proceso *named* en todos los servidores de DNS y verificar que se encuentre dentro de los procesos activos del sistema, esta verificación se realiza a través del comando *ps -f |grep named*, el cual debe dar el siguiente resultado:

```
PID TTY CMD
3579 pts/1 /usr/local/sbin/named
```

Si no se obtiene esta respuesta, es indicativo de que el proceso de BIND no se pudo ejecutar y se deberá revisar su instalación.

El siguiente paso de la primera prueba consiste en observar los mensajes de las bitácoras del sistema operativo en el archivo */var/adm/messages*, para verificar que no existan mensajes de error de BIND, como pueden ser errores de sintaxis o de configuración del servidor. En el caso de que no exista ningún error en la configuración aparecerá un solo mensaje como el que se muestra a continuación:

```
May 9 18:33:41 named: starting (/etc/named.conf) named 8.2.5.
Ready to answer.
```

Este mensaje indica la fecha y hora en que BIND inició su operación y está listo para resolver peticiones.

Esta prueba se realizó en todos los servidores y sin presentar problema al ejecutar el comando *ps* o al verificar las bitácoras, con lo que se comprueba el adecuado funcionamiento del proceso *named*.

La segunda prueba está dedicada a verificar que los servidores resuelvan consultas de sus propias tablas. Para esto se hicieron dos programas, uno que pregunte por todos y cada uno de los nombres para verificar su resolución normal, y otro que verifique su resolución inversa. Para poder ejecutarlos es necesario contar con un archivo de texto que contenga todo el direccionamiento de Uninet dividido en subredes clase C, para que de cada una de ellas se revise la resolución de los *hosts* que las componen. El resultado de los programas se manda a otro archivo y nos indica el número de *hosts* que no tuvieron resolución de cada subred. Estos programas se detallan en el anexo F.

De la ejecución del primer programa se detectaron 300 subredes sin resolución normal, las cuales fueron revisadas en QIP encontrando que no habían sido importadas correctamente, por lo que se volvió a realizar el proceso de importación y se ejecutó nuevamente el programa para comprobar que las correcciones hayan sido hechas. Después de revisar el archivo de resultados no se presentaron subredes sin resolución. En las siguientes líneas se muestra un extracto del archivo de resultados.

```
...
REVISAR 148.233.41 (254 IP's No Resueltas)
OK 148.233.42
```

REVISAR 148.233.43 (254 IP's No Resueltas)
OK 148.233.44
OK 148.233.45
OK 148.233.46
OK 148.233.48
OK 148.233.49
OK 148.233.50
OK 148.233.51
OK 148.233.52
OK 148.233.53
OK 148.233.54
OK 148.233.55
...

En el segundo programa se detectaron 364 subredes sin resolución inversa, las cuales fueron revisadas en QIP, encontrando que una vez más no habían sido importadas correctamente, realizando nuevamente la importación de estos datos para ejecutar el programa otra vez. Después de revisar el archivo de resultados no se presentaron subredes sin resolución. En las siguientes líneas se muestra un extracto del archivo de resultados.

...
OK 148.235.16
OK 148.235.17
OK 148.235.18
OK 148.235.19
OK 148.235.20
OK 148.235.21
OK 148.235.22
OK 148.235.23
OK 148.235.24
OK 148.235.25
OK 148.235.26
OK 148.235.27
OK 148.235.28
OK 148.235.29
OK 148.235.30
OK 148.235.31
...

Con estos resultados se asegura la correcta resolución de todas la redes en todos los servidores y con ello poder continuar con las pruebas de actualización de datos entre servidores.

4.1.6. Pruebas de transferencia

Una vez que no se tienen problemas en la resolución de las redes de Uninet, se tiene que comprobar la correcta actualización de la información que se realiza con las transferencias de BIND del servidor administrativo a los servidores secundarios.

Para las pruebas se necesita configurar en una red aislada el servidor administrativo como servidor primario de todas las redes y dominios, es decir, este equipo debe contar con el archivo *named.conf* y las tablas de resolución normales e inversas, mientras que los otros servidores deben de configurarse como secundarios y sólo contienen los archivos *named.conf*.

Después de configurar los servidores se ejecuta el proceso *named* en todos los equipos y se verifica en el directorio */var/named* de los servidores secundarios, que se hayan transferido las tablas. A continuación se muestra una parte del listado del directorio, al observar los nombres y las fechas de los archivos se comprueba que se están realizando las transferencias.

```
-rw-r--r-- 1 root      other      12845 May  9 12:07 db.148.223.236
-rw-r--r-- 1 root      other       7893 May  9 12:07 db.148.223.237
-rw-r--r-- 1 root      other      12709 May  9 12:07 db.148.223.238
-rw-r--r-- 1 root      other      14247 May  9 12:07 db.148.223.239
-rw-r--r-- 1 root      other      12669 May  9 12:07 db.148.223.24
-rw-r--r-- 1 root      other      7855 May  9 12:07 db.148.223.240
-rw-r--r-- 1 root      other      12916 May  9 12:07 db.148.223.241
-rw-r--r-- 1 root      other      13677 May  9 12:07 db.148.223.242
-rw-r--r-- 1 root      other       8038 May  9 12:07 db.148.223.243
```

Otra forma de verificar que se realizan transferencias en los servidores secundarios, es con el comando *ps -flgrep named-xfer*, el cual muestra todos las transferencias que se están realizando.

```
17273 ?          /usr/local/sbin/named-xfer -z tm3
17251 ?          /usr/local/sbin/named-xfer -z fb
17303 ?          /usr/local/sbin/named-xfer -z la
17271 ?          /usr/local/sbin/named-xfer -z pl-
17306 ?          /usr/local/sbin/named-xfer -z qu
17305 ?          /usr/local/sbin/named-xfer -z av
...
```

Esta prueba comprobó que los servidores secundarios son capaces de transferir tablas nuevas, pero no garantiza la actualización de las mismas cuando hayan modificaciones, por este motivo es necesario realizar una prueba adicional. La prueba consiste en modificar el número de serie de algunas tablas del servidor primario y observar su actualización en los secundarios, lo cual se deberá llevar a cabo en un intervalo de tiempo máximo definido por el parámetro *refresh*.

En diez tablas contenidas en el servidor primario se les incrementó el número de serie de 4 a 5, dado que el parámetro de *refresh* está configurado en 14400 segundos, los ocho servidores secundarios realizaron las transferencias en un lapso máximo de cuatros horas. A continuación se muestra una parte de una tabla que se modificó y de la transferencia.

Tabla en el servidor primario

```
;*****
; Prefix zone extension
```

```

;*****
;
$TTL 3600
;=====
; Reverse Addresses (PTR Records) for zone: 150.33.200.in-addr.arpa
;=====
@      IN      SOA      dns-interno.uninet.net.mx. adm-dns.reduno.com.mx.
(
    5          ; Serial No.
    14400     ; Refresh
    3600      ; Retry
    604800    ; Expire
    3600     ) ; Minimum

;
                                IN      NS      dns.uninet.net.mx.
                                IN      NS      nsmex4.uninet.net.mx.
                                IN      NS      nsmex1.uninet.net.mx.
                                IN      NS      nsgd11.uninet.net.mx.
                                IN      NS      nsmtyl1.uninet.net.mx.

```

Tabla en el servidor secundario después de la transferencia

```

; BIND version named 8.2.5-REL Wed Dec 12 11:40:47 CST 2001
; BIND version root@nsqd12:/var/tmp/bind/src/bin/named
; zone '150.33.200.in-addr.arpa' last serial 4
; from 200.33.150.193:53 (local 200.33.146.217) using AXFR at Sat Jun 8
22:46:1
4 2002
$ORIGIN 33.200.in-addr.arpa.
150 3600 IN      SOA      dns.uninet.net.mx. adm-dns.reduno.com
.mx. (
    5 14400 3600 604800 3600 )
    3600 IN      NS      dns.uninet.net.mx.
    3600 IN      NS      nsmex4.uninet.net.mx.
    3600 IN      NS      nsmex1.uninet.net.mx.
    3600 IN      NS      nsqd11.uninet.net.mx.
    3600 IN      NS      nsmtyl1.uninet.net.mx.
$ORIGIN 150.33.200.in-addr.arpa.

```

Esta prueba confirma que las transferencias de BIND no tienen ningún problema en su funcionamiento y con base en los resultados de las pruebas anteriores se concluye que el sistema está listo para entrar en operación.

4.2. Puesta en operación

Con la terminación de la instalación, configuración y pruebas de los nuevos componentes del servicio de DNS se deben realizar todos los preparativos para ponerlos en operación, entre ellos la adecuación de sitios y la estrategia de migración, para este último se debe tener una especial atención para tener la menor afectación en la operación de los servidores actuales y antes de dar por finalizada la puesta en operación se realizan pruebas del servicio para asegurar su correcto funcionamiento.

4.2.1. Adecuación de sitios

Dado que la mayoría de los servidores serán reubicados en centrales telefónicas, es necesario adecuar los sitios con los requerimientos técnicos de cada servidor que se detallan en el anexo B. Estos preparativos fueron solicitados a las áreas correspondientes de Uninet. A continuación se mencionan las adecuaciones que serán revisadas dos días antes de la puesta en operación:

- Ubicación del espacio asignado para instalar los servidores
- Revisión física del equipo para asegurar que no hubo daños físicos en el traslado a las centrales
- Equipos montados en RACK
- Doble cableado de energía eléctrica de -48V CD
- Doble cableado UTP para conexión a la red
- Puertos de enrutadores asignados y configurados para los servidores de DNS
- Prender equipos para revisar su correcto funcionamiento

4.2.2. Estrategia de migración

Dentro de la puesta en operación se encuentra como punto principal la estrategia de migración, donde se definen las fechas en las que se afecta menos el servicio de DNS para cada uno de los servidores que se encuentran funcionando, y para los servidores nuevos no existe restricción en su fecha y hora de puesta en operación.

Observando las gráficas de consultas por minuto que se muestran en el capítulo II, se selecciona el día y la hora en que cada servidor recibe menos consultas, como resultado de esta revisión se definieron los horarios que se muestran en la tabla 4.4.

SERVIDOR	FECHA Y HORA DE PUESTA EN OPERACIÓN
dnsadm-interno	03 de Mayo de 2002 a las 0:00
nsmex1	10 de Mayo de 2002 a las 0:00
nsmex2	18 de Mayo de 2002 a las 0:00
nsmex3	27 de Mayo de 2002 a las 0:00
nsgdl1	06 de Junio de 2002 a las 0:00
nsmtly1	14 de Junio de 2002 a las 0:00
nsmex4	17 de Junio de 2002 a las 9:00
nsgdl2	21 de Junio de 2002 a las 9:00
nsmtly2	28 de Junio de 2002 a las 9:00

Tabla 4.4. Fechas de puesta en operación.

En la puesta en operación se debe de realizar también el cambio de la configuración de los servidores de acceso, conforme a la puesta en operación de

cada uno de los servidores y realizar la configuración de las listas de acceso en los ruteadores que ya se encontraban definidas. Esta configuración la realizó el área de seguridad de red.

4.2.3. Pruebas de operación

Inmediatamente después de la puesta en operación y configuración de listas de acceso en los ruteadores, se debe probar que cada uno de los servidores tenga conexión a red, que se establezcan las conexiones especificadas en las listas de acceso, que se realicen las transferencias de zona del servidor administrativo, que se resuelvan los nombres de los *hosst* contenidos en las tablas, y que resuelvan los dominios y las direcciones IP que no pertenezcan a sus tablas. Con lo que se confirma que se están realizando todas las funciones del servicio de DNS. Las pruebas realizadas se describen a continuación.

Conexión a la red

Se realizan ping's² a los servidores, desde direcciones dentro de la red de Uninet y desde los ruteadores donde se encuentran conectados.

```
Ping 200.33.150.193
200.33.150.193 is alive
```

```
Ping 200.33.146.193
200.33.150.193 is alive
```

```
Ping 200.33.146.201
200.33.150.193 is alive
```

```
Ping 200.33.146.209
200.33.150.193 is alive
```

```
Ping 200.33.146.217
200.33.150.193 is alive
```

```
Ping 200.23.242.193
200.33.150.193 is alive
```

```
Ping 200.23.242.201
200.33.150.193 is alive
```

```
Ping 200.33.148.193
200.33.150.193 is alive
```

```
Ping 200.33.148.201
200.33.150.193 is alive
```

² Comando del sistema operativo para probar la comunicación entre dos hosts.

Listas de acceso

Estas pruebas son importantes para garantizar la administración remota de los servidores, ya que los equipos no se encuentran en las mismas instalaciones donde están los administradores. Para ello se probarán los servicios habilitados como son TELNET , FTP y *named*.

Este es un ejemplo de una conexión vía FTP

```
>ftp 200.33.150.193
Connected to 200.33.150.193.
220 dnssadm-interno.uninet.net.mx. FTP server (Version 1.1.214.4 Mon Feb
15 08:48:46 GM.
Name (200.33.150.193:usuario): usuario1
331 Password required for root.
Password:
230 User usuario1 logged in.
ftp>
```

Ejemplo de una conexión vía TELNET

```
$ telnet 200.33.150.193
Trying...
Connected to 200.33.150.193.
Escape character is '^]'.
Local flow control off
```

SunOS 5.8

```
login: usuario1
Password:
Dialup Password:
Last login: from usuario1
Sun Microsystems Inc. SunOS 5.8 Generic February 2000
```

Ejemplo las transferencias de zona

```
/usr/local/sbin/named-xfer -z tm3
/usr/local/sbin/named-xfer -z fb
/usr/local/sbin/named-xfer -z la
/usr/local/sbin/named-xfer -z pl-
/usr/local/sbin/named-xfer -z qu
/usr/local/sbin/named-xfer -z av
```

Resolución de nombres en tablas contenidas en el servidor

Ejemplo de las resoluciones de nombres

```
#dns:/>nslookup
Default Server: dns.uninet.net.mx
Address: 200.33.150.193
```

```
> 200.33.150.193
Server: dns.uninet.net.mx
Address: 200.33.150.193

Name: dns.uninet.net.mx
Address: 200.33.150.193

> 148.235.172.132
Server: dns.uninet.net.mx
Address: 200.33.150.193

Name: customer-148-235-172-132.uninet.net.mx
Address: 148.235.172.132

> customer-148-235-172-132.uninet.net.mx
Server: dns.uninet.net.mx
Address: 200.33.150.193

> nsmex4.uninet.net.mx
Server: dns.uninet.net.mx
Address: 200.33.150.193

Name: nsmex4.uninet.net.mx
Address: 200.33.146.217

> 200.33.146.217
Server: dns.uninet.net.mx
Address: 200.33.150.193

Name: nsmex4.uninet.net.mx
Address: 200.33.146.217
```

Resolución de nombres no contenidas en las tablas del servidor

Ejemplo de las resoluciones realizadas

```
> nslookup
Default Server: dnsadm-interno.uninet.net.mx
Address: 200.33.150.193

> www.mama.com
Server: dnsadm-interno.uninet.net.mx
Address: 200.33.150.193

Name: www.mama.com
Address: 66.79.10.211

> www.papa.com
Server: dnsadm-interno.uninet.net.mx
Address: 200.33.150.193

Name: www.papa.com
Address: 207.201.163.94
```

```

> www.yahoo.com
Server: dnsadm-interno.uninet.net.mx
Address: 200.33.150.193

Name: www.yahoo.akadns.net
Addresses: 66.218.71.81, 66.218.71.84, 66.218.71.87, 66.218.71.80,
66.218.71.88, 66.218.71.89, 66.218.71.86, 66.218.71.83
Aliases: www.yahoo.com

```

Todos los servidores pasaron correctamente las pruebas antes mencionadas, su funcionamiento ha sido correcto y no se han tenido problemas con el servicio de DNS.

4.2.4. Monitoreo de los servidores de DNS

Los servidores ya puestos en operación se monitorean con *Firehunter* y *SMC*, para verificar el comportamiento y la calidad del nuevo sistema. Cada uno utiliza el agente previamente instalado para realizar mediciones y después envía la información colectada a un servidor central para que sea procesada.

Firehunter se configuró para medir la disponibilidad y el tiempo de respuesta de los servidores en intervalos de tres minutos durante la semana siguiente a la instalación de cada uno de ellos, esta información se presenta en la tabla 4.5. Es importante mencionar que este monitoreo no aplica al servidor dnsadm-interno por ser administrativo y no aceptar consultas de ningún equipo.

Para complementar las mediciones hechas por *Firehunter*, se utiliza *SMC*, a fin de obtener el comportamiento de los parámetros más importantes de cada uno de los servidores, como son: utilización de CPU, RAM y SWAP. Esta información se muestra en las figuras 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 y 4.9.

SERVIDOR	DISPONIBILIDAD	TIEMPO DE RESPUESTA PROMEDIO
nsmex1	100%	0.01 seg
nsmex2	100%	0.015 seg
nsmex3	100%	0.025 seg
nsmex4	100%	0.03 seg
nsgdl1	100%	0.03 seg
nsgdl2	100%	0.02 seg
nsmt1	100%	0.028 seg
nsmt2	100%	0.024 seg

Tabla 4.5. Resumen de mediciones de Firehunter.

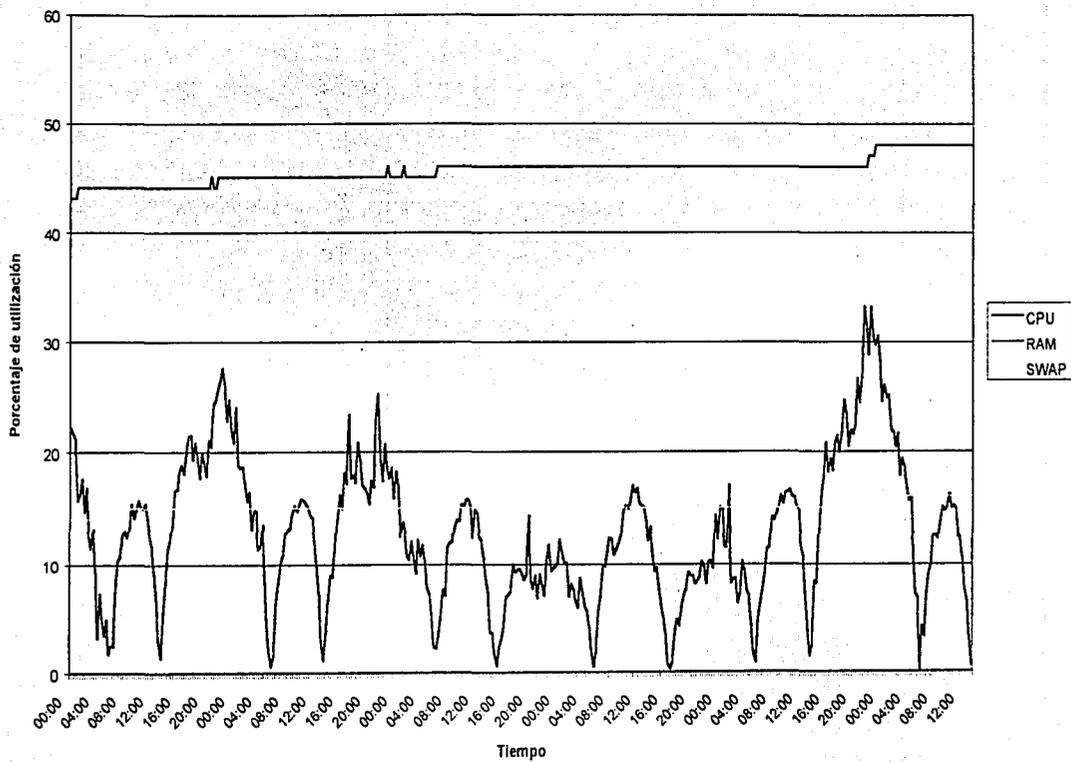


Figura 4.3. Parámetros de desempeño del servidor nsmex1.

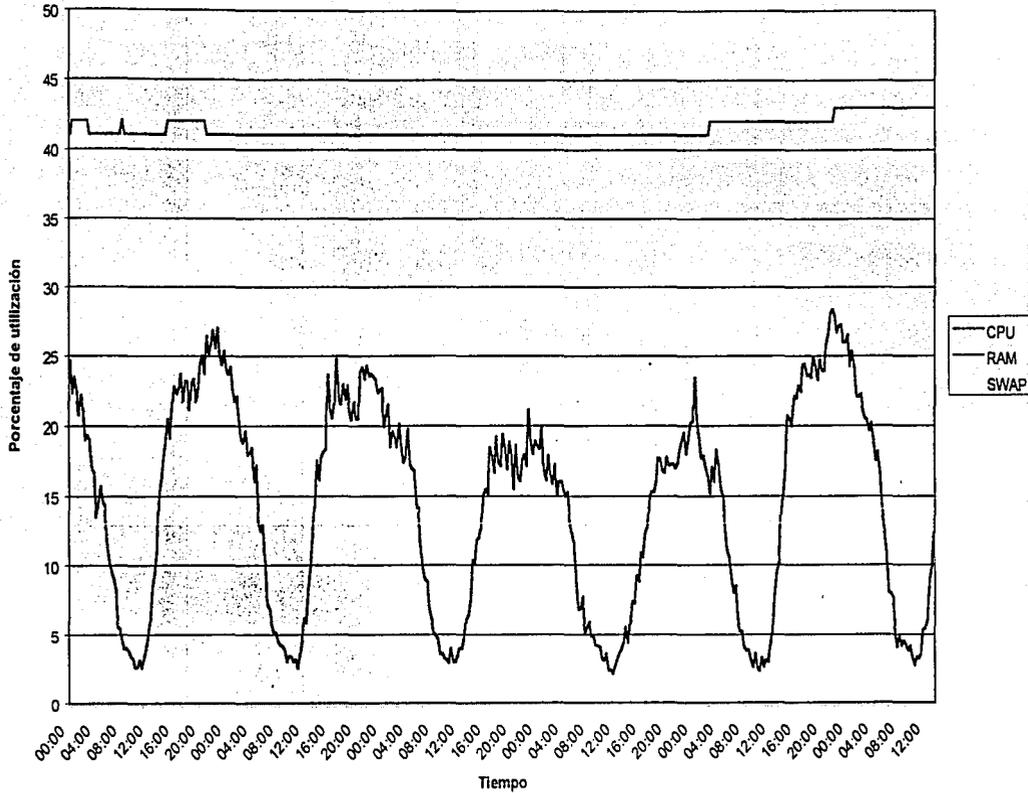


Figura 4.4. Parámetros de desempeño del servidor nsmex2.

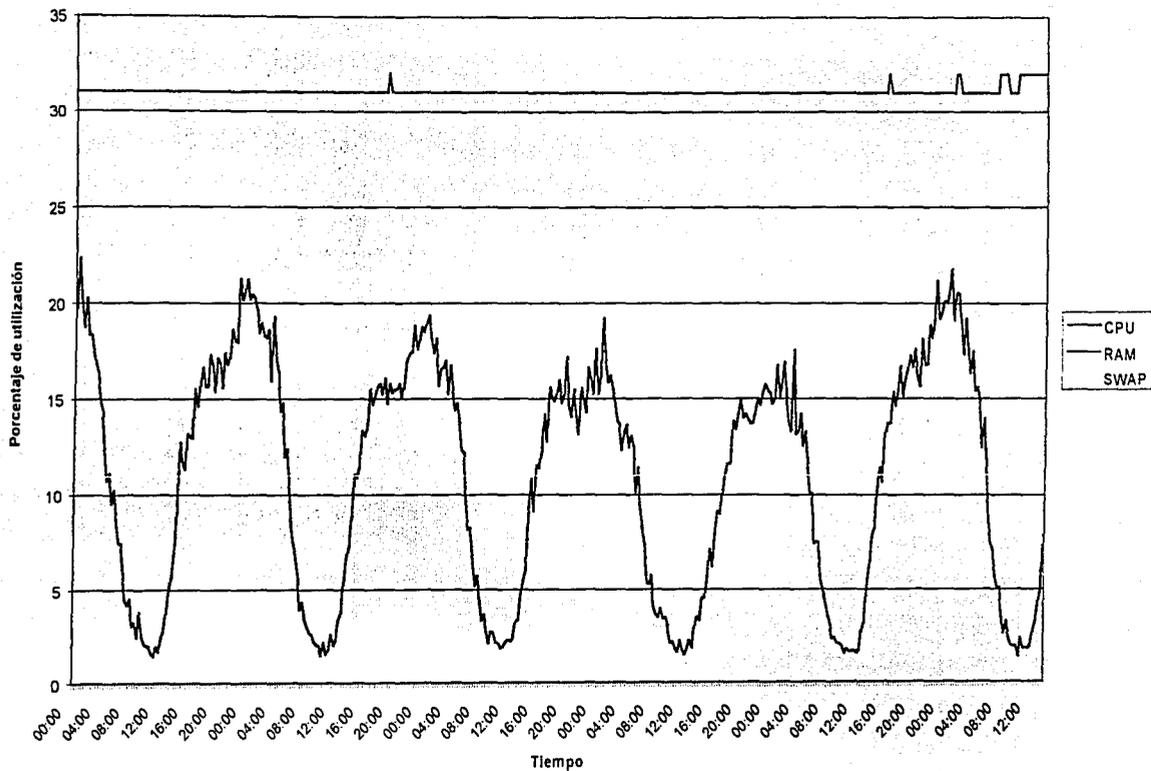
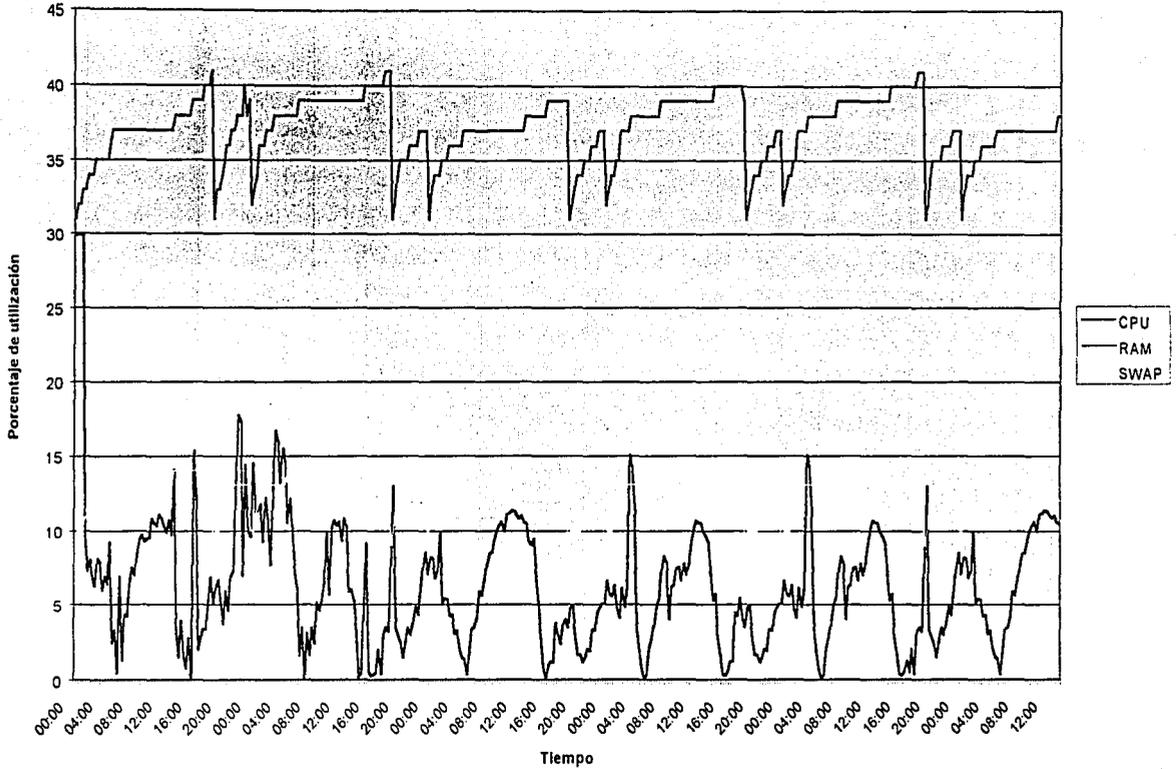


Figura 4.5. Parámetros de desempeño del servidor nsmex3.



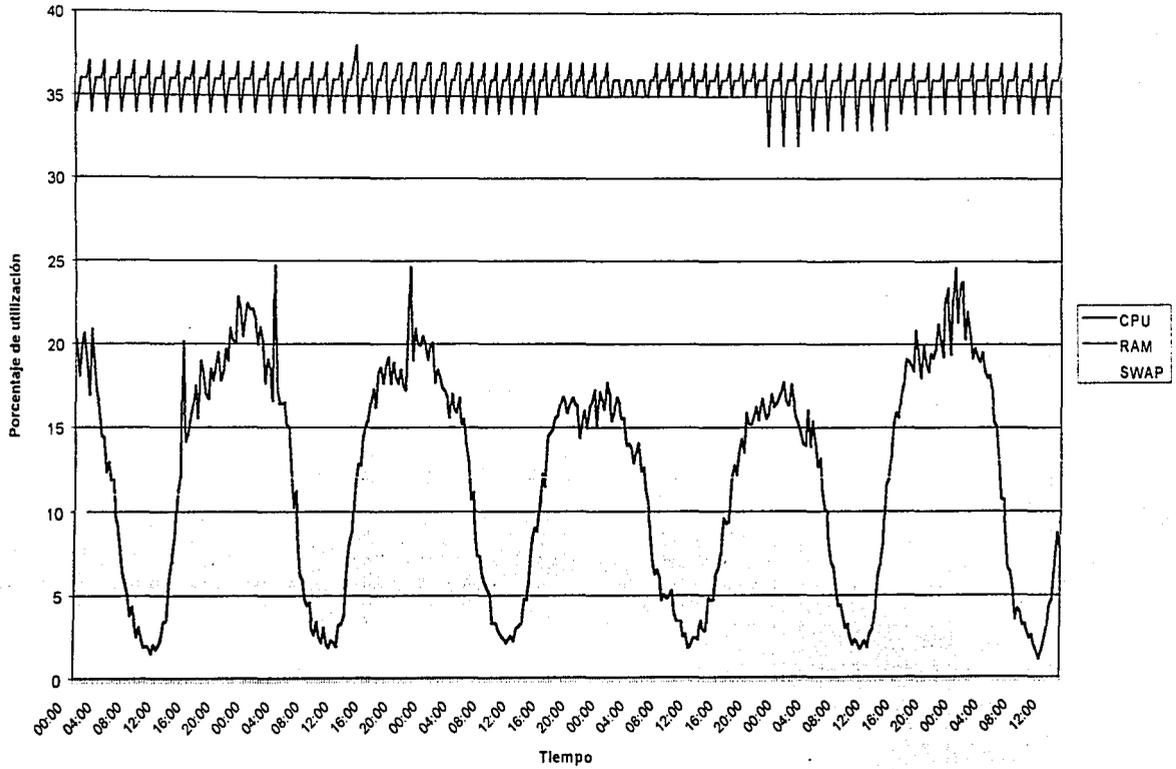


Figura 4.7. Parámetros de desempeño del servidor nsgd11.

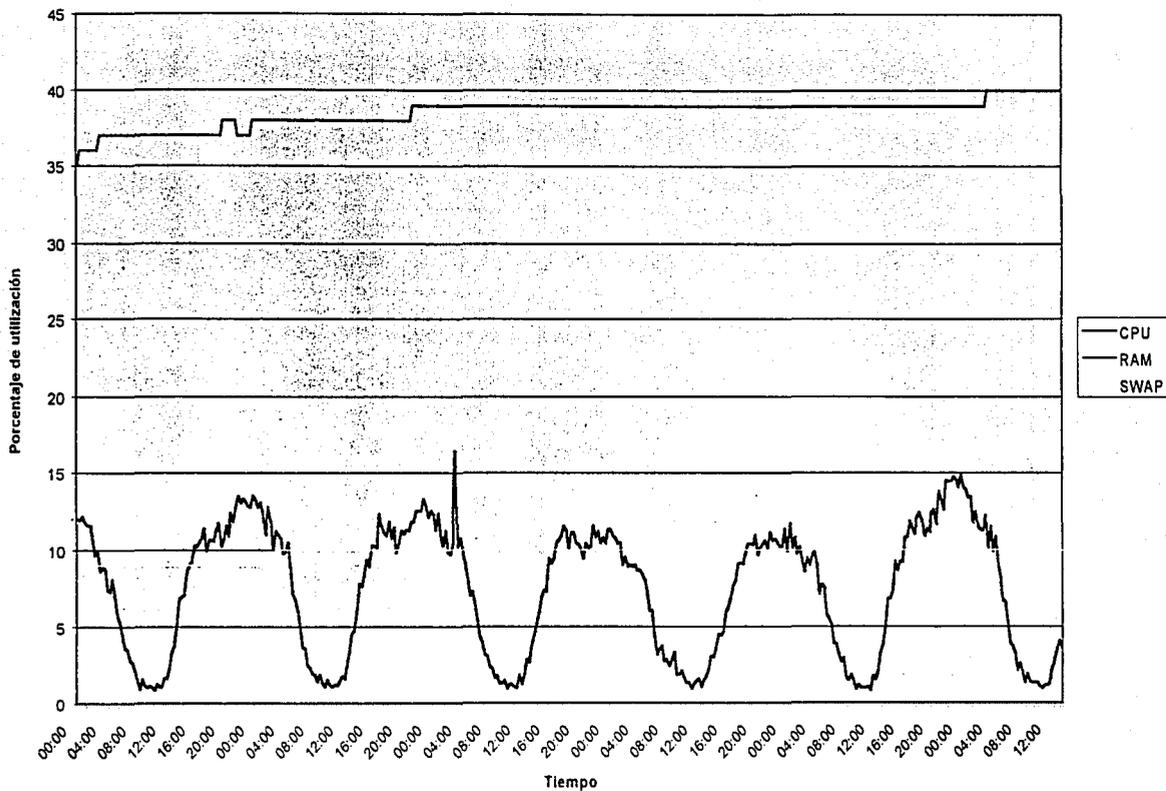


Figura 4.8. Parámetros de desempeño del servidor nsgdl2.

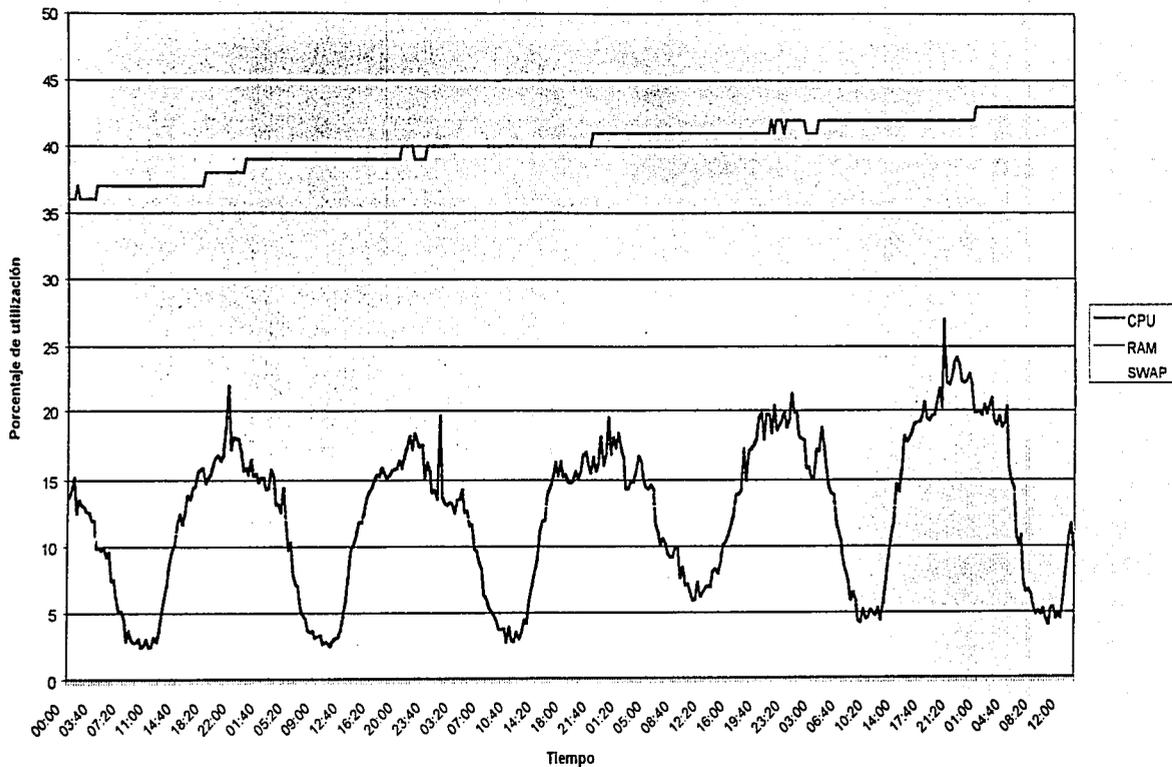


Figura 4.9. Parámetros de desempeño del servidor nsmtty1.

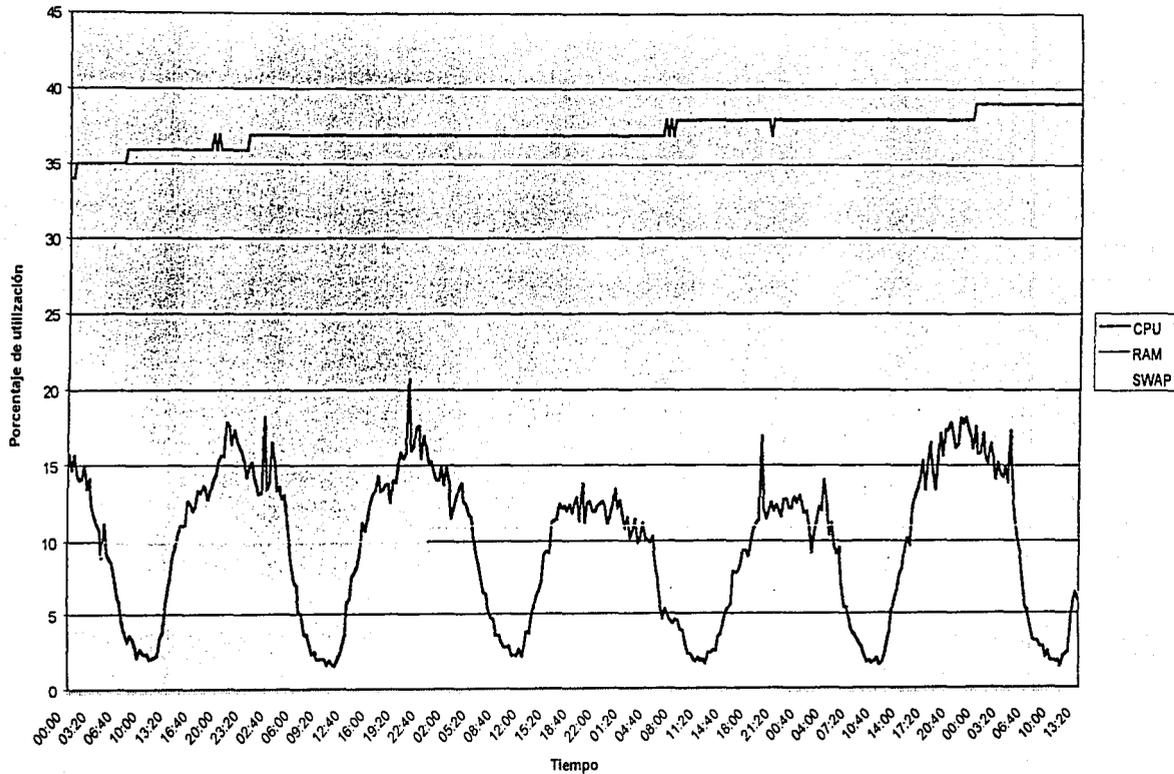


Figura 4.10. Parámetros de desempeño del servidor nsmtly

De forma general se observa un compartimiento similar en todos los equipos, sus porcentajes de utilización son bajos y están respondiendo adecuadamente, lo cual se ve reflejado de forma positiva en el servicio. A manera de resumen se presenta la tabla 4.6 con los promedios de cada uno de los parámetros medidos para cada servidor en la semana siguiente a su puesta en operación.

SERVIDOR	PROMEDIO DE PORCENTAJE DE UTILIZACIÓN DE CPU	PROMEDIO DE PORCENTAJE DE UTILIZACIÓN DE RAM	PROMEDIO DE PORCENTAJE DE UTILIZACIÓN DE SWAP
dnsadm-interno	12.30 %	32.60%	8.20 %
nsmex1	12.73 %	45.66%	15.07 %
nsmex2	14.30 %	41.45%	13.12 %
nsmex3	10.86 %	31.04%	11.00 %
nsmex4	6.33 %	37.16%	12.38 %
nsgdl1	12.05 %	35.71%	11.94 %
nsgdl2	7.41 %	38.55%	9.80 %
nsmt1	12.32 %	40.19%	12.46 %
nsmt2	9.49 %	37.28%	10.61 %

Tabla 4.6. Promedio de parámetros de servidores.

Todas la métricas mencionadas comprueban el buen funcionamiento de los sistemas de monitoreo, los cuales le dan a los operadores la información requerida para mantener operando el sistema. Completando con esto el último paso de la puesta en operación del sistema.

Una vez terminadas las pruebas de monitoreo, el nuevo sistema se da por implantado y pasa al proceso de mantenimiento operativo para evitar que se salga de sus parámetros adecuados de funcionamiento y degrade nuevamente el servicio.

CAPÍTULO V

Resultados y Conclusiones

Después de terminar la puesta en operación y de analizar el comportamiento del servicio de DNS, se presentan los resultados y las conclusiones finales de este trabajo.

5.1. Resultados

El servicio de resolución de nombres de Uninet presentaba graves problemas en su operación, afectando directamente a sus clientes de Internet, por tal motivo se realizó un análisis del servicio y de sus componentes para detectar las causas de tal comportamiento. Habiendo identificado la problemática que afectaba al servicio, se realizó un plan para reestructurarlo, actualizar la infraestructura que lo conforma y monitorear todos sus componentes.

En el análisis se encontraron diversos factores que propiciaban la degradación del servicio, los importantes fueron:

- La falta de control en los servidores, que se debió al cambio frecuente de administradores, a la falta de documentación y de planificación en el crecimiento de dominios.
- Los errores de sintaxis que tienen los archivos de configuración de BIND en los servidores de DNS.
- La capacidad insuficiente de los servidores, con espacio suficiente en disco duro para almacenar los datos de DNS y las bitácoras del sistema, además de presentar falta de memoria y sobrecarga de CPU.
- A que las versiones del sistema operativo y de BIND son viejas y no son soportadas por los fabricantes.
- A diferentes versiones de BIND entre los servidores.
- La mala distribución de los servidores en la red y a que algunos de ellos compartían los enlaces con otros servidores ajenos al servicio de DNS.
- La saturación de los enlaces por la mala distribución de los equipos dentro de la red.
- La inequitativa distribución de las consultas a los diferentes servidores.
- La falta de monitoreo del servicio y de los servidores.

Es importante mencionar que existen muchos servidores de DNS en el mundo, que no presentan ninguno de los problemas mencionados, pero la mayoría de ellos manejan un volumen muy bajo de información y realizan muy pocos cambios en su información por mes, lo que les permite llevar el control de sus nombres de manera tradicional, es decir, editando de forma manual los archivos de configuración y sin necesidad de utilizar complejos sistemas de administración. Pero en el caso de Uninet ya no es posible utilizar la administración tradicional, debido al gran número de direcciones administradas (500,000), al número de dominios de clientes manejados (400) y sobre todo al número de cambios de configuración que se realizan por día, resultado de la operación propia de Uninet.

Estos motivos obligaron a realizar una reestructura total en la forma de operar el servicio de DNS de manera que no se volviera a presentar las mismas fallas. Para realizar el diseño se tomaron como premisas:

- Tener un sistema amigable para los administradores que les permita tener un control de la información.

- Un sistema de administración con filtros en su interfaz gráfica para evitar errores de sintaxis en la configuración de los archivos de BIND.
- Un sistema que lleve el control de los cambios realizados por los operadores y administradores.
- Que realice actualizaciones automáticas que puedan ser programadas en horarios definidos por los administradores.
- Tener diferentes perfiles de usuarios con diferentes niveles de seguridad.
- Hacer una distribución de manera más equitativa de las funciones y de la información entre los servidores.
- Que cuente con enlaces suficientes para soportar el tráfico de los servidores.
- Un sistema de monitoreo eficiente que permita observar el desempeño de todos los componentes del servicio y en caso de falla notificar automáticamente a los administradores del sistema.

Tomando en cuenta estas premisas se propusieron los siguientes puntos como solución integral al servicio de DNS:

- Crecimiento del número de servidores de seis a nueve, con dos servidores por región y uno administrativo que no resuelva consultas de usuarios.
- Definición de funciones a cada servidor dependiendo del tipo de consulta.
- Asignación equitativa y redundante de servidores de DNS a usuarios finales.
- Contar con dos enlaces dedicados por cada servidor para soportar el tráfico de DNS.
- Utilizar *QIP de Lucent Technologies* como sistema de administración y configuración, siendo la única interface en los operadores y los servidores de DNS.
- Definición de la estrategia de seguridad para cada servidor.
- Monitorear el servicio con *Firehunter de Agilent Technologies*.
- Medir el desempeño de los servidores con *SMC de Sun Microsystems*.

Para poder llevar a cabo el diseño propuesto, fue necesario realizar un plan de implantación y puesta en operación para garantizar que se aplicara completamente el diseño propuesto con todas sus premisas. En este plan se contemplaron las siguientes actividades:

- Instalación del sistema operativo, parches de *kernel*, agentes de monitoreo, BIND y software de restricción de entrada a los servidores de DNS.
- Configuración de parámetros de espacio en discos, red, usuarios y servicios, incluyendo los discos en espejo para dar mayor redundancia.
- Instalación y configuración de la herramienta de administración QIP de Lucent.
- Realización de inventarios de la información configurada en los servidores en operación.
- Importación de datos a QIP, utilizando utilerías de carga.
- Pruebas de generación y actualización de archivos de BIND por QIP.
- Configuración de archivos, tablas y parámetros de BIND.
- Pruebas de resolución de nombres, sintaxis, transferencia de zonas a servidores secundarios y actualización de servidores secundarios por cambios en la información.
- Adecuación de los sitios en las centrales telefónicas.
- Definición de horarios de la puesta en operación para afectar lo menos posible el servicio de DNS prestado por los servidores anteriores.
- Pruebas después de la puesta en operación de conexión a la red Internet y resolución de nombres.
- Comienzo del monitoreo del servicio por *Firehunter*.
- Recopilación de mediciones de los parámetros más importantes de los servidores por medio de SMC.

Para poder evaluar los resultados de todas estas actividades se obtuvieron las métricas que se mencionan a continuación.

5.1.1. Desempeño de la CPU

De acuerdo a las recomendaciones de los desarrolladores de BIND el porcentaje de utilización de la CPU para un servidor de DNS en condiciones normales no debe de exceder el 40%, de ser así, es un indicador de problemas en el sistema. En Uninet se detectaron porcentajes superiores al umbral recomendado por periodos de tiempos muy grandes. Con el nuevo sistema se tienen promedios del 10% y el pico más alto no supera el 35% en un intervalo de tiempo pequeño. La tabla 5.1 se muestra una comparación del comportamiento de la CPU entre los servidores antes y después del cambio. Los servidores nsmex4, nsmdl2 y nsmt2 no existían por lo que la columna de antes no contiene información.

SERVIDOR	ANTES	DESPUÉS
dns	41.53%	12.30%
nsmex1	47.48%	12.73%
nsmex2	12.23%	14.30%
nsmex3	19.59%	10.86%
nsmex4	n/a	6.33%
nsmdl1	60.42%	12.05%
nsmdl2	n/a	7.41%
nsmt1	53.01%	12.32%
nsmt2	n/a	9.49%

Tabla 5.1. Porcentajes de CPU.

Cabe mencionar que anteriormente se necesitaba reiniciar el proceso de *named* dos veces al día para evitar que se saturara el servidor y dejara de resolver las consultas. En los nuevos servidores ésta acción ya no es necesaria.

5.1.2. Desempeño de la memoria RAM

Todo equipo de cómputo debe contar con memoria RAM suficiente para soportar los procesos propios del sistema, esto incluye a los servidores de DNS, en donde la falta de memoria RAM se refleja en un comportamiento inestable del proceso *named* dejando de responder consultas.

En la tabla 5.2 se muestra una comparación de los promedios de utilización de memoria RAM de los anteriores servidores y los nuevos.

Debido a la falta de capacidad de memoria RAM de los servidores anteriores era necesario reiniciar dos veces al día el proceso *named* para liberar la memoria RAM y poder continuar respondiendo consultas. Con los servidores actuales ya no es necesario reiniciar *named* y se mantiene un porcentaje bajo de utilización de memoria

SERVIDOR	ANTES	DESPUÉS
dns	95.65%	32.60%
nsmex1	73.36%	45.66%
nsmex2	74.72%	41.45%
nsmex3	73.23%	31.04%
nsmex4	n/a	37.16%
nsgdl1	86.89%	35.71%
nsgdl2	n/a	38.55%
nsmtly1	91.70%	40.19%
nsmtly2	n/a	37.28%

Tabla 5.2 Porcentajes de RAM.

5.1.3. Desempeño de SWAP

La utilización de la memoria SWAP está relacionada con el uso de la memoria SWAP, el hecho de que los procesos tengan que hacer uso frecuente de este tipo de memoria afecta al tiempo de respuesta de los procesos de BIND.

El comportamiento de la memoria SWAP varía sustancialmente en los servidores anteriores y los nuevos, por lo que no ha afectado la resolución de consultas este de parámetro. En la tabla 5.3 se muestra la comparación de la memoria SWAP.

SERVIDOR	ANTES	DESPUÉS
dns	28.15%	8.20%
nsmex1	17.76%	15.07%
nsmex2	16.19%	13.12%
nsmex3	17.96%	11.00%
nsmex4	n/a	12.38%
nsgdl1	19.75%	11.94%
nsgdl2	n/a	9.80%
nsmtly1	23.20%	12.46%
nsmtly2	n/a	10.61%

Tabla 5.3 Porcentaje de SWAP.

5.1.4. Sistema de administración

El sistema de administración fue uno de los factores clave para el éxito del trabajo realizado, ya que muchas de las funciones realizadas manualmente por los operadores fueron delegadas al nuevo sistema de administración implantado para optimizar recursos y evitar errores humanos que afecten al servicio.

A través de QIP se controló el acceso a los nombres de los dominios y a la configuración de cada dominio administrado, controlando los movimientos del

operador y aplicándole filtros en la captura de datos para evitar que se incurran en errores de sintaxis.

Dado que QIP fue diseñado pensando en la administración de grandes volúmenes de datos para DNS, se pudo llegar a tener el control total del direccionamiento de Uninet y de todos los dominios administrados.

También se configuró y probó la actualización automática de las tablas de BIND en los servidores de DNS a través de QIP, resultando ser un sistema eficiente que ejecuta estas tareas sin ningún problema, lo cual beneficia a los operadores que redujeron considerablemente los tiempos de configuración y distribución de información.

El uso de un servidor administrativo (dnsadm-interno) dio como resultado que ya no fuera necesario detener frecuentemente los procesos de BIND en los demás servidores para que pudiera actualizar la información, ya que esto afecta al servicio al sacar de operación a los servidores varias veces al día.

En la tabla 5.4 se resumen las parámetros administrativos que se midieron antes y después de la implantación del nuevo sistema. Como se puede observar los resultados son muy positivos, reflejando de forma notoria la mejoría del sistema.

MÉTRICA	ANTES	DESPUÉS
Errores promedio de sintaxis por servidor	2300	0
Tiempo de activación de un servicio	25 Minutos	3 Minutos
Interrupción del servicio por actualización de datos	3 por día	0
Porcentaje de direcciones que resuelven correctamente	60%	100%
Disponibilidad por servidor	n/a	100%
Frecuencia de reinicio de servidores	2 por día	0

Tabla 5.4 Métricas de administración.

5.1.5. Enlaces

Para asegurar que los servidores tengan los recursos de red necesarios para cursar el tráfico de DNS, se definió que cada servidor debe tener dos enlaces E1 dedicados, los resultados de la medición de dichos enlaces durante una semana completa después de la puesta en operación, se resumen en la tabla 5.5. Estos

números nos indican que los enlaces tienen capacidad suficiente para soportar el tráfico de DNS y en caso de que ocurriera un falla en alguno de ellos el otro enlace puede soportar la carga de trabajo de los dos sin ningún problema.

SERVIDOR	ENLACE	PROMEDIO DE UTILIZACIÓN	PICO MÁXIMO DE UTILIZACIÓN	PROMEDIO DE ERRORES	PROMEDIO DE DESCARTES
dnsadm-interno	Uno	22.32%	30.50%	0%	0%
	Dos	24.45%	35.22 %	0%	0%
nsmex1	Uno	29.50%	40.23%	0%	0%
	Dos	26.10%	35.69%	0%	0%
nsmex2	Uno	27.35%	39.45%	0%	0%
	Dos	25.52%	35.63%	0%	0%
nsmex3	Uno	21.86%	31.25%	0%	0%
	Dos	19.86%	30.85%	0%	0%
nsmex4	Uno	23.68%	37.56%	0%	0%
	Dos	21.96%	33.96%	0%	0%
nsgdl1	Uno	22.36%	32.56%	0%	0%
	Dos	23.55%	36.23%	0%	0%
nsgdl2	Uno	25.30%	33.45%	0%	0%
	Dos	25.96%	35.66%	0%	0%
nsmtly1	Uno	27.85%	38.12%	0%	0%
	Dos	28.32%	41.35%	0%	0%
nsmtly2	Uno	21.23%	35.22%	0%	0%
	Dos	22.56%	36.20%	0%	0%

Tabla 1.5. Porcentajes promedio de utilización.

Cabe señalar que no se puede realizar una comparación entre los enlaces de los antiguos servidores y de los nuevos, dado que las condiciones en que operaban son muy distintas.

5.1.6. Distribución de cargas

Con el objetivo de que todos los servidores tengan una carga de trabajo más equitativa, se realizó una distribución de los dominios y de los nombres de dominios que resolverá cada servidor. Esta iniciativa trajo como beneficio que no existan servidores con poca carga de trabajo mientras otros están saturados.

Otro factor importante que contribuye a esta mejor distribución de cargas es la configuración de los servidores de acceso donde se conectan 900,000 usuarios, a los cuales se les asignan dos servidores de acceso para que realicen sus consultas. Esta asignación se diseñó para que las consultas les lleguen de forma equitativa a todos los servidores.

Los resultados de esta distribución se ven reflejados en el desempeño de la CPU y de la memoria RAM antes mencionados, donde vemos que el comportamiento es muy similar en todos los servidores.

5.1.7. Versiones de BIND

El haber realizado un cambio en las versiones de BIND en todos los servidores, dio un resultado muy positivo, ya que al estar todas homologadas no existe ningún problema de transferencias de zona entre los servidores.

La versión instalada contiene mejoras muy grandes en el aspecto de seguridad, corrigiendo los *bugs* detectados hasta el momento.

5.1.8. Sistema de monitoreo

El último de los resultados que será revisado es el sistema de monitoreo, a través del cual se puede conocer el estado del servicio y de sus componentes, y que sin él no se podrían tener los datos necesarios para revisar su comportamiento a lo largo del tiempo corriendo el riesgo de perder el control del sistema y volver a caer en los mismos errores.

El sistema de monitoreo fue verificado a través de pruebas a los agentes y de comparación con los parámetros de cada elemento. Encontrando una gran confiabilidad en la recolección de datos que después son procesados en los servidores de las aplicaciones de monitoreo para presentarlos en gráficas y alarmas del sistema.

Para fines de pruebas se configuraron alarmas con umbrales bajos para verificar la eficacia del sistema en la notificación de eventos que puedan llegar a afectar al desempeño del servicio.

Estas pruebas resultaron exitosas y los resultados se ven reflejados en todas las métricas mencionadas en este capítulo.

Todos los resultados mencionados en incisos anteriores dan base suficiente para poder dar una conclusión del éxito del trabajo.

5.2. Conclusiones

Tomando como referencia el estado anterior del servicio de DNS, el diseño realizado y los resultados de la implantación, podemos concluir que las mejoras en el servicio han sido notorias desde la implantación del nuevo sistema y que se cumplió totalmente con las mejoras previstas en el diseño.

Después de todos los cambios realizados concluimos que el servicio se queda con las siguientes características:

- Servicio estable, ya no es necesario reiniciar los servidores por saturación ni por configuración de nuevos nombres o dominios.

- Servicio con redundancia en servidores y en el hardware de los servidores, que da una mayor disponibilidad al servicio.
- Control de los dominios y del direccionamiento, con lo que se resuelven todos los nombres y dominios que administra Uninet.
- Servidores robustos y escalables que pueden soportar la demanda del servicio.
- Servicio con mejores tiempo de respuesta en la resolución de nombres.
- Generación y distribución automática de los archivos de configuración de BIND para evitar que por errores humanos se degrade la calidad del servicio.
- Distribución más equitativa de la carga de trabajo entre los servidores.

El DNS al igual que todo sistema es necesario revisarlo y darle mantenimiento periódicamente para asegurarse que no salga de sus parámetros operativos y degrade la calidad del servicio. Esta función queda a cargo de los operadores y administradores del servicio que son los responsables del sistema.

Una de las ventajas de este sistema es que el personal requerirá cada vez menos la interacción directa con BIND, agilizando sus funciones; esto tiene la desventaja que en caso de presentarse una falla y de necesitar que el personal modifique manualmente la configuración de BIND, se dificulte su labor por la falta de experiencia y de conocimiento.

El sistema puede ser mejorado al automatizar la búsqueda de mensajes en sus bitácoras para catalogarlos por su tipo (informativo, error y seguridad), lo que permitiría darle atención oportuna a los eventos que pudieran presentarse.

Actualmente todas las estadísticas que se generan de BIND son configuradas e interpretadas de forma manual. Una mejora importante que se podría realizar al sistema sería la creación de programas que realicen la generación e interpretación automática de las estadísticas de BIND.

Con las mejoras hechas al sistema sería posible hacer un desarrollo que controle automáticamente la asignación de direcciones a todos los dispositivos de la red.

BIBLIOGRAFÍA

- Craig Hunt, "TCP/IP Network Administration", O'Reilly, United States of America, 1997.
- Paul Albitz, Cricket Liu, "DNS and BIND Help for System Administrators", O'Reilly, United States of America, 2001.
- Kevin Loney, "ORACLE, DBA Handbook", McGraw-Hill, United States of America, 1994.
- Barrie Sosinsky, Carol Tanielu, "Solaris 8", SYBEX, United States of America, 2001.
- Sunmicrosystems, "Solaris 8 System Administration", Sunmicrosystems, Broomfield, Colorado U.S.A., 2001.
- Lucent Technologies, "VitalQIP Instalation Guide for Unix", Lucent Tenologies, Malven, PA U.S.A, 2001.
- Lucent Technologies, "VitalQIP User's Guide for Unix", Lucent Tenologies, Malven, PA U.S.A, 2001.
- Stuart McClure, Joel Scambray, George Krurt, "Hacking exposed", Foundstone, McGraw-Hill, USA, 2001.

Páginas Web

- www.webopedia.com/TERMI/IP_address.html
- www.nic.cu/informacion
- www.cisco.com/univercd/cc/td/doc

- www.oac3.hsc.uth.tmc.edu/staff/snnewton/TCP_tutorial
- www.kblabs.com
- www.yale.edu/pclt/comm/TCPIP.htm
- www.nic.unam.mx
- <http://www.isc.org/products/BIND/>
- http://www.acmebw.com/askmrdns/bind-messages.htm#idx_d
- www.sun.com
- www.sunsolve.sun.com

ANEXO A

VitalNet

Es un software que ayuda a optimizar las operaciones de la red mediante el monitoreo, el análisis, la administración y la predicción del funcionamiento de toda la infraestructura de la red. El software recolecta datos de los elementos de la red y presenta la información de una manera ordenada, indicando el estado en que se encuentran los recursos de la red. El sistema también genera reportes y un índice que provee de un método para seguirlo a través de las tecnologías de redes en tiempo real y en operaciones críticas como el tráfico.

Adicionalmente con *VitalNet* se puede personalizar el programa para obtener datos específicos que el administrador de red necesite:

- Estadísticas en tiempo real de problemas y problemas resueltos.
- Información continua de las operaciones y datos históricos para monitorear los niveles del servicio.
- Resumen de alto nivel describiendo de principio a fin las operaciones de la red.
- Analiza la tendencia de la red actualizando lo planeado.

El sistema proporciona las herramientas necesarias para administrar y mantener la infraestructura de red y brinda la seguridad para crecer la red y planear otros servicios en un futuro.

Características

Monitoreo del tráfico: Usa un conjunto de herramientas pro-activas de monitoreo, que continuamente revisan el tráfico de la red con un gran número de mediciones y análisis.

Reportes visuales y gráficos

Provee una visión personalizada de reportes con la capacidad de identificar rápidamente las condiciones de la red.

Web

Incluye un navegador basado en una interface gráfica para reportes de administración y presentaciones de la red, desde cualquier lugar.

ANEXO B

Especificaciones del equipo Netra T1120

Este equipo es recomendable para aplicaciones de telecomunicaciones, cuenta con una arquitectura abierta y un sistema operativo robusto y escalable. En las tablas B.1, B.2, B.3, B.4, B.5, B.6 y B.7 se detallan sus especificaciones.

PROCESADORES	
Arquitectura	Dos procesadores a 440 MHz
Administración de la memoria	MMU con 64 entradas I-TLB y 64 entradas D-TLB
Cache	4 MB, 16KB de datos y 16 KB de Instrucciones en chip secundario: 2 MB externos (por CPU)
Memoria principal	2 GB máximo (con 128-MB en pares de SIMMs) (Nota: Instalar SIMMs en sets de 4 para mejores resultados)

Tabla B.1. Especificaciones de los procesadores.

INTERFACES	
Red	Ethernet/ Fast Ethernet, STP (10-Base T y 100-Base T)
I/O	40-MB/sec UltraSCSI (SCSI-3)
Puerto serial	Dos RS-232C/RS-423 (DB25)
Puerto Paralelo	Centronics-Compatible (DB25) (Modo inteligente ECP)
Bus de Expansión	4 full-size slots PCI compatible con PCI version 2.1; 3 slots operando a 33 MHz, 32-o 64 bits de bus de datos; 1 slot operando a 33 o 66 MHz
Consola	Conector DB de 15 pins; 3 contactos de salida (menor, mayor, crítico); entrada para reset externo

Tabla B.2. Especificaciones de las interfaces.

ALMACENAMIENTO	
CD interno	DVD-ROM interno (48x CD-ROM)
Unidad de cinta interna	Opcional 12 a 24GB DDS-3 4mm, 14GB DDS-2
Discos internos	Dos discos de 36GB
Unidades externas	Todos los aparatos con Ultra SCSI

Tabla B.3. Especificaciones almacenamiento.

SOFTWARE	
Sistema operativo	Solaris 8 con Licencia
Herramientas de programación	4/97 Solaris NEO 2.0, Solaris Open Step 1.0 C, C++, Pascal, FORTRAN, Java, todos los leguajes estándar que soporta Sun
Protocolos	ONC,NFS,TCP/IP,SunLink OSI, MHS, IPX/SPX, DCE, SS7, ATM, FDI
Alarmas	No tiene Alarmas

Tabla B.4. Especificación de software.

AMBIENTE	
Alimentación de	-48/60VDC nominal, 350W, entrada dual

CD	
Condiciones de operación	5C a 40C (41F a 104F) de 5% a 85% de humedad relativa, sin condensar, sujeto a una humedad absoluta máxima de 0.024 kg de agua / kg de aire seco
Operación de tiempo corto (96 horas consecutivas)	-5C a 55C (23F A 131F)(a máxima altura de 1800 m) 5% a 90% de humedad relativa, sin condensar
Fuera de operación	-40C a 70C (-4F a 158F) 10% a 95% de humedad relativa, sin condensar, sujeto a humedad absoluta máxima de 0.024Kg de agua / kg de aire seco
Cinta	Libre de errores de 0° C a 40° C (32° F a 104° F)
Variaciones de temperatura	30° C / hr máxima
Elevación	Operando:-300 a +3000 metros Sin Operar: -300 a +12000 metros
Ruido acústico	Menos de 60dbA a una distancia de 600 mm y una altura de 1500mm, medidos a 25° C
Terremotos	Zona 4 de Terremotos

Tabla B.5. Especificaciones de ambiente.

REGULACIONES	
Seguridad	UL 1950 3ra Edición, CSA C22.2 No. 950
RFI/EMI	FCC Clase A, EN 55022 Clase A, EN 61000-3-2
Certificación	NEBS (Network Equipment Building Systems, Sistema de Construcción de Equipos para Redes), UL, Cul

Tabla B.6. Regulaciones del equipo.

DIMENSIONES Y PESO	
Altura	17.70 cm (6.97 in)
Ancho	43.50 cm (17.13 in)
Profundidad	49.60 cm (19.53 in)
Peso	23.18 kg (51 lb)
Tamaño del rack	19 in. (requiere kit de instalación)

Tabla B.7. Especificaciones de dimensiones y peso.

ANEXO C

QIP Enterprise

QIP de Lucent es una herramienta para la administración de servicios de IP para organizaciones en expansión, que ayuda a la asignación y distribución de servicios de DNS y de direccionamiento.

Para eliminar fallas en el servicio de DNS QIP ofrece actualizaciones sincronizadas en tiempo real y autónomas. Permite una base de datos centralizada, basada en Oracle o Sybase, con servicios remotos en las siguientes plataformas: Microsoft Windows NT y Windows 2000, Hewlett-Packard HP-UX, Sun Solaris, Linux remote e IBM AIX.

También soporta servidores primarios y secundarios de DNS con alto desempeño y disponibilidad. QIP construye la configuración remotamente, permite definir varias organizaciones y reducir el número de administradores del DNS. Es compatible con BIND 4.x y 8.x, soportando información de *resource records* y archivos de zonas para transferencia incrementales.

Requerimientos

Plataforma

- Microsoft Windows NT Server 4.0 (Service Pack 4 o mayor)
- Hewlett-Packard HP-UX 10.20, 11

- Sun Microsystems Solaris 2.5.1, 2.6, 2.7
- IBM AIX 4.2, 4.3
- Windows 2000 Advanced Server

Hardware

- Procesador de 75 MHz o mejor
- Memoria RAM mínimo de 128 MB, recomendación de más de 256 MB
- Espacio en Disco de 350 MB

Bases de Datos

- Sybase 11.9.2.4
- Oracle 7.3.4 en HP-UX 10.20, AIX 4.2, Solaris 2.5.1
- Oracle 8.1.6 en HP-UX 11, AIX 4.3, Solaris 2.6.7, Windows NT/2000

Servidor de QIP

Plataforma:
HP-UX 10.20

Hardware:
HP C3000
552MHz cpu, 120MHz SDRAM, Ultra 2 SCSI LVD hard disk, CD-ROM, HP *fxe*,
fx5 pro or *fx10 pro* graphics, HP 10/100Mbps, keyboard, mouse, power cord.

Base de Datos:
Oracle versión 7.3.4

Interface de Usuario

En la pantalla de presentación, se conecta el usuario dependiendo de su perfil.

Lucent Technologies
Bell Labs Innovations



X

Login Server : qipdev

User Name : qipman

Password : *****

QIP Enterprise Version 5.0

It's an IP World I Manage It

Copyright © 1999 Lucent Technologies Inc. All Rights Reserved

Use Subject to License Agreement

Interface de usuario, proporciona una excelente navegación y solo los administradores tienen acceso a áreas de responsabilidad.

The screenshot displays the QIP Management System web interface. The main window shows a navigation menu with options like File, Policies, Infrastructure, Import, Management, Network Services, Reports, View, and Help. A tree view on the left shows the network hierarchy, including External, QIP Organization, Domains, Networks, OSPF Areas, Subnet Organizations, Regional Offices, Serial Links, and Subnets. A central pane shows a list of subnets, with two windows open for configuration: '172.16.2.0 (Building A) Subnet' and '172.16.10.128 (London) Subnet'. The '172.16.10.128 (London) Subnet' window shows a grid of IP addresses and a DHCP configuration section. The interface includes search filters for Exact match and Begins with, and buttons for Refresh and Close.

Auditoría y Reportes

Provee al administrador de reportes ya hechos y la posibilidad de obtener sólo la información que se requiera.

The screenshot displays the 'Object Management: All Objects' window. It features a menu bar with 'File', 'Edit', 'View', and 'Options'. Below the menu is a 'Search Pattern' input field. A table lists various objects with columns for IP Address, Object Name, MAC Address, Object ID, Status, Domain, Description, and Network. The object 'apollo' with IP 172.16.10.142 is selected. An 'Object Audit History' window is open, showing a search for 'Object: 172.16.10.142'. The audit report for 'QIP Object Audit Report' is displayed, including details like Date (1999-03-02 16:15), Organization (QIP Organisation), User Name (qipman), and contact information for Justin Clark.

IP Address	Object Name	MAC Address	Object ID	Status	Domain	Description	Network
172.16.10.141	pc0073	PC	D DHCP	qivc1kc
172.16.10.142	apollo	Workstation	Static	lucent.com
172.16.10.143	pc0074	PC	D DHCP	qivc1kc
172.16.10.144	pc0075					
172.16.10.145	pc0076					
172.16.10.146	pc0077					
172.16.10.147	pc0078					
172.16.10.148	pc0079					
172.16.10.149	pc0080					
172.16.10.150	tc0081					

Object Audit History

Object
 Address: 172 | 16 | 10 | 142
 Name: _____

Date Range
Start Date: 03/02/1999 00:00
End Date: 03/02/1999 23:59

Report Format
 Text CSV

Screen File Cancel Help

QIP Object Audit Report
Date: 1999-03-02 16:15
Organization: QIP Organisation
User Name: qipman
ADDED: 03/02/1999 16:09 qipman
Object Name: apollo
IP Address: 172.16.10.142
Domain: lucent.com
Class: Workstation
Application: Sales
Name: services.N
TTL: time.
Contact: Justin Clark Email: jclark@lucent.com
Phone: 011-44-1262-866554 Pager: 011-44-1262-355667
Location: 24 Hyde Park

Configuración de DNS, el servicio de DNS es configurado centralmente en una base de datos y ésta puede ser enviada a servidores remotos. Varios software de DNS son soportadas como BIND 8.x y 4.9.x.

Server Profile : Modify

Server Type : [111]

Server Name : [dns.quaditek.com]

Server Address : [180] [200] [3] [6]

Parameters / Value(s)

Parameter	Values	Value(s)
Server Vendor/Version	BIND-4.9X	BIND-4.9X
Domain Root	False	BIND-8.1
Default Directory	c:\named	MICROSOFT-NT4 0
Scheduled Automatic Updates	False	QIP DNS LDAP
Prefix Corporate Extension		XYLAN DNS LDAP
Suffix Corporate Extension		
Zone E-mail	root@quaditek...	
Expire Time	604800 seconds	
Refresh Time	21600 seconds	
Minimum Time	86400 seconds	

OK

Cancel

Help

ANEXO D

Firehunter

Para evaluar el desempeño de los servidores de DNS, es necesario saber como lo perciben los usuarios, para esto se deben de tomar mediciones que sigan un consulta de resolución desde una aplicación que se configure como usuario. *Firehunter* ofrece estas pruebas:

Se mide la disponibilidad del proceso de DNS, la habilidad de resolver un nombre, y los tiempos de respuesta para nombres, como se muestra en la tabla 1.

Medición	Descripción
<i>Availability</i>	Determina si, el proceso de DNS está disponible.
<i>Resolved</i>	Habilidad de resolver la dirección de un nombre
<i>ResolutionTime</i>	Tiempo requerido para resolver un nombre

Tabla 1. Mediciones.

Mediciones del servidor

Las mediciones tomadas en un servidor de aplicaciones de Internet pueden producir abundante información que es de importancia crítica para definir problemas. El agente de *Firehunter* ejecuta en el servidor un rastreo pasivo de estadísticas de interés como la siguiente:

Vmstat, esta prueba pasiva reporta estadísticas acerca de los procesos, memoria virtual, y actividad de CPU. En la tabla 2 se describen las mediciones.

Medición	Descripción
<i>RunQLength</i>	Número de procesos esperando para correr.
<i>BlockedQLength</i>	Número de procesos bloqueados, esperando para habilitarse.
<i>FreeMemory</i>	Paginación de memoria libre en el tiempo de la prueba.
<i>PercentCPUIdle</i>	Porcentaje de CPU disponible en tiempo.

Tabla 2. Mediciones del servidor.

Agente Remoto

El agente remoto de *Firehunter* es el que colecciona los datos de medición y los pasa al DMS (*Diagnostic Measurement Server*) corriendo en otro sistema. Aunque *Firehunter* operara con un simple agente local en el sistema DMS, se puede adicionar agentes en sistemas remotos para coleccionar datos de mediciones pasivas o correr pruebas activas en diferentes puntos. *Firehunter* no requiere un agente en todos los sistemas para monitorearlos; sólo las pruebas pasivas requieren agentes remotos.

Requerimientos

En la tabla 3, se presentan los requerimientos del agente remoto.

<i>Operating Systems</i>	<ul style="list-style-type: none">• Windows NT 4.0 Server or Workstation (Service Pack 5)• Windows 2000• Solaris 2.6,2.7,2.8(SPARC)• Solaris x86 2.6,2.7,2.8 (intel)• HP-UX 10.20, 11.X• RedHat Linux 6.2 with glibc version 2.1.2, or greater• FreeBSD 3.1,3.2,3.3,3.4,4.0
--------------------------	---

Tabla 3. Requerimientos del agente remoto. (Continúa)

Disk Space	Varies by operating system Minimum: 40 MB
Physical Memory	Varies by operating system. Minimum: 64 MB More may be required for large services models.
Virtual Memory	256 MB or an amount equal to the installed physical memory, whichever is greater.
Swap Space (HP-UX only)	HP-UX 11.X Minimum: 512 MB Recommended: 1GB or more HP-UX 10.20 Minimum: 512 MB Recommended: 1 GB or more if running active test

Tabla 3. Requerimientos del agente remoto.

ANEXO E

Sun Management Center 3.0

El software de *Sun Management Center* incluye tres niveles de componentes: consola, servidor y agente, y está basado en una arquitectura administrador/agente en la que:

- La consola es la interfaz con la que el usuario inicia las tareas de administración.
- El servidor (administrador) ejecuta las aplicaciones de administración y envía peticiones a los agentes para efectuar la labor de control solicitada por el usuario.
- Los agentes (que se ejecutan en los nodos administrados) acceden a la información de administración, supervisan los recursos locales y responden a las peticiones del administrador.

A continuación se describen los principales niveles de *Sun Management Center* y sus funciones.

Es posible tener varias consolas (que dan servicio a varios usuarios) para el mismo servidor *Sun Management Center*. Las consolas proporcionan lo siguiente:

- Representaciones visuales de los objetos administrados (por ejemplo, sistemas y redes).
- La posibilidad de manipular atributos y propiedades asociadas a los objetos administrados (por ejemplo, la creación de umbrales de alarmas).
- La posibilidad de iniciar tareas de administración (por ejemplo la reconfiguración dinámica).

Niveles de Sun Management Center

Nivel consola

El nivel consola de *Sun Management Center* constituye la interfaz entre el usuario y los otros niveles de componentes del software de *Sun Management Center*.

Nivel servidor

El nivel servidor acepta las peticiones del usuario a través de la consola y las envía al agente adecuado. A continuación transmite al usuario la respuesta del agente. Por ejemplo, si un administrador desea información sobre el número de usuarios que están accediendo a un sistema, el nivel servidor recibe esta petición desde la consola y la envía al agente de ese sistema. El agente localiza la respuesta y la envía al servidor, que a su vez, transmite la información a la persona que ha emitido la petición (a través de la consola). Igualmente, si se produce una condición de error en uno de los sistemas, el agente de ese sistema envía una notificación del error (un evento) al servidor, que se encarga de reenviar esa información al administrador (a través de la consola) en forma de alarma.

Por último, este nivel proporciona a la consola un punto de acceso seguro para establecer la comunicación con los agentes.

El nivel servidor de *Sun Management Center* incluye cinco componentes que son:

1. Servidor *Sun Management Center*.
2. Administrador de topología.
3. Manejador de alarmas.
4. Administrador de configuración.
5. Administrador de eventos.

El componente servidor es el núcleo del nivel servidor. Está basado en la tecnología Java y puede manejar múltiples peticiones de datos procedentes de distintos usuarios de *Sun Management Center*.

El Administrador de topología de *Sun Management Center* proporciona diversos servicios, como la gestión de los dominios administrativos de los usuarios y la disposición de los objetos administrados dentro de la topología.

El Manejador de alarmas es un receptor central de alarmas SNMP que registra y reenvía todos los avisos de errores a los componentes afectados. Es el componente del nivel servidor responsable de recibir todas las notificaciones de alarmas.

El Administrador de configuración de *Sun Management Center* se encarga de los servicios de seguridad del servidor y de los agentes *Sun Management Center*.

El Administrador de eventos envía y recibe la información de eventos procedente de los agentes *Sun Management Center*. Éstos pueden activar alarmas que luego se reenvían a la consola.

Nivel Agente

El nivel agente se encarga de recopilar la información, supervisar y administrar los objetos en los nodos administrados por *Sun Management Center*. El nivel servidor interactúa con el nivel agente para acceder a los objetos administrados utilizando SNMP.

Los agentes *Sun Management Center* son escalables, ampliables y están basados en SNMP. Supervisan y administran objetos que incluyen el hardware, el sistema operativo y las aplicaciones, para lo cual cargan los módulos específicos de cada aspecto del sistema. Asimismo, se encargan de vigilar el rendimiento y el buen estado de las aplicaciones.

Los agentes utilizan reglas para determinar el estado de los objetos administrados. Cuando las condiciones especificadas en una regla son verdad, el software genera automáticamente las alarmas pertinentes o emprende las acciones indicadas en la regla.

Contexto De Servidor

El contexto de servidor *Sun Management Center* se define como el nivel servidor y los niveles agente correspondientes. Al iniciar la consola, se entra en un determinado contexto de servidor y los objetos administrados cuyos agentes envían información al mismo servidor pertenecen a ese mismo contexto de servidor.

Un objeto administrado puede pertenecer al mismo contexto o a un contexto de servidor remoto (los objetos administrados en contextos remotos envían la información a otro servidor, mientras que los objetos administrados dentro del mismo contexto envían la información al sistema servidor que está conectado a la consola del administrador).

El software de *Sun Management Center* administra de forma predeterminada los objetos situados en el mismo contexto de servidor, pero sólo supervisa objetos situados en contextos remotos.

ANEXO F

Programas de verificación de resolución de nombres

```
#####  
#Verifica la resolución de inversos de las clases C indicadas  
#Lenguaje: awk  
#Autor: sistemas de Administración  
#Fecha: 15/Abril/2002  
#Descripción: Verifica cada una de las 254 direcciones IP de las clases C  
#resuelvan sus inversos generando archivos de texto de salida con el  
#resultado y resumen de las consultas realizadas por el programa.  
#salidas: resumen y resultado  
#entrada: archivo de texto con las clases C a verificar.  
#####  
#!/bin/sh  
  
#Modificar variable de path a su equivalente de directorio de trabajo del  
#programa  
  
path=/export/home/verificar  
  
#####  
fecha=`date '+%d-%m-%y.%H:%MM'`  
  
#Archivo de entrada
```

```
in='echo "$path"/clases-c.txt'
server=$1
result='echo "$path"/result-detalle.$server'
resumen='echo "$path"/resumen-result.$server'
tempo='echo "$path"/tempo'
tempo2='echo "$path"/tempo2'
```

```
rm $result $resumen
cat $in | \
```

#Conformación de la dirección IP

```
while read ip; do
mien='echo "$ip" | awk '{print $1}'"
resto='echo "$ip" | awk '{print $2}'"
o1='echo "$mien" | cut -d"." -f1'
o2='echo "$mien" | cut -d"." -f2'
o3='echo "$mien" | cut -d"." -f3'
o4='echo "$mien" | cut -d"." -f4'
```

#Dominio por el que va a preguntar

```
domain='echo "$o3.$o2.$o1.in-addr.arpa"'
conta=1
ok=0
echo $ok > $tempo2
```

#Se hace la consulta al servidor

```
while [ $conta -lt 255 ]; do
dominio='echo "$conta.$domain"'
ip='echo "$o1.$o2.$o3.$conta"'
sleep 1
/usr/sbin/nslookup -norecurse -querytype=ptr -retry=1 -timeout=2 $dominio
$server > $tempo 2>&1
```

*#Si no contesta la consulta se envía un mensaje al archivo result
#Si responde la consulta envía al archivo result el resultado*

```
falla='cat $tempo | head -1 | cut -d" " -f1'
if [ "$falla" = "" ]; then
temp='cat $tempo | head -1 '
echo "$fecha $ip $dominio $temp" >> $result
ok='expr $ok + 1'
echo $ok > $tempo2
else
nombre='cat "$tempo" | grep $dominio | awk '{print $4}'"
longitud='echo "$nombre" | wc -c'
if [ $longitud -lt 2 ]; then
```

```
ok='expr $ok + 1'  
echo $ok > $tempo2  
fi  
echo "$fecha $ip $dominio $nombre" >> $result  
fi  
conta='expr $conta + 1'  
done  
checar='cat $tempo2'
```

*#Se tiene otra salida con un resumen de como se contestaron las consultas por
#clase C*

```
if [ $checar -eq 0 ]; then  
echo "OK $mien " >> $resumen  
else  
echo "CHECAR $mien ($checar IP's NOResueltas)" >> $resumen  
fi  
done  
rm $tempo $tempo2
```

```

#####
#Verifica la resolución normales de las clases C indicadas
#Lenguaje: awk
#Autor: Sistemas de Administración
#Fecha: 18/Abril/2002
#Descripción: Verifica cada una de las 254 direcciones IP de las clases C
# resuelvan normalmente generando archivos de texto de salida con el
# resultado y resumen de las consultas realizadas por el programa.
#salidas: resumen y resultado
#entrada: archivo de texto con las clases C a verificar.
#####
#!/bin/sh
#####
#Modificar variable de path a su equivalente de directorio de trabajo del
programa

path=/export/home/verifica
#####
#Archivo de entrada

in=`echo "$path"/clases-c`

#Tabla del dominio

archivo=`echo "$path"/db.uninet.net.mx`
dominio=uninet.net.mx
#####
fecha=`date '+%d-%m-%y.%HH:%MM'`
server=$1

#Definición de archivos de salida

result=`echo "$path"/result-detalle.$server`
resumen=`echo "$path"/resumen-result.$server`
tempo=`echo "$path"/tempo`
tempo2=`echo "$path"/tempo2`

rm $result $resumen
cat $in | \

#Definición de la dirección IP

while read linea; do
  mien=`echo "$linea" | awk '{print $1}'`
  resto=`echo "$linea" | awk '{print $2}'`
  o1=`echo "$mien" | cut -d"." -f1`
  o2=`echo "$mien" | cut -d"." -f2`
  o3=`echo "$mien" | cut -d"." -f3`
  conta=1
  ok=0

```

```
echo $ok > $tempo2
```

```
#Consulta realizada a cada una de las direcciones IP
```

```
while [ $conta -lt 255 ]; do
  ip='echo "$o1.$o2.$o3.$conta"'
  name='grep " A $ip" "$archivo" | grep -w $ip |awk '{print $1}'"
  nameall='echo "$name.$dominio"'
  lname='echo "$name" | wc -c | awk '{print $1}'"
  if [ $lname -gt 1 ]; then
    /usr/sbin/nslookup -norecurse -querytype=a -retry=1 -timeout=1 $nameall
  $server > $tempo 2>&1

```

```
#Si falla la consulta se envia un mensaje a uno de los archivos de salida
```

```
falla='cat $tempo | head -1 | cut -d" " -f1'
if [ "$falla" = "****" ]; then
  temp='cat $tempo | head -1 '
  echo "$ip $name $temp" >> $result
  ok='expr $ok + 1'
  echo $ok > $tempo2

```

```
#Si no envia el resultado de la consulta
```

```
else
  dirname='cat "$tempo" | grep "Name:" | awk '{print $2}'"
  longitud='echo "$dirname" | wc -c'
  if [ $longitud -lt 2 ]; then
    ok='expr $ok + 1'
    echo $ok > $tempo2
  fi
  echo "$ip $name ANSWER=$dirname" >> $result
fi
else
  echo "$ip No-Entrada-En-Archivo-de-IN-A" >> $result
fi
conta='expr $conta + 1'
done
checar='cat $tempo2'

```

```
#Envia los resultados por clase C al archivo resumen
```

```
if [ $checar -eq 0 ]; then
  echo "OK $mien " >> $resumen
else
  echo "CHECAR $mien ($checar IP's NOResultas)" >> $resumen
fi
done

```

```
rm $tempo $tempo2
```