

96
25j



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE CIENCIAS

**ESTUDIO PROBABILISTICO DE FALLAS EN
UN SISTEMA HIBRIDO DE COMPUTO,
SU MONITOREO Y CORRECCION**

T E S I S

QUE PARA OBTENER EL TITULO DE:

A C T U A R I A

P R E S E N T A :

MARIA SUSANA SORIANO RAMIREZ



**TESIS CON
FALLA DE ORIGEN**

DIRECTOR DE TESIS: ING. RAFAEL LACAMBRA MACEDO

MAT. ANA LUISA GONZALEZ COSIO



1996

**FACULTAD DE CIENCIAS
EBOCION ESCOLAR**

**TESIS CON
FALLA 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.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

M. en C. Virginia Abrín Baule
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
P r e s e n t e

Comunicamos a usted que hemos revisado el trabajo de Tesis:
Estudio Probabilístico de Fallas en un Sistema Híbrido de Cómputo, su Monitoreo
y Corrección.

realizado por María Susana Soriano Ramírez

con número de cuenta 8727479-4 , pasante de la carrera de Actuaría

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis

Propietario Ing. Rafael Lacantra Macedo

Propietario Mat. Ana Luisa Solís González-Cosío

Propietario M. en C. Héctor Perales Valdivia

Suplente M. en C. María Guadalupe Ibarrera González

Suplente Mat. Hugo Villaseñor Hernández



Consejo Departamental de Matemáticas

Act. Claudia Carrillo Quiroz

A mis padres:

Doña Emma Ramírez:

Por su apoyo, cariño, grandeza, inteligencia, valor y su admirable corazón. Por enseñarme todo lo básico para seguir luchando en la vida. Y porque la quiero con todo mi corazón.

Don Ignacio Soriano:

Por su apoyo, cariño, por ese corazón que siempre esconde, por sus desveladas por acompañarme, por aguantarme en las épocas más difíciles de esta tesis. Y porque lo quiero mucho.

Agradecimientos

Son tantas las personas que influyeron en esta tesis que si las nombrara una por una, esta tesis tendría el doble de grueso. Varias de estas personas son mencionadas, algunas de ellas no lo serán pero todas y cada una de ellas saben que tienen todo mi agradecimiento, respeto y afecto.

Angel Soriano por apoyarme siempre moralmente y por sus valiosos consejos para la construcción, realización y término de esta tesis.

Rafael Lacambra mi director de tesis, por su apoyo y por ser el que ídeo el origen de esta tesis.

Martin Romero por revisar la parte matemática de esta tesis y por su colaboración.

Ara Luisa Solís mi co-directora de tesis por la revisión de esta tesis y por estar dispuesta siempre a ayudar para que este trabajo saliera adelante.

Lupita Ibarquengolita, Hugo Villaseñor y Héctor Perales por revisar la tesis.

Al Departamento de Supercómputo y Laboratorio de Visualización por todas las facilidades prestadas para realizar esta tesis, en especial a *Alejandro Viruega* por su disposición y ayuda por que este trabajo se terminara lo antes posible.

A doña Emma y don Ignacio por apoyarme siempre y por quererme tanto.

A mis hermanos: Jose Luis, Rosa, Angel, Abel, Tere, Oscar, Israel, Melissa y Miriam, por todo el apoyo y cariño.

Y a una persona en especial cuyo nombre esta escrito en cada una de las páginas de esta tesis,

Gracias.

Índice General

I INTRODUCCIÓN	6
1 Introducción	7
1.1 Historia de los sistemas en cómputo	7
1.2 Análisis de un sistema híbrido de cómputo	8
1.3 Objetivo de la Tesis	10
1.4 Hilo Conductor	10
II ANTECEDENTES Y DESCRIPCIÓN DE UN SISTEMA DE CÓMPUTO REAL	12
2 Descripción de un sistema de cómputo real.	13
2.1 Descripción General	13
2.2 Problemas presentados inicialmente en el sistema.	14
2.2.1 Tráfico de red	15
2.2.2 Insuficiente espacio en disco	15
2.3 Grupos de máquinas que componen el sistema	16
2.3.1 Primer grupo	16
2.3.2 Segundo grupo	17
2.3.3 Tercer grupo	17
2.3.4 Cuarto grupo	17
2.3.5 Servidor de disco.	17
2.4 Descripción de los problemas generados por la nueva estructura del sistema.	18
2.4.1 Disponibilidad de la información	18
2.4.2 Interactividad del sistema	18
2.4.3 Seguridad en la información	21
2.5 Cambios físicos iniciales en el sistema	21

III CONCEPTOS BÁSICOS DE CÓMPUTO 24

3 Servicios de Red (NFS, NIS) 25

- 3.1 Remote Procedure Call (RPC) 25
 - 3.1.1 Modo de operación 25
 - 3.1.2 Independencia del RPC con los protocolos de Transporte . 28
 - 3.1.3 Protocolo Port mapper 28
 - 3.1.4 Modelo Cliente-Servidor 29
- 3.2 Network File System (NFS) 29
 - 3.2.1 Protocolo Port mapper trabajando con RPC para NFS . . . 31
 - 3.2.2 Características especiales de NFS 31
 - 3.2.3 Entrada/Salida Asíncrona en el cliente de NFS 31
 - 3.2.4 Restricciones en la operación de NFS 33
- 3.3 Network Information Service (NIS) 34
 - 3.3.1 Ejemplo de ejecución de una petición de NIS 36

IV MODELOS PROBABILÍSTICOS EN PROBLEMAS DE COMPUTACIÓN 37

4 Construcción de los Modelos Probabilísticos. 38

- 4.1 Preliminares 38
 - 4.1.1 Procesos de Poisson 38
 - 4.1.2 Procesos de Nacimiento-Muerte 44
- 4.2 Redes de Colas 48
 - 4.2.1 Teorema de Little, (Little's Theorem) 52
 - 4.2.2 Análisis de la cola de tipo M/M/1 54
 - 4.2.3 Modelo de una cola de tipo M/G/1 57
 - 4.2.4 Análisis de la cola M/M/K 61
 - 4.2.5 Sistemas de redes de colas 65
- 4.3 Modelo del sistema de Cómputo de tiempos distribuidos
(Modelo del servicio de red NIS). 69
 - 4.3.1 Errores y retardos 71
- 4.4 Modelo de sistemas de archivos distribuidos remótante
(Modelo del servicio de red NFS). 71
 - 4.4.1 Evaluación de Errores y Fallas 75
- 4.5 CSMA con Detector de Colisiones,(CSMA/CD) 76
 - 4.5.1 Características del (CSMA/CD) 76

4.5.2	Análisis de un CSMA/CD no-persistente	77
4.5.3	Análisis de un sistema CSMA/CD no-persistente con intervalos de tiempo	80
4.5.4	Análisis para el esquema CSMA/CD 1-persistente	83
4.5.5	Evaluación de los retardos en un CSMA/CD.	85
4.5.6	Recuperación de Errores para un CSMA/CD	88
4.6	Modelo de FDDI	89
4.6.1	Notación para el modelo de una red FDDI.	89
4.6.2	Redes con una sola clase de tipo asincrónico por estación	91
4.6.3	Evaluación de los retardos	91
4.7	Recuperación de Error	92
4.7.1	Recuperación de las rupturas	93
4.8	Modelo de un Ruteador	94
4.8.1	Evaluación de los Retardos y Errores	96
4.9	Modelo de un Sistema Híbrido de Cómputo.	96
4.9.1	Notación	97
4.9.2	Análisis del tiempo de retardo del servidor y clientes conectadas en una misma red	98
4.9.3	Análisis de los retardos de sistema de cómputo híbrido en diferentes redes	100
4.9.4	Análisis del rendimiento del sistema híbrido de cómputo	100
	Análisis del rendimiento de un sistema de cómputo en una red local	101
	Análisis del rendimiento de un sistema híbrido conectados en más de una red.	102

V APLICACION DE LOS MODELOS PROBABILÍSTICOS EN UN SISTEMA DE CÓMPUTO REAL 103

5	Aplicación de los modelos estocásticos en un problema real	104
5.1	Análisis de resultados de NFS en clientes comunicadas en una de las redes 159 y 161	108
5.2	Análisis de resultados de NFS en clientes comunicados en distintas redes	113
5.3	Análisis de resultados de NFS en clientes comunicadas en la red fddi 160	115

5.4	Análisis de resultados de NFS en clientes comunicadas en las redes 170 y 204.	118
5.5	Comparación entre tiempos de retardo de clientes	124
5.6	Comparación entre el desempeño de un Ethernet y un Fddli	126
VI	Conclusiones	129
6	Conclusiones	130
6.1	Recomendaciones para mejorar el desempeño de un sistema de cómputo híbrido	130
6.2	Conclusiones Generales	131
VII	APÉNDICES	133
A	Monitoreo de un Sistema Híbrido de Cómputo	134
A.1	Herramientas y Técnicas de Monitoreo de un Sistema de Cómputo	134
A.2	Monitoreo del servicio de red NFS	134
A.2.1	Monitoreo de NIS	135
A.2.2	Monitoreo de Red	136
B	Protocolos de Comunicación en red.	138
B.1	Preliminares	138
B.2	Tareas principales de un protocolo	140
B.3	Protocolos de comunicación directa y de comunicación indirecta	141
B.4	Arquitectura de TCP/IP	141
B.4.1	Internet Protocol (IP)	142
B.4.2	Transmission Control Protocol (TCP)	144
B.4.3	User Datagram Protocol (UDP)	144
C	Respaldos	146
C.1	Necesidad de hacer respaldos	146
C.2	Formas de respaldo	147
	Dispositivos de respaldo	148
	Red	148
C.3	Automatización de respaldos	148
C.3.1	Descripción de rtar.	149

	Archivos de configuración: .rtar.src y otros	149
	Archivos de log	149
C.3.2	Opciones para el manejo de rtar,	149
	Control local de los dispositivos	151
C.3.3	Localización de la herramienta rtar	151
C.3.4	Automatización de respaldo para tareas de administrador	151
C.4	Otras especificaciones acerca de la implementación de la automa- tización de respaldos	153
C.5	Recuperación de los respaldos	153
C.6	Comandos básicos para el respaldo de información y recuperación de la misma	157
C.6.1	tar	157
C.6.2	dd	158
C.6.3	mt	159
C.6.4	rsh	160

Parte I
INTRODUCCIÓN

Capítulo 1

Introducción

1.1 Historia de los sistemas en cómputo

Con la llegada de las computadoras, se comenzaron a crear arquitecturas con el fin de cumplir tareas distintas. En la década del 70, se crearon los *sistemas centrales*, constituidos de sistemas grandes, voluminosos, de propósito general, y como características principales grandes cantidades de disco y memoria. Los puntos de acceso a ellas se constituían de terminales en línea*, en su mayoría, máquinas con un monitor y teclado; totalmente esclavas de las máquinas centrales.

Esta configuración tenía sus desventajas, ya que cuando alguna terminal no estaba disponible (problemas principalmente de hardware), los puntos de conexión hacia las máquinas centrales disminuían; la total dependencia de las terminales a las computadoras centrales, causaba que cuando estas últimas tenían problemas internos (problemas de hardware o software), las terminales tampoco podían ser utilizadas, aunque estas no tuvieran ningún tipo de problema.

El concepto del sistema descrito, vendió gran cantidad de computadoras centrales y terminales. Las desventajas principales para los usuarios que requerían de aplicaciones particulares fue el elevado tiempo de ejecución, tiempo de aprendizaje, tiempo de depuración, mantenimiento, etc.

Desde la mitad de la década del 70 a la fecha, se comenzaron a crear *sistemas de cómputo de uso personal*, integrados por una computadora disponible a un solo usuario, esta arquitectura originó un aumento de población con acceso a una computadora, debido a que ya no era necesario comprar todo un equipo de máquinas para ejecutar una aplicación o tarea y el costo era más accesible.

*Terminales conectadas directamente a la máquina central

Capítulo 1

Introducción

1.1 Historia de los sistemas en cómputo

Con la llegada de las computadoras, se comenzaron a crear arquitecturas con el fin de cumplir tareas distintas. En la década del 70, se crearon los *sistemas centrales*, constituidos de sistemas grandes, voluminosos, de propósito general, y como características principales grandes cantidades de disco y memoria. Los puntos de acceso a ellas se constituían de terminales en línea*, en su mayoría, máquinas con un monitor y teclado; totalmente esclavas de las máquinas centrales.

Esta configuración tenía sus desventajas, ya que cuando alguna terminal no estaba disponible (problemas principalmente de hardware), los puntos de conexión hacia las máquinas centrales disminuían; la total dependencia de las terminales a las computadoras centrales, causaba que cuando estas últimas tenían problemas internos (problemas de hardware o software), las terminales tampoco podían ser utilizadas, aunque estas no tuvieran ningún tipo de problema.

El concepto del sistema descrito, vendió gran cantidad de computadoras centrales y terminales. Las desventajas principales para los usuarios que requerían de aplicaciones particulares fue el elevado tiempo de ejecución, tiempo de aprendizaje, tiempo de depuración, mantenimiento, etc.

Desde la mitad de la década del 70 a la fecha, se comenzaron a crear *sistemas de cómputo de uso personal*, integrados por una computadora disponible a un solo usuario, esta arquitectura originó un aumento de población con acceso a una computadora, debido a que ya no era necesario comprar todo un equipo de máquinas para ejecutar una aplicación o tarea y el costo era más accesible.

*Terminales conectadas directamente a la máquina central

La principal desventaja de diseño de esta arquitectura que hasta la fecha ha mostrado, es el contar con pocos recursos básicos, como es el caso de espacio en disco y memoria. Esto da como resultado que la ejecución de un proceso sea lenta y a veces imposible.

Con la ampliación del uso de *redes de computadoras*, se crearon las *Estaciones de trabajo*, que son arquitecturas cuyo diseño permite manejar volúmenes de espacio en disco, memoria, procesadores, etc. Estas arquitecturas funcionan en modo multiusuario y multiproceso, debido a esto, es posible ejecutar más de una tarea al mismo tiempo, lo cual reduce el tiempo de espera de ejecución. La mayoría se comunican mediante redes de computadoras, es por esto que la influencia de este factor de comunicación es de gran importancia.

Existen en el mercado distintas arquitecturas de estaciones que se apegan a fines específicos, tal es el caso de las máquinas con características de una alta resolución de gráficos, suficientes recursos para funcionar como servidor o proveedor de recursos compartidos, etc. A un grupo de estaciones con arquitecturas diferentes se le llama sistema híbrido de cómputo.

1.2 Análisis de un sistema híbrido de cómputo

El análisis de un sistema híbrido de cómputo, se enfoca principalmente, a mejorar el desempeño y rendimiento de cada una de las estaciones de trabajo que lo integran. Este análisis por lo general, se comienza con el estudio del comportamiento del sistema, teniendo como principales puntos:

- El espacio en buffer que el sistema provee para la entrada y salida de datos.
- Cuál es el tiempo de respuesta esperado para ejecutar un servicio en una cola.
- Cuántos procesos en promedio esperan para su ejecución.
- Cuál es la estructura de la red para efectuar la retransmisión de mensajes.

Algunos de los métodos que proporcionan una idea para conocer estos puntos son:

- **Benchmarks.** En las pruebas de rendimiento (*Benchmarks*) se aplican, para cada componente de la computadora, tareas específicas; de tal forma que es posible tener una idea del funcionamiento y grado de utilización o subutilización para cada uno de sus componentes en estudio.

La ventaja principal de estas pruebas, es que el rendimiento estimado es preciso. Las desventajas son: el costo y la rigidez de la técnica. En ocasiones, es muy caro adquirir el equipo para propósitos de evaluación y mantener personal especializado para el análisis de los resultados.

- Simulación. Este modelo tiene como fin validar las funciones del sistema y de esta forma conocer el origen de sus fallas. Al igual que los benchmarks, este método forma parte de un experimento científico, ya que a través de la modificación de parámetros es posible obtener diferentes resultados.

La simulación tiene la propiedad de poder definir detalles del sistema, esto representa una ventaja y una desventaja. La ventaja consiste en que cualquier rasgo del sistema puede implementarse y observar el impacto que tiene en el rendimiento del sistema. La desventaja es que al incluir tantos rasgos o características del modelo, se incrementa el tiempo de construcción del mismo y su grado de dificultad.

- Análisis basado en predicciones. En este análisis se construye un modelo matemático del sistema basado en la Teoría de las Probabilidades y en Teoría de Colas.

Este último método es el que se usó en esta tesis, las razones principales por las que se optó por crear un análisis basado en predicciones fueron:

- Muchos de los procesos son aleatorios.
Por ejemplo, la transmisión de un paquete de información por la red es un evento aleatorio, ya que el tiempo varía dependiendo del tráfico de red y de las fallas en ésta, en el cliente o en el servidor.
- Es mucho más conveniente
Es el mejor medio para representar las características esenciales del sistema con algunos parámetros, pues permite evaluar el rendimiento y demostrar los efectos que producen los cambios de los parámetros en el sistema.

El origen de la aplicación de la Teoría de la Probabilidad en el campo de los sistemas de cómputo fue en el año de 1917 con el Ingeniero A.K. Erlang (Ver [Pe]), quien analizó el comportamiento de colas simples al diseñar un sistema de comunicación. En 1960 este estudio fue retomado por Scherr y aplicado en los sistemas de cómputo. Fue hasta el final de los setenta y extendida hasta los últimos veinte años cuando la Teoría de Colas comenzó a

ser una parte fundamental en el estudio del rendimiento en un sistema de cómputo.

El análisis de predicciones se implantará en un sistema híbrido de cómputo real, ubicado en el Departamento de Supercómputo y el Laboratorio de Visualización, de la Dirección General de Servicios de Cómputo Académico en la UNAM.

1.3 Objetivo de la Tesis

La principal desventaja en los sistemas híbridos de cómputo comunicados por red y ejecutando servicios de red, se produce cuando la estación que brinda dichos servicios falla, o en su defecto, cuando el desempeño de la red es pobre, pues el sistema se desestabiliza reduciendo su rendimiento total.

Es por esto que se planteó un estudio del sistema y con ello se llegó a definir el objetivo principal de esta tesis:

Definir un modelo matemático de un sistema híbrido de cómputo comunicado por red que permita predecir las fallas y rendimiento de éste, con la característica de mantener dos sistemas de red llamados NIS y NFS. Además de construir programas basados en el modelo, que permitan hacer el análisis de fallas que afectan a los dos servicios de red en el Sistema Híbrido de Cómputo.

1.4 Hilo Conductor

La tesis comienza con la introducción de los sistemas híbridos de cómputo, donde se da una breve historia del origen de éstos y sus antecesores. En la segunda parte de este estudio se describe el sistema híbrido de cómputo de la Dirección General de Servicios de Cómputo Académico, en donde se originó el objetivo de esta tesis.

En la tercera parte, se mencionan los protocolos que este sistema híbrido de cómputo utiliza para su funcionamiento. Este capítulo es importante pues contiene los conceptos de cómputo básicos para entender el desempeño y con base en este construir el estudio de las predicciones.

Posteriormente se construyen los modelos probabilísticos basados en los protocolos utilizados por el sistema híbrido de cómputo, los cuales tienen como fin estimar los tiempos de retardo y fallas en el sistema mediante la Teoría de Colas.

Por último se aplican los modelos construídos en el sistema de cómputo híbrido ya mencionado, con el fin de obtener la principal causa de las fallas en los servicios de red. De esta aplicación se mencionan recomendaciones para el mejoramiento del sistema y se dan las conclusiones obtenidas de todo el estudio.

Al final de esta tesis, se agregan apéndices cuyo contenido son conceptos técnicos útiles para el fácil manejo de un sistema que cumpla con estas características. Entre ellos se agregan las herramientas creadas para el monitoreo las cuales se basan en los modelos probabilísticos construídos.

Parte II

ANTECEDENTES Y DESCRIPCIÓN DE UN SISTEMA DE CÓMPUTO REAL

Capítulo 2

Descripción de un sistema de cómputo real.

Con la puesta en marcha de los sistemas de cómputo híbridos, surgieron las ventajas y desventajas que este traía consigo, y con ello se hizo necesario un estudio profundo para el mejoramiento y buen rendimiento del sistema.

El objetivo de esta tesis, surgió de la necesidad del estudio de un sistema híbrido de cómputo en particular, con la característica principal de mantener un ambiente distribuido y compartir recursos, mediante redes de cómputo.

2.1 Descripción General

En el Laboratorio de Visualización, y el Departamento de Supercómputo de la Dirección General de Servicios de Cómputo Académico en la Universidad Nacional Autónoma de México, se encuentra un sistema compuesto por veinte máquinas de varios tipos, marcas y versiones de sistemas operativos Unix. Cada una de ellas conectadas por redes locales, las cuales se comunican con redes exteriores de la UNAM, por medio de dispositivos de red llamados ruteadores.

El ruteador sirve para guiar paquetes de información en el camino correcto, es decir, es un dispositivo que se encarga de ordenar la información que llega a través de las redes, de acuerdo a la dirección especificada en el paquete.

Inicialmente se tenían tres ruteadores que conectaban a dos redes, donde las máquinas que en este momento existían en los dos departamentos se conectaban entre sí, con una supercomputadora de la UNAM y con el exterior.

El tipo de máquina que predominaba era el que se caracterizaba por su alta re-

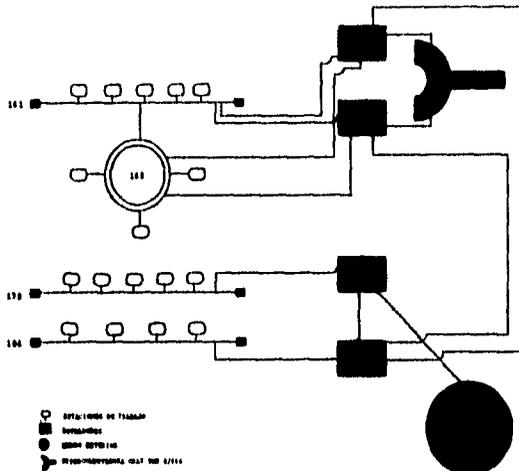


Figura 2.1: Esquema del sistema inicial

solución en la visualización de gráficos, esto se debía principalmente a que desde sus inicios este sistema estaba orientado para el área de investigación*.

2.2 Primeros problemas en el sistema inicial.

El primer problema fue que el número de máquinas en los dos departamentos no cubrían la demanda de los usuarios y el personal simultáneamente; los usuarios necesitaban tiempo de acceso en las máquinas para sus investigaciones, pero simultáneamente, tenían dudas de como manejar las aplicaciones o requerían de la compilación de otras más. El personal por otro lado, necesitaba de tiempo de acceso para resolver estas dudas o para compilar las aplicaciones y herramientas que los usuarios utilizaban a su vez para la representación visual de sus datos.

Este problema se solucionó, con la adquisición de un grupo de máquinas, suficientes para cubrir la demanda del personal y así remotamente, este pudiera [†] te-

*Los investigadores después de generar datos (en su mayoría obtenidos de la supercomputadora), necesitaban máquinas con alta resolución para visualizarlos.

†Se considera que algo se ejecuta remotamente, cuando la tarea es enviada a la estación utili-

ner acceso a las máquinas iniciales, para realizar sus tareas. La característica más relevante de este grupo fue que solo una de ellas contaba de sobra con suficiente espacio en disco y capacidad de memoria para las tareas que se le encomendaban y además disponía de estos recursos para compartir con las demás.

De acuerdo con las características anteriores se ideó un sistema de trabajo entre este grupo. A la máquina que contaba con los suficientes recursos para abastecer a las demás, se le encargó la tarea de compartir el espacio en disco con sus compañeras de grupo, así como información de usuarios.

A la primer máquina se le denominó *servidor* y a las restantes *clientes*.

Con este nuevo cambio y aumento de máquinas se manifestaron otros problemas los cuales fueron:

- Tráfico de red.
- Insuficiente espacio en disco en las primeras máquinas.
- La no-disponibilidad de la información

2.2.1 Tráfico de red

En el sistema en cuestión, el factor que intervino en un principio fue el aumento del tráfico de información, debido a que la máquina *servidor*¹ estaba conectada en una sola red y la mayoría de las máquinas *clientes*² no se encontraban bajo esa misma conexión de red. Así, el nivel de competencia por conectarse al servidor entre este grupo de máquinas *clientes* aumentaba, dando como resultado un tráfico de información denso y con ello ralentizar la velocidad de flujo.

2.2.2 Insuficiente espacio en disco

Este problema es uno de los más frecuentes en un sistema o conjunto de sistemas, debido a que afecta directamente al usuario y con ello se convierte en uno de los puntos a resolver para el administrador. En este caso para el grupo de máquinas que ya eran *clientes* del servidor, el problema se resolvió, sin embargo las máquinas restantes del sistema presentaban este problema pues la demanda de espacio en

zando una sesión no física a esta, esto es posible cuando la estación esta conectada en una red de cómputo.

¹Recuerdese que esta máquina comparte particiones de sus discos físicos con las máquinas que tienen necesidad de espacio en disco, es por esto que fue nombrada *servidor*.

²Al grupo de máquinas que comparten recursos con el servidor se les llamó *clientes*.

disco no disminuía. Así se pensó en crear grupos de máquinas que cumplieran con una tarea específica y cuya necesidad de espacio en disco coincidiera.

Los puntos que a continuación se mencionan, forman parte de las características requeridas para que una máquina formara parte de un grupo:

- Uso del espacio en disco.
- Usuarios.
- Aplicaciones.
- Necesidad de espacio temporal.

De acuerdo a estas características se formaron cuatro grupos de máquinas. Esto contribuyó a plantear el análisis de una forma más organizada y eficiente.

2.3 Grupos de máquinas que componen el sistema

Para conocer a fondo el camino que seguirá el análisis, es necesario saber las características de cada una de las máquinas, esto se reduce a conocer los patrones que predominan en cada grupo, es por ello que en esta sección se da una descripción de cada grupo.

2.3.1 Primer grupo

Las máquinas que integran el primer grupo, tienen la característica de carecer de recursos suficientes de disco y memoria. Un ejemplo claro es la insuficiencia de espacio en disco para instalar el sistema operativo y para mantener la información de usuarios locales.

Con estas características se pensó en crear un ambiente distribuido, esto es, un sistema en donde la información se almacene en una máquina y esta distribuyera a las demás, copias de dicha información.

2.3.2 Segundo grupo

En este grupo se tomó en cuenta, que la principal función de las máquinas integrantes se utilizan por lo general para ejecutar aplicaciones científicas y todas ellas requieren de un espacio temporal para guardar los resultados o para ejecutar las

aplicaciones. Además cuentan con un gran número de usuarios por lo que el espacio temporal de cada una de ellas tenía que ser grande para abastecer a todos los usuarios simultáneamente.

2.3.3 Tercer grupo

Este grupo se caracterizó por requerir espacio para almacenar sus aplicaciones, ya que una vez más, el problema del insuficiente espacio en disco se presentaba, había que almacenar aplicaciones científicas para varias áreas, como química, física, medicina, etc., así como herramientas para las máquinas como las páginas de manual y otros archivos.

2.3.4 Cuarto grupo

El último se integró por máquinas que requerían espacio temporal, debido a que son servidores de impresión. Para estas máquinas se necesitaba de espacio en disco para almacenar los archivos y peticiones de impresión. Este espacio por lo general debía ser lo suficientemente grande para almacenar un número promedio de archivos de impresión.

2.3.5 Servidor de disco.

De acuerdo a las necesidades que se tenían en cada uno de los grupos, se idearon nuevas tareas para que la máquina con grandes recursos de disco y memoria funcionara como servidor de un ambiente distribuido para las máquinas que carecían de estos recursos, y como servidor de disco para las restantes.

Con el fin de obtener un mayor control y orden en este servidor, se crearon particiones¹ en cada uno de los discos del servidor, de manera que se pudiera exportar solo el espacio requerido por cada conjunto.

2.4 Descripción de los problemas generados por la nueva estructura del sistema.

De los problemas enumerados al comienzo, con la adquisición del servidor de disco, fueron resueltos la duplicación de información y el insuficiente espacio en dis-

¹Áreas lógicas independientes de espacio en disco

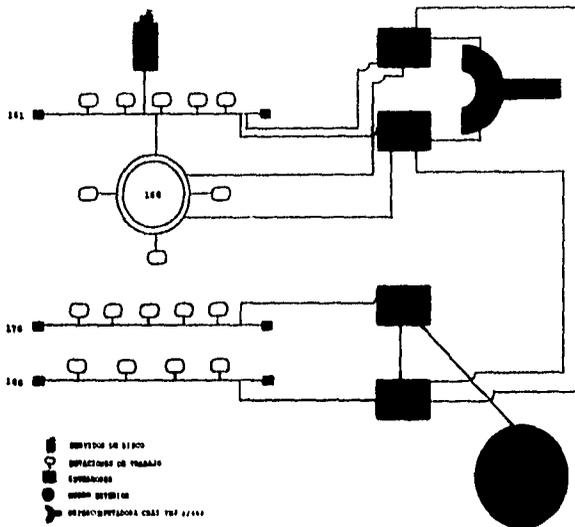


Figura 2.2: Esquema del sistema a la llegada del servidor de disco

co, sin embargo aumentó en gran medida el problema de tráfico de red. Generando problemas tales como:

- Disponibilidad de la información
- Interactividad del sistema
- Seguridad de la información.

2.4.1 Disponibilidad de la información

De acuerdo a la red en que cada máquina se encuentra conectada, así como al servidor de disco, depende el tráfico de información.

Como se observa en la figura 2.2, no todas las máquinas se conectaban en la red local del servidor de disco, por lo cual se establecieron niveles de competencia que se determinaron dependiendo de cuantos caminos tenían que pasar los paquetes de

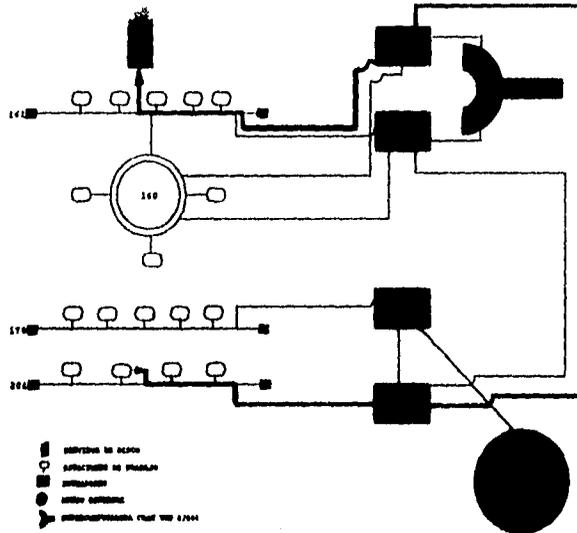


Figura 2.3: Esquema del camino que recorre una máquina para acceder información del servidor

información. Por ejemplo, si una máquina conectada en la red cuatro quería acceder información del servidor, tenía que competir primero con las máquinas de la misma red, segundo con los paquetes de información que llegaban tanto del exterior como de las redes locales y que contenían los dos ruteadores por los que tenía que pasar y por último con la red local del servidor (ver figura 2.3).

De esta forma, la comunicación entre el servidor de disco y las máquinas clientes, tenían que competir con otros paquetes externos dando como resultado dificultades en el tráfico en la red y ralentizando transferencia de información.

Este problema, disminuía si el servidor de disco se comunicaba con sus clientes en la misma red, ya que únicamente tenía que competir con máquinas y no con ruteadores y redes externas. Por lo que el acceso de información al servidor de disco era rápida y confiable.

2.4.2 Interactividad del sistema

En cuanto al ambiente distribuido, el problema de interactividad del sistema se notó claramente, pues la mayoría de las máquinas que conformaban el ambiente distribuido, no se conectaba directamente con la misma red del servidor, esto provocaba que si las redes externas tenían un alto porcentaje de carga de trabajo o su rendimiento era muy pobre, cualquier usuario en sesión dentro del ambiente distribuido no podía continuar con su sesión o sesiones, ya que la máquina daba la apariencia de no reconocer su entrada, o si el usuario quería entrar a sesión como la máquina cliente no podía acceder la información desde el servidor no reconocía al usuario.

2.4.3 Seguridad en la información

El hecho de que la información del servidor a los clientes tuviera que recorrer caminos externos y no administrados por el personal del Laboratorio de Visualización y el Departamento de Supercómputo, se consideró un problema. Aunque no visible, fue tomado en cuenta, pues el temor principal era la seguridad de autenticación de la información, producida por las fallas en la red. Por ejemplo si un ruteador fallaba y no mandaba la información completa o difícil si por una falla física de la red la información se perdiera.

2.5 Cambios físicos iniciales en el sistema

Una medida que redujo los problemas de hardware planteados, fue la implementación de una conexión de red más al servidor (Ver figura 2.4), y bajo esta misma conectar un grupo de máquinas cliente de los dos departamentos.

En la práctica se vio que este cambio mejoró el problema de seguridad en la información ya que ahora para estas máquinas, la transferencia de información no tenía que depender de ruteadores o redes externas para llegar al servidor.

La interactividad del ambiente distribuido de las máquinas clientes conectadas en esta nueva red mejoró notablemente, sin embargo, no ocurrió lo mismo para las máquinas no locales a cualquiera de las dos redes.

En resumen, con la nueva conexión de red, solo se intuía que la disponibilidad de información para las máquinas cliente integrantes del ambiente distribuido había mejorado y que el nivel de autenticación se había elevado.

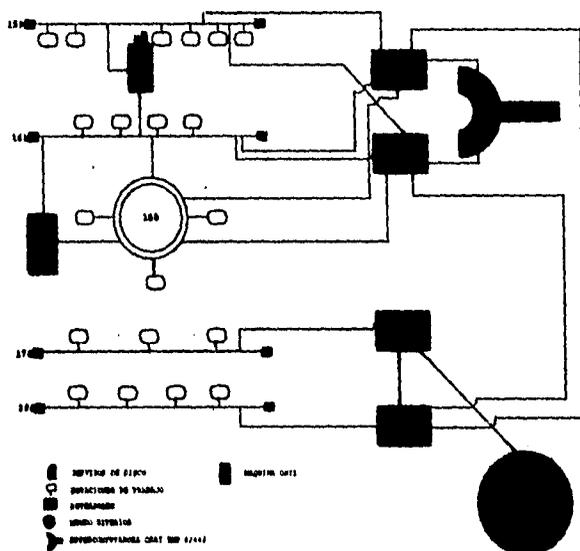


Figura 2.4: Esquema del sistema con el servidor conectado a dos redes

Sin embargo, era necesario tener bases sólidas que corroboraran estas intuiciones y además conocer el tiempo de retardo de transmisión de las peticiones de los clientes al servidor con esta nueva topología, para saber que porcentaje había mejorado este cambio y de ser posible conocer si este se podía mejorar aún más.

Lo anterior dió como resultado la necesidad de crear herramientas basadas en modelos probabilísticos, que describan con antelación los tiempos de retardo de transmisión o tiempos de retardo de ejecución de una petición del ambiente distribuido o de uso de disco, los cuales generan fallas en el sistema híbrido de cómputo, y con ello poder mejorar el uso del mismo.

FALTA PAGINA

No.

23

Parte III

**CONCEPTOS BÁSICOS DE
CÓMPUTO**

Capítulo 3

Servicios de Red (NFS, NIS)

Para detectar las fallas en el sistema híbrido de cómputo anteriormente descrito, es necesario conocer su funcionamiento. Es por esto que es importante saber los conceptos básicos con los que se apoya para establecer la comunicación en red de sus integrantes, así como los servicios de red que aplica en su desarrollo.

3.1 Remote Procedure Call (RPC)

El protocolo *Remote Procedure Call (RPC)* se clasifica de acuerdo con el modelo OSI¹, dentro de la capa de sesión. Fue especialmente diseñado para un servicio de red llamado *Network File System, NFS*, que distribuye sistemas de archivos sin embargo debido al buen resultado que dió para transportar llamadas remotas se convirtió en un protocolo utilizado para otros servicios de red.

Como su nombre lo indica este protocolo sirve como guía para ejecutar llamadas remotas o procedimientos a un alto nivel de programación. Este modo de trabajo facilita en mucho las tareas del programador en el diseño e implementación de tareas distribuídas.

3.1.1 Modo de operación

El método de trabajo del RPC, es similar al método de una llamada local¹. En el caso de una llamada local, los parámetros son asignados al procedimiento local en

¹Ver apéndice B

¹Se considera llamada local, a la ejecución de un proceso en el mismo sistema en donde se generó la petición

un punto específico del sistema, en pocas palabras, el cliente procesa y ejecuta una tarea en su propio espacio.

La ejecución de un RPC consiste de los siguientes pasos:

1. Activación del programa cliente. Los parámetros de la petición son empaquetados.
2. Manda la petición y se desempaquetan los parámetros de la solicitud en el programa del servidor.
3. Se ejecuta la petición en el servidor.
4. El servidor empaqueta los resultados y los envía al cliente.
5. El cliente desempaqueta los resultados, y los ejecuta en tareas específicas.

En la figura 3.1 se muestran en forma gráfica estos pasos. La manera de como empaqueta y desempaqueta la información enviada es posible a través de rutinas de librerías que están ligadas a el programa.

El programador puede definir un número arbitrario de procesos entre el cliente y el servidor, con parámetros opcionales y resultados igualmente arbitrarios. Al grupo de procesos que se forman se le llama servicio de RPC; a cada uno de estos servicios se les asigna un número de programa (*program number*). Todos los paquetes de este tipo tienen un encabezado en el cual se especifica:

- Número de programa (*program number*)
- Número de proceso (*procedure number*)
- Número de versión (*version number*)

La numeración del proceso es secuencial al del número de programa, mientras que el número de versión corresponde a que el servidor puede soportar varias formas del mismo programa, y este asigna el número de versión a utilizar. Por convención el proceso 0 es asignado para propósito de pruebas de los programas, este contiene una rutina nula, así es muy fácil probar si los servidores de RPC funcionan bien.

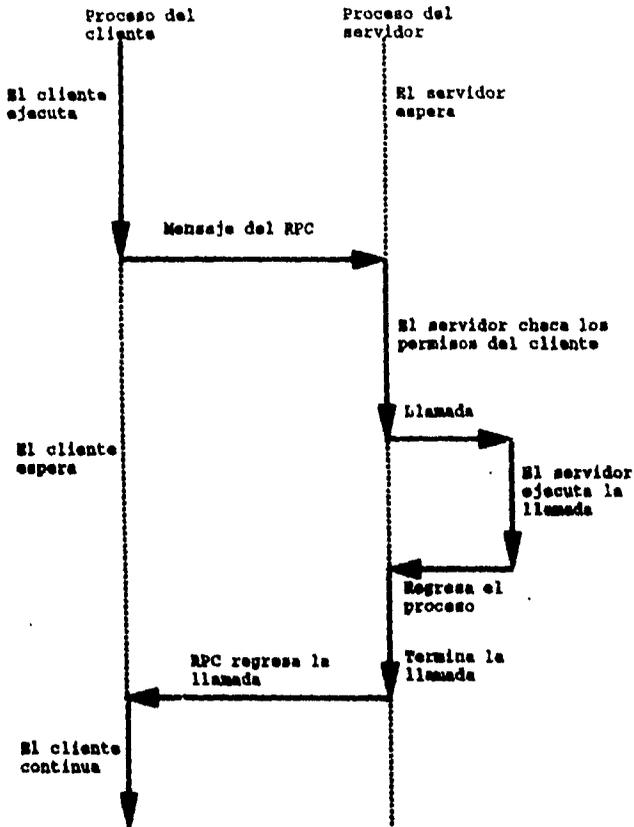


Figura 3.1: Ejecución de un RPC

3.1.2 Independencia del RPC con los protocolos de Transporte

El protocolo *RPC* es independiente del tipo con el cual el mensaje es transportado, razón por la cual, un programa en *RPC* puede ser ejecutado bajo *TCP* o *UDP*. Algunos servidores ofrecen sus servicios de *RPC* bajo los dos protocolos de transporte.

En la mayoría de los casos, *RPC* envía los mensajes cortos a través de *UDP*, ya que *UDP* no puede transportar paquetes arbitrariamente grandes pues como es de comunicación orientada, al transportar mensajes muy largos la información puede perderse ya que *UDP*, no garantiza la integridad de la información.

Por este hecho, cuando *RPC* requiere mandar mensajes cuyo tama no exceda el límite de *UDP*, estos mensajes son enviados a través de *TCP*.

Con base en la confiabilidad de la información que ofrece *TCP*, para el protocolo *RPC* se creo un modo de operación que actua de forma no interactiva ente el cliente y el servidor. Las peticiones del *RPC*, son enviadas de una manera rápida sin esperar una respuesta a una llamada previa.

Los resultados de velocidad de transmisión son considerablemente altos pues se omiten los tiempos de espera de peticiones y respuestas.

3.1.3 Protocolo Port mapper

El protocolo *Protocolo Port mapper*, junto con el protocolo *RPC*, permiten la comunicación entre un cliente y un servidor. El protocolo *Port mapper* se encarga principalmente de enviar información del tipo de versión del servicio que en la máquina cliente esta configurado, del identificador del procedimiento (Como se llama el servicio de red que desea el cliente) y el protocolo de transporte utilizado (*UDP*, *TCP*).

Cuando se dió a conocer la primer versión de este protocolo, muchas estaciones lo implementaron en sus sistemas, sin embargo con el paso del tiempo y de acuerdo a la experiencia fueron mejorandolo de esta forma surgieron las siguientes versiones.

El problema de la variedad de versiones es que como han ido transformandose dos versiones distintas no se comprenden y con ello el servicio de red no se satisface, de esta forma este protocolo le tiene que informar a la máquina remota el tipo de versión para que ésta pueda corresponderle con la misma.

3.1.4 Modelo Cliente-Servidor

Por la construcción de RPC es posible establecer un modelo Cliente-Servidor, es decir, el cliente ejecuta una llamada remota mediante un proceso local, este envía el mensaje a otra máquina que cuente con los recursos necesarios para ejecutarla y una vez que termina, manda el resultado a la máquina quien realizó la petición; es por eso que a la primera se le llama cliente y a la segunda se le llama servidor, usualmente los clientes son máquinas que no cuentan con los recursos suficientes para ejecutar localmente las llamadas o procesos.

Los dos ejemplos más comunes que representan este modelo son los servicios de red¹: *Network File System (NFS)* y *Network Information System (NIS)*. Estos servicios de red fueron construídos por *Sun Microsystems Inc.* e implementados en la mayoría de las arquitecturas Unix.

Nuestro estudio se basa en un sistema de cómputo híbrido con ambientes distribuídos de información y de espacio en disco basados en estos dos servicios de red.

A continuación se da una descripción de los puntos más importantes de *Network File System (NFS)* y *Network Information System (NIS)*.

3.2 Network File System (NFS)

Network File System (NFS), fue desarrollado y distribuído por *Sun Microsystems Inc.* La filosofía de este servicio es distribuir espacio en disco a un número de máquinas que lo requieran.

Un servidor de **NFS**, tiene uno o más sistemas de archivos que son compartidos por clientes de **NFS** ambos comunicados por red. En las máquinas clientes los sistemas de archivos externos funcionan de la misma manera que los sistemas de archivos locales.

La principal tarea de **NFS** es facilitar el acceso a disco remotamente de una forma transparente, básicamente ejecuta programas que permiten el acceso de lectura y escritura a archivos en el servidor mediante la red que comunica al servidor con sus clientes. De esta forma el usuario puede acceder a la información de un disco vía red sin necesidad de hacer uso de las transferencias de archivos o abrir una sesión remota.

La característica de la transparencia de este servicio se basa en el hecho que aunque depende de la red de comunicación entre el cliente y el servidor, el acce-

¹Un servicio de red, se define como uno o más programas ejecutados remotamente

Capa	Protocolo	Estandar
Física/Enlace	CSMA/CD	Ethernet IEEE 802.3
Red	IP	RFC 791
Transporte	TCP	RFC 793
	UDP	RFC 768
Sesión	RPC	RFC 1050
Presentación	XDR	RFC 1014
Aplicación	NFS MOUNT	RFC 1094

Tabla 3.1: Protocolos utilizados para el funcionamiento de NFS.

so a disco es tan rápido, que cuando los factores que intervienen, tienen un buen rendimiento, la diferencia entre un disco local y un disco via NFS es despreciable, y cuando el desempeño del disco local es bajo, el uso de los discos NFS resulta el más apropiado y más barato.

Para su funcionamiento *NFS* requiere de un conjunto de protocolos, en la figura 3.2 se muestra la estructura de *NFS* comparada con el modelo OSI; de la comparación resulta suponer que teóricamente *NFS* es un protocolo independiente, sin embargo en la práctica se considera como una aplicación pues en la implementación se incorporan los mismos protocolos que en cualquier aplicación. La tabla 3.1, muestra una lista de los protocolos estandar usados en *NFS*.

En la capa física y de enlace se tiene como estandar a la red de tipo *ethernet*⁵, la cual se basa en el protocolo denominado *CSMA/CD*⁶, sin embargo existen arquitecturas operar con otro tipo de redes, tal es el caso de la red *fdi*⁷.

NFS se basa en el protocolo *TCP/IP* (Ver sección B.4), ya que este software es uno de los más comunes protocolos en sistemas Unix, utilizando los dos tipos de transportes llamados *UDP* y *TCP*.

Además de estos protocolos, *NFS* utiliza un protocolo en la capa de sesión el

⁵Ethernet es una Área de Red Local (Local Area Network), que pertenece a la capa de enlace en el modelo OSI, transmite información de 10 Mb/s y cuyo número máximo de estaciones conectadas a ella 1024.

⁶Carrier Sense Multiple Access with Collision Detection, es un protocolo de envío de información a través de un medio físico. Véase sección

⁷FDDI (Fiber Distributed Data Interface), pertenece al grupo de Áreas de Red Local (Local Area Network), ideada para ofrecer un alto rendimiento, transmitiendo 100(Mb/s), con un máximo de 500 estaciones.

cual actúa como chasis de las peticiones entre el cliente y el servidor, este protocolo es el llamado **Remote Procedure Call (RPC)** (Véase sección 3.1).

RPC forma las bases necesarias para el intercambio de mensajes en todas las aplicaciones de NFS.

3.2.1 Protocolo Port mapper trabajando con RPC para NFS

Este protocolo establece la comunicación entre un cliente y el servidor de NFS, enviando peticiones de NFS y su versión.

En Unix este protocolo está implementado dentro del programa llamado **portmap**, las conexiones entre el cliente y el servidor sin él son prácticamente imposibles. Existe una herramienta en Unix en la cual es posible conocer la tabla de información que este protocolo revisa al conectarse con otra máquina y resulta un buen detector de errores en un sistema con el servicio de red NFS.

3.2.2 Características especiales de NFS

Los creadores de NFS, planearon un servicio robusto de accesos de archivos remotos. La solución fue idear un protocolo en el cual el servidor no tuviera almacenada ninguna información acerca del estado o progreso del diálogo con el cliente, por ejemplo la idea era que si el cliente abría o cerraba un archivo el servidor no registrara este procedimiento. Contrariamente a los sistemas Unix estándar, en donde cualquier función de entrada o salida, son supervisadas por el servidor.

En la práctica esta función se implementó con una función llamada *shandle*. De este modo un servidor puede fallar y en seguida reestablecerse, ejecutando operaciones de lectura transparentemente en cuanto al cliente respecta. Mientras el servidor no está disponible, el programa del cliente retransmite la petición de lectura al servidor en intervalos de tiempo, hasta que el servidor responde.

El tiempo que tarda entre los intentos de peticiones incrementa de manera exponencial y el número de retransmisiones puede limitarse mediante herramientas especiales de Unix tal es el caso del comando *mount*. **.

3.2.3 Entrada/Salida Asíncrona en el cliente de NFS

Para mejorar el desempeño de información, en Unix los bloques de datos son mantenidos en buffers cuyo nombre es *buffer cache*. Este mecanismo es utilizado en

**Véase [S].

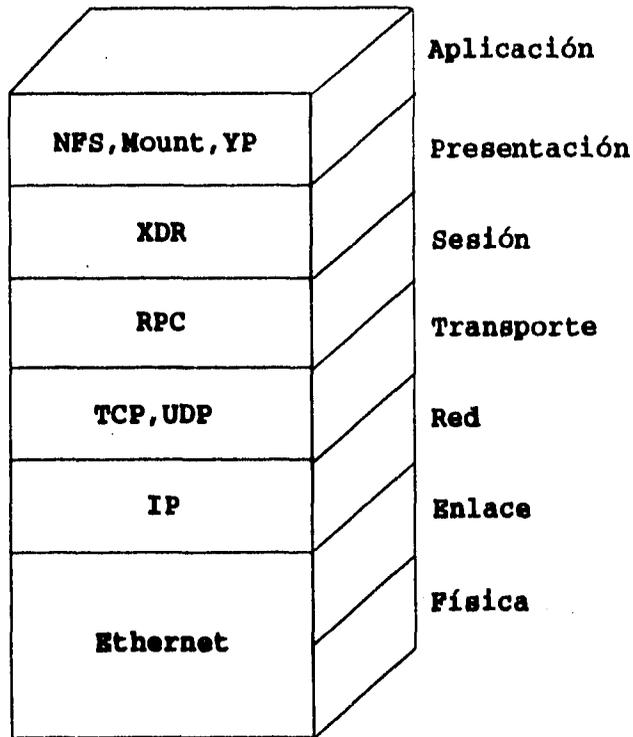


Figura 3.2: Modelo OSI de la estructura de NFS

las operaciones de lectura y escritura vía NFS.

Usualmente, las operaciones de entrada y salida físicas se realizan de manera asíncrona, esto es, para una rutina de lectura el sistema de entrada/salida lee el próximo bloque de un archivo por adelantado. Contrariamente para la rutina de escritura donde los bloques de datos son escritos en el buffer y más tarde transferidos a disco cuando el límite del bufer sea alcanzado.

Estos procedimientos son conocidos como *los que leen por adelantado y los que escriben más tarde (read ahead and delayed write)*,

Dichos procedimientos son implementados en NFS por trabajos que se ejecutan continuamente llamados *daemons*, en NFS se les conoce como *biobd*. El *daemon biobd*, mejora el desempeño de datos enviados por un cliente de NFS, pero no es necesario para la operación de NFS. Sin este demonio las lecturas y escrituras podrían no ejecutarse de tipo asíncrono lo que daría como resultado el lento acceso a los archivos y con ello la no transparencia.

Con el fin de aumentar el rendimiento en las tareas de NFS, normalmente son cuatro *biobd* que operan en paralelo; ya que cuando una petición es de solo lectura, el disco duro se encuentra ocioso y tiene que esperar hasta que esta petición sea servida para ejecutar la próxima petición. De esta manera si la próxima petición es de escritura el disco duro puede comenzar a trabajar en el mismo tiempo que la otra petición también esta siendo atendida, consumiendo con esto menos tiempo que el normal. Sin embargo, también es importante tomar ciertas medidas con respecto a cuantos *daemons* estaran dados de alta, por ejemplo hasta cuatro *daemons* el rendimiento crece sin embargo si se tiene una cantidad considerablemente mayor que cuatro al ejecutarse un número de peticiones pueden "sofocar" el medio de transporte de red, dando como resultado que el rendimiento del sistema se degrade.

3.2.4 Restricciones en la operación de NFS

El diseño de NFS, es el permitir operaciones equivalentes a los sistemas de archivos locales; sin embargo debido a que el servidor no sabe las operaciones que sus clientes estan ejecutando en sus discos, no puede saber si el cliente tuvo problemas con las tareas; estos errores pueden ser corregidos por el lado del cliente.

Un ejemplo que muestra este problema se presenta cuando se quiere borrar un archivo abierto por otro cliente. En un sistema de archivos local es posible saber que un archivo se encuentra abierto y no puede ser borrado ya que el sistema sabe que tiene un proceso con el archivo abierto. Mediante NFS, eso no es posible pues el servidor no puede saber las condiciones de los archivos y tareas de sus clientes.

Otras restricciones son mencionadas a continuación:

- Operaciones de escritura, cuya longitud exceda el máximo propuesto por el protocolo RPC.
- El tiempo aproximado que tarda un buffer en hacer los cambios en un disco es de 30 segundos, en este caso si un cliente cambia los permisos de un directorio existe un intervalo de 30 segundos para que otro cliente pueda tener los permisos anteriores.
- Cuando un super-usuario de una máquina cliente crea archivos dentro de los discos del servidor y este no reconoce el identificador de la cuenta cliente, los archivos creados por el super-usuario cliente aparecen con dueños erróneos.
- El acceso simultáneo de más de un cliente a un mismo archivo en el servidor puede generar inconsistencias.

Es así como se tiene que estudiar la implementación de un sistema con NFS, pues muchas veces se sacrifica rendimiento por obtener un bajo costo de equipo. Si se tiene un sistema con NFS, se debe de analizar si el beneficio que ofrece es mayor que las desventajas que genera.

3.3 Network Information Service (NIS)

El servicio de red **Network Information Service (NIS)**, al igual que NFS, fue desarrollado y distribuido por *Sun Microsystems Inc.*. La filosofía de NIS, es compartir información en un grupo de máquinas de manera transparente. Dicha información se ubica en las bases de datos de una o más máquinas que pertenecen al grupo, las cuales exportan la información a las restantes integrantes del grupo vía red.

Las bases de datos comprenden información del tipo de protocolos que utilizan, estaciones de trabajo, aplicaciones y usuarios.

La principal tarea de NIS, es distribuir información en un grupo de máquinas a través de la red de manera transparente, sin embargo esta tarea se combina con NFS para brindar un ambiente de acceso de lectura de información y escritura de sistemas de archivos.

NIS utiliza para su funcionamiento el modelo *Cliente-Servidor*, las máquinas que almacenan la información se les llama servidores y al conjunto de máquinas que accesan esta información se les llama clientes.

Capa	Protocolo	Estandar
Física/Enlace	CSMA/CD	Ethernet IEEE 802.3
Red	IP	RFC 791
Transporte	UDP	RFC 768
Sesión	RPC	RFC 1050
Presentación	XDR	RFC 1014
Aplicación	NIS(YP)	RFC 1094

Tabla 3.2: Protocolos utilizados para el funcionamiento de NIS.

El grupo de servidores, se compone de un *servidor central y esclavos*. La población construída por los clientes y servidores se le llama *dominio*.

NIS fue construído con base en el protocolo **Remote Procedure Call (RPC)**. Utiliza UDP para transportar las peticiones del cliente al servidor. Estas peticiones son de sólo lectura por lo que para su funcionamiento no requiere utilizar espacio en disco. Comparado con el modelo *OSI* la estructura de NIS se muestra en la tabla 3.2

Las bases de datos que NIS utiliza son archivos en texto claro llamados *mapas*, estos mapas contienen la información de los clientes y servidores acreditados para acceder la información.

NIS esta implementado en el sistema Operativo Unix mediante librerías y por un daemon llamado *ypserv*. Las solicitudes de este servicio se agrupan en tres conjuntos que son: *llamadas internas de NIS, de mapas y de clientes*.

El servicio de NIS utiliza para las solicitudes de los clientes, una sola operación la cual sirve las peticiones de los clientes. Esto es se tiene un solo servidor que atiende cada una de las llegadas provenientes de los clientes. La principal función es encontrar para qué dominio la petición tiene que ser servida, si el dominio no esta funcionando entonces da por terminada esa petición y comienza a servir otra.

En la práctica se recomienda que un servidor de NIS, sea un cliente de NIS; de esta forma se pueden hacer pruebas de reconocimiento de hosts, sin intervenir los efectos de red u otros, aislando el problema y asegurando que el servidor esta dando un buen servicio. Esto trae consigo un mejor comportamiento en todo el ambiente distribuído.

Una recomendación más, es tener varios servidores esclavos dados de alta, pues cuando uno de ellos no este disponible, los clientes pueden tomar la información

de cualquier otro, dando como resultado un mejor servicio.

Este último punto sirve también para desahogar el nodo de red y distribuir los clientes con el fin de evitar la carga de tráfico y con ello los errores subsecuentes.

3.3.1 Ejemplo de ejecución de una petición de NIS

Este ejemplo pretende ilustrar como funcionan las librerías y llamadas al sistema en el servicio de red NIS.

Supongase que un usuario ejecuta un `ls -l` desde una cuenta en NIS:

- Cuando el usuario corre este comando, el proceso `ls` necesita encontrar el identificador correspondiente del propietario de cada archivo. En este caso el comando llama al proceso `getpwuid` este proceso tiene la función de encontrar la contraseña del propietario del archivo y con ello su identificador.
- El archivo local de la estación es checado, al no encontrar respuesta busca en la información de la base de datos de NIS.
- La función `getpwuid` graba el nombre del dominio donde encontró la información y busca el servidor que esta a cargo de este. Con el proceso `ypbind` sabe si el servidor esta disponible, si este lo esta entonces el proceso sigue, otro caso espera de dos a tres minutos y si el servidor no le responde entonces suspende el envío.
- El cliente procesa la llamada y la envía al servidor.
- El servidor recibe la llamada, checa si la información existe. Si es así envía el identificador del usuario y su contraseña. Si no lo encuentra la manda de regreso a través de la función `getpwuid`.

Cuando NIS se conjuga con NFS y el rendimiento de ambos es bueno, se ejecuta un ambiente remoto distribuido semejante a un ambiente local en cada máquina cliente, de esta manera se logra el beneficio de dar servicio a los usuarios sin necesidad de utilizar espacio en disco en cada máquina cliente. En caso contrario y si el grupo de máquinas es pequeño se recomienda establecer una sesión local.

En resumen el uso del servicio de red NIS es útil para un grupo de máquinas grande con información de usuarios, máquinas, protocolos, etc., en común, y cuyo rendimiento de estos es estable.

Parte IV

**MODELOS PROBABILÍSTICOS
EN PROBLEMAS DE
COMPUTACIÓN**

Capítulo 4

Construcción de los Modelos Probabilísticos.

Una vez que ya se tienen las bases necesarias para entender el funcionamiento de comunicación y de los servicios de red del sistema de cómputo. En este capítulo, se construyen los modelos probabilísticos de cada uno de los factores que afectan la comunicación y el desempeño de los servicios de red, en el sistema híbrido de cómputo.

4.1 Preliminares

Para construir los modelos probabilísticos es necesario recordar algunos conceptos, por ejemplo: los Procesos de Poisson, los Procesos de Nacimiento y Muerte y algunos otros contenidos en la Teoría de Colas.

4.1.1 Procesos de Poisson

Una aproximación a los procesos de Poisson comienza con los procesos Bernoulli, que se caracterizan por las siguientes propiedades:

1. El experimento consiste en n ensayos repétidos.
2. Los resultados de cada uno de los ensayos pueden clasificarse como: un éxito o un fracaso.
3. Los ensayos son independientes.

La variable aleatoria número k de éxitos en n experimentos Bernoulli recibe el nombre de **binomial**. La distribución de probabilidad de esta variable aleatoria discreta se llama **distribución binomial** y sus valores se representan por $h(k; n, p)$. Si la probabilidad de un éxito es p (la de los fracasos es $1 - p$) entonces la probabilidad de tener una secuencia de k éxitos es

$$p^k(1-p)^{n-k}$$

El número total de puntos muestrales en el experimento que tiene k éxitos es igual al número de particiones en n en dos grupos, con k en un grupo y $n-k$ en el otro, esto se expresa de la siguiente forma:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Debido a que estas particiones son mutuamente excluyentes, se suman las probabilidades de todas las diferentes particiones para obtener la fórmula general. Entonces la probabilidad de obtener k éxitos en n eventos es:

$$\mathbf{P}[k \text{ éxitos en } n \text{ eventos}] = h(k; n, p) \triangleq \binom{n}{k} p^k (1-p)^{n-k} \quad (4.1)$$

Dos de las propiedades importantes de la distribución binomial son: El número de éxitos representados por k , los k éxitos se distribuyen como una Binomial, y la probabilidad de que el evento sea un éxito es $\frac{k}{n}$. La segunda propiedad es que carece de memoria, por lo que una secuencia de eventos de tipo Bernoulli es una cadena de Markov. Una cadena de Markov es una sucesión de variables aleatorias con un número numerable de estados y con la propiedad de *pérdida de memoria*, es decir, la probabilidad del estado de un evento depende solo del estado anterior. Esto se expresa mejor en la ecuación 4.2 para el estado $N+1$, donde S_i es el evento cuyo resultado fué un éxito en el i -ésimo estado.

$$\mathbf{P}(S_{N+1} = k_{N+1} / S_1 = k_1, S_2 = k_2 \dots S_N = k_N) = \mathbf{P}(S_{N+1} = k_{N+1} / S_N = k_N) \quad (4.2)$$

Si se tienen n experimentos Bernoulli en un segundo de tiempo y el número de experimentos comienza a incrementarse sin límite y la probabilidad de un éxito para estos experimentos disminuye a cero, entonces la intensidad del número de

éxitos en un segundo se denotará por λ y por lo tanto se puede interpretar como la tasa de ocurrencia de los éxitos en n eventos (np), esto es:

$$\lambda \triangleq np$$

entonces se tiene que:

$$\lim_{n \rightarrow \infty, p \rightarrow 0} b(0; n, p) = \lim_{n \rightarrow \infty, p \rightarrow 0} \left(1 - \frac{\lambda}{n}\right)^n = e^{-\lambda} \quad (4.3)$$

Demstración

De acuerdo a la distribución binomial y a la relación $\lambda \triangleq np$

$$\begin{aligned} b(0; n, p) &= \binom{n}{0} p^0 (1-p)^{n-0} \\ &= \left(1 - \frac{\lambda}{n}\right)^n \end{aligned} \quad (4.4)$$

por otra parte

$$\lim_{n \rightarrow \infty, p \rightarrow 0} \left(1 - \frac{\lambda}{n}\right)^n = \lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^n \quad (4.5)$$

para demostrar lo anterior basta con observar que la sucesión

$$s_n = \log \left(1 - \frac{\lambda}{n}\right)^n \quad (4.6)$$

posee el límite $-\lambda$, puesto que la función exponencial es continua.

$$\begin{aligned} s_n &= n \log \left(1 - \frac{\lambda}{n}\right) \\ &= n \int_1^{1 - \frac{\lambda}{n}} \frac{1}{\varpi_n} d\varpi_n \\ &= n \frac{1}{\varpi_n} \left[\left(1 - \frac{\lambda}{n}\right) - 1 \right] \\ &= \frac{-\lambda}{\varpi_n} \end{aligned} \quad (4.7)$$

donde

$$1 \leq \varpi_n \leq 1 - \frac{\lambda}{n}$$

y debido a que

$$\lim_{n \rightarrow \infty} \frac{\lambda}{n} = 0$$

entonces también se tiene

$$\lim_{n \rightarrow \infty} \varpi_n = 1 \Rightarrow \lim_{n \rightarrow \infty} s_n = -\lambda$$

por lo que se concluye que

$$\lim_{n \rightarrow \infty, p \rightarrow 0} \left(1 - \frac{\lambda}{n}\right)^n = e^{-\lambda} \quad (4.8)$$

y

$$\begin{aligned} \lim_{n \rightarrow \infty, p \rightarrow 0} \frac{b(k; n, p)}{b(k-1; n, p)} &= \lim_{n \rightarrow \infty, p \rightarrow 0} \frac{\binom{n}{k} p^k (1-p)^{n-k}}{\binom{n}{k-1} p^{k-1} (1-p)^{n-(k-1)}} \\ &= \lim_{n \rightarrow \infty, p \rightarrow 0} \frac{\frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}}{\frac{n!}{(k-1)!(n-(k-1))!} p^{k-1} (1-p)^{n-(k-1)}} \\ &= \lim_{n \rightarrow \infty, p \rightarrow 0} \frac{n-k+1}{k} \frac{p}{1-p} \\ &= \lim_{n \rightarrow \infty, p \rightarrow 0} \frac{np - kp + p}{k(1-p)} \\ &= \lim_{p \rightarrow 0} \left(\frac{\lambda}{k} - p + \frac{p}{k} \right) \frac{1}{1-p} \\ &= \frac{\lambda}{k} \end{aligned} \quad (4.9)$$

por otro lado:

$$\begin{aligned}
 b(k; n, p) &= \binom{n}{k} p^k (1-p)^{n-k} \\
 &= \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \\
 &= \frac{n(n-1) \cdots (n-k+1)}{k!} p^k (1-p)^{n-k} \\
 &= \frac{n(n-1) \cdots (n-k+1)}{k!} \left(\frac{\lambda}{n}\right)^k \left(1 - \frac{\lambda}{n}\right)^{n-k} \\
 &= \left(\frac{\lambda^k}{k!}\right) \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-k} \frac{n(n-1) \cdots (n-k+1)}{n^k} \quad (4.10)
 \end{aligned}$$

Ahora que

$$\frac{n(n-1) \cdots (n-k+1)}{n^k} = \left(1 - \frac{1}{n}\right) \left(1 - \frac{2}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right),$$

y tomando los límites

$$\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right) = 1 \quad (4.12)$$

$$\lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^{-k} = 1 \quad (4.13)$$

*Puesto que

$$\frac{n(n-1) \cdots (n-k+1)}{n^k} = \frac{\overbrace{n(n-1) \cdots (n-k+1)}^{k \text{ términos}}}{\underbrace{n \cdots n}_{k \text{ veces}}}$$

se tiene que

$$\begin{aligned}
 \frac{n(n-1) \cdots (n-k+1)}{n^k} &= \frac{n}{n} \frac{n-1}{n} \cdots \frac{n-k+1}{n} \\
 &= 1 \left(1 - \frac{1}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right) \quad (4.11)
 \end{aligned}$$

De acuerdo a las ecuaciones 4.8, 4.10, 4.12 y 4.13 se tiene que:

$$\lim_{n \rightarrow \infty, p \rightarrow 0} b(k; n, p) = e^{-\lambda} \frac{\lambda^k}{k!} \quad (4.14)$$

Con base en lo anterior se puede observar que las distribuciones binomial y Poisson sirven para modelar las llegadas de eventos ¹. Esto es, en el caso de las fuentes que transmitan un número finito de mensajes las llegadas de estos se distribuyen como una binomial ($b(k; n, p)$) y si el número de mensajes para cada fuente es infinito y la probabilidad de que se transmita este mensaje es p , entonces se distribuyen como una Poisson ($Po(k, \lambda)$)

Un proceso Poisson también se puede representar a través de los intervalos de tiempo de llegada para cada evento:

Si T representa el tiempo (aleatorio) que tarda en llegar el primer mensaje (partiendo de un tiempo inicial $t = 0$), la función de probabilidad para T será:

$$\begin{aligned} F_T(t) = \mathbf{P}[T \leq t] &= 1 - \mathbf{P}[\text{ninguna llegada en el intervalo } (0, t)] \\ &= 1 - P_0(t) \\ &= 1 - e^{-\lambda t} \end{aligned} \quad (4.15)$$

entonces la función de densidad es:

$$f_T(t) dt \triangleq \mathbf{P}[t < T \leq t + dt] = \lambda e^{-\lambda t} dt \quad (4.16)$$

En las ecuaciones 4.15 y 4.16 se tiene una distribución exponencial, esto significa que la esperanza es:

$$E(T) = \int_0^{\infty} t \lambda e^{-\lambda t} dt$$

integrando por partes y tomando $\zeta = t\lambda$ se obtiene

$$\begin{aligned} E(T) &= (-\zeta e^{-\zeta} |_0^{\infty}) - \frac{1}{\lambda} \int_0^{\infty} -e^{-\zeta} d\zeta \\ &= (-e^{-\zeta} |_0^{\infty}) \frac{1}{\lambda} = \frac{1}{\lambda} \end{aligned} \quad (4.17)$$

¹En este caso el evento son las llegadas de peticiones de los clientes al servidor

Esto muestra la relación existente entre la distribución poisson y la exponencial. La ecuación 4.18 da un ejemplo de la propiedad de la no memoria donde se toma en cuenta la distribución exponencial. Por ejemplo si llega una petición al tiempo $t = 0$ y ninguna en el intervalo $(0, t)$ entonces la probabilidad de que una nueva petición llegue en el intervalo $(0, t + \theta)$, es la probabilidad de una petición en el intervalo $(t, t + \theta)$ dado que no hubo ninguna petición en el intervalo $(0, t)$, i.e.:

$$\begin{aligned} \mathbf{P}\{T \leq t + \theta | T \geq t\} &= \mathbf{P}\{t < T \leq t + \theta | T \geq t\} \\ &= \frac{F_T(t + \theta) - F_T(t)}{1 - F_T(t)} \\ &= \frac{1 - e^{-\lambda(t + \theta)} - 1 + e^{-t\lambda}}{e^{-t\lambda}} = 1 - e^{-t\lambda} \quad (4.18) \end{aligned}$$

La ecuación 4.18 confirma nuevamente la propiedad de carencia de memoria de la exponencial, i.e. la probabilidad de que una petición llegue depende únicamente de la longitud del intervalo y no de las llegadas anteriores. Una característica propia de estos procesos es que la probabilidad de que más de un resultado ocurra en un intervalo de tiempo muy corto o en una región muy pequeña es despreciable, y por ello los procesos de poisson pueden representarse la forma:

$$P_0(\delta) = 1 - \lambda\delta + \sigma(\delta) \quad (4.19)$$

$$P_1(\delta) = \lambda\delta + \sigma(\delta) \quad (4.20)$$

$$P_i(\delta) = \sigma(\delta), \quad i \geq 2 \quad (4.21)$$

donde $\sigma(\delta)$ representa los términos de orden mayor en t tal que el $\lim_{\delta \rightarrow 0} [\sigma(\delta)/\delta] = 0$. Las ecuaciones 4.19, 4.20 y 4.21 dicen que para un intervalo con un incremento muy corto el que se tenga más de una llegada resulta despreciable y la probabilidad de la llegada de una solicitud es proporcional a la duración de este intervalo (δ) .

4.1.2 Procesos de Nacimiento-Muerte

La generalización de los Procesos de Poisson da como resultado los Procesos de Nacimiento, estos sirven para representar modelos de llegadas. Sin embargo en el estudio de sistemas de cómputo es necesario modelar no solo las llegadas sino las

Esto muestra la relación existente entre la distribución poisson y la exponencial. La ecuación 4.18 da un ejemplo de la propiedad de la no memoria donde se toma en cuenta la distribución exponencial. Por ejemplo si llega una petición al tiempo $t = 0$ y ninguna en el intervalo $(0, t)$ entonces la probabilidad de que una nueva petición llegue en el intervalo $(0, t + \theta)$, es la probabilidad de una petición en el intervalo $(t, t + \theta)$ dado que no hubo ninguna petición en el intervalo $(0, t)$, i.e.:

$$\begin{aligned} \mathbf{P}[T \leq t + \theta | T \geq t] &= \mathbf{P}[t < T \leq t + \theta | T \geq t] \\ &= \frac{F_T(t + \theta) - F_T(t)}{1 - F_T(t)} \\ &= \frac{1 - e^{-\lambda(t + \theta)} - 1 + e^{-t\lambda}}{e^{-t\lambda}} = 1 - e^{-t\lambda} \quad (4.18) \end{aligned}$$

La ecuación 4.18 confirma nuevamente la propiedad de carencia de memoria de la exponencial, i.e. la probabilidad de que una petición llegue depende únicamente de la longitud del intervalo y no de las llegadas anteriores. Una característica propia de estos procesos es que la probabilidad de que más de un resultado ocurra en un intervalo de tiempo muy corto o en una región muy pequeña es despreciable, y por ello los procesos de poisson pueden representarse la forma:

$$P_0(\delta) = 1 - \lambda\delta + \sigma(\delta) \quad (4.19)$$

$$P_1(\delta) = \lambda\delta + \sigma(\delta) \quad (4.20)$$

$$P_i(\delta) = \sigma(\delta), \quad i \geq 2 \quad (4.21)$$

donde $\sigma(\delta)$ representa los términos de orden mayor en t tal que el $\lim_{\delta \rightarrow 0} [\sigma(\delta)/\delta] = 0$. Las ecuaciones 4.19, 4.20 y 4.21 dicen que para un intervalo con un incremento muy corto el que se tenga más de una llegada resulta despreciable y la probabilidad de la llegada de una solicitud es proporcional a la duración de este intervalo (δ).

4.1.2 Procesos de Nacimiento-Muerte

La generalización de los Procesos de Poisson da como resultado los Procesos de Nacimiento, estos sirven para representar modelos de llegadas. Sin embargo en el estudio de sistemas de cómputo es necesario modelar no solo las llegadas sino las

salidas de procesos, también es necesario tomar en cuenta más de una llegada o una salida en un intervalo incremental de tiempo (δ); estas necesidades las satisfacen los Procesos de Nacimiento-Muerte. Si se supone que (δ) es muy pequeño, entonces a lo más una sola llegada con probabilidad $\lambda_n \delta^1$ puede ocurrir cuando la población es de tamaño n^2 ; la probabilidad de una salida en un intervalo incremental δ es $\mu_n \delta$. La probabilidad de una llegada y una salida en el mismo intervalo incremental es entonces del orden δ^2 y consecuentemente se puede no considerar. Entonces para el caso en que $n > 0$,

$$\begin{aligned} P_n(t + \delta) &= P \{ n \text{ procesos en el sistema al tiempo } t \} \\ &\times P \{ 0 \text{ llegadas y salidas en el intervalo de tiempo } (t, t + \delta) \mid \text{ existen } n \text{ procesos en el sistema} \} \\ &+ P \{ n - 1 \text{ procesos en el sistema al tiempo } t \} \\ &\times P \{ \text{una llegada en el intervalo } (t, t + \delta) \mid \text{ existen } n - 1 \text{ procesos en el sistema} \} \\ &+ P \{ n + 1 \text{ procesos en el sistema al tiempo } t \} \\ &\times P \{ \text{una salida en el intervalo } (t, t + \delta) \mid (n + 1) \text{ procesos en el sistema} \} \\ &= (1 - \lambda_n \delta - \mu_n \delta) P_n(t) + \lambda_{n-1} \delta P_{n-1}(t) + \mu_{n+1} \delta P_{n+1}(t) \quad (4.22) \end{aligned}$$

donde μ_n y λ_n representan el promedio del flujo de carga.

De acuerdo a la ecuación 4.22 la probabilidad de que no exista ninguna llegada ni salida en el intervalo de tiempo ($t, t + \delta$) es $(1 - \lambda_n \delta - \mu_n \delta)$; entonces para el caso $n = 0$ como no existen salidas, la probabilidad de este evento se puede escribir como:

$$\begin{aligned} P_0(t + \delta) &= P \{ 0 \text{ peticiones en el sistema al tiempo } t \} \\ &\times P \{ 0 \text{ llegadas en el intervalo } (t, t + \delta) \mid \text{ hay } 0 \text{ peticiones en el sistema} \} \\ &+ P \{ \text{una llegada en el sistema al tiempo } t \} \\ &\times P \{ \text{una salida en el intervalo } (t, t + \delta) \mid \text{ hay una petición en el sistema} \} \\ &= (1 - \lambda_0 \delta) P_0(t) + \mu_1 \delta P_1(t) \quad (4.23) \end{aligned}$$

Si el intervalo incremental lo hacemos lo más pequeño posible ($\delta \rightarrow 0$) y obtenemos la derivada de las ecuaciones 4.22 y 4.23:

$$\frac{\partial P_0(t)}{\partial t} = -\lambda_0 P_0(t) + \mu_1 P_1(t) \quad (4.24)$$

¹De acuerdo a la ecuación 4.14 las llegadas de procesos se distribuyen como una Poisson por ello la probabilidad es $\lambda_n \delta$.

²Es decir, existen ya n procesos en el sistema.

$$\frac{\partial \mathbf{P}_n(t)}{\partial t} = -(\lambda_n + \mu_n)\mathbf{P}_n(t) + \lambda_{n-1}\mathbf{P}_{n-1}(t) + \mu_{n+1}\mathbf{P}_{n+1}(t), \quad n > 0. \quad (4.25)$$

Es posible mostrar que al menos una solución positiva existe para estas ecuaciones si:

$$\sum_{n=0}^{\infty} \mathbf{P}_i(t) \leq 1 \text{ y } \mu_n \geq 0, \lambda_n \geq 0 \quad (4.26)$$

Para mostrar una solución positiva en las ecuaciones 4.24 y 4.25. Para el caso en que $\sum_{n=0}^{\infty} \mathbf{P}_i(t) < 1$ es solo de interés teórico, es decir, no es aplicable en la práctica ya que es un caso análogo de un proceso de nacimiento divergente. Para efectos prácticos que son los que más nos interesan en esta tesis la solución positiva puede obtenerse mediante la condición $\sum_{n=0}^{\infty} P_i = 1$.

Para encontrar una distribución de equilibrio de la población en el sistema al tiempo t se toma

$$\frac{\partial \mathbf{P}_i(t)}{\partial t} = 0, \text{ para } i = 0, 1, 2 \dots \quad (4.27)$$

de acuerdo con la ecuación 4.24 y la ecuación 4.27 se obtiene la igualdad:

$$\lambda_0 \mathbf{P}_0 = \mu_1 \mathbf{P}_1 \quad (4.28)$$

del mismo modo, de 4.25 y 4.27 se tiene:

$$(\lambda_n + \mu_n)\mathbf{P}_n = \lambda_{n-1}\mathbf{P}_{n-1} + \mu_{n+1}\mathbf{P}_{n+1} \quad (4.29)$$

donde $\mathbf{P}_i, i = 0, 1, 2 \dots$ indican la probabilidad que existan i peticiones en un sistema en equilibrio.

En resumen, las ecuaciones de equilibrio para un sistema están representadas por 4.28 y 4.29 y la condición de normalización por:

$$\sum_{n=0}^{\infty} \mathbf{P}_i(t) = 1. \quad (4.30)$$

Así mediante las ecuaciones de equilibrio se demuestra por inducción que:

$$\mu_n \mathbf{P}_n = \lambda_{n-1} \mathbf{P}_{n-1}, n = 0, 1, 2 \dots \quad (4.31)$$

Demostración

La ecuación de equilibrio 4.28 prueba para $n = 1$.

Se supone válido para $n - 1$

$$(\mu_{n-1}P_{n-1} = \lambda_{n-2}P_{n-2})$$

y se desea demostrar que

$$\mu_n P_n = \lambda_{n-1} P_{n-1}$$

como

$$(\mu_{n-1}P_{n-1} = \lambda_{n-2}P_{n-2})$$

entonces:

$$\begin{aligned} P_{n-1} &= \frac{\lambda_{n-2}P_{n-2} + \mu_n P_n}{(\lambda_{n-1} + \mu_{n-1})} \\ &= \frac{\mu_{n-1}P_{n-1} + \mu_n P_n}{\lambda_{n-1} + \mu_{n-1}} \end{aligned}$$

$$\Rightarrow P_{n-1}\lambda_{n-1} + \mu_{n-1}P_{n-1} = \mu_{n-1}P_{n-1} + \mu_n P_n$$

$$\Rightarrow P_{n-1}\lambda_{n-1} = \mu_n P_n$$

es así como la ecuación 4.31 resulta válida. Esta representa el equilibrio entre el flujo de entrada en un estado anterior y el flujo de servicio en el estado actual.

Además de la ecuación 4.31 se obtiene la relación del estado actual n con el primer estado 0 a saber, esto se representa:

$$P_n = P_0 \prod_{j=1}^n \left(\frac{\lambda_{j-1}}{\mu_j} \right) \quad (4.32)$$

Demostración

De acuerdo con la ecuación 4.31 se obtienen las siguientes igualdades:

$$\mu_n P_n = \lambda_{n-1} P_{n-1} \quad (4.33)$$

$$\mu_{n-1} P_{n-1} = \lambda_{n-2} P_{n-2} \quad (4.34)$$

$$\mu_{n-2} P_{n-2} = \lambda_{n-3} P_{n-3} \quad (4.35)$$

$$\vdots = \vdots \quad (4.36)$$

$$\mu_2 P_2 = \lambda_1 P_1 \quad (4.37)$$

$$\mu_1 P_1 = \lambda_0 P_0 \quad (4.38)$$

de acuerdo con las ecuaciones anteriores:

$$\begin{aligned}
 \mathbf{P}_n &= \frac{\lambda_{n-1}}{\mu_n} \mathbf{P}_{n-1} = \frac{\lambda_{n-1} \lambda_{n-2}}{\mu_n \mu_{n-1}} \mathbf{P}_{n-2} \\
 &= \frac{\lambda_{n-1} \lambda_{n-2} \lambda_{n-3}}{\mu_n \mu_{n-1} \mu_{n-2}} \mathbf{P}_{n-3} \\
 &= \dots \\
 &= \frac{\lambda_{n-1} \lambda_{n-2} \lambda_{n-3} \dots \lambda_1}{\mu_n \mu_{n-1} \mu_{n-2} \dots \mu_2} \mathbf{P}_1 = \frac{\lambda_{n-1} \lambda_{n-2} \lambda_{n-3} \dots \lambda_1 \lambda_0}{\mu_n \mu_{n-1} \mu_{n-2} \dots \mu_2 \mu_1} \mathbf{P}_0 \\
 &= \mathbf{P}_0 \prod_{j=1}^n \frac{\lambda_{j-1}}{\mu_j} \quad (4.39)
 \end{aligned}$$

y de acuerdo a la condición de normalización (ver 4.30) se obtiene que:

$$\sum_{n=0}^{\infty} \mathbf{P}_n = 1 \Rightarrow \sum_{n=1}^{\infty} \mathbf{P}_n + \mathbf{P}_0 = 1$$

Por otro lado

$$\begin{aligned}
 \sum_{n=1}^{\infty} \mathbf{P}_n + \mathbf{P}_0 &= \sum_{n=1}^{\infty} \mathbf{P}_0 \prod_{j=1}^n \frac{\lambda_{j-1}}{\mu_j} + \mathbf{P}_0 \\
 &= 1 \quad (4.40)
 \end{aligned}$$

$$\Rightarrow \mathbf{P}_0 = \left[1 + \sum_{n=1}^{\infty} \prod_{j=1}^n \frac{\lambda_{j-1}}{\mu_j} \right]^{-1}$$

4.2 Redes de Colas

En general los procesos de Nacimiento-Muerte son útiles para modelar las llegadas o partidas de peticiones del cliente al servidor, cada uno de estos clientes espera en una cola a ser atendidas sus peticiones y parte el cliente cuando el servicio fue realizado exitosamente. Los tiempos esperados para realizar estos servicios naturalmente se consideran que se distribuyen como una exponencial, así la dinámica de los procesos es medida por la variación del número de peticiones que los clientes manden en el sistema (servidor). Los tipos de colas varían de acuerdo al servicio o al sistema, así se tienen:

de acuerdo con las ecuaciones anteriores:

$$\begin{aligned}
 P_n &= \frac{\lambda_{n-1}}{\mu_n} P_{n-1} = \frac{\lambda_{n-1} \lambda_{n-2}}{\mu_n \mu_{n-1}} P_{n-2} \\
 &= \frac{\lambda_{n-1} \lambda_{n-2} \lambda_{n-3}}{\mu_n \mu_{n-1} \mu_{n-2}} P_{n-3} \\
 &= \dots \\
 &= \frac{\lambda_{n-1} \lambda_{n-2} \lambda_{n-3} \dots \lambda_1}{\mu_n \mu_{n-1} \mu_{n-2} \dots \mu_2} P_1 = \frac{\lambda_{n-1} \lambda_{n-2} \lambda_{n-3} \dots \lambda_1 \lambda_0}{\mu_n \mu_{n-1} \mu_{n-2} \dots \mu_2 \mu_1} P_0 \\
 &= P_0 \prod_{j=1}^n \frac{\lambda_{j-1}}{\mu_j} \tag{4.39}
 \end{aligned}$$

y de acuerdo a la condición de normalización (ver 4.30) se obtiene que:

$$\sum_{n=0}^{\infty} P_n = 1 \Rightarrow \sum_{n=1}^{\infty} P_n + P_0 = 1$$

Por otro lado

$$\begin{aligned}
 \sum_{n=1}^{\infty} P_n + P_0 &= \sum_{n=1}^{\infty} P_0 \prod_{j=1}^n \frac{\lambda_{j-1}}{\mu_j} + P_0 \\
 &= 1 \tag{4.40}
 \end{aligned}$$

$$\Rightarrow P_0 = \left[1 + \sum_{n=1}^{\infty} \prod_{j=1}^n \frac{\lambda_{j-1}}{\mu_j} \right]^{-1}$$

4.2 Redes de Colas

En general los procesos de Nacimiento-Muerte son útiles para modelar las llegadas o partidas de peticiones del cliente al servidor, cada uno de estos clientes espera en una cola a ser atendidas sus peticiones y parte el cliente cuando el servicio fue realizado exitosamente. Los tiempos esperados para realizar estos servicios naturalmente se consideran que se distribuyen como una exponencial, así la dinámica de los procesos es medida por la variación del número de peticiones que los clientes manden en el sistema (servidor). Los tipos de colas varían de acuerdo al servicio o al sistema, así se tienen:

- El primero en llegar es el primero en salir (first-come-first-served,FCFS).
- El último en llegar es el primero en salir (last-come-first-served,LCFS).
- Servirse de un modo aleatorio.

para cualquiera de estos casos como el tiempo esperado de servicio no tiene memoria, el servidor puede interrumpir un servicio de un cliente antes de servir a otro cliente sin cambiar la dinámica. Un ejemplo claro es la estrategia de tipo *Round Robin*, donde el servidor crea un ciclo a través de todos los miembros de la cola y por cada intervalo de tiempo (generalmente es de un segundo) son servidos hasta que exceden el tiempo de CPU asignado. La clave de todos estos casos mencionados es que el servidor nunca se encuentra ocioso mientras la cola no esta vacía.

Resulta interesante observar el comportamiento del tiempo esperado para que un servicio sea ejecutado exitosamente.

El paso inicial en el diseño del sistema de cómputo (SC) en estudio, es considerar los recursos para su desempeño con respecto a diferentes versiones de la estructura organizacional. Esas versiones pueden ser diferentes en sus componentes y características de módulos de acoplamiento, modos de operación, y algoritmos de control. Por ello se utilizan los modelos de sistemas de cómputo.

Por regla general, los usuarios acceden aleatoriamente a los sistemas de cómputo. El espacio de memoria requerido, el tiempo de operación del procesador, las interfaces, los aditamentos I/O, el tiempo que tarda una petición de un cliente en llegar al servidor y otras unidades son también aleatorios. Las fallas y descomposuras en el equipo del SC ocurren en instantes aleatorios del tiempo e inclusive lo que dura su reparación es aleatorio.

Debido a la contingencia de los instantes en los cuales se demanda un servicio y la duración arbitraria de éste ocurre un fenómeno de colas de espera. La operación del sistema de cómputo en sus diferentes recursos, incluyendo colas para los procesadores, aditamentos I/O uso de unidades externas, y la transmisión de datos en los canales (por ejemplo ver subsección 2.3).

Esta investigación esta dirigida en la evaluación de los retardos y longitudes de las colas de acceso en las fuentes del sistema de cómputo y retransmisión de mensajes entre computadoras remotas; las probabilidades de las fallas, congestiones, tiempo de retención de mensajes en la red factores de utilización de aditamentos separados (procesadores, memoria, canales) y software; etc., dando como aplicación específica los servicios de red NIS Y NFS.

En la mayoría de los casos, los modelos de los sistemas de cómputo pueden ser representados como una colección de servidores interconectados con colas de-

nominados Sistemas de Colas (SC). Tal colección de sistemas de colas es llamado una *Red de Colas* (RC).

El modelo abstracto de un sistema de cómputo que es una RC no debe ser confundido con la llamada *complejidad* del hardware-software en las redes de computadoras.

Dependiendo del tipo de fuentes de solicitudes, los modelos en la forma de RC se pueden clasificar como sigue: Una fuente es llamada *finita* si genera un número finito de solicitudes, por ejemplo, si hay un número finito de terminales en las que los usuarios trabajan, un usuario que ejecuta comandos o programas en modo interactivo puede introducir una nueva solicitud solo después de haber recibido una respuesta a una previa. Las RC con fuentes finitas son conocidas como *cerradas* (el procesador necesita retornar a las fuentes para confirmar la ejecución de la solicitud).

Una fuente que contiene un número infinito de solicitudes, entonces puede ser llamada *infinita*, y la RC relevante es conocida como *abierta*. Un flujo de solicitudes desde una fuente infinita es llamada una *corriente externa de la red*. Por lo tanto las RC cerradas contienen un número fijo de solicitudes las cuales se generan en la RC mas nuevas solicitudes que llegan desde fuera. En una RC, pueden existir solicitudes de servicios hacia dentro de la RC y para servicios externos a ésta, razón por la que el número de solicitudes en la red pueden variar.

Generalmente, las RC cerradas sirven como modelos de sistemas de cómputo procesando un número fijo y limitado de solicitudes. Las RC abiertas son utilizadas para modelar sistemas de cómputo con un número ilimitado y variable de solicitudes o llamadas.

Cada sistema de colas de la RC incluye uno o mas servidores similares en capacidad o que ofrecen el mismo tipo de servicios y solicitudes específicos. Como ejemplos se pueden citar: las llamadas de lectura y/o escritura de un sistema de archivos de una máquina cliente a una máquina servidor remotamente (NFS), ejecución de un programa remoto o local (NIS), etc. Los servidores de sistema de colas se supone que trabajan concurrentemente e independientemente. Si el número de servidores es mayor que el número máximo probable de solicitudes en un sistema de colas dado, entonces este sistema de colas no tiene una línea de solicitudes en espera.

Tradicionalmente, para definir un tipo de sistema de colas, se utiliza la designación del tipo $X/Y/k$, donde la primera letra significa la distribución de los tiempos entre dos solicitudes consecutivas de ejecución o de compartir un recurso; la segunda letra es la distribución del servicio solicitado (lo que se tarda en ejecutar la tarea solicitada); la tercera letra establece el número de servidores. Cuando

$k = 1$, el sistema de colas es denominado *unilineal*, cuando $k > 1$ se conoce como *multilineal*, o k -lineal. La letra M designa a la exponencial y la G distribuciones aleatorias. Por ejemplo, un sistema de colas del tipo $M/M/k$ existen k servidores, y la corriente de solicitudes es una Poisson-estacionaria y los intervalos de tiempo entre dos solicitudes consecutivas son exponencialmente distribuidas cuya función de distribución es (Ver sección 1.1)

$$F(t) = 1 - e^{-\lambda t}, \quad t > 0.$$

El parámetro λ de ésta distribución define la tasa media de solicitudes que llegan, y es igual al valor inverso de la media del intervalo entre los arribos de solicitudes, $\lambda = 1/t_{\text{media del intervalo}}$. La duración del servicio de las solicitudes por cada servidor se distribuye exponencialmente con parámetro μ la cual se define como la intensidad de la media de los servicios prestados por un servidor y es igual a la inversa del valor de la duración de servicio media: $1/\mu_{\text{servicio media}}$. Un importante parámetro del sistema de colas es el factor de carga del servicio

$$\rho = \lambda/\mu k. \quad (4.41)$$

con k , representando al número de servidores.

Un comportamiento estacionario en el sistema de colas es posible, si la intensidad de la cadena o corriente de solicitudes de entrada es menor que la intensidad del servicio solicitado, i.e. si

$$0 \leq \rho < 1 \quad (4.42)$$

De otra manera la cola se incrementa infinitamente. El comportamiento estacionario de operación es siempre posible en RC cerradas. Mientras que, en RC abiertas un comportamiento estacionario es factible, siempre que la condición (4.42) sea satisfecha en cada sistema de colas de la red.

En el comportamiento estacionario del sistema de colas, las intensidades de las cadenas de entrada y salida de solicitudes son iguales.

Al diseñar analíticamente el sistema de colas, se puede suponer sin excepción que los intervalos entre las llegadas de las solicitudes y duración del servicio son distribuidos exponencialmente. En la práctica, la cadena de entrada de las solicitudes en las redes reales es a veces muy semejante a la distribución Poisson, como lo exhibe la observación 4.2.1 puesto que esto es usualmente una superposición de cadenas o corrientes separadas e independientes.

La suposición que las duraciones de los servicios correspondientes a las solicitudes son exponencialmente distribuidas a veces no corresponden a la situación

real. Sin embargo experiencia en la simulación, muestra que la aproximación a los resultados obtenidos bajo esas suposiciones son aceptables para su uso en cálculos de ingeniería. Justamente esas redes de colas son conocidas como exponenciales y además son ampliamente usadas en modelos de sistemas de cómputo.

Observación 4.2.1 Si se considera una partición de un intervalo de tiempo $[a, b]$ por ejemplo $\{t_0 = a, t_1, \dots, t_{n-1}, t_n = b\}$, considérese la variable aleatoria de tipo Bernoulli definida por la observación en cada subintervalo $[t_i, t_{i+1}]$ que registra como éxito E la presencia de una llamada al sistema de cómputo en ese lapso de tiempo, consecuentemente el fracaso F se define por la ausencia de llamada o petición al sistema. Estos eventos tienen asociadas probabilidades de ocurrencia p y q . Por lo que al analizar el intervalo completo, existen k ($k = 0, 1, \dots, n$) llamadas distribuidas en los n subintervalos. Lo cual ocurre con probabilidad

$$\binom{n}{k} p^k q^{n-k}$$

(que es una variable aleatoria Binomial (n, p)), cuando se refina la partición (los subintervalos se hacen suficientemente pequeños en longitud) y la probabilidad de una llegada se hace pequeña, de tal manera que la intensidad $\lambda = np$ sea constante entonces es bien conocido que

$$\lim_{n \rightarrow \infty, p \rightarrow 0, np = \lambda} \binom{n}{k} p^k q^{n-k} = \frac{\lambda^k e^{-\lambda}}{k!}$$

que es precisamente una densidad poisson. El parámetro λ es precisamente la intensidad de las llegadas de las llamadas al sistema (Ver sección 1.1) \square

En los próximos puntos se describirán algunos tipos de colas que se aplicarán en el estudio del modelo de un sistema de cómputo, este modelo se enfocará a dos servicios de red (NFS y NIS), y al comportamiento de los tipos de redes.

Para comenzar este estudio, es necesario mencionar un resultado que auxilia al análisis de colas y es el llamado Teorema de Little.

4.2.1 Teorema de Little, (Little's Theorem)

La función principal de este resultado es relacionar dos de los parámetros en los sistemas de colas que son: el tiempo esperado de servicio en el sistema y el tiempo promedio que los clientes esperan en la cola para un servicio.

Cuatro de las cantidades que afectan estos dos parámetros son: la media del número total de clientes en el sistema N y en espera en la cola N_q , la media de los arribos λ , y los tiempos promedios de espera en el sistema y en la cola T y T_q , respectivamente.

Así el Teorema de Little muestra que:

$$N = \lambda T \quad (4.43)$$

Demostración

Antes considere la función del sistema de colas, donde en intervalos de tiempo llegan los trabajos, son ejecutados y en seguida salen. Notese que no es necesario mencionar la distribución de estos intervalos de tiempo.

Sea $\alpha(t)$ el número de llegadas en el sistema en el intervalo de tiempo $[0, t]$ y $\delta(t)$ el número de partidas en el sistema en el mismo intervalo. Si el sistema está vacío al tiempo $\kappa = 0$, el número de mensajes en el sistema $N(t)$ al tiempo t es:

$$N(t) = \alpha(t) - \delta(t) \quad (4.44)$$

El tiempo total $\gamma(t)$ de consumo del sistema por los mensajes al tiempo t es el área acumulada de $[0, t]$ lo cual es representado como:

$$\gamma(t) = \int_0^t N(\kappa) d\kappa \quad (4.45)$$

Lo cual es equivalente a:

$$\sum_{i=1}^{\delta(t)} T_i + \sum_{i=\delta(t)+1}^{\alpha(t)} (t - t_i) \quad (4.46)$$

donde T_i es el tiempo que el mensaje i consumió en el sistema⁴. Dividiendo 4.45 y 4.46 entre t :

$$\begin{aligned} N_t &= \frac{1}{t} \int_0^t N(\kappa) d\kappa \\ &= \text{Promedio de mensajes en el sistema} \end{aligned} \quad (4.47)$$

⁴Ver [Ge].

$$T_i = \frac{1}{\alpha(t)} \left[\sum_{i=1}^{\delta(t)} T_i + \sum_{i=\delta(t)+1}^{\alpha(t)} (t - t_i) \right]$$

= Tiempo promedio de consumo de un mensaje en el sistema (4.48)

$$\lambda_i = \frac{\alpha(t)}{t}$$

= Promedio de arribos de mensajes (4.49)

De esta forma se obtiene:

$$N_i = \lambda_i T_i \quad (4.50)$$

Lo anterior fue probado solo para un intervalo finito $[0, t]$, en el caso de que exista un estado de equilibrio ¹¹, se puede tomar a t muy larga ($t \Rightarrow \infty$), asumiendo que N_i , λ_i y T_i permanecen finitos y se aproximan a sus valores de equilibrio N , λ y T respectivamente se tiene:

$$N = \lambda T \quad (4.51)$$

La ecuación 4.51 muestra que el número promedio de mensajes en el sistema se puede definir, como el promedio de llegadas multiplicado por el tiempo total de consumo en el sistema, este resultado es válido con las suposiciones descritas anteriormente para este estudio sin embargo con otras suposiciones puede no ser general.

Similarmente se tiene la relación:

$$N_q = \lambda T_q \quad (4.52)$$

Esto es el número promedio de mensajes esperando en la cola es igual al número promedio de llegadas multiplicado por el tiempo promedio de consumo generados por los mensajes que esperan en la cola.

4.2.2 Análisis de la cola de tipo M/M/1

El objetivo de analizar la cola de tipo M/M/1, es encontrar la probabilidad de que n mensajes estén presentes en el buffer al tiempo t ($P_n(t)$).

¹¹El número de peticiones es semejante al número de salidas

Para comenzar el análisis, se asume que las llegadas de las peticiones se distribuye como una Poisson con promedio de llegadas λ_n por segundo, para cada estado n , la cantidad λ_n es la misma denotada por λ , y que cada cliente requiere un promedio de servicio de tiempo distribuido exponencialmente con media $\frac{1}{\mu}$.

Sea ρ la intensidad de función de λ y μ de la forma:

$$\rho = \frac{\lambda}{\mu}$$

, debido a que el promedio de las llegadas durante el tiempo promedio de servicio es $\frac{\lambda}{\mu} < 1$, se tiene que $\rho < 1$.

Con estas definiciones y dadas $\lambda_n = \lambda$ y $\mu_n = \mu$ y tomando la ecuación 4.39 se llega a:

$$\begin{aligned} P_n &= P_0 \prod_{j=1}^n \frac{\lambda_{j-1}}{\mu_j} \\ &= P_0 \prod_{j=1}^n \frac{\lambda}{\mu} \\ &= P_0 \rho^n \end{aligned} \quad (4.53)$$

Para encontrar P_0 se toma el resultado de la ecuación 4.40:

$$\begin{aligned} P_0 &= \left[1 + \sum_{n=1}^{\infty} \prod_{j=1}^n \frac{\lambda_{j-1}}{\mu_j} \right]^{-1} \\ &= \left[1 + \sum_{n=1}^{\infty} \rho^n \right]^{-1} \\ &= \left[\sum_{n=0}^{\infty} \rho^n \right]^{-1} \\ &= \left[\frac{1}{1-\rho} \right]^{-1} \\ &= (1-\rho) \end{aligned} \quad (4.54)$$

Entonces la ecuación 4.53 se puede simplificar a:

$$P_n = (1-\rho)\rho^n \quad (4.55)$$

Nótese que esta última ecuación representa a la distribución del número de clientes en el sistema, la cual asemeja a una distribución geométrica cuya probabilidad de tener un sistema vacío es $P_0 = 1 - \rho$.

Tomando en consideración este resultado se obtiene la función generadora:

$$P(z) = \frac{1 - \rho}{1 - z\rho} \quad (4.56)$$

Derivando y haciendo $z = 1$ se llega a que la media de esta distribución es:

$$\begin{aligned} N &= E[\tilde{N}] \\ &= \frac{\rho}{(1 - \rho)} \end{aligned} \quad (4.57)$$

Para ρ menores que 0.5, el número promedio de mensajes N , es menor que 1. Para altos valores de ρ , el número de mensajes en espera incrementa rápidamente.

Usando la fórmula de Little, se encuentra el tiempo promedio de retardo en el sistema:

$$\begin{aligned} T &= \frac{N}{\lambda} \\ &= \frac{\rho}{\lambda(1 - \rho)} \\ &= \frac{1}{\mu(1 - \rho)} \\ &= \frac{1}{\mu}(1 + N) \end{aligned} \quad (4.58)$$

Este último resultado muestra que el comportamiento del tiempo promedio que un mensaje consume en el sistema esta en función de la intensidad de tráfico ρ , al igual que en número promedio de mensajes en el sistema. Cuando $\rho = 0$, el valor de T es exactamente el tiempo promedio de servicio, esto es, el mensaje no tiene que esperar en la cola para ser servido en $\frac{1}{\mu}$ segundos en promedio.

Cuando ρ se aproxima a 1, tanto el tiempo promedio de consumo en el sistema y el número promedio de mensajes en la cola incrementa dramáticamente.

Como el tiempo promedio de servicio por mensaje es $\frac{1}{\mu}$, y ya que T es el tiempo promedio consumido por los mensajes en servicio T_s , más el tiempo promedio gastado por los mensajes en la cola T_q , así T también se representa como:

$$T = \frac{1}{\mu} + T_q \quad (4.59)$$

Así se tiene que:

$$\begin{aligned} T_q &= T - \frac{1}{\mu} \\ &= \frac{\rho}{\mu(1-\rho)} \end{aligned} \quad (4.60)$$

Aplicando la fórmula de Little al número promedio de mensajes en la cola:

$$\begin{aligned} N_q &= \lambda T_q \\ &= \frac{\rho^2}{1-\rho} \end{aligned} \quad (4.61)$$

En ocasiones es necesario poder representar el tamaño del bufer, esto con el fin de calcular la probabilidad de una saturación en este. Así la probabilidad de que el número de mensajes que esperen en una cola M/M/1 sea más grande que un número esperado N es:

$$\begin{aligned} P(n > N) &= \sum_{n=N+1}^{\infty} P_n \\ &= (1-\rho) \sum_{n=N+1}^{\infty} \rho^n \\ &= (1-\rho) \left(\sum_{n=0}^{\infty} \rho^n - \sum_{n=0}^N \rho^n \right) \\ &= (1-\rho) \left(\frac{1}{1-\rho} - \frac{1-\rho^{N+1}}{1-\rho} \right) \\ &= \rho^{N+1} \end{aligned} \quad (4.62)$$

ya que $\rho < 1$, la probabilidad de que el número de mensajes en el sistema exceda a N , es una función que decrece geoméricamente con respecto a N .

4.2.3 Modelo de una cola de tipo M/G/1

El modelo analítico de un servidor cuyas llegadas se distribuyen como una Poisson con λ como valor medio y cuya función de distribución de los servicios se distribuyen de manera general con media μ y varianza σ^2 se representa por medio de la cola de tipo M/G/1.

Un ejemplo que se aplica en este modelo y que se utilizará para el modelo de la red en estudio son los ruteadores.

La llegada de las peticiones enviadas por las estaciones de trabajo se suponen con distribución Poisson con un promedio de llegadas λ por segundo, el proceso de servicio sigue el orden "El primero en llegar, es el primero en ser servido (First Come, First Served, FCFS)".

La distribución del tiempo de servicio para cada petición es arbitraria, esta será denotada por $B(t)$ con media $\frac{1}{\mu}$ y varianza σ_b^2 .

Los resultados para obtener el promedio de mensajes y el promedio de tiempo de retraso en un sistema de cola de tipo M/G/1, son tomados de *Pollaczek-Khinchin (P-K) mean value formulas* (Vease [Pe]). Se define la razón de la varianza y la media de la distribución de servicio de los paquetes por:

$$C_b^2 = \frac{\sigma_b^2}{\mu^2} \quad (4.63)$$

Dadas las siguientes variables aleatorias:

- $N(t)$ = Número de mensajes en el servidor al tiempo t .
- $N_q(t)$ = Número de mensajes en la cola al tiempo t .
- $N_s(t)$ = Número de mensajes que están siendo procesados al tiempo t .

La relación que existe entre estas variables es:

$$N(t) = N_q(t) + N_s(t) \quad (4.64)$$

Tomando en cuenta los valores esperados respectivamente:

- N = Número promedio de mensajes en el sistema al tiempo t .
- N_q = Número promedio de los peticiones en la cola.
- N_s = Número promedio de mensajes que están siendo procesados.

La relación que existe entre estos valores en un estado estable es:

$$N = N_q + N_s \quad (4.65)$$

* Los paquetes que están esperando en la cola más los que están siendo procesados o en servicio

El promedio de número de mensajes en el sistema $E[\tilde{N}]$ se basa en la fórmula (P-K) y es la siguiente:

$$\begin{aligned} N &= E[\tilde{N}] \\ &= \rho + \rho^2 \frac{1 + C_b^2}{2(1 - \rho)} \end{aligned} \quad (4.66)$$

la cual es definida por el factor de intensidad de tráfico ρ ($\rho = \frac{\lambda}{\mu}$), el promedio de longitud de los mensajes $\frac{1}{\mu}$ ¹¹, y por la varianza de la distribución del tiempo de servicio.

Así usando la ecuación 4.65, podemos obtener el número promedio de mensajes en la cola N_q , dados N_s , ρ , se tiene:

$$\begin{aligned} N_q &= N - N_s \\ &= N - \rho \\ &= \rho^2 \frac{(1 + C_b^2)}{2(1 - \rho)} \end{aligned} \quad (4.67)$$

Definiendo T como el tiempo promedio de retardo en procesar el sistema una petición se obtiene de la siguiente manera:

Recuérdese que la fórmula de Little para el caso de una cola de tipo (FCFS) en un intervalo finito es el número de llegadas multiplicado por el retardo en un intervalo de tiempo t . Para entender mejor este concepto, supongase que se tiene un túnel en el cual un número promedio de 5 bolitas arriba por minuto ($\lambda = 5bol$), cada bolita atraviesa el túnel rodando en un tiempo promedio de 2 minutos ($T = 2seg$); el primer grupo de bolitas (el túnel esta vacío), comienzan a rodar, al transcurrir un minuto, arriba el segundo grupo de bolitas y este espera a que termine el primer grupo de bolitas para atravesar el túnel, a la llegada del tercer grupo de bolitas el primer grupo ya logró atravesar el túnel y comienzan a rodar el segundo grupo de bolitas, etc.

De esta manera se tienen siempre 10 bolitas en el servidor (5 en espera y 5 rodando por el túnel).

Es así como se representa el número total de peticiones en el servidor:

¹¹También se puede definir como el promedio de tiempo de servicio $\frac{1}{\mu}$

$$N_t = \lambda_t T_t \quad (4.68)$$

Cuando t es muy larga ($t \rightarrow \infty$), se asume que T_t , N_t/λ_t permanecen finitos y se aproximan a sus valores de equilibrio N , T y λ respectivamente.

Así la fórmula de Little que se aplica es:

$$N = \lambda T \quad (4.69)$$

Igualmente el número promedio de peticiones en la cola es igual al promedio de arribos de los clientes multiplicados por el promedio de tiempo que un cliente espera para ser procesado en la cola y esto se representa de la siguiente forma:

$$N_q = \lambda T_q \quad (4.70)$$

El tiempo promedio que tarda en procesar una petición el servidor de acuerdo con las ecuaciones 4.66 y 4.69 es entonces:

$$\begin{aligned} T &= \frac{1}{\lambda} E[\tilde{N}] \\ &= \frac{1}{\lambda} \left[\rho + \rho^2 \left(\frac{1 + C_b^2}{2(1 - \rho)} \right) \right] \\ &= \frac{1}{\mu} + \frac{\lambda}{\mu^2} \left(\frac{1 + C_b^2}{2(1 - \rho)} \right) \quad (4.71) \\ &= \frac{1}{2\mu(1 - \rho)} \left[2(1 - \rho) + \rho \left(1 + \frac{\sigma^2}{\mu^2} \right) \right] \\ &= \frac{1}{2\mu(1 - \rho)} \left[2 - \rho \left(1 - \frac{\sigma^2}{\mu^2} \right) \right] \end{aligned}$$

Se define el rendimiento total del servidor directamente proporcional al número de llegadas de peticiones e indirectamente proporcional a el tiempo promedio de retardo de un servicio ¹¹. Esto es:

$$S = \frac{\lambda}{T} \quad (4.72)$$

¹¹En el ejemplo del túnel y bolitas, el rendimiento total es 2.5 bolitas por minuto.

4.2.4 Análisis de la cola M/M/K

Considerese un cierto i -ésimo sistema de colas, del tipo $M/M/K_i$ con una cola de longitud infinita. Sea λ_i la intensidad de la cadena de entrada de solicitudes, μ_i la intensidad de los servicios solicitados ejecutados por cada servidor, y la condición (4.42) es cierta.

Siendo $P_i(x_i)$ la probabilidad estacionaria del estado en el cual existen x_i solicitudes en el i -ésimo sistema de colas. El conjunto de ecuaciones lineales para encontrar las probabilidades estacionarias están dadas por las ecuaciones de equilibrio (Ver sección 1.1, ecuaciones 4.28 y 4.29)

$$\begin{aligned} \lambda_i P_i(0) &= \mu_i P_i(1) \\ [x_i \mu_i + \lambda_i] P_i(x_i) &= \lambda_i P_i(x_i - 1) + [x_i + 1] \mu_i P_i(x_i + 1) \quad \text{para } 1 \leq x_i < k_i \\ [k_i \mu_i + \lambda_i] P_i(x_i) &= \lambda_i P_i(x_i - 1) + k_i \mu_i P_i(x_i + 1) \quad \text{para } x_i \geq k_i \end{aligned} \quad (4.73)$$

La solución de este conjunto de ecuaciones (4.73) toma la forma

$$P_i(x_i) = A_i(x_i) (\lambda_i / \mu_i)^{x_i} P_i(0) \quad (4.74)$$

donde

$$A_i(x_i) = \begin{cases} (x_i!)^{-1} & \text{si } x_i \leq k_i, \\ (k_i! k_i^{x_i - k_i})^{-1} & \text{si } x_i > k_i, \end{cases} \quad (4.75)$$

Demostración

Para $1 \leq x_i < k_i$ se tiene que: Para $x_i = 1$:

$$\begin{aligned} P_i(1) &= (1) \left(\frac{\lambda}{\mu} \right)^1 P_i(0) \\ &\Rightarrow \lambda P_i(0) = \mu P_i(1) \text{ lo cual es cierto de acuerdo a las ecuaciones de equilibrio} \\ &= \frac{\lambda}{\mu} P_i(0) \end{aligned} \quad (4.76)$$

Ahora, se supone cierto para $n - 1$. P.D. que

$$P_i(n_i) = (x_i!)^{-1} \left(\frac{\lambda_i}{\mu_i} \right)^{x_i} P_i(0) \quad (4.77)$$

De acuerdo a las ecuaciones 4.73

$$\begin{aligned}
 P_i(n) &= \frac{[\mu_i(n-1) + \lambda_i]P_i(n-1) - \lambda_i P_i(n-2)}{n\mu_i} \\
 &= \frac{[\mu_i(n-1) + \lambda_i] \left[\frac{1}{(n-1)!} \left(\frac{\lambda_i}{\mu_i}\right)^{n-1} P_i(0) \right] - \lambda_i P_i(n-2)}{n\mu_i} \\
 &= \frac{[\mu_i(n-1) + \lambda_i] \left[\frac{1}{(n-1)!} \left(\frac{\lambda_i}{\mu_i}\right)^{n-1} P_i(0) \right] - \lambda_i \left[\frac{1}{(n-2)!} \left(\frac{\lambda_i}{\mu_i}\right)^{n-2} P_i(0) \right]}{n\mu_i} \\
 &= P_i(0) \left(\frac{\lambda_i}{\mu_i}\right)^{n-2} \left(\frac{1}{(n-2)!}\right) \left[\frac{(n-1)\mu_i + \lambda_i}{n\mu_i} - \lambda_i \right] \quad (4.78) \\
 &= P_i(0) \left(\frac{\lambda_i}{\mu_i}\right)^{n-2} \left(\frac{1}{(n-2)!}\right) \left[\frac{(n-1)\mu_i\lambda_i + \lambda_i^2 - \lambda_i\mu_i(n-1)}{n(n-1)\mu_i^2} \right] \\
 &= P_i(0) \left(\frac{\lambda_i}{\mu_i}\right)^{n-2} \left(\frac{1}{(n-2)!}\right) \left(\frac{\lambda_i}{\mu_i}\right)^2 \frac{1}{n(n-1)} \\
 &= P_i(0) \left(\frac{\lambda_i}{\mu_i}\right)^n (n!)^{-1}
 \end{aligned}$$

En el caso de $x_i \geq k_i$ P.D.

$$P_i(n) = (k_i! k_i^{n-k_i})^{-1} \left(\frac{\lambda_i}{\mu_i}\right)^n P_i(0)$$

De acuerdo a las ecuaciones de equilibrio 4.73 se tiene

$$\begin{aligned}
 (k\mu_i + \lambda_i)P_i(n) &= \lambda_i P_i(n-1) + k\mu_i P_i(n+1) \\
 \Rightarrow P_i(n+1) &= \frac{(k\mu_i + \lambda_i)P_i(n) - \lambda_i P_i(n-1)}{k\mu_i}
 \end{aligned}$$

Para $n = 1$ se tiene la ecuación de equilibrio

$$\lambda_i P_i(0) = \mu_i P_i(1)$$

Para $n = 2$:

$$\begin{aligned}
P_i(2) &= \frac{(k\mu_i + \lambda_i)P_i(1) - \lambda_i P_i(0)}{k\mu_i} \\
&= \frac{(k\mu_i + \lambda_i)\frac{\lambda_i}{\mu_i} P_i(0) - \lambda_i P_i(0)}{k\mu_i} \\
&= P_i(0) \left[\frac{k\lambda_i\mu_i + \lambda_i^2 - \lambda_i\mu_i}{\frac{\lambda_i}{\mu_i}} \right] \\
&= P_i(0) \left[\frac{k\lambda_i\mu_i(k-1) + \lambda_i^2}{k\mu_i^2} \right] \quad (4.79) \\
&= \frac{P_i(0)}{k!} \left(\frac{\lambda_i}{\mu_i} \right)^2 \left[\frac{(k-1)!\mu_i(k-1) + \lambda_i(k-1)!}{\lambda_i} \right] \\
&= P_i(0) \left(\frac{\lambda_i}{\mu_i} \right)^2 \frac{1}{k!}
\end{aligned}$$

Suponiendo válido para $n-1$, P.D. para n , entonces:

$$\begin{aligned}
P_i(n) &= \frac{(k\mu_i + \lambda_i)P_i(n-1) - \lambda_i P_i(n-2)}{k\mu_i} \\
&= \left[\frac{(k\mu_i + \lambda_i)(k!k^{n-1-k})^{-1} \left(\frac{\lambda_i}{\mu_i} \right)^{n-1} P_i(0) - \lambda_i \left[(k!k^{n-2-k})^{-1} \left(\frac{\lambda_i}{\mu_i} \right)^{n-2} P_i(0) \right]}{k\mu_i} \right] \\
&= P_i(0) \left(\frac{\lambda_i}{\mu_i} \right)^{n-1} \left[\frac{(k\mu_i + \lambda_i)(k!k^{n-1-k})^{-1}}{k\mu_i} - \frac{(k!k^{n-2-k})^{-1}}{k} \right] \\
&= P_i(0) \left(\frac{\lambda_i}{\mu_i} \right)^{n-1} \left[\frac{k\mu_i + \lambda_i - k\mu_i}{(k!k^{n-k})\mu_i} \right] \quad (4.80) \\
&= P_i(0) \left(\frac{\lambda_i}{\mu_i} \right)^{n-1} \frac{1}{k!k^{n-k}} \left[\frac{\lambda_i}{\mu_i} \right] \\
&= P_i(0) \left(\frac{\lambda_i}{\mu_i} \right)^n \frac{1}{k!k^{n-k}}
\end{aligned}$$

Pero además se debe tener presente la condición de normalización

$$\sum_{r_i=0}^{\infty} P_i(r_i) = 1 \quad (4.81)$$

y sumando los términos de la progresión geométrica (4.74) con factor común $\lambda_i/r_i\mu_i$, entonces se obtiene

$$P_i(0) = \left[\sum_{r_i=0}^{k_i-1} \frac{(\lambda_i/\mu_i)^{r_i}}{r_i!} + \frac{(k_i\lambda_i/\mu_i)^{k_i}}{k_i!(1-\lambda_i/\mu_i)} \right]^{-1} \quad (4.82)$$

Demostración

Dada la ecuación de normalidad 4.81 y de acuerdo a las ecuaciones 4.78 4.80 se tiene que:

$$\begin{aligned} \sum_{n=0}^{\infty} P_i(x_i) &= \sum_{r_i=0}^{\infty} \frac{P_i(0)}{k_i!} \left(\frac{\lambda_i}{\mu_i}\right)^{r_i} + \sum_{n=0}^{\infty} P_i(x_i) \frac{P_i(0)}{k_i!k_i^{n-k_i}} \\ &= P_i(0) \sum_{r_i=0}^{k_i-1} \frac{1}{k_i!} \left(\frac{\lambda_i}{\mu_i}\right)^{r_i} + P_i(0) \sum_{n=0}^{\infty} \frac{\left(\frac{\lambda_i}{\mu_i}\right)^{r_i} \left(\frac{\lambda_i}{\mu_i}\right)^{k_i} \left(\frac{\lambda_i}{\mu_i}\right)^{-k_i}}{k_i!k_i^{n-k_i}} \\ &= P_i(0) \left[\sum_{r_i=0}^{k_i-1} \frac{1}{r_i!} \left(\frac{\lambda_i}{\mu_i}\right)^{r_i} + \left(\frac{\lambda_i}{\mu_i}\right)^{k_i} \frac{1}{k_i!} \sum_{r_i=0}^{\infty} \left(\frac{\lambda_i}{\mu_i}\right)^{r_i-k_i} \right] \quad (4.83) \\ &= P_i(0) \left[\sum_{r_i=0}^{k_i-1} \frac{1}{r_i!} \left(\frac{\lambda_i}{\mu_i}\right)^{r_i} + \left(\frac{\lambda_i}{\mu_i}\right)^{k_i} \frac{1}{k_i!} \frac{1}{1-\frac{\lambda_i}{\mu_i}} \right] \\ &= 1 \end{aligned}$$

$$\Rightarrow P_i(0) = \left[\sum_{r_i=0}^{k_i-1} \frac{1}{r_i!} \left(\frac{\lambda_i}{\mu_i}\right)^{r_i} + \frac{k_i \left(\frac{\lambda_i}{\mu_i}\right)^{k_i}}{k_i!(k_i - \frac{\lambda_i}{\mu_i})} \right]^{-1}$$

4.2.5 Sistemas de redes de colas

Los modelos de sistemas de computadoras en la forma de redes de colas exponenciales algunas veces contienen sistemas de colas, de tipo $M/M/1$ y $M/M/\infty$. En el primer caso, las fórmulas anteriores se reducen a

$$P_i(x_i) = (1 - \lambda_i/\mu_i) (\lambda_i/\mu_i)^{x_i} \quad (4.84)$$

mientras que en el segundo se obtiene

$$P_i(0) = \left[\sum_{x_i=0}^{\infty} \frac{(\lambda_i/\mu_i)^{x_i}}{x_i!} \right]^{-1} = e^{-\lambda_i/\mu_i} \quad (4.85)$$

$$P_i(x_i) = e^{-\lambda_i/\mu_i} \frac{(\lambda_i/\mu_i)^{x_i}}{x_i!} \quad (4.86)$$

Si en un sistema de colas hay un sistema de colas k_i -lineal y N el número total de solicitudes circulando en esta red no excede* a k_i , entonces no hay cola en el sistema de colas k_i -lineal y este puede ser tratado como un sistema de colas del tipo $M/M/\infty$.

Se supone que la cadena o corriente de entrada tiene distribución Poisson con intensidad λ , las duraciones del servicio (tiempo de espera) en las colas unilínea $SC1$ y $SC2$ son exponencialmente distribuidas con intensidades de servicio μ_1 y μ_2 , respectivamente, las longitudes de la cola son ilimitadas, y la red es estacionaria (i.e. $k < \min(\mu_1, \mu_2)$). El estado de la red queda definido por el vector $\bar{r} = (x_1, x_2)$, donde x_1 y x_2 son los números de solicitudes encontrados en las colas unilínea $SC1$ y $SC2$.

Sea $P(x_1, x_2)$ la probabilidad estacionaria del estado $\bar{r} = (x_1, x_2)$.

El conjunto de ecuaciones lineales para la probabilidad del estado estacionario o ecuaciones de balance se escriben como sigue:

$$\begin{aligned} \lambda P(0, 0) &= \mu_2 P(0, 1) \\ [\lambda + \mu_2] P(0, x_2) &= \mu_1 P(1, x_2 - 1) + \mu_2 P(0, x_2 + 1) \\ [\lambda + \mu_1] P(x_1, 0) &= \mu_2 P(x_1, 1) + \lambda P(x_1 - 1, 0) \\ [\lambda + \mu_1 + \mu_2] P(x_1, x_2) &= \mu_1 P(x_1 + 1, x_2 - 1) \\ &\quad + \mu_2 P(x_1, x_2 + 1) + \lambda P(x_1 - 1, x_2) \end{aligned} \quad (4.87)$$

*Se van atendiendo inmediatamente según van llegando.

donde $x_1, x_2 \geq 1$. En este caso la condición de normalización

$$\sum_{\mathcal{T}} P(x_1, x_2) = 1,$$

la solución del conjunto de ecuaciones (4.87) adopta la forma

$$P(x_1, x_2) = \left(1 - \frac{\lambda}{\mu_1}\right) \left(\frac{\lambda}{\mu_1}\right)^{x_1} \left(1 - \frac{\lambda}{\mu_2}\right) \left(\frac{\lambda}{\mu_2}\right)^{x_2} P_1(x_1) P_2(x_2). \quad (4.88)$$

Esta fórmula significa que (comparela con (4.84)) cada sistema de colas puede ser considerado independientemente como un sistema del tipo $M/M/1$.

Un sistema de cola del tipo $M/M/1$, con una cola ilimitada y con un comportamiento estacionario de operación que tiene una distribución Poisson en su cadena o corriente de salida [Bu]. Asimismo como en los intervalos entre llegadas sucesivas, los intervalos entre las partidas de las solicitudes después del servicio tienen una distribución exponencial con parámetro λ .

Por lo tanto, se sigue que n sistemas de colas conectados en serie con colas ilimitadas, con probabilidad estacionaria $\bar{x} = (x_1, \dots, x_n)$ en las cuales x_1, \dots, x_n son el número de solicitudes en los sistemas de colas números $1, \dots, n$, respectivamente, se sigue que

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P_i(x_i) \quad (4.89)$$

donde, $P_i(x_i)$ es la probabilidad estacionaria del estado estacionario en el i -ésimo sistema de colas en el que el sistema de colas contiene x_i solicitudes. Cada sistema de colas puede ser considerado separadamente, como un sistema del tipo $M/M/1$ con la intensidad λ de llegadas de solicitudes e intensidad de servicios μ_i ($\lambda < \mu_i$).

Del mismo modo como los sistemas de colas pueden estar conectados en serie, la probabilidad estacionaria, $P(x_1, \dots, x_n)$ de un estado de la RC puede ser presentada en forma de un producto de las probabilidades de los estados de los SC por separado, $P_i(x_i)$, i.e., en la forma (4.89). Es interesante notar que el comportamiento de la Red de colas RC es el mismo que si se considera los comportamientos por separado de los SC con distribución Poisson, observando el hecho que las corrientes o cadenas en la RC con loops no tienen distribuciones Poisson.

Para utilizar este resultado en el caso general de una red que contenga n SC, se tienen que calcular las intensidades λ_i^l de arribos de solicitudes en el i -ésimo sistema de colas. Siendo λ_0 una intensidad resultante de la cadena o corriente de solicitudes que vienen a una red abierta desde una fuente externa. Introduzcanse

los factores c_i ($i = 1, \dots, n$) tal que $\lambda_i t = c_i \lambda_0$. La intensidad $\lambda_i t$ es el resultado de la intensidad λ_i de arribos de solicitudes desde una fuente externa ($\lambda_i = \rho_{0i} \lambda_0$) y las intensidades λ_j / μ_j de arribos de solicitudes en el i -ésimo sistema de colas desde el j -ésimo sistema de colas. Por lo tanto

$$c_i = \rho_{0i} + \sum_{j=1}^n c_j p_{ji} \quad i = 1, \dots, n. \quad (4.90)$$

Resolviendo este conjunto de ecuaciones para c_i , se define el estado con probabilidades estacionarias de cada sistema de colas SC por separado por las expresiones (4.74), (4.84) y (4.86) en los cuales la intensidad $c_i \lambda_0$ podría ser sustituida por la intensidad λ_i . La c_i , determina el número medio de las solicitudes que pasan sobre el i -ésimo sistema de colas.

Se considerará el caso de una RC cerrada que comprende dos sistemas de colas unilíneales. Cada sistema de colas puede ser considerado como una fuente finita. Es supuesto que N solicitudes circulan en la RC, las duraciones de los servicios solicitados en SC1 y SC2 son exponencialmente distribuidos con parámetros μ_1 y μ_2 . El estado de la red es determinado por el vector $\bar{x} = (x_1, x_2)$, con $x_1 + x_2 = N$.

El conjunto de las ecuaciones lineales para obtener las probabilidades de la distribución de los estados adopta la siguiente forma:

$$\begin{aligned} \mu_1 P(N, 0) &= \mu_2 P(N-1, 1); & \mu_2 P(0, N) &= \mu_1 P(1, N-1) \\ [\mu_1 + \mu_2] P(x_1, x_2) &= \mu_1 P(x_1+1, x_2-1) + \mu_2 P(x_1-1, x_2+1) \end{aligned} \quad (4.91)$$

donde $0 < x_1, x_2 < N$.

Donde, ahora tomando en cuenta la condición de normalización

$$\sum_{\bar{x}} P(x_1, x_2) = 1,$$

se obtiene

$$\begin{aligned} P(x_1, x_2) &= P(N, 0) \left(\frac{\mu_1}{\mu_2} \right)^{x_2} \\ &= P(N, 0) \mu_1^{x_1+x_2} \left(\frac{1}{\mu_1} \right)^{x_1} \left(\frac{1}{\mu_2} \right)^{x_2} \\ &= C(N) \left(\frac{1}{\mu_1} \right)^{x_1} \left(\frac{1}{\mu_2} \right)^{x_2} \\ &= C(N) P(x_1) P(x_2) \end{aligned} \quad (4.92)$$

donde

$$C(N) = P(N, 0) \rho_i^N = \left[\sum_{i=0}^N \left(\frac{1}{\rho_1} \right)^i \left(\frac{1}{\rho_2} \right)^{N-i} \right]^{-1}$$

es la constante de normalización.

La expresión para las probabilidades de la distribución de los estados de una red cerrada son independientes de las intensidades λ_i y dependen de los factores c_i . En el ejemplo en cuestión $c_1 = c_2 = 1$.

Como en el caso de RC abiertas, las probabilidades de la distribución de los estados de una RC cerrada, donde el i -ésimo sistema de colas puede tener k_i líneas ($i = 1, \dots, n$) son el resultado de la multiplicación conjunta de la probabilidad $P_i(x_i)$ de cada estado por separado de los sistemas de colas (SC). El único hecho a distinguir es el factor de normalización $C(N)$, pues debe tomarse en cuenta que $\sum_{i=1}^n x_i = N$ [GN].

Puesto que no hay corrientes o cadenas de solicitudes externas en una RC cerrada ($\rho_{0i} = 0$ y $\lambda_i = 0$, con $i = 1, \dots, n$) la ecuación (4.90) adopta la forma

$$c_i = \sum_{j=1}^n c_j \rho_{ij}. \quad (4.93)$$

Para obtener una única solución de este sistema se supone que $c_1 = 1$. En este caso, la cantidad c_i puede ser interpretada como el número medio de solicitudes que pasan por el i -ésimo sistema de colas.

Por ejemplo, sea una RC con $c_1 = 1$, $c_2 = \rho$ y $c_n = q$. Para esta red cerrada, la probabilidad de un estado estacionario específico dado por el vector $\bar{x} = (x_1, \dots, x_n)$ para $\sum_{i=1}^n x_i = n$ determinada por la expresión

$$P(\bar{x}) = C(N) \prod_{i=1}^n P_i(x_i) = C(N) \prod_{i=1}^n A_i(x_i) (c_i / \rho_i)^{x_i} \quad (4.94)$$

donde $P_i(x_i)$ es la probabilidad del estado estacionario con x_i solicitudes en el i -ésimo sistema de colas; $A_i(x_i)$ está determinado por (4.75).

La constante de normalización es

$$C(N) = \left[\sum_{\bar{x}} \prod_{i=1}^n P_i(x_i) \right]^{-1}. \quad (4.95)$$

Para todas las RC que han sido consideradas la expresión de las probabilidades del estado estacionario tienen la forma de un producto convencional (o una forma multiplicativa). Sin embargo esta forma multiplicativa se preserva para una amplia

clase de RC con solicitudes heterogéneas que se mueven en la RC en diferentes caminos y pueden ser servidas en diferentes sistemas de colas con diferentes intensidades. Las solicitudes heterogéneas son divididas en clases. Las RC de este tipo pueden ser cerradas, abiertas o combinadas: cerradas para solicitudes de algunas clases y abiertas para algunas de otras clases [BCM].

En la modelación de sistemas de cómputo, es importante tener en cuenta la heterogeneidad de las solicitudes bajo el servicio. Algunos ejemplos de tales solicitudes pueden ser referencias a la memoria en diferentes operaciones que utilicen diferentes procesadores (algunas son las operaciones escalares o vectoriales), diferentes programas de sistema y variedad de usuarios que llaman para un servicio desde diferentes dispositivos, varios tipos de mensajes transmitidos en una red de computadoras, etc.

4.3 Modelo del sistema de Cómputo de tiempos distribuidos (Modelo del servicio de red NIS).

El modelo probabilístico del servicio de red NIS, cuenta con una sola cola de servicio de solicitudes de las máquinas clientes.

Sea t_n , el tiempo aleatorio empleado por la máquina cliente, y sea t_{pr} , el tiempo aleatorio (de duración) que se toma el servidor en realizar una tarea relativa a una solicitud del cliente. Se supondrá que son exponencialmente distribuidos con valores medios $E(t_n)$ y $E(t_{pr})$, respectivamente. Además, el cliente no envía una nueva solicitud (o llamada) al sistema mientras que no reciba la respuesta del sistema a una solicitud previa.

Se supone que el programa interruptor es configurado para ser instantáneo, es decir, sin pérdidas de tiempo en las operaciones de interrupción y desconexión.

El modelo es construido para determinar las siguientes características básicas: el número medio de clientes esperando una respuesta y el tiempo medio de espera de la respuesta a una solicitud hecha por un cliente. Los parámetros iniciales son el tiempo medio tomado por el cliente para la preparación de la solicitud $E(t_n) = 1/\mu_1$, el tiempo medio que se lleva el procesador en manejar la solicitud $E(t_{pr}) = 1/\mu_2$, y el número de usuarios N en las máquinas cliente.

Este modelo se representa como una RC cerrada con dos sistemas de colas: una N -línea sin cola (fase de cliente), donde N , representa el número de clientes y una

cola unilineal (fase del servidor)¹. El estado de esta RC es determinado por el par $(N - j, j)$, donde j es el número de solicitudes que corren en la fase de servidor y $N - j$ es el número de solicitudes en la fase de cliente.

De acuerdo con (4.94), las probabilidades del estado estacionario de una RC dada están determinadas por la siguiente expresión:

$$P(N - j, j) = C(N) / (N - j)! \mu_1^{N-j} \mu_2^j. \quad (4.96)$$

En este caso se obtiene una expresión compacta para la constante de normalización $C(N)$. De (4.96), después de una transformación, se tiene que

$$\begin{aligned} P(N - j, j) &= C(N) \left(\frac{1}{N! \mu_1^N} \right) \frac{N!}{(N-j)!} \left(\frac{\mu_1}{\mu_2} \right)^j \\ &= P(N, 0) \frac{N!}{(N-j)!} \left(\frac{\mu_1}{\mu_2} \right)^j \end{aligned} \quad (4.97)$$

En concordancia con las condiciones de normalización,

$$\sum_{j=0}^N P(N - j, j) = 1.$$

Por lo tanto, la probabilidad de que el procesador este ocioso es

$$P(N, 0) = \left[\sum_{i=0}^N \frac{N!}{(N-i)!} \left(\frac{\mu_1}{\mu_2} \right)^i \right]^{-1}. \quad (4.98)$$

Como paso inicial, desde (4.97) y (4.98) se obtiene

$$P(N - j, j) = \frac{\frac{N!}{(N-j)!} \left(\frac{\mu_1}{\mu_2} \right)^j}{\sum_{i=0}^N \frac{N!}{(N-i)!} \left(\frac{\mu_1}{\mu_2} \right)^i}. \quad (4.99)$$

El número medio de clientes esperando respuestas es igual al número medio de solicitudes encontradas en el SC2. Por otra se tiene que

$$M_2 = \sum_{j=0}^N j P(N - j, j) = \sum_{j=0}^N \frac{\frac{N!}{(N-j)!} \left(\frac{\mu_1}{\mu_2} \right)^j}{\sum_{i=0}^N \frac{N!}{(N-i)!} \left(\frac{\mu_1}{\mu_2} \right)^i}. \quad (4.100)$$

¹Se supone un solo servidor de ahí que la cola sea unilineal

El tiempo medio que le toma al cliente esperar la respuesta a una solicitud es igual al tiempo medio que se quedan las solicitudes en la fase de sistema, i.e. en SC2 (Vease [BCM]):

$$T_2 = \frac{M_2}{\mu_2}. \quad (4.101)$$

4.3.1 Errores y retardos

El problema principal de este servicio al igual que de NFS, es la comunicación entre la máquina servidor y la máquina cliente. Esta comunicación puede verse entorpecida por la ineficiencia de la red o de los protocolos.

En el caso de la ineficiencia de los protocolos, no hay forma de atacarlos con tiempo, ya que solo dependen de la configuración interna y su construcción, y debido a esto la predicción de los errores en este punto no puede ser predecible. De esta forma para su corrección se auxilia de herramientas básicas de Unix para conocer que tipo de versión se esta manejando y si los programas base estan ejecutandose bien.

Si las fallas tienen origen desde el protocolo de transporte, entonces su recuperación puede ser predecible y es dada en la sección 4.6 y 4.5.

La recuperación rápida de las fallas y retardos es esencial, debido a que el usuario esta ejecutando la aplicación interactivamente. Los problemas que pueden generarse en el servidor en el retardo de lectura en los discos, es despreciable por lo que su análisis no se incluye.

Otro de los problemas que afectan a este servicio se da cuando el NFS tiene fallas. Esto se debe a la relación que existe entre estos dos servicios, ya que si NFS no puede ser accedido, tampoco lo será la información de los usuarios y con ello NIS no podrá funcionar debidamente.

4.4 Modelo de sistemas de archivos distribuidos remotamente (Modelo del servicio de red NFS).

Un servidor de NFS además de basarse en protocolos de red, codificación de datos, etc. requiere del auxilio de un protocolo para poder compartir el recurso de espacio

¹Ver [BCM].

4.4 Modelo de sistemas de archivos distribuidos remotamente (Modelo del servicio de red NFS).

72

en disco a estaciones de trabajo externas de manera transparente. El protocolo que auxilia en esta tarea es el llamado **RPC**⁵. Debido a que la transmisión de datos con **TCP**⁶ es confiable, **RPC** manda las peticiones sin esperar a que la respuesta de una petición anterior sea devuelta por el servidor; este procedimiento se realiza con un previo acuerdo entre el servidor y los clientes para que el servidor no mande mensajes de respuesta, y esta es una de las funciones que hacen que el rendimiento aumente.

Una función más para obtener mayor rendimiento, es la llamada *Entrada y Salida Asíncrona de un cliente NFS*, esto es, **NFS** mantiene bloques de datos en buffers (buffer cache) vía lectura y escritura. El sistema de entrada/salida lee por adelantado el próximo bloque del archivo (los bloques de datos deberán estar en el bufer) de una petición de un cliente. Por el contrario, cuando la petición es de escritura, los bloques de datos son escritos en el buffer y más tarde son transferidos a disco, cuando se requiere espacio suficiente en el buffer. Este proceso es conocido como "leer por adelantado (*read ahead*)" y escribir más tarde (*delayed write*). Debido a esta función el retardo de lectura y escritura a disco es despreciable.

Para ejecutar un **RPC**, este proceso es requerido por **NFS**. En **Unix** opera el daemon **iod**, el cual puede leer por adelantado y escribir después. Generalmente operan cuatro procesos al mismo tiempo para ejecutar operaciones en paralelo.

Ya que se tienen cuatro colas que procesan las peticiones de **NFS** de los clientes (Vease sección 3.2), que cada una de estas se distribuye como una Poisson y que la distribución del servicio se distribuye como una exponencial, el modelo de una cola de tipo **M/M/k** (Ver sección anterior), se ajusta al modelo de **NFS**.

Se asume que las llegadas se distribuyen como Poisson con media λ_n , debido a que cada petición se procesa de igual forma en cualquier buffer se supone que esta λ_n para cualquier estado n es igual a λ .

Para obtener el valor promedio de servicio se define:

$$\begin{aligned} \mu_n &= \text{mínimo}[n\mu, k\mu] \\ &= \begin{cases} n\mu & \text{para } 0 \leq n \leq k \\ k\mu & \text{para } m \leq n \end{cases} \end{aligned} \quad (4.102)$$

Se denotará a ρ como el factor de carga en el servidor dado por:

$$\rho = \frac{\lambda}{\mu} \quad (4.103)$$

⁵Remote Procedure Call es el amazón en donde se transportan las peticiones al servidor.

⁶ver B

Sea U definida como el factor de utilización de servicio para un sistema con múltiples servidores:

$$U = \frac{\rho}{k} \quad (4.104)$$

donde k es el número de servidores en el sistema. El factor U puede interpretarse como la fracción esperada de servidores ocupados cuando cada servidor tiene la misma función de distribución de tiempo de servicio, o alternativamente puede definirse como la fracción esperada de capacidad del sistema que esta en uso.

Los estados de transiciones de un sistema se definen como los cambios del mismo en un lapso de tiempo y se denotaran por p_n , donde n es el estado de transición n del sistema. Para obtener la probabilidad de los estados de transición p_n , se analizará en dos partes, cuando $n \geq k$ y cuando $n \leq k$ de acuerdo con las ecuaciones de equilibrio (ver capítulo 4 primera parte ecuaciones 4.28 y 4.29) se tiene que:

$$(\lambda + n\mu)p_n = \lambda p_{n-1} + (n+1)\mu p_{n+1} \text{ para } n \geq 1 \quad (4.105)$$

o

$$(\rho/n)p_n = \rho p_{n-1} + (n+1)p_{n+1} \quad (4.106)$$

De acuerdo con la ecuación 4.32 de la primera parte:

$$p_n = \frac{\rho^n}{n!} p_0 \text{ para } n \leq k \quad (4.107)$$

Similantemente se tiene para $n \geq k$, la ecuación de equilibrio:

$$(\lambda + k\mu)p_n = \lambda p_{n-1} + k\mu p_{n+1} \quad (4.108)$$

o

$$(\rho + k)p_n = \rho p_{n-1} + k p_{n+1} \text{ para } n \geq k \quad (4.109)$$

Así de acuerdo a la ecuación 4.80 se tiene:

$$p_n = \frac{\rho^n p_0}{k! k^{n-k}} \text{ para } n \geq k \quad (4.110)$$

**4.4 Modelo de sistemas de archivos distribuidos remotamente
(Modelo del servicio de red NFS).**

y tomando la ecuación 4.82:

$$\rho_0 = \left[\sum_{n=0}^{k-1} \frac{\rho^n}{n!} + \frac{\rho^k}{k!} \frac{1}{(1 - \frac{\rho}{k})} \right]^{-1} \quad (4.111)$$

Para obtener el rendimiento de un servidor es necesario tomar en cuenta los factores que intervienen en su modificación, estos son:

- N Número promedio de peticiones en el sistema (los que estan en espera y los que estan en servicio).
- N_q Número promedio de peticiones en espera a ser servidas.
- T Tiempo promedio que un mensaje espera en el sistema (incluye el tiempo esperado en la cola más el tiempo que fue servido).
- T_q Tiempo promedio que una petición espera en la cola.

La longitud de la cola se puede estimar considerando los estados para los cuales $n \geq k$, debido a que una cola puede formarse si los k servidores estan ocupados. Por definición:

$$N_q = \frac{\rho^k (\frac{\rho}{k})}{k! (1 - \frac{\rho}{k})^2} \rho_0 \quad (4.112)$$

Tomando en cuenta la fórmula de Little:

$$T_q = \frac{N_q}{\lambda} \quad (4.113)$$

y

$$N = T\lambda \quad (4.114)$$

donde:

$$\begin{aligned} T &= T_q + T_s \\ &= T_q + E[s] \\ &= T_q + \frac{1}{\mu} \end{aligned} \quad (4.115)$$

debido a que el tiempo promedio de procesamiento del mensaje es: $E[s] = \frac{1}{\mu}$. Así el rendimiento estará representado por:

$$S = \frac{\lambda}{T} \quad (4.116)$$

4.4.1 Evaluación de Errores y Fallas

Por ser NFS un servicio de red, las fallas y errores se deben en su mayoría al estado de la red. Los principales problemas en el servicio de red NFS, son:

- Fallas en el protocolo *Port mapper*.
- Problemas a nivel de transporte.
- Fallas en el servidor.

En cuanto a las fallas del protocolo de comunicación, se detectan a través de monitoreo y no son predecibles, pues depende de la configuración interna del servidor y sus clientes. Estas fallas en su mayoría se dan porque las versiones no son compatibles entre sí, por lo que el protocolo no puede comunicarse.

Los problemas a través de transporte ya fueron evaluados en la sección de red, la cual se debe al tráfico de información en las redes generado por el retardo de envío (**fdi**) o el número de colisiones presentadas (**ethernet**) y retransmisiones o las llegadas de los tiempos límite para transmitirlos.

Como evaluar las fallas en el servidor, solo se puede hacer a través de los retardos generados por la demanda de peticiones hacia él.

La probabilidad de que todos los servidores k este ocupados es:

$$\begin{aligned} P[\text{servidores ocupados}] &= \sum_{n=k}^{\infty} P_n \\ &= \sum_{n=k}^{\infty} P_0 \frac{\rho^n}{k! k^{n-k}} \\ &= \frac{\rho^k}{k! (1-\rho)} \\ &= \frac{\rho^{k-1}}{\sum_{n=0}^{k-1} \frac{\rho^n}{n!} + \frac{\rho}{k! (1-\rho)}} \end{aligned} \quad (4.117)$$

Esta expresión es conocida como *Fórmula de retardo de tipo Erlang*.

Como se puede observar en la ecuación 4.117, esta probabilidad esta en función de la carga de trabajo ρ en el sistema. Debido a esto si la carga es alta, la probabilidad de que los servidores esten ocupados es casi uno. Si cada cliente se encuentra activa enviando peticiones las probabilidades de error por tiempo límite aumentan y con ello los errores en el sistema de cómputo en general.

Debido a que en este servicio es despreciable el tiempo que tarda en la lectura y escritura de disco el modelo de NFS, no lo incluye.

4.5 CSMA con Detector de Colisiones,(CSMA/CD)

Para poder definir este protocolo, es necesario enunciar la base para su construcción que es el protocolo **Carrier Sense Multiple Access (CSMA)**. Se define como un protocolo de envío de información a través de un medio físico y funciona de la siguiente manera: cuando una estación desea enviar un paquete mediante algún medio, primero se cerciora si alguna otra transmisión se encuentra en progreso, así, cuando una estación manda un mensaje, las restantes se dan cuenta y se esperan a que el canal se desocupe para enviar algún mensaje. Esta técnica se basa en el método de *Primero escucho y después hablo (Listen Before Talk, LBT)*.

Con efectos de llevar este protocolo a la implementación, se hicieron modificaciones creando así el protocolo llamado *CSMA con Detector de Colisiones, (Carrier Sense Multiple Access with Collision Detection, CSMA/CD)*, a diferencia del CSMA este protocolo se basa en el método *Escucho Mientras Hablo (Listen While Talk, LWT)*, esto quiere decir, que si una máquina necesita enviar un paquete, lo envía y al mismo tiempo esta alerta, por si una transmisión interrumpe su envío exitoso. En este protocolo de envío de información, se basa un software implementado por las compañías Digital y Xerox llamado **Ethernet**.

4.5.1 Características del (CSMA/CD)

Dentro de la *CSMA/CD* existen tres tipos de metodologías diferentes: no-persistente, 1-persistente, p-persistente. Cuando una *CSMA/CD* detecta una colisión, inmediatamente manda una breve señal de alarma, notificando a todas las estaciones la colisión, cada colisión es detectada mediante un monitoreo que se realiza directamente en el canal.

Cuando el *CSMA/CD* esta listo para transmitir mensajes, el comportamiento para el envío de estos varía de acuerdo al tipo de metodología utilizada. Por ejemplo supongase que la estación *Ei* envía un mensaje, entonces:

- Si el canal esta desocupado ("el canal esta libre"):
El paquete se transmite inmediatamente si la metodología es no-persistente o 1-persistente, en el caso de ser p-persistente, el paquete es enviado con probabilidad p .
- Si el canal esta ocupado y la metodología es:
 - no-persistente El paquete retrocede y el algoritmo se repite.

- 1-persistente La estación retrasa la transmisión hasta que el canal este ocioso e inmediatamente transfiere el mensaje.
- p-persistente La estación suspende la transmisión hasta que el canal este ocioso y entonces transmite el mensaje con una probabilidad p , o prorroga la transmisión con un espacio de $(1 - q)$ de probabilidad.

Si una colisión es detectada, la estación aborta el mensaje y manda una señal de alerta en el canal. Una vez que esta señal ha sido transmitida, la estación toma su paquete y espera una cantidad de tiempo aleatorio y vuelve a intentar mandar el mensaje, por su parte las demás estaciones saben que ocurrió una colisión y no envían ningún mensaje hasta un tiempo aleatorio; de esta forma disminuye la probabilidad de que vuelvan a colisionar las máquinas anteriores, esto esta garantizado por un método llamado *Retroceso Exponencial Binario Truncado*, (*Truncated Binary Exponential Backoff*), donde se toman en cuenta hasta 16 intentos de transmisión (el original y 15 más), si el mensaje no logró transmitirse en este número de intentos, el paquete es desechado y el evento es reportado como un error.

Generalmente, si un paquete ha sido transmitido n veces, la próxima transmisión se retrasa por un entero r , este entero es seleccionado en el rango de $0 \leq r \leq 2^k$, donde k es el mínimo del número de transmisiones n y un entero 10, ($k = \min(n, 10)$).

4.5.2 Análisis de un CSMA/CD no-persistente

Se sabe que las llegadas de peticiones se distribuyen como una Poisson, sea δ el tiempo de detección de colisión y sean las retransmisiones consideradas como arribos Poisson independientes.

En la figura 4.1, se muestra una secuencia de eventos en una colisión. Comenzando con $t = 0$ la estación A transmite un paquete, durante el período de vulnerabilidad¹¹ a , la estación B envía un paquete al tiempo $t = \mu_i$, y al tiempo $t = a + \delta$, B detecta el mensaje que A esta transmitiendo e inmediatamente suspende el envío y manda una señal de alarma de longitud b . En el tiempo $t = \mu_i + a + \delta$, A detecta la colisión y suspende la transmisión, mandando de igual forma una señal de alarma de longitud b , así el canal llega a estar desocupado en el tiempo $t = 2a + \mu_i + b + \delta$.

Sea $E[B]$ el valor esperado de la duración de un período ocupado, $E[I]$ el valor esperado de un período desocupado y $E[U]$ el promedio esperado de tiempo que

¹¹El período de vulnerabilidad es el tiempo entre envío de mensajes que existe, en caso de que alguna máquina mande un mensaje durante este tiempo se sufre una colisión.

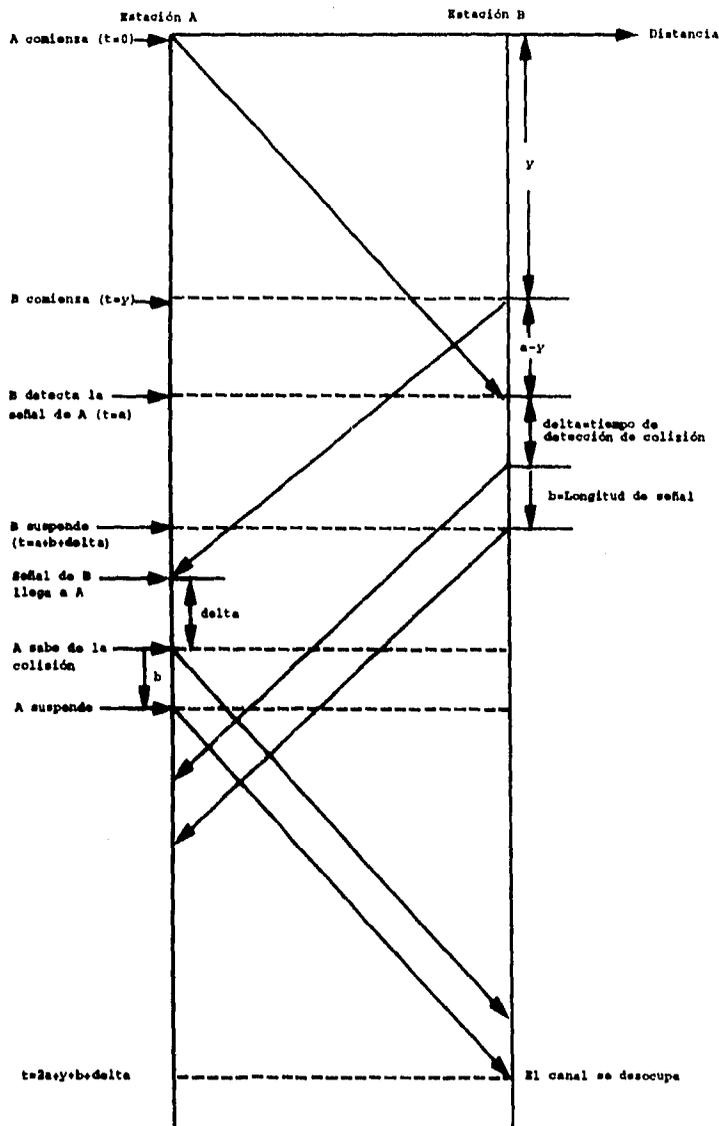


Figura 4.1: Secuencia de eventos en una colisión usando el protocolo CSMA/CD

dura un canal sin colisiones y sea S el número de paquetes transmitidos exitosamente, entonces la relación que existe entre estos términos es:

$$S = \frac{E[U]}{E[B] + E[I]} \quad (4.118)$$

Para encontrar $E[U]$ notese que la probabilidad de que un mensaje se transmita exitosamente (sin colisiones) es igual a la probabilidad de que ninguna estación transmita un mensaje durante el período de vulnerabilidad a , definiendo a G como la carga presentada en el canal. De acuerdo a que los arribos se distribuyen como una Poisson ** se tiene que:

$$E[U] = e^{-aG} \quad (4.119)$$

El período que el canal se encuentra desocupado se puede definir como el período entre un canal ocupado y el siguiente arribo de mensaje a la red, así este período se puede representar como:

$$E[I] = \frac{1}{G} \quad (4.120)$$

Ahora, solo falta examinar el período en donde el canal se encuentra ocupado $E[B]$. Un período está ocupado si está transmitiendo un mensaje o si existe una colisión. En el caso de que la transmisión del mensaje sea exitosa, el canal no estará desocupado hasta $1 + a$ ††, y si existe una colisión, no estará desocupado hasta el tiempo $2a + b + y + \delta$, tomando $\delta \rightarrow 0$ se tiene que:

$$\begin{aligned} E[B] &= P(\text{transmisión exitosa})E[1 + a] \\ &+ P(\text{transmisión incompleta})E[2a + b + y_i] \\ &= e^{-aG}(1 + a) + (1 - e^{-aG})(2a + b + E[y_i]) \end{aligned} \quad (4.121)$$

para evaluar $E[y_i]$ es necesario saber su función de densidad por lo que se define la probabilidad $P(y \geq y_i)$, $0 \leq y \leq a$:

$$f(y) = P(y \geq y_i) = 1 - P(y < y_i) \quad (4.122)$$

**Véase [Ge], pág 218

††Se supone 1 la longitud de transmisión del paquete

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

donde $P(y < y_i)$ es la probabilidad de que no exista ninguna llegada en el intervalo $[0, a]$, esto puede ser escrito como:

$$\begin{aligned} P(y < y_i) &= P(\text{no arribos en } [0, y] | \text{hubo un arribo al final del intervalo } [0, a]) \\ &= \frac{P(\text{no arribos en } [0, y])P(\text{hubo un arribo en el intervalo } [y, a])}{P(\text{hubo un arribo al final del intervalo } [0, a])} \\ &= \frac{e^{-yG}[1 - e^{-G(a-y)}]}{1 - e^{-aG}} \end{aligned} \quad (4.123)$$

de acuerdo con la ecuación 4.121 se tiene:

$$f(y) = \frac{1 - e^{-yG}}{1 - e^{-aG}}, 0 \leq y \leq a \quad (4.124)$$

y para encontrar $E(y_i)$:

$$\begin{aligned} E(y_i) &= \int_0^a yf(y) dy \\ &= \frac{1}{G} - \frac{ae^{-aG}}{1 - e^{-aG}} \end{aligned} \quad (4.125)$$

Insertando la ecuación 4.125 en 4.121 y sustituyendo estos resultados en la ecuación 4.118 se tiene que el número de paquetes transmitidos exitosamente para un modelo de tipo **CSMA/CD no-persistente** es:

$$S = \frac{Ge^{-aG}}{Ge^{-aG} + bG(1 - e^{-aG}) + 2aG(1 - e^{-aG}) + (2 - e^{-aG})} \quad (4.126)$$

4.5.3 Análisis de un sistema CSMA/CD no-persistente con intervalos de tiempo

Es importante aclarar que este modelo se ajusta al **CSMA/CD no-persistente**, sin tomar en cuenta que los envíos de mensajes pueden hacerse solo en fracciones de tiempo (unslotted).

Para un modelo **CSMA/CD no-persistente**, con intervalos de tiempo para envíos de mensajes (slotted), se redefinen algunas variables importantes:

El promedio de tiempo que dura el canal sin colisiones para este punto se representa por $E[U]$ y además:

$$E[U] = P_s \quad (4.127)$$

donde P_s es una probabilidad condicional que representa la fracción de tiempo que una transmisión es exitosa, sea a una fracción de tiempo, entonces:

$$\begin{aligned} P_s &= P(\text{un paquete llegue en el tiempo } a | \text{algunos arribos ocurran}) \\ &= \frac{P(\text{un paquete llegue en } a \text{ y algunos arribos ocurran})}{P(\text{algunos arribos ocurran})} \\ &= \frac{P(\text{un paquete llegue en } a)}{P(\text{algunos arribos ocurran})} \end{aligned} \quad (4.128)$$

usando que las llegadas se distribuyen como una Poisson se tiene que:

$$P(\text{un paquete llegue en } a) = aG e^{-aG} \quad (4.129)$$

y que algunos arribos ocurran:

$$P(\text{algunos arribos ocurran}) = 1 - e^{-aG} \quad (4.130)$$

entonces la ecuación 4.128 se define como:

$$E[U] = \frac{aG e^{-aG}}{1 - e^{-aG}} \quad (4.131)$$

El tiempo libre del canal (tiempo de ociosidad de este), esta representado siempre por un entero positivo $l \geq 0$.

En el caso de que un mensaje llegue al terminar el período a entonces en la siguiente fracción de tiempo para envío de mensajes ya existirá una petición, por lo que el período de ociosidad del canal será 0¹¹. Si en el primer período de tiempo a , no hay ninguna llegada en los siguientes $l - 1$ períodos estarán también desocupados, hasta que exista una petición en el período l , así, el período de ociosidad del canal estaría dado por $P(l-1)$, sea p la utilización o eficiencia del canal, definido

¹¹El canal estará ocupado por lo menos con una llegada

El promedio de tiempo que dura el canal sin colisiones para este punto se representa por $E[U]$ y además:

$$E[U] = P_s \quad (4.127)$$

donde P_s es una probabilidad condicional que representa la fracción de tiempo que una transmisión es exitosa, sea a una fracción de tiempo, entonces:

$$\begin{aligned} P_s &= P(\text{un paquete llegue en el tiempo } a | \text{algunos arribos ocurran}) \\ &= \frac{P(\text{un paquete llegue en } a \text{ y algunos arribos ocurran})}{P(\text{algunos arribos ocurran})} \\ &= \frac{P(\text{un paquete llegue en } a)}{P(\text{algunos arribos ocurran})} \end{aligned} \quad (4.128)$$

usando que las llegadas se distribuyen como una Poisson se tiene que:

$$P(\text{un paquete llegue en } a) = aG e^{-aG} \quad (4.129)$$

y que algunos arribos ocurran:

$$P(\text{algunos arribos ocurran}) = 1 - e^{-aG} \quad (4.130)$$

entonces la ecuación 4.128 se define como:

$$E[U] = \frac{aG e^{-aG}}{1 - e^{-aG}} \quad (4.131)$$

El tiempo libre del canal (tiempo de ociosidad de este), esta representado siempre por un entero positivo $l \geq 0$.

En el caso de que un mensaje llegue al terminar el período a entonces en la siguiente fracción de tiempo para envío de mensajes ya existirá una petición, por lo que el período de ociosidad del canal será 0 ¹¹. Si en el primer período de tiempo a , no hay ninguna llegada en los siguientes $l - 1$ períodos estarán también desocupados, hasta que exista una petición en el período l , así, el período de ociosidad del canal estaría dado por $P(l-1)$, sea p la utilización o eficiencia del canal, definido

¹¹El canal estará ocupado por lo menos con una llegada

como el promedio de fracción de tiempo que un canal esta ocupado. Entonces de la ecuación 4.130 es:

$$\rho = 1 - e^{-aG} \quad (4.132)$$

y así:

$$P(I = 0) = \rho = 1 - e^{-aG} \quad (4.133)$$

para el caso en que $I = 1$ se tiene, una probabilidad conjunta que es: la probabilidad de que no hayan arribos en la última fracción de tiempo que el canal estuvo ocupado, por la probabilidad de que existan algunos arribos en la siguiente fracción de tiempo a , esto es dado por:

$$P(I = 1) = (1 - \rho)\rho \quad (4.134)$$

extendiendo este argumento para i períodos seguidos de ociosidad del canal:

$$P(I = i) = (1 - \rho)^i \rho \quad (4.135)$$

lo cual describe la función de distribución geométrica, para encontrar el valor esperado de período de ociosidad del canal; sea la variable Q con media:

$$E[Q] = \frac{1 - \rho}{\rho} \quad (4.136)$$

entonces:

$$\begin{aligned} E[I] &= aE[Q] \\ &= \frac{ae^{-aG}}{1 - e^{-aG}} \end{aligned} \quad (4.137)$$

En este esquema, los paquetes solo son mandados al comienzo de los períodos de tiempos de envíos de mensajes, en el caso de una colisión la longitud del tiempo que tarda en desocuparse el canal es $(2a + b)$, donde b es un número total de los períodos de envío.

$$\begin{aligned} E[B] &= P(\text{transmisión exitosa})E(1 + a) \\ &\quad + P(\text{transmisión no exitosa})E[2a + b] \\ &= P_s(1 + a) + (1 - P_s)(2a + b) \end{aligned} \quad (4.138)$$

Sustituyendo los resultados anteriores, el número total de mensajes transmitidos exitosamente es:

$$S = \frac{aGe^{-aG}}{aGe^{-aG} + b(1 - e^{-aG} - aGe^{-aG}) + a(2 - e^{-aG} - aGe^{-aG})} \quad (4.139)$$

4.5.4 Análisis para el esquema CSMA/CD 1-persistente

En el esquema **CSMA/CD 1-persistente** la máquina aplaza la transmisión de un mensaje hasta que el canal este desocupado, y una vez se cumpla esto inmediatamente lo transmite, esto es el paquete es transmitido con probabilidad 1, de ahí que sea 1-persistente.

Una aproximación a este esquema utiliza tres estados de una cadena de Markov dados por Soharaby, Molle y Venetsanopoulos * del cual se genera el modelo SMV, se asume un numero infinito de usuarios (usuarios que transmiten bastante información en períodos cortos de tiempo), donde colectivamente generan tráfico con distribución Poisson y con una velocidad de G paquetes por tiempo de transmisión.

El análisis se basa en una secuencia de períodos "subocupados", que ocurren en cada período ocupado, todas las máquinas que generan paquetes mientras el canal esta ocupado en el j -ésimo período subocupado, puede transmitirlos en el $j + 1$ período subocupado.

El modelo SMV define los siguientes tres tipos de períodos subocupados:

1. Períodos en los cuales el canal esta ocioso, es decir se tienen transmitiendose cero paquetes.
2. Períodos en los que se tiene un solo paquete a transmitir.
3. Períodos en donde se tiene más de un paquete a transmitir †

Sea cada uno de estos períodos denotados por $i = 0, 1$ y 2 , al comienzo de los períodos subocupados, entonces la probabilidad de transición P_{01} (la probabilidad de pasar del estado 0, período ocioso; al estado 1, período en el cual se tiene un paquete) es igual a 1 dado que el esquema es 1-persistente

*Para más detalles consultar [Ge], apéndice E

†Generalmente el resultado es una colisión

Sea $E[T_i]$ el promedio de tiempo que el protocolo dedica en el estado i y sea $\pi = \{\pi_0, \pi_1, \pi_2\}$ la función de distribución estacionaria de la cadena de Markov con los tres estados. $E[T_0]$ es el valor medio del período en el cual el canal es ocioso de acuerdo a la ecuación 4.120 se tiene:

$$E[T_0] = \frac{1}{G} \quad (4.140)$$

y $E[T_1]$ se deriva de la ecuación 4.121:

$$E[T_1] = (1 - e^{-aG}) \left(2a + b + \frac{1}{G} \right) + e^{-aG} \quad (4.141)$$

Para obtener las probabilidades estacionarias P_{10} y P_{11} se tiene:

$$\begin{aligned} P_{10} &= P(\text{ninguna llegada en el intervalo } t, \text{ y el anterior fue éxito}) \\ &\quad + P(\text{ninguna llegada en el intervalo } a + b + y, \text{ el anterior fue colisión}) \\ &= e^{-G} e^{-aG} + \int_0^a e^{-G(a+b+y)} G e^{-yG} dy \\ &= e^{-G(1+a)} + \frac{1}{2} e^{-G(a+b)} (1 - e^{-2aG}) \end{aligned} \quad (4.142)$$

$$\begin{aligned} P_{11} &= P(\text{una llegada durante el intervalo } t, \text{ dado que fue un éxito}) \\ &\quad + P(\text{una llegada durante el intervalo } (a + b + y), \text{ dado que fue una colisión}) \\ &= G e^{-G} e^{-aG} + \int_0^a G(u + b + y) e^{-G(a+b+y)} G e^{-Gy} dy \\ &= G e^{-G(1+a)} + \frac{1}{3} e^{-G(a+b)} [(1 - e^{-2aG})(1 + 2G(u + b)) - 2aG e^{-2aG}] \end{aligned} \quad (4.143)$$

Por último, el estado $i = 2$ se genera cuando dos o más paquetes son transmitidos simultáneamente, la longitud de este período es: la fracción del período ocupado a , más el intervalo que dura la señal de alarma de colisión b y el segundo intervalo de tiempo del período ocupado a , lo que se representa como:

$$E[T_2] = 2a + b \quad (4.144)$$

Las probabilidades de transición P_{2n} , se obtienen de acuerdo a que se requiere la probabilidad de que n paquetes lleguen con media G en el intervalo t , donde $t =$

$b + a$, ya que es el período de tiempo en donde las llegadas no generan colisiones puesto que el período de vulnerabilidad es a , entonces:

$$P_{20} = P(\text{ninguna llegada en el intervalo } (a + b), \text{ dado que hubo colisión en el último intervalo}) \\ = e^{-C(a+b)} \quad (4.145)$$

y P_{21} se define como:

$$P_{21} = P(\text{un arribo durante el intervalo } (a + b), \text{ dado que hubo colisión}) \\ = (a + b)C e^{-(a+b)C} \quad (4.146)$$

Las distribuciones estacionarias son $\pi_1, \pi_2 / \pi_0$:

$$\pi_1 = \frac{P_{20} + P_{21}}{(1 - P_{10} - P_{11})(1 + P_{20}) + (1 + P_{10})(P_{20} + P_{21})} \quad (4.147)$$

$$\pi_2 = \frac{1 - P_{10} - P_{11}}{(1 - P_{10} - P_{11})(1 + P_{20}) + (1 + P_{10})(P_{20} + P_{21})} \quad (4.148)$$

$$\pi_0 = 1 - \pi_1 - \pi_2 = \frac{(1 - P_{11})P_{20} + P_{10}P_{21}}{(1 - P_{10} - P_{11})(1 + P_{20}) + (1 + P_{10})(P_{20} + P_{21})} \quad (4.149)$$

donde S resulta ser:

$$S = (P_{20} + P_{21})e^{-aC} \\ \times \frac{1}{\left\{ \frac{(1 - P_{11})P_{20} + P_{10}P_{21}}{C} + (2a + b)(1 - P_{10} - P_{11}) + [(1 - e^{-aC})(2a + b + \frac{1}{C}) + e^{-aC}] (P_{20} + P_{21}) \right\}}$$

4.5.5 Evaluación de los retardos en un CSMA/CD.

En la sección 4.5 se definieron y analizaron tres tipos del protocolo CSMA/CD. El Ethernet usa el protocolo CSMA/CD 1-persistente, con el algoritmo *binary exponential backoff*, el cual decide cuando transmite los paquetes que estuvieron involucrados en una colisión. El algoritmo es muy delicado de analizarse; la mayoría

de los expertos en redes, ha llegado a la conclusión de que el retardo de la retransmisión después de la primera colisión es la única variable que afecta el desempeño de este protocolo.

Con el fin de hacer una generalización se construirá el modelo para un retardo de un CSMA/CD p -persistente. En el caso del Ethernet solo se aplicara con $p=1$.

En un CSMA/CD p -persistente, las estaciones con mensajes transmiten con probabilidad p en el periodo seguido a una colisión. Después de una transmisión exitosa, cada estación puede transmitir con probabilidad 1, pues el canal esta ocioso.

De acuerdo con la ecuación 4.135, la duración de un período de ociosidad del canal se distribuye geoméricamente y:

$$P_0 = e^{-\lambda T} \quad (4.151)$$

donde λ es el promedio de llegadas de los mensajes al canal.

Se define el período de contención como el período que comienza después de una transmisión exitosa cuando hay más de un mensaje en el sistema o después de un período de ociosidad del sistema cuando hubo una colisión. Este período termina cuando comienza una transmisión que es exitosa (cuando solo una estación transmite). Ya que la probabilidad de que un mensaje sea transmitido exitosamente es independiente del número de mensajes en el sistema, así la duración de un período de contención se distribuye geoméricamente y se tiene:

$$P_r[C_{i+1} = k | \text{Hubo una colisión}] = (1 - \theta)^{k-1} \theta \quad (4.152)$$

donde C_{i+1} , es la duración del $(i + 1)$ intervalo de contención y θ es la probabilidad de un mensaje exitoso.

Sea τ , el período que separa la transmisión de un mensaje a otro y sea I_{i+1} , el intervalo $i + 1$ de ociosidad del canal. Después de que un mensaje ha sido transmitido transcurre un período de tamaño θ segundos en el cual las estaciones pueden transmitir.

Sea a_{i+1} , el número de mensajes que llegan en el intervalo $I_{i+1} + C_{i+1}$, b_{i+1} el número de mensajes que llegan durante la transmisión del $(i + 1)$ ésimo mensaje y n_i el número de mensajes que estan esperando en el sistema cuando el i ésimo mensaje termina de ser servido. Entonces se tiene que:

$$n_{i+1} = n_i + a_{i+1} + b_{i+1} - 1 \quad (4.153)$$

Sea R_k , la probabilidad de k llegadas durante el período de contención. Si $n_i = 0$, se tiene entonces que:

$$P[a_{i+1} = 1 | n_i = 0] = P = P_1 \theta, \quad j = 0, 1, 2, \dots \quad (4.154)$$

donde $P_j t$ es la probabilidad de j llegadas en un período de tiempo.

y

$$P\{a_{i+1} = j + k | n_i = 0\} = P_j t R_k, \quad j > 1, k = 0, 1, \dots \quad (4.155)$$

En el caso de que $n_i = 1$, entonces no se presenta un período de ociosidad ni de contención y se tendría:

$$P\{a_{i+1} = 0 | n_i = 1\} = 1 \quad (4.156)$$

Para $n_i > 1$, no hay período de ociosidad ni de contención tampoco, así:

$$P\{a_{i+1} = k | n_i > 1\} R_k \quad (4.157)$$

Debido a que la duración de un período de contención se distribuye geoméricamente y las llegadas de las solicitudes como Poisson, se tiene que la función generadora del número de llegadas es¹:

$$R(z) = \vartheta [e^{T\lambda(1-z)} - (1 - \vartheta)]^{-1} \quad (4.158)$$

donde $T = 2\vartheta$.

La función generadora para b_{i+1} es (Ver [Ha]):

$$B(z) = M(\lambda(1-z)) e^{-T\lambda(1-z)} \quad (4.159)$$

donde $M(s)$ es la transformada de Laplace de la función de densidad del tiempo requerido para transmitir un mensaje.

La función generadora del número de mensajes en el sistema, se obtiene de la siguiente ecuación:

$$Q_{i+1}(z) = \frac{\left[q_0 \left(P_1 t z + R(z) \sum_{j=2}^{\infty} P_j t z^j \right) + z q_1 - R(z)(q_0 + q_1 z) \right] B(z)}{z - R(z)B(z)} \quad (4.160)$$

(Vease [Ge], pág. 230-246). Ahora, quienes son q_0 y q_1 , q_0 se define como la probabilidad de ningún mensaje en el sistema y q_1 se define como la probabilidad de un mensaje en el sistema.

Que un sistema este vacío quiere decir que:

¹Ver [Ha], pág. 229)

- El sistema se vacía cuando después de un período de tiempo llegó un mensaje el cual fue servido rápidamente.
- Había solo un mensaje en el sistema y ningún mensaje más llegó en ese lapso.

Considerando los últimos puntos se tiene que:

$$q_0(1 - e^{-\lambda T}) = q_0 \lambda T e^{-\lambda T} B(0) + q_1 B(0)(1 - e^{-\lambda T}) \quad (4.161)$$

o también

$$q_1 B(0) = q_0 \left[1 - \frac{\lambda T e^{-\lambda T}}{1 - e^{-\lambda T}} B(0) \right] \quad (4.162)$$

donde $B(0)$ es la probabilidad de cero llegadas durante el mensaje de transmisión.

Si se toma $z = 1$ en la ecuación 4.160 y se relaciona con la ecuación 4.161 se tiene:

$$q_0 = \frac{1 - \lambda(\frac{T}{d} + \bar{m} + \tau)}{\left[\frac{\lambda T}{1 - e^{-\lambda T}} \right] - \left[\frac{\lambda T}{d B(0)} \right]} \quad (4.163)$$

donde \bar{m} es el número de períodos de tiempo T requeridos para transmitir el mensaje. De las ecuaciones 4.160, 4.161 y 4.162 y aplicando el teorema de Little el retardo de un CSMA/CD es:

$$\begin{aligned} \bar{D} = & \bar{m} + \tau + T \left(\frac{1}{2} + \frac{1}{d} \right) \\ & - \left\{ \frac{1 - e^{-\lambda T}}{2[B(0)d - (1 - e^{-\lambda T})]} \right\} \left[\frac{2}{\lambda} + dT - 3T \right] \\ & + \frac{\lambda[\bar{m}^2 + 2\bar{m}\tau + \tau^2 + 2(\bar{m} + T)(\frac{T}{d}) + T^2(1 + \frac{2(1-d)}{d^2})]}{1 - \lambda(\bar{m} + \tau - \frac{T}{d})} \end{aligned} \quad (4.164)$$

4.5.6 Recuperación de Errores para un CSMA/CD

El objetivo de esta sección es distinguir los errores de este protocolo y las características para su recuperación sin mayores pérdidas, todo esto en condiciones normales, es decir, comunicando estaciones de trabajo.

El error más cotidiano son las colisiones, generadas por la intervención de más de una máquina para transmitir un mensaje en un período de tiempo fijo. Como ya se mencionó en la sección 4.5, cuando una colisión es detectada, la estación que detecta la presencia de la segunda aborta el mensaje que planeaba transmitir mandándole una señal a la segunda la cual al recibirlo aborta el mensaje que transmitía mandando al igual una señal de alarma la cual notifica la colisión presentada. Debido a que el tiempo que esperan para volver a enviar el mensaje es aleatorio la probabilidad de una segunda colisión disminuye pero no así termina con el problema.

De acuerdo al algoritmo *Retroceso Exponencial Binario Truncado*, se cuentan hasta 15 intentos más el original para poder transmitir exitosamente el mensaje. Si el mensaje no es transmitido en este lapso entonces se considera un error. Con base en lo anterior, se puede considerar que el aumento del número de intentos es directamente proporcional al tráfico en toda la red.

El tiempo promedio que una red CSMA/CD requiere para recuperarse de una colisión será:

- Para una red de tipo CSMA/CD no-persistente es:

$$\begin{aligned} T_{\text{rec,smard}} &= P(\text{Transmisión incompleta})E[2a + b + y_i] \\ &= (1 - G^{-aG})(2a + b + \frac{1}{G} - \frac{ae^{-aG}}{1 - e^{-aG}}) \end{aligned} \quad (4.165)$$

- Para una red de tipo CSMA/CD 1-persistente es:

$$\begin{aligned} T_{\text{rec,smard}} &= P(\text{Transmisión incompleta})E[2a + b + y_i] \\ &= (1 - G^{-aG})(2a + b + \frac{1}{G}) \end{aligned} \quad (4.166)$$

4.6 Modelo de FDDI

4.6.1 Notación para el modelo de una red FDDI.

El rendimiento de una red FDDI S , se define como el número total de datos en bits recibidos en el destino por segundo, expresados como fracciones de anchos de banda⁵.

⁵Un ancho de banda es el número de datos que se pueden transmitir por unidad de tiempo (bit por segundo).

La utilización de una red representada por U , se define como la fracción de tiempo que la red se ocupa en transmitir los paquetes de datos, esto incluye la carga adicional en bits, entonces $(1 - U)$ es la fracción de tiempo que el token consume en su transmisión.

La carga existente G esta definida como el número total de datos en bits generados por todas las máquinas por unidad de tiempo⁴.

El rendimiento S_i , la carga existente G_i y la utilización de la red U_i por clase i es dada por:

$$S_i = \alpha_i U_i \quad (4.167)$$

donde

$$\alpha_i = \frac{[T_m(i) - T_{th}]}{T_m(i)} \quad (4.168)$$

el total de utilización de la red y el rendimiento es:

$$S = \sum_i S_i \quad (4.169)$$

$$U = \sum_i U_i \quad (4.170)$$

Cuando la carga adicional es despreciable comparado con los campos de información, el rendimiento y la utilización son los mismos.

El ciclo del token C , esta definido como el tiempo transcurrido desde el instante que la máquina recibió el token hasta la próxima vez que vuelve a ocurrir el mismo evento.

Sea C_0 el tiempo de circulación del token en un "camino libre", esto es; el tiempo que tarda en recorrer el token, desde que llega a una máquina hasta que vuelve a la misma sin que ninguna de ellas transmita un mensaje (ausencia de tráfico en la red). Este valor C_0 se obtiene de la suma del recorrido de una máquina a otra por el cable (fddi) más el retardo de cada máquina al dejar libre el token.

Sea T_i el retardo distribuido de la red entonces:

$$T_i = \frac{C_0}{N} \quad (4.171)$$

Entonces la utilización de la red es:

$$U = \frac{(\bar{C} - C_0)}{\bar{C}} \quad (4.172)$$

⁴Expresadas también en fracciones de ancho de banda.

Se define el retardo de una cola como el arribo de un paquete a la cola hasta que comienza su transmisión. El retardo promedio de una clase esta evaluado por el valor promedio de retardo de la cola con paquetes que pertenecen a una clase particular en la red.

Para cargas pequeñas en este tipo de red, el rendimiento es igual a la carga existente, pero cuando la carga se incrementa, el rendimiento comienza a saturarse.

4.6.2 Redes con una sola clase de tipo asíncrono por estación

El modelo matemático para este tipo de redes fue propuesto y evaluado por *Dykeman* y *Bax*. El modelo consiste en lo siguiente:

Sea m el número de clases asincronas y $n(i)$ el número de máquinas con prioridad i , con $1 \leq i \leq 8$, donde la clase 1 tiene la más baja prioridad y la prioridad 8 es la más alta ($T_{Pri}(i+1) \geq T_{Pri}(i)$), se asume que todos los nodos de la red transmiten cargas pesadas de mensajes. Las prioridades de clases pueden no ser capaces de transmitir paquetes dado que la carga de prioridad es más alta.

Sea l la prioridad más baja con rendimiento no cero, y n el total de actividad de transmisión de las estaciones, se tiene:

$$n = \sum_i^m n(i) \quad (4.173)$$

entonces l es mínimo entero el cual satisface:

$$(n - n(i) + 1)T_{Pri}(i) - \sum_{i=l+1}^m n(i)T_{Pri}(i) - C_0 > 0 \quad (4.174)$$

y el rendimiento garantizado por cada clase i es:

$$t(i) = \frac{r(i)}{(n+1)C_0 + \sum_{j=l}^m r(j)} \quad (4.175)$$

4.6.3 Evaluación de los retardos

El tiempo esperado que un paquete de información tiene que esperar a ser transmitido esta definido como el intervalo de tiempo desde la llegada de este paquete al nodo (máquina) hasta que comienza su transmisión. Sea W este tiempo esperado.

Cuando se tienen varias clases de tráfico^{||}, el análisis es complejo pues trata con más de un comportamiento de transmisión de mensajes, en este caso se trata con una sola clase (una sola prioridad), de tipo asíncrono.

Las aproximaciones que ya fueron evaluadas para este caso se basan en la suposición de que una red FDDI puede considerarse de tipo **Token Ring, con un servidor exhausto (Ver [Be])**, dos de los algoritmos ya evaluados se muestran a continuación:

La máxima utilización de una red con k , ($k = 1 \dots N$), con estaciones transmitiendo y con prioridad $T_{Pri}=T$ esta denotada por $\tilde{p}(k)$ donde:

$$\tilde{p}(k) = \frac{k(T - C_0)}{kT + C_0} \quad (4.176)$$

Sean \bar{X} y $\overline{X^{(2)}}$ la media y el segundo momento respectivamente del tiempo de transmisión de un paquete, λ el valor esperado de la distribución de los arribos de mensajes a ser transmitidos y sea ρ la carga, donde $\rho = \lambda\bar{X}$, entonces el primer algoritmo define a W como:

$$\begin{aligned} W &= \frac{\rho\overline{X^{(2)}}}{2\bar{X}(1-\rho)} + \frac{(1-\frac{\rho}{N})C_0}{2(1-\rho)} \text{ para } \rho < \tilde{p}(1) \\ &= \frac{\rho\overline{X^{(2)}}}{2\bar{X}(1-\rho)} + \frac{(1-\frac{\rho}{N})C_0(k+1)}{2(1-\rho)} \text{ para } \tilde{p}(k) \leq \rho \leq \tilde{p}(k+1) \end{aligned} \quad (4.177)$$

4.7 Recuperación de Error

En esta sección se analiza y caracteriza el comportamiento de una red FDDI, en donde están conectadas estaciones de trabajo, las cuales reciben y transmiten mensajes en ella, el tipo de tráfico es asíncrono, con token no restringido, los errores incluyen los token perdidos y los CRC **.

Uno de los errores en este tipo de red de mayor importancia, ocurre cuando el anillo tiene una ruptura. Generalmente esta ruptura se da por problemas en alguna máquina integradora o de un cable físico; debido a la topología de FDDI el anillo se

^{||}Clases de transmisión de los mensajes

**Cyclic Redundancy Codes, este da el bit de detección de error en la capa de enlace de un modelo OSI

autoreconfigura inmediatamente recuperando la estabilidad, cuando estas rupturas no duran más de un segundo la red no se ve afectada.

Por ello esta simulación se basa en la presencia y ausencia de errores usando resultados reales.

Todos los nodos de esta red se comunican, las estaciones de trabajo accesan a sistemas de archivos pero también envían mensajes y archivos a cada una de las otras y a ruteadores, estos por estar conectados a otras redes generan tráfico.

Por ser el tráfico de tipo asíncrono, los tamaños de los paquetes son generados aleatoriamente. La frecuencia de los paquetes es controlada por un factor de carga asíncrona.

Los errores CRC son disminuidos en el nivel físico, cuando llegan a un nodo este los rechaza y los envía a un nivel más alto. La decisión de cómo y cuándo se retransmite el paquete es invisible para las capas de bajo nivel.

En condiciones estables, cuando el tiempo de rotación del token llega a su límite, la primera estación manda una alarma de este evento y comienza el proceso de negociación del token. La recuperación del token está determinada por el valor del tiempo de rotación del token.

4.7.1 Recuperación de las rupturas

Debido a la topología de una red FDDI, la recuperación de errores es muy corta; por esta razón el modelo de recuperación de las rupturas del anillo se basa en que el intervalo de falla se encuentra por debajo del segundo y medio.

Una vez que el anillo es restablecido, la red se recupera rápidamente y el proceso de token se reinicializa, así los nodos subastan el derecho para inicializar el token.

En general una red puede o no contener tráfico de tipo síncrono, el tiempo de recuperación de un tipo síncrono (T_{sinc}), es el promedio de tiempo que la red requiere para regresar a un estado estable después de que la red ha sido corregida. Sea (SB_{sinc}), el ancho de banda de un estado estable y se define como el porcentaje del ancho de banda total que se encuentra en dispositivos síncronos. Sea T_B la duración de la ruptura, AB el ancho de banda disponible.

Entonces el ancho de banda disponible es igual al máximo ancho de banda menos el resto de un ancho de banda asíncrono ($AB = 90\%$). El ancho de banda restante es el porcentaje de ancho de banda que el tráfico asíncrono está dispuesto

a durar mediante un proceso síncrono. Así se tiene:

$$T_{sinc} = \frac{TB}{(AB - SB_{sinc})} \quad (4.178)$$

El tiempo de recuperación de tipo asíncrono (T_{asinc}) es el promedio de tiempo que la red requiere para regresar a un estado estable despues de que se corrigió una ruptura. El modelo de recuperación es semejante a la recuperación de tipo síncrono más el ancho de banda de un estado estable asíncrono (SB_{asinc}):

$$T_{asinc} = T_{sinc} + \frac{TB}{(AB - SB_{asinc})} \quad (4.179)$$

4.8 Modelo de un Ruteador

El modelo analítico de un ruteador, se basa en el análisis de una cola de tipo M/G/1. El esquema básico de un ruteador se construye con buffers separados que son usados para cada dirección en donde se transmiten los mensajes; si los buffers estan vacíos el conector envía los paquetes de inmediato de acuerdo a su dirección IP, cuando el ruteador se retrasa (no encuentra la dirección IP o aún no decide por que camino enviarlo), los paquetes que van llegando se almacenan en estos buffers encolandose las peticiones (Ver figura 4.2).

La llegada de los paquetes enviados por las estaciones de trabajo se suponen con distribución Poisson con un promedio de llegadas λ por segundo, el proceso de servicio sigue el orden "El primero en llegar, es el primero en ser servido (First Come, First Served, FCFS)".

De acuerdo a lo ya estudiado en la sección de la cola M/G/1, se recuerda lo siguiente:

El promedio de número de mensajes en el ruteador $E[\tilde{N}]$ se basa en la fórmula (P-K) (Ver sección anterior) y es la siguiente:

$$\begin{aligned} N &= E[\tilde{N}] \\ &= \rho + \rho^2 \frac{1 + C_b^2}{2(1 - \rho)} \end{aligned} \quad (4.180)$$

El tiempo promedio de retardo para pasar un mensaje por un ruteador de acuerdo con las ecuaciones 4.180 y 4.69 se tiene:

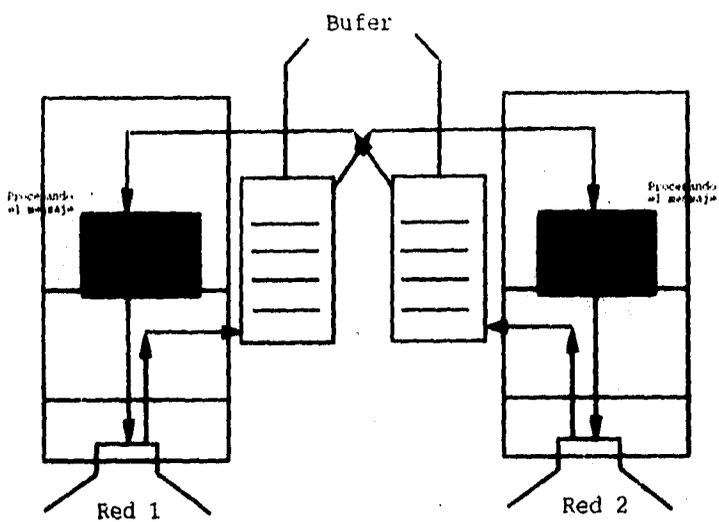


Figura 4.2: Modelo general de un ruteador

$$T = \frac{1}{2\mu(1-\rho)} \left[2 - \rho \left(1 - \frac{\sigma^2}{\mu^2} \right) \right] \quad (4.181)$$

Y el rendimiento total de un ruteador es:

$$\begin{aligned} S &= \frac{\lambda}{T} \\ &= \frac{2\mu\lambda(1-\rho)}{2 - \rho \left(1 - \frac{\sigma^2}{\mu^2} \right)} \end{aligned} \quad (4.182)$$

4.8.1 Evaluación de los Retardos y Errores

El tiempo total de retardos en un ruteador fue dado por la ecuación 4.181, de acuerdo al modelo de la cola M/G/1.

La característica principal de un ruteador es poder seleccionar una o más rutas para enviar un mensaje. Esta decisión puede estar basada en parámetros de retardo de tránsito, congestión y otros ruteadores disponibles, o el número de ruteadores entre las estaciones fuente y destino.

Ya que puede mantener esta característica, un ruteador está diseñado para mantener fallas de enlaces, estaciones y otras redes. Sin embargo el introduce retardo pues puede elegir una ruta larga, creando con esto retardo en la transmisión del mensaje. Esto se corrige en la implementación de la estructura de la red, configurando de tal manera que este retardo se corrija al máximo.

Un ruteador tiene también la propiedad de desechar los paquetes que sean enviados con errores o duplicados, de esta forma reduce la probabilidad de encontrar errores en las redes, así como desecha aquellos paquetes cuyo tiempo de vida expire. Los errores de origen del ruteador generalmente son de configuración o de hardware por lo que solo es posible mediante la probabilidad predecir sus retardos.

4.9 Modelo de un Sistema Híbrido de Cómputo.

En las secciones pasadas se crearon modelos de cada uno de los factores que intervienen en la función del sistema de cómputo, en esta sección se agrupan estos

modelos creando el modelo principal para un sistema híbrido de cómputo enfocado a dos servicios de red ya descritos NIS y NFS, prediciendo su rendimiento y tiempo promedio de retardo.

Como ya se menciono anteriormente, la estructura del sistema de cómputo en estudio consta de cuatro redes de tipo Ethernet y una de tipo FDDI, conectadas mediante cuatro **ruteadores** ¹¹.

En el caso del sistema híbrido de cómputo en estudio, el servidor de NFS y NIS+, se comunica al exterior mediante dos redes de tipo ethernet, debido a que el comportamiento de transmisión varía considerablemente, se realizará un estudio separado para las máquinas cliente que pertenezcan a estas redes y uno más para las estaciones clientes que se comuniquen al servidor mediante conectores de red.

4.9.1 Notación

Para efectos de simplificación se definen las siguientes variables:

- T_{NFS} Representa el tiempo promedio que una petición de tipo NFS, tarda en procesarse desde su llegada al servidor, hasta ser servida y entregada a la cola que modela la red.
- R_{NFS} Se define como el rendimiento promedio que el servicio NFS genera, en la práctica este es medido en mensajes por segundo.
- T_{NIS} Es el tiempo promedio de retardo, que una petición de tipo NIS tarda en ejecutarse desde que llega al servidor hasta que es procesada.
- R_{NIS} Representa el rendimiento promedio que una petición de NIS, genera por segundo.
- T_{esmacl} Esta variable se define, como el tiempo promedio de retardo que un mensaje tarda en ser transmitido por una red de tipo Ethernet (CSMA/CD)
- R_{esmacl} Se define, como el rendimiento promedio de transmisión de mensajes en Ethernet por segundo.
- T_{FDDI} Esta definida, como el tiempo promedio de retardo que un mensaje tarda en ser transmitido por una red de tipo FDDI.

¹¹Dos de ellos conectados directamente en la red fddi y los restantes conectados a dos redes ethernet.

- R_{FDDI} Representa el rendimiento promedio de transmisión de mensajes sobre FDDI por segundo.
- T_{rut} Se define, como el tiempo promedio de retardo que un mensaje tarda en ser enviado de la red origen a la ruta correcta.
- R_{rut} Esta variable representa, el rendimiento promedio de transmisión de mensajes a través de un ruteador.
- $T_{servidor}$ Representa el tiempo total de retardo en un servidor con respecto al tiempo de proceso de las solicitudes de los servicios.
- $R_{servidor}$ Esta variable se define, como el rendimiento promedio del servidor para ejecutar las peticiones de los servicios de red.
- T_{tot} Representa el tiempo promedio de retardo total, considerando todos los factores que intervienen en la petición de un servicio de red.
- R_{tot} Se define como el rendimiento promedio total, considerando las variables de rendimiento anteriores que intervienen en la petición de un servicio de red.

4.9.2 Análisis del tiempo de retardo del servidor y clientes conectadas en una misma red

Como en este estudio, las dos redes a las que se conecta el servidor son del mismo tipo *ethernet*, se construirá un solo modelo que se aplica a cualquiera de las dos.

Cuando el servidor atiende peticiones en una red local, el rendimiento y el porcentaje de fallas del servicio de red, se reduce al estudio de la red local más el comportamiento del servidor.

Sea $E\gamma_s$ la red que comunica al servidor con los clientes, y sean $c_{\gamma_1} \dots c_{\gamma_k}$, las máquinas cliente conectadas en la red $E\gamma_s$.

El tiempo promedio de retardo para que una petición de un cliente se procese en el servidor es:

$$T_{tot} = \text{Tiempo promedio de retardo de la red} + \text{Tiempo promedio de retardo del servidor} \quad (4.183)$$

En la sección 4.5 se definieron y analizaron tres tipos del protocolo *CSMA/CD*. El Ethernet usa el protocolo *CSMA/CD 1-persistente*.

El tiempo promedio de retardo de un CSMA/CD 1-persistente esta dado por el tiempo ocupado de esta misma que fue denotado en la sección 4.5 por la ecuación 4.164

$$\begin{aligned} \bar{D} = & \bar{m} + \tau + T\left(\frac{1}{2} + \frac{1}{\beta}\right) \\ & - \left\{ \frac{1 - e^{-\lambda\tau}}{2[B(0)\beta - (1 - e^{-\lambda\tau})]} \right\} \left[\frac{2}{\lambda} + \beta T - 3T \right] \\ & + \frac{\lambda[\bar{m}^2 + 2\bar{m}\tau + \tau^2 + 2(\bar{m} + T)\left(\frac{\tau}{\beta}\right) + T^2\left(1 + \frac{2(1-\beta)}{\beta^2}\right)]}{1 - \lambda(\bar{m} + \tau - \frac{\tau}{\beta})} \quad (4.184) \end{aligned}$$

El tiempo promedio de retardo del servidor varía de acuerdo al tipo de solicitud. Para una solicitud de tipo NFS el retardo del sistema se consideran los siguientes resultados obtenidos en las secciones anteriores:

$$T_{NFS} = \frac{\left(\frac{\rho^k(\beta)}{k!(1-\beta)^2 I_0} \right)}{\lambda} + \frac{1}{\mu} \quad (4.185)$$

El modelo que se busca para el tiempo total de retardo en el sistema considerando que se requiere el tiempo de retardo de ejecución de las solicitudes del servicio NFS es: re

$$T_{serv} = T_{NFS} \quad (4.186)$$

Así el tiempo de retardo promedio para una petición de NFS cuando el cliente y el servidor se comunican por la misma red de tipo ethernet es:

$$T_{tot} = T_{serv} + T_{csma/cd} \quad (4.187)$$

Para una petición de NIS, los factores que intervienen en el retardo del procesamiento de la petición son:

$$T_{NIS} = \frac{1}{\mu}(1 + N) \quad (4.188)$$

el retardo de $T_{csma/cd}$ esta definido en la ecuación 4.184.

De esta manera, el modelo para encontrar el tiempo de retardo promedio para una petición de NIS, será:

$$T_{tot} = T_{NIS} + T_{csma/cd} \quad (4.189)$$

4.9.3 Análisis de los retardos de sistema de cómputo híbrido en diferentes redes

Si el servidor se comunica con sus clientes a través de medios secundarios (ruteador, bridge, etc); el análisis se complica más, pues se debe tomar en cuenta el medio o los medios auxiliares entre las redes ya que el tiempo que ocupan en transmitir los mensajes, afecta en el tiempo de retardo total.

Para este estudio son tres, las redes que a través de ruteadores comunican con las redes del servidor de NIS y NFS. Una de estas es de tipo FDDI que se comunica con un sólo ruteador, a las redes locales del servidor. Las dos restantes, son de tipo Ethernet las cuales se comunican mediante dos ruteadores a las redes del servidor.

En general, el tiempo promedio de retardo para procesar un servicio de red, donde las máquinas clientes se comunican con el servidor mediante más de dos redes, de igual o diferente tipo será:

- Para el caso de una petición de NFS

$$T_{tot} = T_{NFS} + T_{red_1} + \dots + T_{red_n} + T_{rut_1} + \dots + T_{rut_k} \quad (4.190)$$

- Cuando las solicitudes son de NIS:

$$T_{tot} = T_{NIS} + T_{red_1} + \dots + T_{red_n} + T_{rut_1} + \dots + T_{rut_k} \quad (4.191)$$

donde n y k , representan el número de las redes y ruteadores, por las que los mensajes son transmitidos para llegar a la red local del servidor.

Si alguna red es de tipo FDDI, el tiempo promedio de retardo será de acuerdo a la ecuación 4.171 el siguiente:

$$T_i = \frac{C_0}{N} \quad (4.192)$$

donde C_0 es el tiempo de circulación del token en un "camino libre" (Vease sección 4.6).

4.9.4 Análisis del rendimiento del sistema híbrido de cómputo

El análisis de un sistema de cómputo, está fijado por el mínimo rendimiento de los factores que afectan en este. Por ejemplo en el estudio en cuestión, el rendimiento

depende de que tan eficiente sea el desempeño de la red, del sistema, etc. Por ejemplo, si el funcionamiento de la red es óptimo, pero el rendimiento de un ruteador es bajo, los mensajes llegarán al ruteador de una manera rápida pero serán servidos de acuerdo a la velocidad fijada por el desempeño del ruteador, con lo que el rendimiento ganado por la red se pierde, ajustándose al rendimiento del ruteador.

De acuerdo a este razonamiento el rendimiento del proceso exitoso de una petición esta definido por:

$$Rend_{tot} = Min(Rend_{factor_1}, \dots, Rend_{factor_n}) \quad (4.193)$$

donde $factor_1, \dots, factor_n$ son los factores que influyen en el rendimiento de una petición.

Análisis del rendimiento de un sistema de cómputo en una red local

Cuando los clientes se comunican con el servidor mediante una sola red, el rendimiento se ajusta a:

$$Rend_{tot} = Min(Rend_{sistema}, Rend_{red}) \quad (4.194)$$

De acuerdo con lo ya analizado se tienen los siguientes resultados: El rendimiento de una red Ethernet estará dado por:

$$Rend_{esmaed} = \frac{G e^{-aG}}{G e^{-aG} + bG(1 - e^{-aG}) + 2aG(1 - e^{-aG}) + (2 - e^{-aG})} \quad (4.195)$$

donde G representa la carga presentada en el canal (Véase sección 4.5).

$$Rend_{NFS} = S = \frac{\lambda}{T} \quad (4.196)$$

donde T esta dado por la ecuación 4.115.

Como el tipo de red local del servidor es Ethernet el rendimiento de un sistema de cómputo con servicio NFS es:

$$Rend_{tot} = Min(Rend_{esmaed}, Rend_{NFS}) \quad (4.197)$$

Para el servicio NIS:

$$Rend_{tot} = \text{Min}(Rend_{esmaod}, Rend_{NIS}) \quad (4.198)$$

donde el rendimiento de NIS es:

$$Rend_{NIS} = \frac{\lambda}{T} \quad (4.199)$$

donde T se representa por la ecuación 4.188.

Análisis del rendimiento de un sistema híbrido conectados en más de una red.

Cuando más de una red interviene en la comunicación de los clientes con el servidor, el rendimiento es menor pues los factores aumentan, por ejemplo en el sistema en estudio, intervienen los ruteadores y más redes ethernet y fddi, debido a esto la eficiencia de cada uno de estos influye en gran medida. Para el mejor de los casos se tendrá que el rendimiento de cada uno de estos es el máximo rendimiento, en este caso el rendimiento total dependerá del mínimo de los mejores rendimientos. El problema se presenta en el peor de los casos, pues la eficiencia de los servicios baja considerablemente. Para una petición de tipo NFS, que pasa sobre más de una red y más de un ruteador es:

$$Rend_{tot} = \text{Min}(Rend_{red_1}, \dots, Rend_{red_n}, Rend_{rut_1}, \dots, Rend_{rut_k}, Rend_{NFS}) \quad (4.200)$$

donde n y k es el número de redes y ruteadores por los que el mensaje es transmitido para encontrar la ruta del servidor fijada.

En el caso del servicio de red NIS:

$$Rend_{tot} = \text{Min}(Rend_{rut_1}, \dots, Rend_{red_n}, Rend_{rut_1}, \dots, Rend_{rut_k}, Rend_{NIS}) \quad (4.201)$$

Parte V

**APLICACION DE LOS MODELOS
PROBABILÍSTICOS EN UN
SISTEMA DE CÓMPUTO REAL**

Capítulo 5

Aplicación de los modelos estocásticos en un problema real

En este capítulo, se aplican los modelos propuestos en el sistema real descrito en la cuarta parte de esta tesis. Los modelos probabilísticos se aplicaron a través de programas, que fueron construidos para generar los tiempos de retardo y desempeño a partir de las ecuaciones obtenidas en ellos. El objetivo principal, es mostrar el tiempo de retardo para transmitir peticiones de NFS, en el sistema de cómputo híbrido de la DGSCA.

En la figura 5.1, se recuerda la estructura actual del sistema híbrido de cómputo en estudio. El servidor está conectado en dos redes denominadas 159 y 161, ambas de tipo ethernet, que a su vez conectan aproximadamente el 51% de las máquinas clientes de NFS y NIS. Tres de las restantes redes (204 y 170 de tipo ethernet y 160 de tipo fddi), conectan el 49% de las máquinas clientes de NFS y NIS.

Con base en el modelo del sistema híbrido de cómputo, el análisis del tiempo de retardo se divide en dos partes:

- Máquinas cliente conectadas en cualquiera de las dos redes del servidor.
El análisis de tiempo de retardo en el servicio NFS, basado en clientes conectados en la misma red que el servidor, toma en cuenta solo dos factores que son:
 - Servidor de NFS
 - Red local (que conecta al cliente y servidor)
- Máquinas cliente no conectadas en ninguna de las dos redes del servidor.

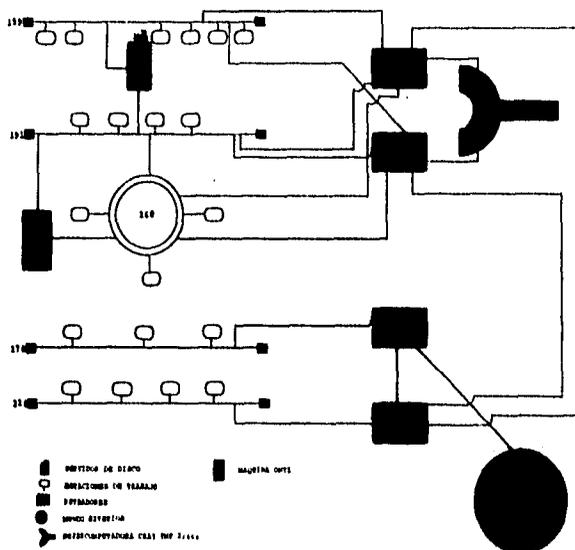


Figura 5.1: Esquema del sistema híbrido de cómputo en estudio.

En cuanto al análisis de máquinas cliente conectadas en distintas redes a las del servidor, los factores que la afectan son:

- Servidor de NFS
- Red local del cliente
- Un número n de ruteadores
- Red local del servidor

Se generaron datos de NFS, cada hora a partir del mes de enero a finales de junio. La información que proporcionaron dichos datos se resume en los siguientes puntos:

- Número de llamadas exitosas de rpc del cliente al servidor (*calls rpc*).
- Número de llamadas erróneas de rpc del cliente al servidor (*badcalls rpc*).
- Número de retransmisiones de solicitudes nfs (*retrans*).
- Número de identificadores no registrados del cliente al servidor* (*badxids*).
- Número de solicitudes cuya transmisión excedió el tiempo límite fijado (*timeouts*).
- Número de llamadas exitosas de nfs del cliente al servidor (*calls nfs*).
- Número de llamadas erróneas de nfs del cliente al servidor (*badcalls nfs*).

A partir de estos datos se calcularon las variables λ y μ de cada mes, λ se calculó como el número de llamadas de rpc por minuto y μ se calculó a través del número de llamadas de nfs por minuto.

Debido a la falta de información de la red, se supusieron los valores de las variables λ y μ para la red y los ruteadores. Sin embargo para un sistema que cuente con datos reales suficientes para la evaluación de los parámetros basta sustituirlos en los datos de entrada del programa de monitoreo construidos que se basan en los modelos probabilísticos y son descritos en el apéndice A.

En los modelos probabilísticos de la sección 4.9, se construyeron fórmulas para estimar el retardo de las dos redes ethernet y fddi respectivamente, así como del

*Los identificadores son etiquetas que el cliente coloca en cada una de las solicitudes enviadas al servidor. Los identificadores no registrados son aquellos que el cliente no reconoce en su lista de solicitudes enviadas.

servicio de red NFS. Con base en ellas se crearon programas mediante los cuales se calcularon los retardos de los factores que intervienen en el servicio NFS.

Un criterio que va muy ligado al análisis del retardo y desempeño del servicio NFS, es el porcentaje de retransmisiones de solicitudes de este servicio, ya que si este porcentaje rebasa el 5%, es una muestra que el rendimiento de la red o del servidor es pobre. Debido a que el desempeño de NFS es inversamente proporcional al tiempo de retardo, si el desempeño disminuye, el retardo aumenta.

Los criterios utilizados para definir el origen de las retransmisiones se enuncian a continuación:

- Desempeño pobre del servidor.

Se considera que el desempeño de un servidor de NFS es pobre, si procesa las peticiones de los clientes en un tiempo mayor al tiempo límite fijado, de tal forma que ocasiona una retransmisión por parte del cliente. Cada que el cliente transmite una petición de NFS, etiqueta esta con un identificador, si se retransmite esta petición el cliente no cambia el identificador. De esta manera, si el servidor procesa dos veces la misma petición provoca que el cliente reciba dos veces el mismo proceso y aumente el contador de identificadores no registrados por el cliente. Si el porcentaje de retransmisión supera el 5% y la diferencia entre el número promedio de transmisiones que rebasaron los tiempos límite y el número de identificadores no registrados es relativamente pequeña, se sabe que el problema radica en el servidor.

- Desempeño pobre de la red.

El desempeño de una red es pobre, si el tiempo para transmitir una petición excede un tiempo límite fijado, lo cual genera que el cliente retransmita la petición. Si el número promedio de tiempos límite es mayor al número de identificadores no registrados por el cliente, indica que el servidor no ha procesado las peticiones transmitidas, por lo que los paquetes no se transmitieron exitosamente.

Si el porcentaje de retransmisiones supera el 5% y el número de identificadores no registrados por el cliente es pequeño con respecto al número de transmisiones que excedieron el tiempo límite de transmisión, el problema se le atribuye al desempeño de la red.

Los puntos anteriores proporcionaron los criterios para separar las fallas entre el servidor y la red. De acuerdo a estos criterios se calculó el número de veces que la red y el servidor excedieron el porcentaje de retransmisiones del 5%.

Con el fin de aclarar aún más los resultados, se hicieron clasificaciones entre el número de veces que los porcentajes de retransmisiones fueron excedidos entre 5% - 10%, 10% - 15% y 15% ... respectivamente, por el servidor o la red.

Los resultados mostraron, que la diferencia entre el número de transmisiones que excedieron los tiempos límite para su transmisión y el número de identificadores no registrados del cliente al servidor, fue relativamente grande en todo el período de tiempo monitoreado. Debido a lo anterior se obtuvo que el servidor de NFS procesó a tiempo las peticiones de dicho servicio de red¹, y el número de veces que el porcentaje de retransmisiones excedió el 5% por fallas en el servidor fue nulo.

Ya que el número de identificadores no registrados por el cliente fue relativamente pequeño en comparación con el número total de transmisiones que excedieron los tiempos límite, el problema de falla se le atribuyó a la red. De esta forma se comenzó un análisis en el retardo de transmisión de una petición desde una máquina cliente al servidor de NFS y con base en la ecuación 4.164 de la sección 4.9, se obtuvo el retardo estimado de la red.

El análisis de resultados se clasificó en redes locales y redes externas al servidor, los cuales se describen a continuación.

5.1 Análisis de resultados de NFS en clientes comunicadas en una de las redes 159 y 161

El monitoreo en la red 159, se ejecutó en dos máquinas clientes de NFS, la primera de ellas llamada *pulque*, servidor de impresión y la segunda, llamada *merak* la cual además de ser cliente de NFS es cliente de NIS del mismo servidor.

Estas máquinas mostraron un alto número de veces que el porcentaje de retransmisiones, rebasó el 5% por problemas de rendimiento en la red¹. Los resultados del número de porcentajes entre el 5% - 10%, 10% - 15% y 15%... se muestran en las gráficas 5.2, 5.3 y 5.4.

De acuerdo con la ecuación 4.164 de la sección 4.5, se calculó el promedio estimado de retardo que la red 159 tardó en transmitir una solicitud de NFS desde la máquina *merak* al servidor. De igual forma se calculó el promedio estimado de

¹Cuando un servidor procesa a tiempo las solicitudes de NFS, muestra que la lectura y/o escritura a disco es rápida, así como corrobora que la capacidad por el procesador de ejecutar las peticiones es estable

²Se considera un problema de red o de servidor si el porcentaje de fallas, rebasa el 5%.

5.1 Análisis de resultados de NFS en clientes comunicadas en una de las redes 159 y 161 109

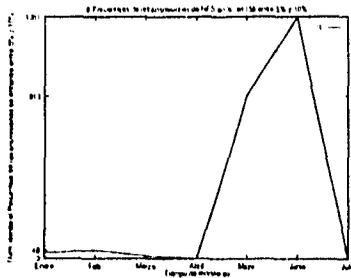


Figura 5.2: Número de veces que el porcentaje de retransmisiones se mantuvo entre el 5% y 10%, en la red 159.

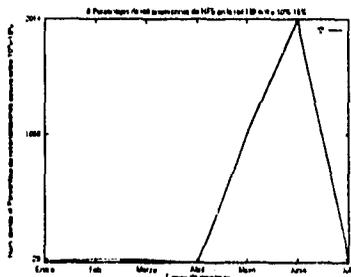


Figura 5.3: Número de veces que el porcentaje de retransmisiones se mantuvo entre el 10% y 15%, en la red 159.

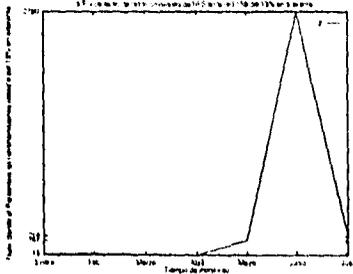


Figura 5.4: Número de veces que el porcentaje de retransmisiones se mantuvo entre el 15% en adelante, en la red 159.

Red	Retardo
159	43.387418

Tabla 5.1: Tiempo de retardo total de la red 159

retardo que la red 159 tardó en transmitir una petición de NFS desde la máquina *pulque* al servidor. En las figuras 5.5, 5.6 se muestran los resultados.

Nótese que de acuerdo al modelo, el retardo de la red 159 aumentó en los períodos donde se presentaron mayores cantidades de porcentajes de retransmisiones que rebasaron el 5%. Esto muestra que el modelo propuesto en la sección 4.5, para el tiempo de retardo de una red *Ethernet* (*CSMACD*) concuerda con el comportamiento real del sistema.

En la tabla 5.1, se resume el tiempo promedio estimado total de retardo el cual se obtuvo de la ecuación 4.164 de la sección 4.5, aplicado a la red 159.

En cuanto a la red 161, se analizó con base en el monitoreo de dos máquinas. La primera máquina es cliente de NFS llamada *deneb* y la segunda llamada *diphda*, es el servidor de NFS y NIS, además funciona como cliente de NIS cuyo servidor es ella misma.

El índice de porcentaje de retransmisiones de peticiones de NFS, para la red 161, fue relativamente pequeño comparado con la red 159. Estos resultados se muestran en las gráficas 5.7, 5.8 y 5.9.

Ya que esta red es de tipo *ethernet*, se tomó la ecuación 4.164 de la sección 4.5, calculándose el tiempo estimado que tarda en transmitir la red 161, peticiones

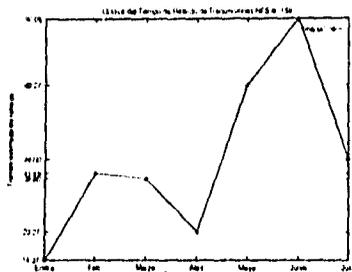


Figura 5.5: Comportamiento del tiempo promedio de retardo de transmisión de una petición NFS desde "merak" al servidor, en la red 159.

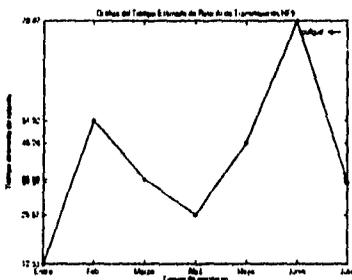


Figura 5.6: Comportamiento del tiempo promedio de retardo de transmisión de una petición NFS desde "pulque" al servidor, en la red 159.

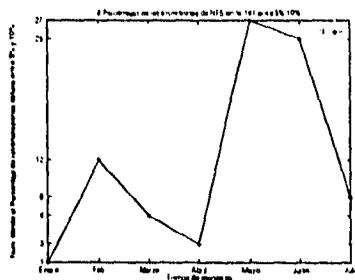


Figura 5.7: Número de veces que el porcentaje de retransmisiones se mantuvo entre el 5% y 10%, en la red 161.

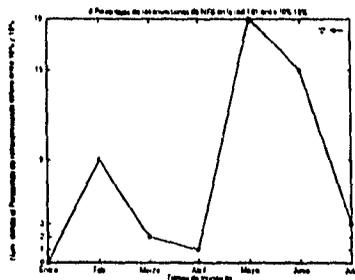


Figura 5.8: Número de veces que el porcentaje de retransmisiones se mantuvo entre el 10% y 15%, en la red 161.

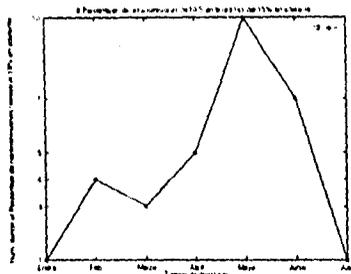


Figura 5.9: Número de veces que el porcentaje de retransmisiones se mantuvo del 15% en adelante, en la red 161.

Red	Retardo
161	29.888842

Tabla 5.2: Retardo estimado para la red 161

de NFS al servidor. En las gráficas 5.10 y 5.11 se muestran los tiempos de retardos estimados para transmitir una solicitud de la máquina *deneb* al servidor y de la máquina *diphda* a ella misma pero transmitiéndose por red.

Nótese que al igual que en el análisis anterior, los períodos que presentan mayor índice de retardo según el modelo concuerdan con los períodos reales donde los porcentajes de retransmisiones atribuidos a la red fueron altos. El tiempo total estimado de retardo obtenido de aplicar la ecuación 4.164 de la sección 4.5, en la red 161, se resume en la tabla 5.2:

5.2 Análisis de resultados de NFS en clientes comunicados en distintas redes

De acuerdo a la ecuación 4.190 de la sección 4.9, el tiempo total de retardo de transmisión de una petición NFS de una máquina cliente conectada en una red externa al servidor es igual a la suma de sus factores. Como ya se vió en la introducción de esta sección el servidor NFS, mostró un número de fallas nulo, es por esto que

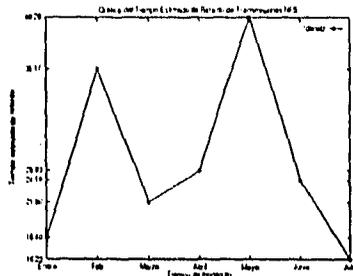


Figura 5.10: Comportamiento del tiempo promedio de retardo de transmisión de una solicitud NFS desde deneb al servidor, en la red 161.

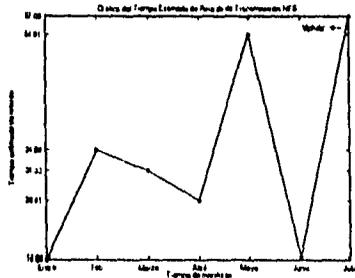


Figura 5.11: Comportamiento del tiempo promedio de retardo de transmisión de una solicitud NFS desde diphda a ella misma por la red 161.

Ruteador	μ
1	0.8
2	0.85
3	0.56
4	0.4

Tabla 5.3: Valores de μ propuestos para los ruteadores

el retardo estimado se reduce a la suma de los factores que intervienen en la transmisión de red. Estos factores son:

- Red local del cliente
- Ruteador o ruteadores
- Red local del servidor

Los promedios de tiempos de servicio de cada ruteador son fijados y propuestos en la tabla 5.3.

5.3 Análisis de resultados de NFS en clientes comunicadas en la red fddi 160

Son cuatro máquinas conectadas en la red 160, pero solo dos de ellas pertenecen al grupo de clientes de NFS. Con base en la figura 5.1, las peticiones de NFS, desde un cliente conectado a la red 160 al servidor, viajan por esta red, hasta llegar a cualquiera de los dos ruteadores 1 o 2, donde la probabilidad de pasar por cualquiera de los dos es la misma, y por alguno de ellos el mensaje es transmitido. Al llegar el mensaje al ruteador, este decide el camino óptimo que seguirá el mensaje transmitiéndolo con probabilidad uno.

Las dos máquinas las cuales fueron sometidas al monitoreo son clientes de NFS, llamadas *polaris* y *mira*.

De acuerdo con los datos generados por cada mes, se obtuvo que los porcentajes de retransmisiones fueron nulos. Esto dió como resultado que la red y el servidor permanecieron estables.

Por lo anterior, para la red 160, el análisis se centra en estimar el tiempo de retardo de transmisión de una petición de NFS, desde cualquiera de las dos clientes

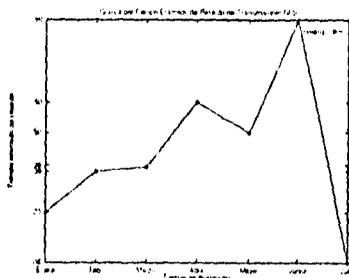


Figura 5.12: Comportamiento del tiempo promedio de retardo de transmisión de una petición de NFS desde polaris al servidor, en la red fddi 160.

Ruteador	Tiem_Retardo
1	1.186337
2	1.031872

Tabla 5.4: Tiempos de retardo estimados para la red 160

del servidor. Con base en la ecuación 4.177 de la sección 4.6, se calculó el retardo promedio de la red fddi para transmitir un mensaje de la máquina *polaris* o *mira* al servidor en la red local 159, el cual es representado a través de las gráficas 5.12 y 5.13.

Para obtener el tiempo de retardo total, además de obtener el retardo de la red 160, se calculó con base en la ecuación 4.181 de la sección 4.8 el retardo estimado que tarda el ruteador en transmitir los mensajes de una red a otra. En la tabla 5.4, se resumen los valores de retardo obtenidos por cada ruteador.

Nótese que como el ruteador decide a donde dirigirse, el tiempo de retardo mínimo estimado será cuando pase por un solo ruteador:

$$T_{ret} = T_{red-local-cliente} + T_{red-local-servidor} + T_{rut-unico} \quad (5.1)$$

En la tabla 5.5, se resume el tiempo de retardo mínimo estimado de transmisión de una solicitud NFS, generado desde una máquina conectada en la red 160 al servidor, pasando por un solo ruteador. Este retardo se obtuvo de la suma del cálculo de las ecuaciones 4.177 (sección 4.6), 4.181 (sección 4.8) y 4.164 (sección 4.5).

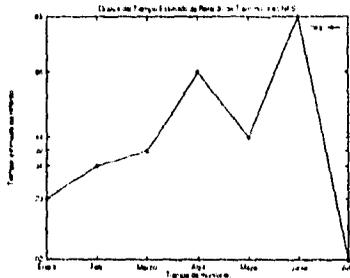


Figura 5.13: Comportamiento del tiempo promedio de retardo de transmisión de una petición de NFS desde mira al servidor de NFS, en la red fddi 160.

Red	Ruteador	Tiem_Retardo
159	1	45.006847
159	2	44.852182
161	1	31.508271
161	2	31.353606

Tabla 5.5: Tiempos de retardo total estimados para la red 160

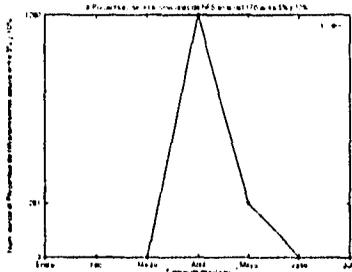


Figura 5.14: Número de veces que el porcentaje de retransmisiones se mantuvo entre el 5% y 10%, en la red 170.

5.4 Análisis de resultados de NFS en clientes comunicadas en las redes 170 y 204.

El análisis para las redes ethernet 170 y 204, es semejante al análisis para la red 160, la diferencia es que para cualquiera de estas dos redes la transmisión de una solicitud NFS pasa por lo menos por dos ruteadores.

La red 170⁵, transmite los mensajes no locales por el ruteador 3, este envía el mensaje con la misma probabilidad a los ruteadores 1 o 2, de ahí el ruteador 1 o 2 deciden por que ruta enviar el mensaje. Para la red 204, se utiliza el ruteador 4 para envíos de mensajes no internos, el cual de igual manera que el ruteador 3 envía el mensaje a cualquiera de los ruteadores 1 y 2. De acuerdo con la ecuación 4.190 (sección 4.9), el tiempo mínimo estimado de retardo estará dado por la suma del retardo de dos ruteadores más la suma del retardo de la red origen y la red destino respectivamente.

En la red 170 se monitorearon dos máquinas, una de ellas llamada *aldebaran* cliente de NFS, la segunda llamada *casiopea* integrante del grupo de los clientes de NFS y NIS.

Los índices de porcentaje de retransmisiones por el bajo ancho de banda de la red, que superaron el 5% de fallas en la red 170 se muestran en las gráficas 5.14, 5.15 y 5.16.

De igual forma que en los análisis anteriores, se calculó el tiempo estimado

⁵(Vease figura 5.1)

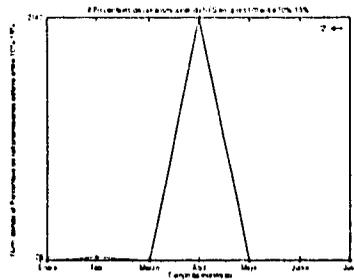


Figura 5.15: Número de veces que el porcentaje de retransmisiones se mantuvo entre el 10% y 15%, en la red 170.

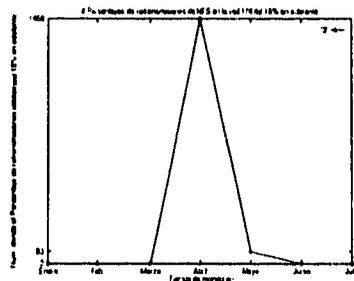


Figura 5.16: Número de veces que el porcentaje de retransmisiones se mantuvo del 15% en adelante, en la red 170.

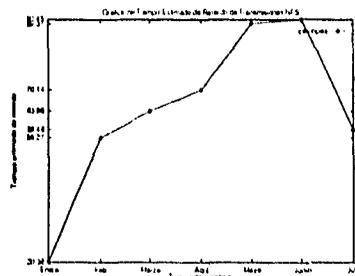


Figura 5.17: Comportamiento del tiempo estimado de retardo para una petición de NFS de casiopea al servidor.

Ruteador	Tiem_Retardo
1	1.254184
2	1.180173
3	1.794310

Tabla 5.6: Tiempos estimados de retardo para cada ruteador que conectan externamente la red 170

de retardo de retransmisión para una petición de NFS, de las máquinas *aldebaran* y *casiopea* obteniéndose los resultados que son mostrados en las gráficas 5.17 y 5.18.

Con base en la ecuación 4.181 de la sección 4.8, se calculó para cada ruteador el retardo estimado que el ruteador se demora en transmitir los mensajes de una red a otra. La diferencia del retardo de los ruteadores entre la red fddi y la red ethernet descrita ahora no varía en mucho, el retardo calculado de acuerdo al modelo. Con los parámetros propuestos para cada uno de los ruteadores se resume en la tabla 5.6:

Los tiempos de retardo estimados se calcularon a partir de las ecuaciones 4.164 (sección 4.5) y 4.181 (sección 4.8). Esta red tiene que transmitir sus mensajes a través de dos ruteadores que son el ruteador 3 y cualquiera de los ruteadores 1 y 2. El tiempo mínimo estimado de retardo de acuerdo con la ecuación 4.190 de la sección 4.9, es la suma de los retardos estimados para cada uno de los ruteadores más la suma del retardo estimado de la red 170, más la suma del retardo estimado

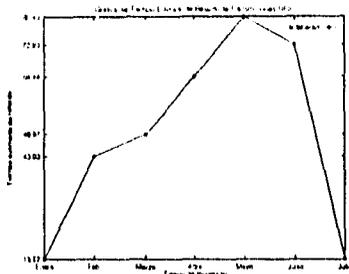


Figura 5.18: Comportamiento del tiempo estimado de retardo para una petición de NFS de aldebaran al servidor.

Red	Ruteador	Tiem.Retardo
159	1	102.862273
159	2	102.788262
161	1	89.363697
161	2	89.289686

Tabla 5.7: Retardos estimados de transmisión de peticiones NFS de la red 170 a las redes locales del servidor

de la red destino (159 ó 161).

Los cálculos obtenidos se mencionan en la tabla 5.7.

Nótese que el retardo de red estimado aumentó considerablemente con respecto a los análisis anteriores ya que transmite mediante dos redes ethernet. Además el período de tiempo con un alto índice de porcentaje de retransmisiones coincide con el tiempo estimado de retardo dado por los modelos probabilísticos.

La red 204, comunica a una sola máquina cliente de NFS llamada *mezcal*, el número de veces que se registraron porcentajes de retransmisiones que superan al 5% están representadas por las gráficas 5.19, 5.20 y 5.21.

El retardo estimado producto del cálculo de las ecuaciones 4.164 (sección 4.5) y 4.181 (sección 4.8) son representadas por la gráfica 5.22.

Los retardos de los ruteadores fueron calculados al igual que los análisis anteriores por el modelo de un ruteador descrito en la sección 4.8. La tabla 5.8, muestra este valor para cada uno de los ruteadores que transmiten un mensaje de NFS, al

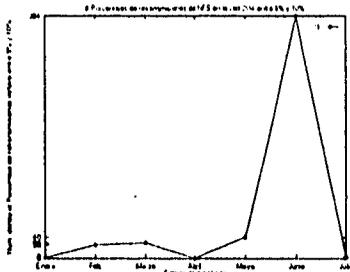


Figura 5.19: Número de veces que el porcentaje de transmisiones se mantuvo entre el 5% y 10%.

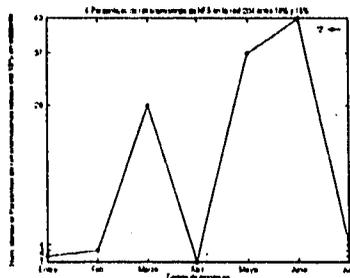


Figura 5.20: Número de veces que el porcentaje de transmisiones se mantuvo entre el 10% y 15%.

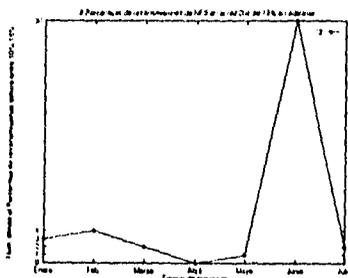


Figura 5.21: Número de veces que el porcentaje de retransmisiones se mantuvo del 15% en adelante.

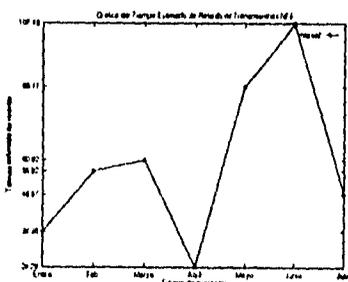


Figura 5.22: Comportamiento del tiempo estimado de retardo para la transmisión de peticiones de NFS, desde la red 201.

Ruteador	Tiem.Retardo
1	1.381527
2	1.231265
3	2.521842

Tabla 5.8: Tiempo de retardo estimado para cada ruteador que conecta externamente a la red 204

Red	Ruteador	Tiem.Retardo
159	1	107.353039
159	2	107.202777
161	1	93.854463
161	2	93.704201

Tabla 5.9: Retardos estimados de transmisión de peticiones NFS desde la red 204 a las dos redes locales del servidor NFS

servidor.

Los resultados obtenidos mediante los modelos probabilísticos construidos para obtener los tiempos mínimos de retardo de transmisión de una petición de NFS, desde la red 204 a cualquiera de las dos redes del servidor y atravesando por cualquiera de los ruteadores 1 y 2, se resumen en la tabla 5.9.

5.5 Comparación entre tiempos de retardo de clientes

La comparación entre los tiempos de retardo estimados de los clientes que están conectados en una red local con los clientes conectados por redes externas, resulta más clara en la siguiente tabla:

De acuerdo con la tabla anterior, son cinco los puntos que se deben notar:

1. Los tiempos de retardo estimados calculados directamente de los modelos probabilísticos coinciden con las observaciones reales de fallas en la red.

Como se notó en los análisis anteriores, el retardo estimado calculado directamente del modelo aumentaba cuando el desempeño real de la red disminuía, esto muestra la efectividad de los modelos.

Red	Red	Retardo
159	159	43.387418
161	161	29.888842
160	159	44.929514
160	161	31.430938
170	159	102.825267
170	161	89.326691
204	159	107.277908
204	161	93.779332

Tabla 5.10: Comparación entre los retardos de las cuatro redes

2. El tiempo de retardo estimado para los clientes de una red no local al servidor solo se puede considerar como el tiempo mínimo.

Esto se debe a que al atravesar por un ruteador no se tiene la certeza de que tome la ruta más corta, es por eso que en esta sección se consideraron solo los tiempos mínimos.

3. La diferencia entre el retardo de una red ethernet local al servidor y una red fddi no local al servidor es muy pequeña.

La explicación es fácil, debido a la construcción de una red fddi, donde no existen las colisiones, no existen los tiempos límite y el ancho de banda teóricamente es de 90 veces más que en una red ethernet.

4. La diferencia entre el retardo de una red ethernet local al servidor y una red ethernet no local a este mismo es aproximadamente el doble.

Esto se debe principalmente a que el retardo local de cada red ethernet es grande, a diferencia de una red fddi cuyo retardo no rebasa a la unidad. Además cabe mencionar que aunque estas dos redes ethernet pasan por dos ruteadores, se vio que por el retardo de estos no se puede considerar un factor causante de la diferencia entre los retardos.

5. Si las dos redes ethernet fueran locales al servidor, el retardo se reduciría al retardo de la red.

Al igual que ocurre con las redes ethernet locales al servidor, disminuiría el tiempo total de retardo y con ello los errores de colisiones y tiempos límite

Parámetro	Cantidad
Velocidad de transmisión	100 Mb/seg
Núm. Estaciones	5
Núm. bit encabezado	32
Retardo de propagación	5.082 μ s/Km
Retardo en el anillo por estación	1 bit
Ecuación de retardo	Ecuación 4.177

Tabla 5.11: Parámetros de la red **fddi** utilizados para la comparación con la red **ethernet**.

excedidos.

El análisis anterior muestra una aplicación de los modelos estocásticos construidos y da la pauta para las bases de un buen diseño de red en un Sistema de Cómputo Híbrido como el estudiado en el Laboratorio de Visualización y el Departamento de Supercómputo.

5.6 Comparación entre el desempeño de un Ethernet y un Fddi

Con base en los modelos probabilísticos de las redes ethernet y fddi, se hizo una comparación entre los retardos de cada uno de ellos.

Los parámetros tomados en cuenta son: Número de paquetes enviados, logitud de cada uno de los paquetes, en una red ethernet y una red fddi con las características como se muestran en las tablas 5.12 y 5.11.

Bajo estos parámetros se estudió el tiempo de retardo de estas dos redes. Los resultados se resumen en la gráfica 5.23.

Esto indica que para una carga de tráfico de información en la red menor o igual al 0.2 el **ethernet**, muestra un buen desempeño. Sin embargo cuando rebasa este número, el retardo se incrementa dramáticamente. Esto indica que para produccio-

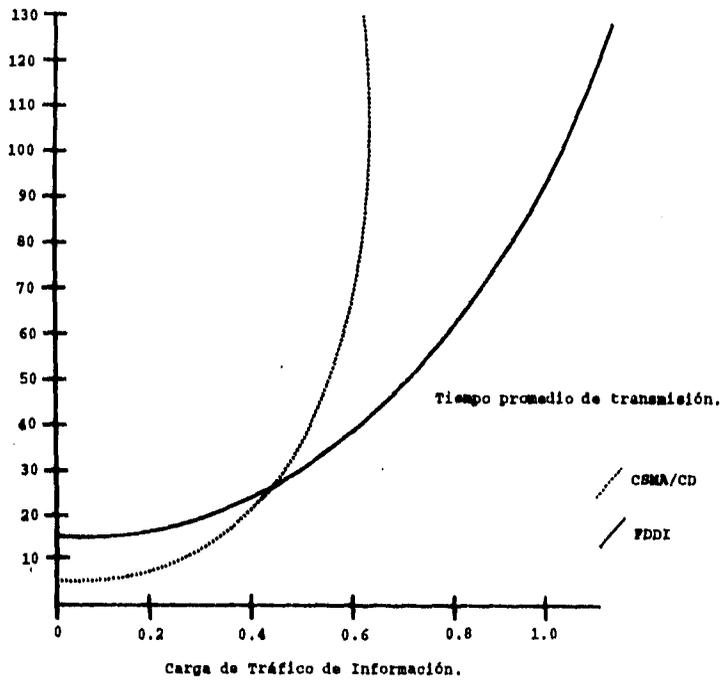


Figura 5.23: Comparación entre una red de tipo Ethernet y una de tipo FDDI.

Parámetro	Cantidad
Velocidad de transmisión	10 Mb/seg
Núm. Estaciones	8
Núm. bit encabezado	32
Retardo de propagación	5 μ s/Km
Ecuación de retardo	Ecuación 4.164

Tabla 5.12: Parámetros de la red ethernet utilizados para la comparación con la red fddi.

nes de mensajes cada vez más grandes, la frecuencia de transmisión se incrementa ocasionando el envío de mensajes dentro del período de vulnerabilidad provocando las colisiones. Debido a que las colisiones generan las retransmisiones, aumenta el tráfico de información y con ello el retardo de transferencia de mensajes.

En cambio para un fddi, el retardo de transmisión para cargas de tráfico de información pequeñas es mayor pero disminuye en cuanto esta carga aumenta. El retardo en una red fddi está determinado por el tiempo que cada máquina debe esperar para tomar el token y poder transmitir. De esta forma si se tiene un número pequeño de transmisión de datos, este retardo será visible pues de cualquier forma la máquina tiene que esperar su turno. En cambio para envíos de grandes cargas de mensajes, el tiempo de espera es insignificante y como el flujo de datos es más rápido el desempeño de la red es muy alto.

Parte VI
Conclusiones

Capítulo 6

Conclusiones

6.1 Recomendaciones para mejorar el desempeño de un sistema de cómputo híbrido

Anteriormente se mostró que el principal problema del sistema de cómputo híbrido es la red, más específicamente los segmentos de red de tipo **ethernet**.

Las recomendaciones que se basan en el modelo y los criterios de monitoreo se resumen en los siguientes puntos.

- Limitar el número de máquinas por red (**ethernet** o **fdi**).

Este punto aunque ya fue discutido anteriormente, es importante recalcarlo, sobre todo en sistemas de cómputo cuyo rendimiento depende del factor red. En el caso del sistema híbrido descrito al principio de este estudio, la recomendación es evitar la sobrepoblación de máquinas conectadas en las redes **ethernet**. Una solución es conectar las máquinas que generen más carga en el sistema en la red **fdi**, ya que esta además de proporcionar un mejor desempeño cuenta hasta el momento con un número pequeño de estaciones conectadas en ella, que si bien es cierto que estas máquinas generan carga de tráfico en la red por ser las máquinas más utilizadas, en la actualidad otras máquinas como el servidor de NIS y NFS y las máquinas de servicios de impresión sobrecargan las redes **ethernet**.

- A mayor número de conectores de redes secundarios (ruteadores, gateways, etc.), mayor probabilidad de error.

Como se mostró en el modelo probabilístico del sistema de cómputo, si los factores que intervienen en la transmisión de los mensajes aumente, el rendimiento disminuye y con ello las posibilidades de error; pues de nada sirve tener un buen desempeño de red, si el rendimiento de alguno de estos conectores de red es pobre.

- Es mejor acceder un archivo grande localmente que remotamente.

Si es necesario leer o escribir un archivo grande es más provechoso transferirlo al disco local. Esto trae consigo que la red no se sobrecargue, el servidor procese más rápido las peticiones necesarias de NIS y NFS provenientes de otros o del mismo cliente, y que el tiempo de retardo se establezca en su mínimo posible.

- Aumentar una interfaz de la red fddi al servidor de NFS y NIS.

La velocidad de transmisión de una red fddi supera en mucho a la velocidad de transmisión de una red de tipo ethernet. Una recomendación que contribuiría en la superación del desempeño es integrar al servidor de NFS y NIS en una red fddi conectada junto con los clientes de NIS, de esta manera se tendría un ambiente distribuido confiable y rápido. Sólo se tendría que considerar la posibilidad de configurarla de tal manera que el servidor contara con una prioridad de transmisión especial.

6.2 Conclusiones Generales

Recordando el objetivo:

Definir un modelo matemático de un sistema híbrido de cómputo comunicado por red que permita predecir las fallas y rendimiento de este, con la característica de mantener dos sistemas de red llamados NIS y NFS. Además de construir programas basados en el modelo, que permitan hacer el análisis de fallas que afectan a los dos servicios de red en el Sistema Híbrido de Cómputo.

Se construyeron modelos matemáticos basados principalmente en la Teoría de Colas y con ello se definió el modelo matemático de un sistema híbrido de cómputo comunicado por red. Dos de los modelos representan los servicios de red NIS y NFS, y están contemplados dentro del modelo general del sistema.

Mediante estos modelos se obtuvieron ecuaciones con las cuales fue posible desarrollar programas que al aplicarlos en el sistema de cómputo en estudio se obtuvo un análisis de tiempos de retardo y rendimiento del sistema y con ello fue posible identificar las fallas que afectan al funcionamiento de los servicios de red, lo cual produce que se afecte el sistema.

Por lo anterior, este estudio cumplió con los objetivos fijados en su comienzo; sin embargo el estudio no termina pues la evolución de cada uno de los factores que se estudiaron se mantiene activa. Para un análisis más avanzado sería interesante recurrir a la simulación para construir un sistema de cómputo híbrido conectado en red óptimo y estudiar las posibles fallas a través de esta tesis. Así, se complementaría o en su defecto se construiría un sistema de cómputo híbrido conectado en red mejorado.

La contribución más importante de esta tesis es auxiliar al administrador de un sistema de cómputo en el monitoreo y análisis de la estructura de red y los factores que esta afecte, con el fin de que el sistema pueda alcanzar el mayor rendimiento y el mínimo de tiempo de retardo en el procesamiento de peticiones vía red.

Parte VII
APÉNDICES

Apéndice A

Monitoreo de un Sistema Híbrido de Cómputo

A.1 Herramientas y Técnicas de Monitoreo de un Sistema de Cómputo

En un sistema de cómputo real, con máquinas conectadas en red, generalmente no es posible identificar plenamente las fallas o errores. Mediante la práctica y la literatura en ocasiones se puede construir una idea pero sin bases firmes que la sustenten.

Los modelos matemáticos presentados en la sección anterior, dan las bases necesarias para determinar el factor que está contribuyendo en la deficiencia del rendimiento total en el sistema.

Con el fin de aplicar estos modelos en la práctica, se crearon herramientas que proporcionan los suficientes datos para conocer el estado de un sistema de cómputo con los servicios de red *NIS* y *NFS*; y conocer que factor está interviniendo en las fallas o errores o en su defecto corroborar el buen desempeño del sistema.

A.2 Monitoreo del servicio de red NFS

En el desempeño del servicio de red *NFS*, intervienen dos factores importantes que son:

- Servidor de NFS

- Red

Se considera que el desempeño de un servidor de NFS es pobre, si procesa las peticiones de los clientes en un tiempo mayor al tiempo límite fijado, de tal forma que ocasiona una retransmisión por parte del cliente. De esta manera, el servidor tiende a procesar por lo menos dos veces la misma petición provocando que el cliente acumule un alto número de identificadores no registrados en sus estadísticas.

Es por esto que si el porcentaje de retransmisión supera el 5% y la diferencia entre el número de retransmisiones y el número de identificadores no registrados es relativamente pequeña (2 o 3 unidades), se sabe que el problema radica en el servidor.

El problema de fallas se le atribuye a la red, si el tiempo de transmisión excede el tiempo límite de transmisión fijado, generando una retransmisión por parte del cliente, y el servidor ejecutando solo una retransmisión exitosa.

Si el porcentaje de retransmisiones supera el 5% y el número de identificadores no registrados es pequeño en comparación con el número de retransmisiones el problema es el desempeño de la red.

La herramienta de monitoreo creada, da a conocer el porcentaje de fallas en el servidor y el cliente, así como el promedio de veces que el porcentaje superó al 5%, 10% y 15%. Además genera el porcentaje de carga en el servicio y el porcentaje de tiempos que rebasen el límite de transmisión. Tomando los datos de las estadísticas generadas por los comandos de NFS. Por otra parte, con base en el modelo, esta herramienta genera el número promedio de llegadas, tiempo de espera en la cola y en el sistema en general.

A.2.1 Monitoreo de NIS

El servicio de red NIS, al igual que el servicio de red NFS, depende de dos factores importantes que son el servidor y la red.

Para construir el monitoreo del servidor de NIS, se tomaron en cuenta, el tipo de operación realizada por todos los clientes en el servidor, el número de llamadas en promedio, el número de errores que el servidor generó, el tiempo promedio de servicio para cada una de las llamadas dependiendo del tipo de operación y por último el tiempo promedio de espera en la cola.

Lo anterior se basa en el modelo matemático y en estadísticas que proporcionan los comandos de NIS. El criterio para considerar que un servidor de NIS tiene un desempeño pobre, lo da el tamaño de los tiempos de fase de servicio y de espera

en la cola. Si el tamaño de estos dos campos es mayor que uno, significa que el servidor no puede procesar las peticiones en el mismo promedio de tiempo que los arribo de estas mismas.

Esto último provoca que el tiempo de espera en la cola aumente, así como el número de retransmisiones por parte del cliente. Debido a que NFS es un servicio de red interactivo, si estos tiempos son muy grandes, el retardo se nota claramente mediante la lentitud de la sesión, así la transparencia de este servicio se ve entorpecida.

En cuanto al monitoreo de red, es importante considerar el tráfico en esta ocasiónado por la carga de trabajo, colisiones y retardos.

Las herramientas para su monitoreo son consideradas en la siguiente sección.

A.2.2 Monitoreo de Red

La red es el factor limitante del rendimiento de cualquier servicio de red. Esto se debe a que permite el desarrollo y funcionamiento de comunicación entre máquinas de todo tipo de arquitecturas y bajo cualquier protocolo. En especial en este estudio se consideran dos tipos de protocolos de red que son el **ethernet** y el **fdi**.

Las herramientas de monitoreo para la red **ethernet** abundan dentro de las diferentes versiones del sistema operativo Unix, sin embargo todas ellas muestran resultados engorrosos y difíciles de identificar a primera vista. Para este estudio se consideró crear una herramienta que proporcionara de una forma más clara los factores que afectan en el desempeño de la red **ethernet**, identificando plenamente el origen del problema.

Debido a que en **ethernet**, las principales fallas son provocadas por el número de colisiones, se creó una herramienta que muestra por cada intervalo de tiempo el número de paquetes emitidos y recibidos por la interfaz de red, el número de errores ocasionados al emitir y recibir cada uno de los paquetes y el número de colisiones, así como el porcentaje de estas.

De esta forma se puede apreciar con menor dificultad el desempeño de cada segmento de red, y los errores ocasionados por las colisiones que se presentan en el momento.

La red **fdi** fue creada y diseñada para evitar las colisiones, pero su principal deficiencia se mide en el retardo de transmisión de los paquetes. Por construcción, este tipo de redes generalmente ofrece un alto desempeño de ancho de banda que supera una red de tipo **ethernet**, además de proporcionar herramientas de monitoreo, que fueron utilizadas para corroborar el buen desempeño de la red **fdi** en el sistema de cómputo en estudio.

Las herramientas mencionadas para **NIS, NFS, Ethernet** y las propias para **Fd-di**, fueron aplicadas en el sistema de cómputo en estudio generando resultados interesantes. La portabilidad del grupo de monitoreo puede ser portado a cualquier sistema de cómputo conectado vía red. Están disponibles en el servidor de ftp del Departamento de Supercómputo **ftp.super.unam.mx://pub/tools/**

Apéndice B

Protocolos de Comunicación en red.

Para detectar las fallas en el sistema híbrido de cómputo anteriormente descrito, es necesario conocer su funcionamiento. Es por esto que es importante saber los conceptos básicos con los que se apoya para establecer la comunicación en red de sus integrantes, así como los servicios de red que aplica en su desarrollo.

B.1 Preliminares

A continuación se describen los protocolos de Comunicación en red, los cuales hacen posible la comunicación en red de un sistema híbrido de cómputo con servicios de red.

Un protocolo de comunicación es un conjunto de reglas que coordinan el intercambio de mensajes entre dos o más sujetos; un ejemplo de un protocolo en el humano es el uso del "Cambio y Fuera" en la radiocomunicación; tanto el emisor como el receptor saben que el mensaje junto con la palabra "cambio" es una señal de que el turno para que el receptor envíe su mensaje está listo, y la palabra "cambio y fuera" para que finalice el envío de mensajes.

Del mismo modo, existen protocolos de comunicación para que los sistemas de cómputo puedan intercambiar mensajes o datos.

Con el objeto de estandarizar la función de los protocolos que existen de este tipo la *International Standardization Organization (ISO)* propuso una arquitectura para construir un modelo de protocolo llamados modelos *Open Systems Interconnection, (OSI)*, la figura B.1 muestra esta arquitectura. El concepto de construir un protocolo por etapas se basa en el hecho de que cualquiera de estas puede usar los servicios de la capa anterior sin necesidad de conocer como mantiene este ser-

vicio.

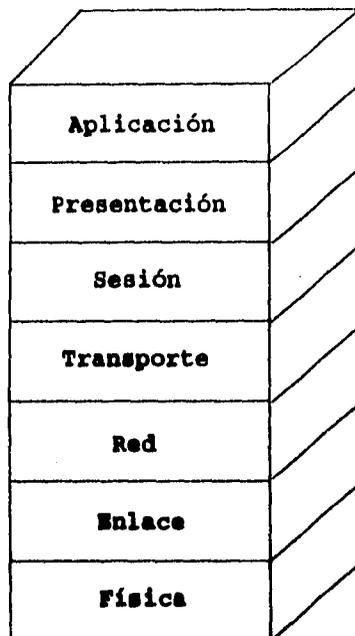


Figura B.1: Estructura de un modelo OSI

B.2 Tareas principales de un protocolo

Además de las tareas ya especificadas, un protocolo tiene el objetivo de proveer servicios de transmisión de mensajes. En la tabla B.1 se muestran los servicios que ofrece.

La alteración de la información durante la transmisión, junto con los datos que se pierden, son de los problemas más frecuentes, en redes cuya distancia es larga, el camino de un mensaje puede llegar a confundirse; por ejemplo cuando una máquina manda un paquete de mensajes a distintas direcciones en la red. Otro de los problemas comunes se da cuando una máquina envía los mensajes más rápido que en lo que esta disponible la máquina receptora, para prevenir esto la máquina

Tarea	Descripción
Detección de Error	Realiza chequeos y testifica las transmisiones de mensajes
Eliminación de Error	Corrige los métodos de transmisión de paquetes
Control de Flujo	Testifica el control de flujo

Tabla B.1: Tareas de un protocolo en la transmisión de mensajes

emisora debe de enviar los paquetes más lentamente.

B.3 Protocolos de comunicación directa y de comunicación indirecta

Existen dos tipos de protocolos de comunicación de datos sobre la red, estos son los llamados *Protocolos de comunicación directa (Connection Oriented Protocol)* y *de comunicación indirecta (Connectionless Protocol)* en el primer caso la comunicación entre la pareja de estaciones se establece antes de enviar el mensaje, mientras que en el segundo caso el mensaje es enviado por una de las partes sin establecer una comunicación directa con la parte que se le envió el mensaje. El protocolo de comunicación directa funciona de la misma forma que una comunicación por teléfono, es decir, la persona marca el número (hosts) en donde quiere transmitir el mensaje si la línea telefónica (línea de red) esta desocupada, entonces se establece la comunicación entre las dos personas (estaciones de trabajo) y el mensaje es transmitido. El protocolo de comunicación indirecta funciona al igual que un telegrama, es decir, la persona (máquina) manda el mensaje a una dirección especificada (hosts), sin necesidad de establecer comunicación directa con la persona.

En la tabla B.2 se muestra un resumen de las funciones de estos dos protocolos.

B.4 Arquitectura de TCP/IP

En 1973 se creó la arquitectura de un protocolo de comunicación llamado *TCP/IP* cuyas especificaciones fueron:

Tipo	Modo de operación	Áreas de aplicación
Comunicación Directa	Tres fases: Establece la conexión, transfiere los datos, cierra la conexión	Sesiones de terminales, transferencia de archivos
Comunicación Indirecta	Transmite el mensaje a la dirección especificada	Bases de datos, transacción de sistemas

Tabla B.2: Funciones de comunicación directa e indirecta

- Independencia del tipo de arquitectura de la máquina y red en la cual se conectaba.
- Comunicación universal a través de la red.
- Estandarizar las aplicaciones de los protocolos existentes.

Con estas especificaciones fue posible que este conjunto de protocolos pudieran tener la estructura de los modelos de referencia de tipo *OSI* en la figura B.2 se muestra más claramente.

A medida que se expandía *TCP/IP* comenzó a mostrar más características las cuales fueron:

- Ruteo dinámico
- Protocolos de transporte con funciones de seguridad
- Un conjunto de programas de aplicaciones comunes
- Protocolos de comunicación indirecta en su capa de red.

B.4.1 Internet Protocol (IP)

Todas las computadoras en Internet entienden este protocolo; la principal tarea del *Internet Protocol (IP)*, es fragmentar los paquetes de información y enviarlos a su próximo destino. En resumen las propiedades de este protocolo se enuncian a continuación:

- Es un protocolo de comunicación indirecta.

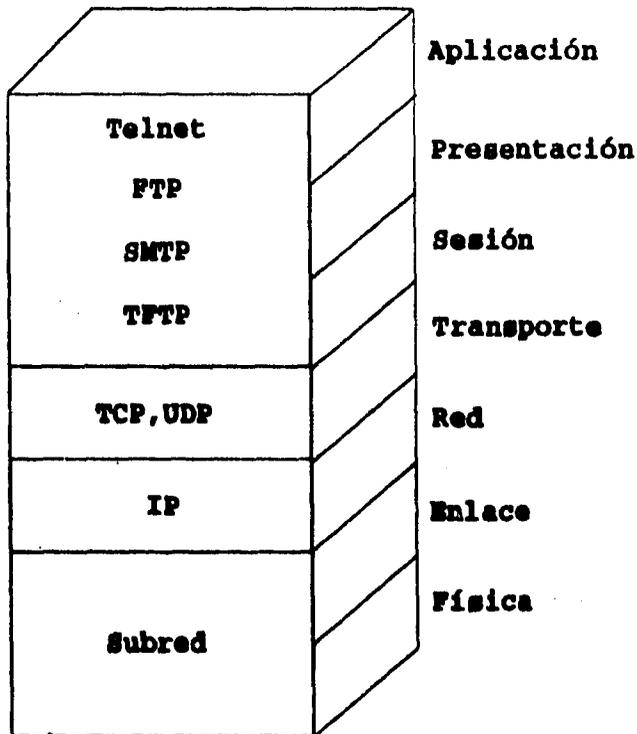


Figura B.2: Arquitectura de TCP/IP

- Pertenece a la capa de red en la estructura OSI.
- Fragmenta los paquetes de información si es necesario.
- El máximo tamaño del paquete de información es de 65535 bytes.
- Tiene un tiempo límite para entregar el paquete.
- Hace su mejor esfuerzo por enviar la información dentro del tiempo establecido.

IP no garantiza que la información que fue enviada no sea alterada o que esta llegue completa, esa tarea es propia de las capas posteriores a ella.

B.4.2 Transmission Control Protocol (TCP)

El protocolo *Transmission Control Protocol (TCP)* pertenece al grupo de protocolos de transporte, es un protocolo de comunicación directa y funciona junto con los servicios del Internet Protocol (IP); su principal tarea es asegurar el transporte de la información a través de la red. Otros atributos de este protocolo son:

- Asegura la transmisión de datos utilizando:
 - secuencia de números
 - chequeos de la transmisión de información
 - retransmisión de segmentos después de checar los tiempos de transmisión de estos.
- Los datos no son transmitidos por bloques
- Asegura la entrega secuencial de paquetes de información.
- Cierra la conexión de transmisión de datos fácilmente.

B.4.3 User Datagram Protocol (UDP)

Este protocolo pertenece a la familia de los protocolos de comunicación indirecta y sus propiedades son:

- Comunicación indirecta

- Es muy simple
- Hace chequeos de datos
- Envía la información por bloques

Apéndice C

Respaldos

C.1 Necesidad de hacer respaldos

Cuando una máquina Unix funciona de una forma estable (no existe anomalía alguna en sus archivos de configuración, en el sistema operativo, alguna aplicación, etc.), la mayoría de los usuarios y administradores no se preocupan por tener una copia de la información.

Muchos de los problemas que los accidentes y fallas provocan, se pueden corregir si tanto los usuarios como administradores hacen respaldos de dicha información. Aún esto no basta, pues los respaldos deben de ser actualizados a medida que la información se modifica o aumenta.

Los problemas más comunes que provoca el no tener respaldada la información se mencionan a continuación:

Cuando se es usuario de una máquina Unix, no se toma en cuenta medidas de prevención para proteger la información contenida en dicha máquina, muchas veces por falta de costumbre o por ignorancia, pero cuando llegan a surgir accidentes como borrarse un archivo porque se ejecutó un comando por error o cuando se modifican los archivos y estas modificaciones afectaron de una forma no deseada la información, es cuando la necesidad de respaldarla llega a la mente del usuario y entonces recurre al administrador. Si este no se había percatado de hacer algún respaldo a tiempo entonces comienzan los problemas, ya que por lo general la información que se perdió se obtuvo invirtiendo tiempo, recursos de la misma máquina y recursos externos como dinero, etc.

Como administrador de una máquina, el no tener precaución de hacer respaldos de la información, es un problema mucho más serio que siendo usuario de ella,

pues en el caso de un administrador, la responsabilidad aumenta al no sólo tener que respaldar información específica, sino que es necesario respaldar toda la que contenga la máquina que administra, desde archivos del sistema hasta archivos pertenecientes a los usuarios.

Los problemas más comunes que se le presentan al administrador como consecuencia de falta de respaldos, ocurren cuando se modifican archivos de configuración, y se producen cambios en los parámetros o en los archivos del sistema, lo cual altera el funcionamiento de este, provocando resultados erróneos y en el peor de los casos que este deje de funcionar. Si el administrador no puede volver a restaurar estos archivos ya sea con ayuda de alguna bitácora, auxilio de algún administrador con más experiencia o mediante manuales y libros, se tendrá que enfrentar a la solución más drástica, el volver a instalar el sistema operativo y con ello arriesgarse a perder más información, en caso de que se presentara alguna falla al momento de la instalación.

Cuando el hardware de una máquina falla, es otro de los problemas serios, pues en caso de da no físico del disco la información puede perderse por completo.

El caso menos drástico, pero de igual importancia, sucede cuando se pierde información de aplicaciones, ya que muchas veces estas requirieron de tiempo y estudio para su compilación.

En cualquiera de los dos papeles, usuario o administrador es una necesidad el tener respaldos de la información en cualquier máquina, debido a que estas están expuestas a todo tipo de accidentes y fallas. Aún más, se debe de tener un mecanismo de actualización en ellos, pues la información se encuentra en modificación constante.

C.2 Formas de respaldo

Las dos formas más óptimas de hacer respaldos de la información en una máquina son:

- Dispositivos de respaldo
- Red

Como dispositivos de respaldo podemos enumerar:

1. Cintas
2. Discos flexibles

3. Impresoras

Dispositivos de respaldo

No todos los dispositivos de respaldo son óptimos para efectuar el respaldo, sobre todo cuando la capacidad de espacio en cada dispositivo es menor al espacio necesario para guardar la información, como es el caso de los discos flexibles, que aunque son prácticos en cuanto a la portabilidad, generalmente su capacidad no cubre la necesidad de respaldar grandes volúmenes de información, o en caso de las impresoras que tampoco son un medio eficaz para el respaldo, puesto que en el caso de archivos de tipo binario, no se puede volver a generarlos a partir de una hoja de impresión, así el tiempo en recuperar la información a partir de este medio sería mayor que volver a generarla.

Por el contrario, existen otros mecanismos de respaldo más óptimos como es el caso de los dispositivos de cintas, puesto que la mayoría de ellas tiene gran capacidad y es de fácil uso para transportar la información, ya sea del disco de la máquina a la cinta o viceversa.

Red

Tener una máquina en red, permite comunicarla con otras, que de igual forma se encuentren conectadas, de esta forma se puede almacenar información de una máquina a otra, la red como medio auxiliar de respaldos desempeña una buena función, las dos desventajas de almacenar los respaldos en disco son, la dependencia que se genera a partir de almacenar toda la información en una máquina, si esta llegara a fallar, los respaldos no pueden estar disponibles en ese tiempo de falla; si el espacio en disco llega a su límite la información corre el riesgo de no ser respaldada completamente.

C.3 Automatización de respaldos

La tarea de hacer respaldos para los usuarios y administradores es en ocasiones muy tediosa, aún más si se trata de una máquina sin dispositivos de respaldos óptimos. Por esta razón se creó un mecanismo de automatización de respaldos, que consiste en almacenar la información mediante dispositivos de cinta o disco remoto llamado **rtar**.

C.3.1 Descripción de rtar.

Este programa fue diseñado para facilitar esta tarea y hacerla amigable para el usuario y para el administrador. Realiza el respaldo ya sea con archivos ordinarios, directorios y es capaz de utilizar un archivo de datos que contenga dichos archivos a respaldar para comodidad del usuario.

Utiliza un programa auxiliar en la máquina remota el cual realiza las tareas locales de control de cinta, este archivo se puede encontrar en el paquete del `rtar` y se encuentra disponible vía ftp anónimo `ftp.super.unam.mx://pub/tools/rtar`.

Archivos de configuración: `.rtar.src` y otros

Se auxilia de archivos de configuración para obtener la máquina remota en donde va a realizar los respaldos, junto con el(los) dispositivo(s) disponibles y la ruta del archivo remoto auxiliar. Existen tres tipos de archivos de configuración: global, local y los definidos bajo la opción `-c`. Por convención tanto el archivo global como el local tienen los nombres de `.rtar.src`, al archivo global generalmente se le encuentra en el mismo directorio donde los archivos auxiliares fueron instalados por default; el archivo local se encuentra en el directorio principal de cada usuario y fue creado con la intención de que el usuario tenga la posibilidad de crear su propio archivo de configuración, así en caso de que el usuario desee agregar una máquina remota más para respaldar, no es necesario modificar el archivo global, por el contrario si el usuario no define su archivo de configuración local, el archivo global es utilizado para conocer en que máquinas puede realizar un respaldo remoto automático. El último tipo de archivo de configuración se define mediante la opción `-c`, la cual funciona de manera independiente y se describe en los siguientes párrafos.

Archivos de log

La primera vez que un usuario ejecuta este comando, inmediatamente `rtar` crea en su directorio principal del usuario dos archivos más llamados `.rtar` y `.rtar/logs` en este último se almacenan los archivos de log que contienen la información de los archivos que fueron respaldados por `rtar`, los archivos de log por default son comprimidos mediante el comando `gzip`.

C.3.2 Opciones para el manejo de rtar,

- **-h hosts**

Especifica el nombre de la *máquina remota* la cual se dio de alta en alguno de los archivos de configuración. si *hosts* no es especificado *rtar* regresa un error.

- **-l login**

Define el login del usuario en la máquina remota, por default toma el login del usuario que ejecutó el comando. Recuerde que este login debe estar dado de alta en el archivo *hosts*.

- **-t dispositivo**

Especifica el dispositivo de cinta o archivo en donde se van a realizar los respaldos, esta opción es necesaria en caso de que la máquina remota contenga más de un dispositivo de cinta o se requiera respaldar en un archivo no especificado en el archivo de configuración. Cuando la máquina remota contiene un solo dispositivo de cinta y esta opción no es especificada toma el dispositivo descrito en el o los archivos de configuración correspondientes a la máquina remota definida por la opción *-h*.

- **-c arch-conf**

Se especifica un archivo de configuración el cual contiene el o los nombres de las máquinas remotas, el o los dispositivos y la ruta del archivo remoto el cual controla localmente los dispositivos de cinta. Si esta opción no es ejecutada por default *rtar* toma los archivos de configuración ya descritos.

El formato de cualquier archivo de configuración es:

```
maq-remota disp1,disp2,...,disp-n arch-remoto
```

- **-f arch-datos**

Especifica los archivos a respaldar en el archivo: *arch-datos* esta opción es útil en caso de que se requieran respaldar archivos y/o directorios en diferentes rutas.

- -c
Reorre la cinta al final del último registro. Si el dispositivo es no-rewind la función de esta opción se cumple satisfactoriamente, en caso contrario si el dispositivo es de tipo rewind por definición regresa la cinta al principio.
- -v Esta opción muestra los archivos, directorios que estan respaldandose en la cinta así como mensajes del estado de esta. En caso de que esta opción no se ejecute la información de los archivos respaldados es almacenada en los archivos de log, que se generan dentro del directorio, estos archivos son creados por default la primera vez que se ejecuta **rtar**
- -d Despliega el contenido de la cinta especificada, esta opción es equivalente a no especificarl al programa ningún archivo o archivos de datos a respaldar.
- -a Opción de ayuda, muestra una breve descripción de las funciones del comando **rtar**.

Control local de los dispositivos

Es importante mencionar que la herramienta **rtar** se auxilia de otros archivos para su funcionamiento, dentro de los cuales destaca el archivo que le ayuda a controlar localmente el manejo de los dispositivos. Es este script se controla la posición de la cinta al momento de almacenar la información, de esta forma si la cinta ya tenía información almacenada, el programa avanza para no da naria.

C.3.3 Localización de la herramienta rtar

Este programa junto con sus auxiliares se estan disponibles vía ftp anónimo en <ftp.super.unam.mx://pub/tools>.

C.3.4 Automatización de respaldo para tareas de administrador

Una de las funciones básicas de un administrador es almacenar toda la información que contenga la máquina, y por cada partición del sistema realizar un respaldo, es aconsejable que cada uno de estos respaldos se almacenen en cintas separadas y por duplicado, pues en el caso de que una cinta se da nara, no se perdería toda la información.

Con el objeto de distribuir tareas, en muchos sistemas de cómputo se cuenta con gente dedicada a realizar los respaldos, pero existen algunos archivos en los cuales solo el superusuario puede tener acceso a ellos. La forma más fácil de solucionar esto es crear un script en donde se ejecute el o los respaldos remotos y darle privilegios de superusuario. Sin embargo esto trae consigo un hueco de seguridad, con efecto de corregir en lo posible este hueco se crea un programa en donde ejecute el script heredando los privilegios de superusuario pero asegurando que el operador o cualquier persona no autorizada no pueda filtrarse como superusuario, pues al ejecutarse el script, el proceso asociado a este programa muere y con ello asegura la confiabilidad y autenticidad de la información.

Un ejemplo de respaldo automático remoto seguro se muestra a continuación:

```
#!/bin/sh
rtar -h diphda -l root -t /dev/rmt/ln -c config -v /users
\begin(verbatim)

\begin(verbatim)
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main(int argc, char *argv[], char *envp[]) {

    setuid(0);
    setgid(0);
    execl("/usr/users/respaldo/usuarios", "", 0);
}
```

Este programa incluye al comienzo, librerías para poder utilizar tres rutinas de las cuales, las dos primeras sirven para especificar la identificación del superusuario y su grupo, la última ejecuta el programa especificado, teniendo como parámetros la ruta completa del programa a ejecutar, una cadena vacía y un 0 que representa el fin de los parámetros.

Se requiere que este programa tenga activado el suid, con esto al ejecutar el segundo script será con privilegios de superusuario.

C.4 Otras especificaciones acerca de la implementación de la automatización de respaldos

La automatización de respaldos requiere de dos cuentas de usuario normal, tanto en el cliente como en el servidor, para el caso en que la máquina es cliente-servidor, solo se requiere de una cuenta.

Si lo que se desea es respaldar información cuyo due no es el superusuario, se siguen las siguientes recomendaciones.

En la cuenta del servidor y cliente-servidor, deberá editarse el archivo `.rhosts`, este **debe** de contener el nombre de las máquinas clientes así como el de la cuenta desde la cual se ejecutará la automatización, como ejemplo se muestra un archivo de una máquina servidor utilizada:

```
polaris root
polaris.labvis.unam.mx root
gate-polaris root
gate-polaris.labvis2.unam.mx root
mezcal root
mezcal.dgsca.unam.mx root
132.248.160.3 root
```

C.5 Recuperación de los respaldos

Después de tener resuelto el problema de automatización de respaldos, la siguiente pregunta es, **Como recuperarlos?**

El procedimiento es más simple que el anterior, pues sólo se utilizan dos programas en el cliente y uno en el servidor, el primero de los que se encuentran en el cliente, al igual que en la automatización de respaldos, es el que se comunica con la máquina servidor para poder efectuar la recuperación de estos; para esto se tiene un programa que recupera la información de máquinas específicas, sin embargo se esta construyendo ya un nuevo programa que pueda ser portable como en la automatización de respaldos, el programa que ya se tiene se describe a continuación.

```
#! /sbin/sh
```

```
TAR="/sbin/tar tvBfb - 20"
RSH="/usr/bsd/rsh
REMOTA1="diphda.labvis2.unam.mx"
REMOTA2="clamato.dgsca.unam.mx"
REMOTA3="ursa.labvis.unam.mx"
REMOTA4="deneb.labvis2.unam.mx"
LOGIN="respaldo"
DD="/sbin/dd
CD=cd
ECHO="/sbin/echo
LOG_FILE="/usr/users/respaldo/logs/log.respaldo.$$
```

```
$ECHO "TIPO DE CINTA:"
$ECHO " a) 8mm (diphda)"
$ECHO " b) 4mm (diphda)"
$ECHO " c) 8mm (clamato)"
$ECHO " d) 4mm (clamato)"
$ECHO " e) 4mm (ursa)"
$ECHO " f) 4mm (deneb)"
read tamano
case $tamano in
  [a]) CINTAS="/dev/rmt/0"
      REMOTA=$REMOTA1
      DEST=/users/respaldo/remoto2
      ;;
  [b]) CINTAS="/dev/rmt/1n"
      REMOTA=$REMOTA1
      DEST=/users/respaldo/remoto2
      ;;
  [c]) CINTAS="/dev/rst1"
      REMOTA=$REMOTA2
      DEST=/Users/respaldo/remoto2
      ;;
  [d]) CINTAS="/dev/rst0"
      REMOTA=$REMOTA2
      DEST=/Users/respaldo/remoto2
      ;;
  [e]) CINTAS="/dev/tape"
```

```

        REMOTA=$REMOTA3
        DEST=/usr/people/respaldo/remoto2
        ;;
    [f]) CINTAS="/dev/tape"
        REMOTA=$REMOTA4
        DEST=/usr/users/respaldo/remoto2
        ;;
esac

$ECHO "No. de Respaldo:"
read NUMERO
( $RSH $REMOTA -l $LOGIN $DEST $CINTAS $NUMERO ) | $STAR

```

Como se observa al principio del programa se listan variables cuyo valor es válido para todos los servidores, no así para las variables CINTAS, REMOTA y DEST que adquieren valores de acuerdo a cada máquina servidor, pero la parte más importante es el final, donde el comando `rsh` ejecuta el script que se encuentra en el servidor, la salida de este comando se convierte en la entrada estándar del siguiente `tar` a través de una tubería y con esto la información es restaurada en la máquina cliente.

En el caso de que la recuperación de respaldos requieran privilegios de superusuario es necesario construir de nuevo un programa en C, el cual tiene como tarea ejecutar el script principal, este programa se muestra a continuación:

```

#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main(int argc, char *argv[], char *envp[]) {

    setuid(0);
    setgid(0);
    execl("/usr/users/respaldo/regreso", "programa", 0);
}

```

Utiliza librerías que contienen rutinas como `setuid`, `setgid` y `exec`, las dos primeras tienen por objetivo especificar la identificación del usuario y grupo, esto con el fin de que pueda el programa en C ejecutar el script con la identidad adecuada, la última ejecuta el script o programa que se le especifique, teniendo como parámetros, la ruta completa de lo que se ejecutará, seguida de una cadena vacía y por último un 0 quien determina que no hay más parámetros.

Este programa deberá tener el `suid` a disposición para poder ser utilizado por cualquier usuario, y con el aumento la probabilidad de tener más seguro el sistema pues muere el proceso asociado a su ejecución y con ello las posibilidades de que el operador o usuario sigan teniendo privilegios especiales.

En el servidor, se requiere del siguiente script:

```
#!/bin/sh -x

MT="/bin/mt -f"
DD="/bin/dd"
ECHO="/bin/echo"

$MT $1 rewind
if [ $2 -eq 0 ]
then
$ECHO "Colocando la cinta desde el principio..." >&2
$MT $1 rewind
else
2) $ECHO "Avanzando al final del primer respaldo la cinta ..." >&2
   $MT $1 fsf
   ;;
3) $ECHO "Avanzando al final del segundo respaldo ..." >&2
   $MT $1 fsf 2
   ;;
*) $ECHO "Opcion invalida o la cinta es nueva..." >&2
   ;;
esac

$DD if=$1 bs=20b
```

Dicho programa tiene como función principal, el hacer los cambios necesarios

para que se pueda recuperar la información, dependiendo de que número de respaldo se desee, como el programa de automatización de respaldos almacena tres veces información, este programa puede recuperarla hasta el tercer respaldo; una vez recorrida la cinta a su posición deseada, entonces se ejecuta el comando **dd**, este lee la información del dispositivo de drive correspondiente y define el tama no de los bloques con los que fue almacenada. Esta lectura pasa al script de la máquina cliente y es entonces como se ejecuta el comando **tar**, el cual con las opciones especificadas recupera la información.

C.6 Comandos básicos para el respaldo de información y recuperación de la misma

C.6.1 tar

El comando **tar**, puede hacer copias de seguridad de archivos y directorios individuales sobre diferentes tipos de medios (no sólo cintas), y posteriormente restaurar esos archivos. Para el propósito de la automatización de respaldos, así como la recuperación de ellos se ejecutó este comando sobre los dispositivos de cinta asociados a cada servidor.

Las opciones más utilizadas de este comando son:

- **c**
Crea un nuevo espacio de cinta, escribiendo al comienzo de este, esta opción no se preocupa por la posición de la cinta, asume que esta se encuentra al comienzo.
- **v**
Muestra los resultados de la ejecución del comando, por default toma la salida estándar.
- **f**
Indica al comando sobre que dispositivo se efectuará el respaldo o se recuperará la información.
- **x**
Extrae los nombres de archivos desde la cinta, en caso de que estos sean directorios, serán recursivamente extraídos.

- t

Lista los nombres de los archivos especificados que estén almacenados en la cinta.

- b

El comando usa el próximo argumento como un factor de bloque para poder almacenar la información en los bloques determinados por el factor.

- B

Forza la entrada y salida a 20 bloques por respaldo y causa que el proceso de entrada ignore los registros de bloque ilimitados.

Nota:

Es importante tomar en cuenta como se van a respaldar los archivos, pues en ocasiones conviene almacenarlos junto con su ruta, por ejemplo archivos de configuración, del sistema, etc.

Pero muchos otros requieren ser almacenados sin su ruta específica, como aplicaciones, etc. Ya que al momento de querer recuperar la información, si la ruta con la que fueron respaldados no existe la información no se podrá recuperar, pues el comando querrá volver almacenarlos en la misma donde los encontró.

C.6.2 dd

Este comando copia la entrada de archivos especificada a una salida especificada. La entrada y salida estándar están por default. Además convierte archivos de acuerdo a varios patrones.

Las opciones más utilizadas son:

- if=

Sirve para especificar el nombre de la entrada.

- of=

Especifica el nombre de la salida.

- **ibs**
Especifica el tamaño en bloques de los archivos de entrada. (el default son 512 bloques)
- **obs**
Especifica el tamaño en bloques de los archivos de salida. (default 512 bloques)
- **bs**
Esta opción suprime a las dos anteriores, ya que especifica el tamaño en bloques de los archivos de la entrada y salida.
- **conv=**
Convierte archivos de tipo EBCDIC a tipo ASCII o viceversa, de acuerdo como se le especifique.

C.6.3 mt

Su principal aplicación es, manipular y controlar cintas magnéticas. Su uso más frecuente es ejecutar comandos para estas, como ver su estado, recorrerla al principio, al final, etc.

Las opciones más útiles son:

- **f**
Sirve para indicarle el nombre del dispositivo que contiene a la cinta.
- **eom**
Recorre la cinta hasta el final de toda la información que almacena. En algunas variantes de sistema operativo esta opción es llamada como feom.
- **fsf**
Recorre la cinta un respaldo apartir de la posición en que se encuentre esta.
- **rewind**
Se posiciona en el comienzo de la cinta.

- status

Despliega información de la cinta: tipo, posición actual, posición respecto a los archivos almacenados en ella, etc.

- erase

Borra el contenido de la cinta.

- exist

Regresa un 0 si la cinta existe en el dispositivo referido, y no cero si esto es falso.

Nota:

La máquina servidor utilizada para respaldar la información deberá ser la misma para recuperarla pues el formato como estas máquinas la almacenan varía.

Además de que para las cintas de tipo dat, existen restricciones pues no todas pueden ser aceptadas por algunas máquinas sobre todo las de marca SGI, pues sólo aceptan de tipo DDS.

C.6.4 rsh

Este comando permite presentarse en un sistema remoto y utilizarlo mientras aún se tiene una sesión abierta en el sistema propio. Cuando se utiliza el sistema remoto, da acceso al sistema local de acuerdo a los argumentos especificados al ejecutar el comando.

Argumentos útiles:

- hosts

• Representa la dirección o el nombre de la máquina remota.

- -l login

Especifica la identificación de la cuenta a la cual se puede acceder de manera remota, si se omite toma por default la misma en la que se encuentra en sesión en el sistema local.

- comando

Determina el comando a ejecutar en la máquina remota, si este se omite actua como si se abriera una sesión con la cuenta especificada o la cuenta en la que se esta en sesión en el sistema local.

Bibliografia

- [Ma] Mayorov, S.A. *Fundamentals of Computer System Theory* p. 408, 1978.
- [BCM] Basket, F., Chandy, K.M. and Muntz, R.R. *Open, Closed and Mixed Networks of Queuing with Different Classes of Computers*, J. Assoc. Comp. Mach., vol. 22, No. 2, pp. 248-260, 1975.
- [Bu] Burke, P.J. *The Output of a Queuing System*, Operations Res., vol. 4, pp. 699-704, 1956.
- [GN] Gordon, W.J. and Newell, G.F. *Closed Queuing Systems with Exponential Servers*, Operations Res., vol. 15, pp. 254-265.
- [Ja] Jackson, J.R. *Job Shop-Like Queuing Systems*, Management Sci., vol. 10, pp. 131-142, 1963.
- [Ha] Jeremiah F. Hayes *Modeling and Analysis of Computer Communications Networks*, R.W. Lucky, vol. 1, pp. 50-63, 1984.
- [Pe] Peter J.B. King *Computer and Communication Systems Performance Modeling*, C.A.R. Hoare, vol. 1, 1990.
- [Ge] Gerd E. Keiser *Local Area Networks*, McGraw-Hill, vol. 1, 1976.
- [Be] Bernhard Albert, Anura P. Jayasumana *FDDI & FDDI II*, Artech House, vol. 1, 1994.
- [Do] Donald Gross, Carl M. Harris *Fundamentals of Queuing Theory*, Wiley Series in Probability and Mathematical Statistics, vol 1, 1985.
- [St] Hal Stern *NFS and NIS*, O'Reilly & Associates, Inc.
- [Lo] Mike Loukides *System Performance Tuning*, O'Reilly & Associates, Inc.

[Ra] Rick Ramsey *NIS+*, SunSoft.