



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE INGENIERÍA

MECÁNICA Y ELÉCTRICA

SECCIÓN DE ESTUDIOS DE POSTGRADO E  
INVESTIGACIÓN

MAESTRÍA EN CIENCIAS CON ESPECIALIDAD EN INGENIERÍA EN  
TELECOMUNICACIONES

TÉSIS:

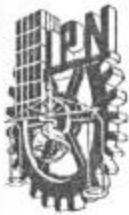
“USO DE TÉCNICAS DE SIMULACIÓN Y MODELADO EN EL DISEÑO Y  
PLANEACIÓN DE REDES DE COMPUTADORAS.”

QUE PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS

PRESENTA EL:  
ING. RAYMUNDO SANTANA ALQUICIRA.

DIRIGIDA POR:  
DR. SALVADOR ALVAREZ BALLESTEROS

MÉXICO D.F. MAYO DE 2003.



# INSTITUTO POLITECNICO NACIONAL

## COORDINACION GENERAL DE POSGRADO E INVESTIGACION

### ACTA DE REVISION DE TESIS

En la Ciudad de México, D. F. siendo las 11:00 horas del día 13 del mes de diciembre del 2002 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de la E.S.I.M.E. para examinar la tesis de grado titulada:

**“USO DE TÉCNICAS DE SIMULACIÓN Y MODELADO EN EL DISEÑO Y PLANEACION DE REDES DE COMPUTADORAS”**

SANTANA  
Apellido paterno

ALQUICIRA  
materno

RAYMUNDO  
nombre(s)

Con registro: 

9	9	1	2	0	3
---	---	---	---	---	---

aspirante al grado de:

MAESTRO EN CIENCIAS


Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

### LA COMISION REVISORA


Director de tesis

  
Dr. SALVADOR ALVAREZ BALLESTEROS

  
Dr. JORGE ISMAEL MONTOYA TENA

  
M. en C. JULIO RAMIRO ALONSO CRUZ.


  
M. en C. RUBEN VAZQUEZ MEDINA

  
M. en C. HECTOR OVIEDO GALDEANO

  
M. en C. MIGUEL SANCHEZ MERAZ

EL PRESIDENTE DEL **COLEGIO**

  
DR. FLORENCIO SANCHEZ SILVA

  
SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION

**IPN**  
ESIME CULHUACAN

DIC. 18 2002

DEPTO. EDUCACION CONTINUA Y CAPACITACION

## RESUMEN

---

La utilización de la tecnología informática y de la comunicación (TIC) es gran parte del quehacer humano, y ha crecido notoriamente en los últimos años en las redes de computadoras. Cabe destacar también la evolución acelerada de los sistemas informáticos en nuevos servicios o en mejoras de los existentes.

Todo esto hace la importancia de considerar el desempeño de las redes de computadoras con su hardware y su software, cuando procesa las tareas y operaciones de los usuarios, obteniendo como consecuencia, una red de computadoras bien diseñada y planeada.

En la sección titulada como “Justificación” se describe la necesidad y la utilidad de manejar herramientas de simulación, que permitan al diseñador de sistemas informáticos, evaluar una red de computadoras antes de implementar o realizar cambios en la misma. Toda esta información se complementa con la parte titulada “Antecedentes”.

Este proyecto integrara algunas de las técnicas de simulación para adquirir una metodología general de análisis de redes para su planeación y diseño.

Por ultimo conformamos conclusiones que se obtuvieron del trabajo realizado y se presentan recomendaciones para trabajos futuros.

## ABSTRACT

---

The use of information technology and communication (TIC) is now essential in our daily activities and has grown significantly in the last few years in terms of computer networking. It's worth to point out also the accelerated evolution of information technology in developing new services or improving the existing ones.

This proves the importance of considering the performance of networks with their hardware and software when it processes the users' tasks resulting in a well-planned and designed network.

In the section "Justificación", the need and usefulness of simulation tools that allow the designer of information technology systems evaluate a network before it is actually implemented or changed is described. This information is complemented with the section "Antecedentes".

This project will integrate some of the simulation techniques in order to acquire a general method of network analysis for planning and design.

Last, conformation of the conclusions obtained of the work performed and recommendations for future projects.

## ANTECEDENTES

---

Actualmente en la planeación y diseño de redes, se desarrollan métodos y herramientas que permitan resolver el problema planteado, como son los monitores de red, que consisten en el seguimiento de la red existente ó la comparación de redes frente a una carga característica de una instalación concreta. Estos métodos y herramientas son bastante costosas y consumen tiempo.

El problema del monitoreo de red tiene una gran dificultad, ya que la carga real de un sistema de red de computadoras es fluctuante en el tiempo y las mediciones no pueden repetirse, salvo que el trabajo se ejecute en un ambiente controlado. Los índices obtenidos de una carga de trabajo en un sistema dado no son los mismos si se procesa dicha carga en otra instalación. Existe una demanda por parte de los usuarios, quienes generan la carga de trabajo, con una serie de requisitos de forma de considerar un funcionamiento aceptable para sus necesidades.

Una herramienta importante es la simulación ya que es una alternativa viable para valuar los servicios de una red de computadoras, con los fines de obtener el mejor desempeño o respuesta a las demandas al mínimo costo.

## OBJETIVO

---

Proporcionar una metodología para encontrar el desempeño de redes de computadoras mediante técnicas de simulación y modelado, que sirvan, con el fin de explorar su utilización en la planeación y diseño de estas.

## JUSTIFICACIÓN

---

Actualmente no es común que el ingeniero mexicano utilice técnicas modernas de planeación y diseño, como lo sería el modelado y la simulación de redes en su trabajo cotidiano, guiándose solamente en su lugar, por métodos empíricos. Esto trae como consecuencia costos mayores cuando se proyecta, diseña e instala una red de computadoras.

La simulación es una excelente herramienta de soporte para predecir el desempeño en un diseño específico de una red de comunicaciones y para probar el impacto de nuevas aplicaciones sobre una red ya existente.

Por lo tanto el generar una metodología que facilite el estudio del posible desempeño que podría tener una red de computadoras es justificable.

## PROLOGO

---

En las comunicaciones de datos, es conveniente tener con precisión el conocimiento del sistema antes de realizar su implementación, prueba y operación real. Sin embargo, algunas veces no es posible probar nuevas soluciones sobre redes ya implementadas debido a la inconveniencia de disturbios sobre el sistema.

Por ello, es conveniente antes de poner en operación una red, aplicar técnicas de simulación y modelado, con el fin de explorar su utilización en la planeación y diseño de redes de computadoras, de modo que el ingeniero mexicano disponga de mecanismos para ponderar el comportamiento de una red en operación antes de su adquisición e instalación. Desde luego esta simulación requerirá previamente de un análisis del desempeño de las redes, mediante la segmentación, el congestionamiento y procesamiento de la comunicación de los recursos que la componen.

Por lo anterior el trabajo de tesis se dividió en 6 capítulos; en cada uno de los cuales se cubrirá lo siguiente:

El primer capítulo explica la importancia de la segmentación de redes, en la planeación y diseño de la misma, permitiendo controlar el número de computadoras conectadas y de esta manera evitar el congestionamiento, teniendo como consecuencia un mejor desempeño de la red.

El segundo capítulo nos permite, saber como mejorar el desempeño de las redes mediante el control de congestión.

Con el tercer capítulo introduce los conceptos básicos del análisis sobre el procesamiento de la comunicación, con la cual es posible comprender las variables que nos permiten medir el desempeño de las redes. Con relación a este tema, se anexa el apéndice A, describiendo el fundamento matemático necesario para entender este tipo de problemas.

En el cuarto capítulo se revisan los aspectos teóricos sobre la aplicación de métodos básicos de modelado y simulación, clasificación, ventajas y desventajas.

Los siguientes capítulos tienen un énfasis fundamentalmente práctico que permiten modelar y simular una red de computadoras, orientándose a cubrir los siguientes aspectos:

Capitulo cinco, tiene el propósito de explicar la metodología, objetivo de este trabajo, donde se mencionan los diferentes pasos para la colección y análisis de las estadísticas de tráfico de red, aplicando técnicas de

simulación y modelado, con el fin de tener un diseño y una administración proactiva de las redes de computadoras.

El capítulo seis, se aplica la metodología propuesta, donde, se analizan los resultados (parámetros de desempeño) que permiten predecir el funcionamiento de la red para determinadas condiciones de carga (aplicaciones de red) simuladas.

Con el objetivo, de corroborar las mediciones del simulador, se construye el caso de estudio en un sistema real.

Se presentan resultados de las mediciones del caso de estudio, mediante la simulación y también los resultados de las mediciones, por medio de analizadores y monitores del sistema real, hechos en el laboratorio de Cabletron de E.S.I.M.E. Zacatenco.

El caso de estudio consiste en la planeación y diseño de un sistema informático, con aplicación cliente / servidor, para la empresa de AXTEL Telecomunicaciones, donde se necesita, encontrar el desempeño en las redes LAN y WAN, que soporten el trabajo requerido y de esta manera saber el comportamiento de la operación de la misma para cambios futuros.

## NOTACIÓN.

Cuando se presenta información que forma parte de la salida producida por la computadora se utiliza texto como este. **Los acrónimos, Siglas de algoritmos, tecnologías y abreviaciones se escriben con mayúsculas o minúsculas y en negritas como en este texto.**

Capitulo 1	
SEGMENTACIÓN DE REDES	1
1.1 “Switches”	2
1.1.1 Mecanismo de control de la congestión	7
1.2 Puentes	8
1.2.1 Puentes a nivel de MAC: 802.1 – Interconexión de redes	9
1.2.2 Estructura de un puente	11
1.2.3 Árbol de expansión (Spanning Tree)	13
1.3 Enrutamiento de origen	13
1.3.1 Propiedades de un algoritmo de enrutamiento	17
1.3.2 Métodos de encaminamiento	18
1.3.3 Clasificación de los métodos de enrutamientos	19
Capitulo 2	
CONTROL DE CONGESTIÓN DE RED	22
2.1 Fenómeno, Definición	23
2.1.1 Causas de la congestión	24
2.1.2 Diferencia entre control de flujo y control de congestión	25
2.2 Soluciones, Mecanismos de control de congestión	26
2.2.1 Soluciones en bucle abierto	26
2.2.2 Soluciones en bucle cerrado	27
2.2.2.1 Monitorización de parámetros	27
2.2.2.2 Reacción: envío de información a los puntos necesarios...	28
2.2.2.3 Ajuste del sistema	28
Capitulo 3	
CONCEPTOS DE MÉTRICAS Y MODELOS OPERACIONALES	29
3.1 Modelo operacional	30
3.2 Definición de variables de entrada o parámetros	32
3.3 Estimación de parámetros con datos observados	32
3.4 Definición de variables de salida, de respuesta del modelo o medidas / índices de desempeño	33
3.5 Leyes operacionales	34
3.6 Métricas de desempeño en un sistema de redes de computadoras	37
3.6.1 Métricas de desempeño en una red	38



Capitulo 4	
TEORÍA DEL MODELADO Y LA SIMULACIÓN	45
4.1 Sistemas, Modelos y Simulación.....	46
4.1.1 Simulación frente a otras técnicas de evaluación.....	47
4.1.2 Simulación en un sistema de eventos discretos.....	48
4.1.3 Componentes y organización de una simulación de un modelo de eventos discretos.....	49
4.2 Modelado en sistemas de eventos discretos.....	51
4.3 Modelos de simulación.....	53
4.3.1 Simulación orientada a sucesos.....	53
4.3.2 Simulación orientada a procesos.....	54
4.4 El proceso de la Simulación.....	56
4.5 Ventajas, desventajas y trampas de la simulación.....	58
4.6 Software de simulación.....	60
4.6.1 Clasificación del software de simulación.....	60
4.6.2 Elementos comunes de simuladores.....	61
4.6.3 Características deseables de software.....	62
 Capitulo 5	
METODOLOGÍA DE DISEÑO Y PLANEACIÓN DE REDES, MEDIANTE ANALISIS DE DESEMPEÑO POR TECNICAS DE SIMULACION	68
5.1 Introducción al diseño y planeación de redes.....	69
5.2 Metodología de diseño y planeación de redes mediante simulación y modelado.....	72
5.2.1 Definiendo el problema.....	74
5.2.2 Formulando la hipótesis.....	74
5.2.3 Definir las pruebas del ambiente.....	79
5.2.4 Desempeño de los experimentos.....	81
5.2.5 Establecer conclusiones.....	84
 Capitulo 6	
EVALUACIÓN DEL CASO DE ESTUDIO	90
6.1 Caso de estudio.....	91
6.1.1 Definiendo el problema.....	91
6.1.2 Formulando la hipótesis.....	91
6.1.2.1 Reunir los datos del problema (modelo).....	91
6.1.2.2 Desarrollo de la topología.....	92
6.1.2.3 Diseñando los flujos de trabajo.....	94
6.1.3 Definir las pruebas del ambiente.....	95
6.1.3.1 Diseñando los procesos.....	95
6.1.3.2 Construir el modelo.....	97
6.1.4 Desempeño de los experimentos.....	98
6.1.4.1 Medida del desempeño en el sistema real.....	98
6.1.4.2 Medidas de desempeño de la simulación.....	99

6.1.5 Estableciendo conclusiones de los resultados de la simulación....	100
6.1.5.1 Conclusiones del caso de estudio simulado.....	101
CONCLUSIONES Y RECOMENDACIONES	102
BIBLIOGRAFIA	105
GLOSARIO	107
FIGURAS Y TABLAS	115
APENDICE A	117
APENDICE B	131
APENDICE C	135
APENDICE D	137

# Capítulo 1

## SEGMENTACIÓN DE REDES

### INTRODUCCIÓN

Uno de los factores más involucrados en el congestionamiento de la red es el número de computadoras conectadas. Mientras la red se mantenga en una dimensión controlada, los parámetros pueden fijarse mejor y de esta manera lograr las configuraciones requeridas. Si los equipos sobrepasan en número la cantidad que se tiene para evitar una competencia desmedida por el derecho a transmitir, es necesario crear mecanismos que permitan mantener el tamaño de cada red de acuerdo a las posibilidades para controlarla. El problema de la transferencia de mensajes de gran longitud a través de la red se soluciona con la fragmentación de los mismos, de igual manera las redes grandes deben ser segmentadas para evitar la competencia excesiva al momento de la transmisión de mensajes. Existen en la actualidad una gran variedad de equipos intermedios como puentes, pasarelas, enrutadores y "switches" que, como equipo intermedio, coadyuvan al proceso de segmentación de redes. En este capítulo se exponen algunas de las ventajas que existen al utilizar estos equipos intermedios para minimizar el congestionamiento en las redes de computadoras, asimismo, al abordar el tema del enrutamiento, se analiza la importancia de crear un vínculo directo entre los mensajes enviados y los dispositivos intermedios por los que pasa el mensaje, para crear así una interacción transparente de las rutas óptimas que debe tomar el mensaje para evitar congestionamientos o por lo menos mantener esta incertidumbre más controlada. Las soluciones para segmentar las redes se abarcan en este capítulo y se hace referencia principalmente a los puentes por ser los dispositivos intermedios de uso más generalizado.

## 1.1 "SWITCHES".

En una red local Ethernet, la capacidad de la red tiene que ser compartida por todas las computadoras conectadas a la red. Las redes conmutadas por medio de "switches" crean circuitos virtuales durante el tiempo que dura la trama en la transmisión y recepción. El cambio de una red local Ethernet a una red conmutada consiste en el cambio del concentrador por un "switch" <sup>9</sup>.

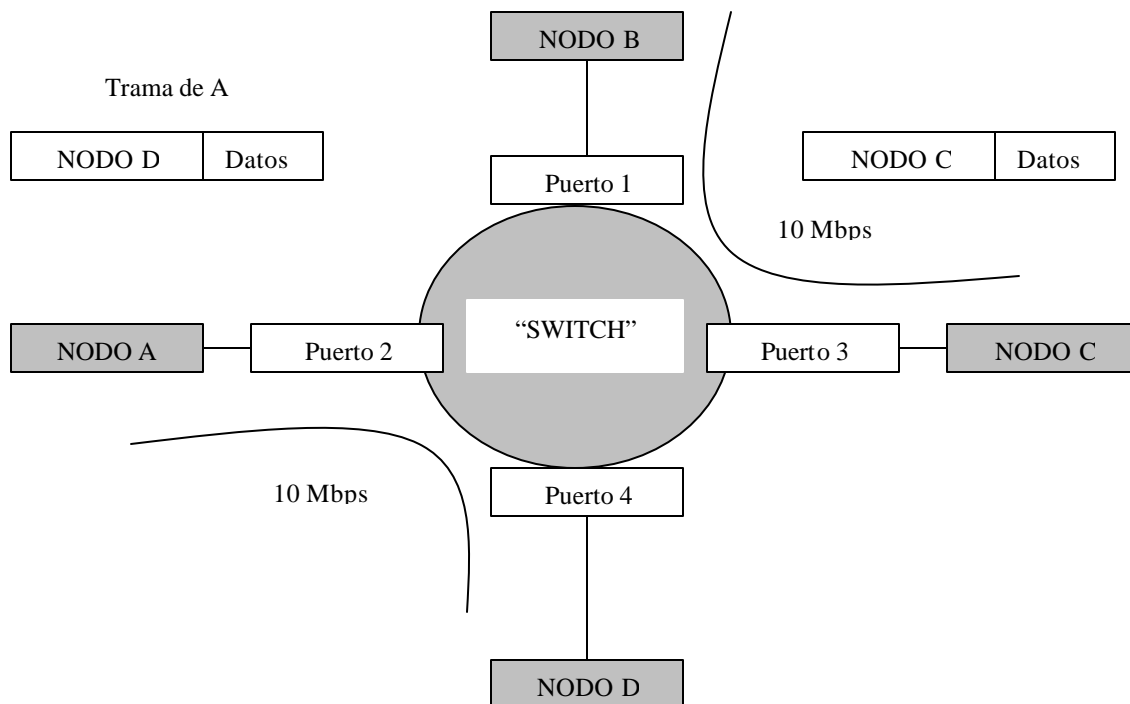


Figura 1.1  
Interconexión en un "switch" de 4 puertos.

Al igual que los puentes, los "switches" crean una tabla de ruteo donde se encuentran las direcciones de las estaciones que tienen conectadas a sus puertos.

Puerto	DIRECCIÓN
1	A
2	B
3	C
4	D

Tabla 1.1  
Tabla de retransmisión de un "switch".

Con la incorporación de "switches" en las redes se puede escalar una red de baja velocidad (10 Mbps) a una red de alta velocidad. Este hecho es posible debido a que el "switch" dedica a cada una de sus estaciones un canal exclusivo; es decir que si tenemos conectadas dos estaciones al "switch" en una red Ethernet por ejemplo, no se tendrán 10 Mbps sino 20 Mbps. En general la velocidad de la red contada, está en función del caudal ("throughput") que se especifica como sigue:

$$\text{CAUDAL ("TROUGHPUT")} = (\text{No. de puertos})(\text{Velocidad del canal}) / 2$$

En el caso de los puentes, el mecanismo empleado para la retransmisión de mensajes es de la forma de almacenamiento y reenvío. En el caso de los "switches " se utilizan otros dos mecanismos para el reenvío de mensajes.

- a). Almacenamiento y reenvío (Store and Forward).
- b). Corte a través (Cut - Through).
- c). Corte a través modificado (Cut - Through Modificado).

Estos mecanismos hacen más eficiente el reenvío de mensajes como lo veremos a continuación en una trama de Ethernet.

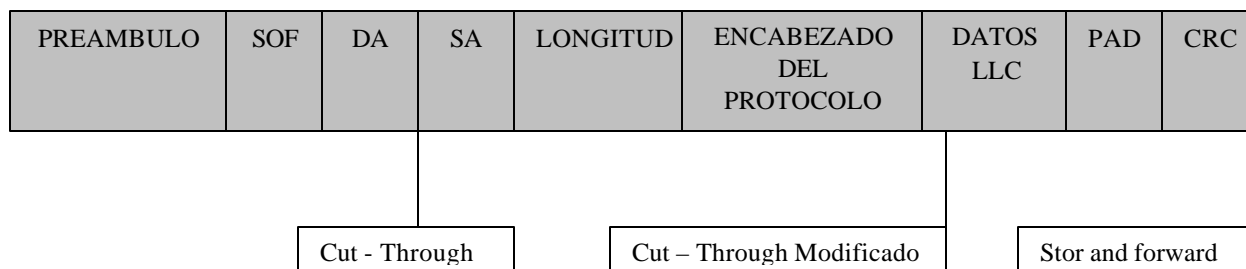


Figura 1.2  
Mecanismos para el reenvío de mensajes en un "switch".

- a) Almacenamiento y reenvío (Store and Forward).- Se analiza toda la trama y se verifica que no exista error. Una vez visto esto, el "switch " manda la trama a su destino por medio de su tabla de retransmisión.
- b) Corte a través (Cut - Through).- En este mecanismo de reenvío el "switch" sólo revisa a donde va dirigido el mensaje sin verificar que esté bien o mal.
- c) Corte a través modificado (Cut - Through Modificado).- En este caso se revisa la trama hasta los primeros 64 bytes de datos y sino se detecta error es muy posible que no existan errores más adelante; de esta forma se hace un balance de compromiso entre las verificaciones de a) y b).

Un "switch" o conmutador de red local es un dispositivo que consiste típicamente de varios puertos que conectan segmentos de una red local (ya sea esta una red Ethernet o de otro tipo) o bien a un puerto rápido (tal como una red Ethernet o FDDI a 100-Mbps) o incluso a una red ATM a 155 Mbps. Un "switch" tiene un ancho de banda dedicado a cada puerto y cada puerto representa un segmento independiente. Los "Switches" son construidos por los diseñadores de redes para mejorar el desempeño de la red.

Cuando un "switch" en una red local empieza a trabajar por vez primera, construye una tabla que asocia la dirección MAC de cada dispositivo dentro de la red con el número de puerto al cual está conectado dicho dispositivo. En términos de la figura 1.3 se puede decir que cuando la computadora A en el puerto 1 necesita transmitir algo a la computadora E que se encuentra en el puerto 5, el "switch" envía tramas desde el puerto 1 al puerto 5.

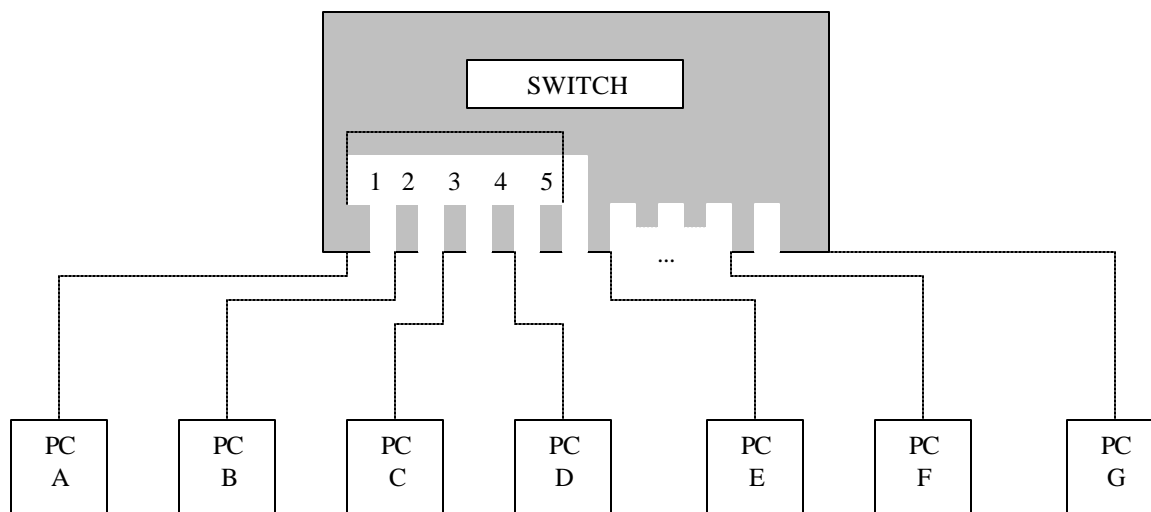


Figura 1.3  
Conexión de dos terminales mediante un "switch" en una LAN.

Cuando un dispositivo que es parte de la red local conmutada quiere mandar un mensaje a una dirección que no se encuentra dentro de la tabla del "switch" (por ejemplo cuando el dispositivo manda un mensaje de difusión general o multidifusión), el "switch" manda un mensaje a todos los puertos excepto para el puerto en el que se origino el mensaje, esta técnica es conocida como inundación ("flooding"). Debido a que los "switches" trabajan como puentes transparentes, tienen un conocimiento de su vecindario. Una red diseñada sólo con "switches" aparece como una red de topología plana que consiste de un dominio de difusión general simple; por consiguiente estas redes suelen padecer de los mismos problemas de los que padecen las redes que trabajan con puentes. No obstante en el caso de las redes computadas a diferencia de lo que ocurre con las redes a base de puentes, se puede tener un mejor escalamiento cuando se trabaja con redes virtuales.

La diferencia fundamental entre un "switch " y un enrutador es que el "switch " opera en la capa 2 del modelo de referencia OSI y el enrutador opera en la capa 3. Esta diferencia afecta la forma en que los "switches " y los enrutadores responden al tráfico. Una buena forma de comparar a los enrutadores con los "switches " es por medio de los siguientes puntos:

1. Ciclos (Bucles).
2. Convergencia.
3. Difusión General.
4. Conectividad entre redes locales.
5. Seguridad.
- 6 Dependencia de Medios.

1. Ciclos. Las topologías de red local conmutada son susceptibles a los *ciclos* tal como se muestra en la figura:

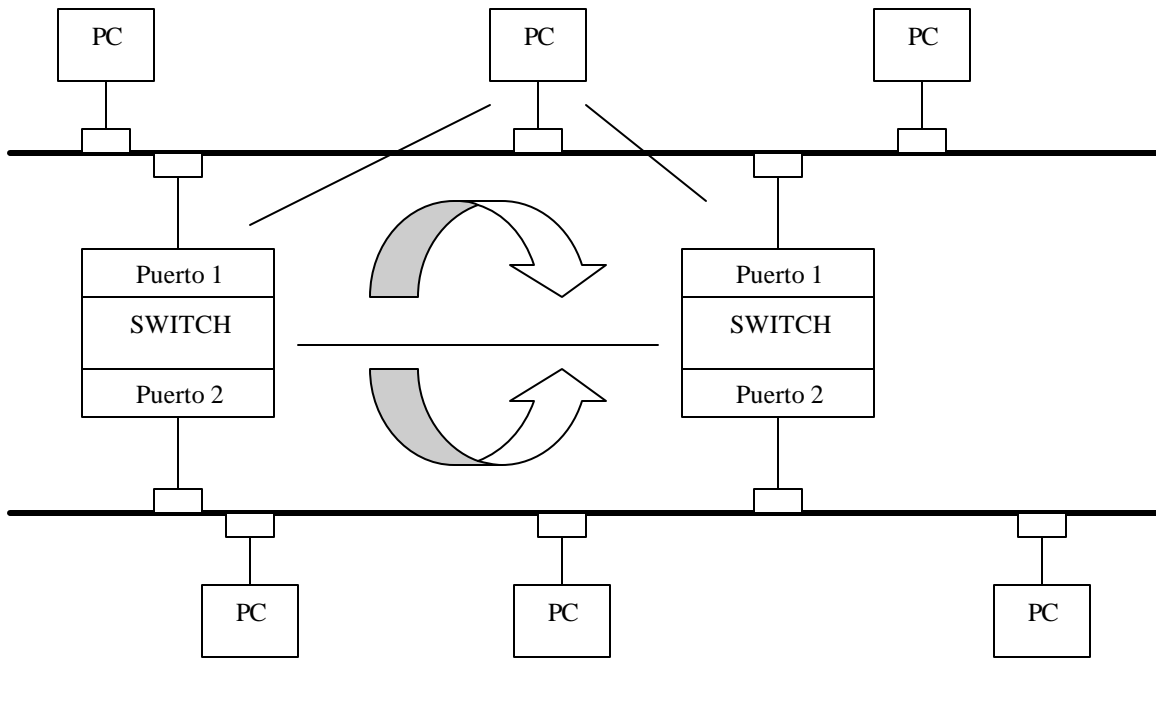


Figura 1.4  
Red LAN switchcada con loops.

En la figura 1.4 podemos apreciar como un mensaje puede quedar atrapado en un ciclo al confundir a que puerto tiene conectada la terminal; aspecto en el que ahondaremos más adelante. Para evitarlo o romper estos *ciclos*, en una red conmutada se hace uso del protocolo del árbol de apertura (Spanning Tree Protocol), que mediante el algoritmo Spanning Tree puede construir una topología sin *ciclos*.

2. Convergencia. En un conmutado transparente los "Switches" realizan decisiones de topología local basándose en el intercambio de PDU's.

3. Difusión General. Los "Switches" no filtran los mensajes de difusión general, multidifusión o cualquier trama de dirección no conocida. La carencia de filtrado puede ser un problema serio en las modernas redes distribuidas donde los mensajes de difusión general se usan para resolver direcciones y describir de una forma dinámica recursos de la red tal como los servidores de archivo. Los mensajes de difusión general originados en cada segmento, son recibidos por cualquier computadora que se encuentre en la red. La mayoría de los dispositivos rechazan los mensajes de difusión general por que son irrelevantes, esto significa que se desperdician grandes cantidades del ancho de banda por la transmisión de difusión general. En ciertas circunstancias, la circulación de mensajes de difusión general puede saturar a la red, dejando sin ancho de banda a las aplicaciones de datos.

4. Conectividad entre redes. Las redes de redes se componen de segmentos separados físicamente, pero se considera que forman una red grande (por ejemplo, una subred IP). Este comportamiento es inherente a la manera en que los "Switches" trabajan.

5. Seguridad. Se dispone de los mecanismos para que los "Switches" puedan ser usados para crear redes más seguras. Los "Switches" pueden usarse para proporcionar acceso a la información mediante filtros basados en la dirección de destino y fuente, el tipo de protocolo y la longitud de la trama.

6. Dependencia de Medios. Deben considerarse dos factores relacionados con los medios mixtos en las redes de redes. El primero es la unidad máxima de transferencia (MTU) que difiere en diversos medios de la red.

En la tabla 1.2<sup>10</sup> se muestran las especificaciones de los tamaños de trama máximos y mínimos para diversos tipos de red.

Medios	Mínimo Válido	Máximo Válido
Ethernet	64 octetos	1518 octetos
Token Ring	32 octetos / 4Kb normal	16 Kb octetos
FDDI	32 octetos	4400 octetos
ATM	53 octetos	1518 octetos
ATM/IP Clásico	64 octetos	9180 octetos
HDLC consecutivo	14 octetos / 4.5 Kb normal	Ningún limite

Tabla 1.2  
Tamaños mínimos y máximos de mensajes según el tipo de red.



Cuando las redes locales de medios no semejantes se intercambian información, los anfitriones deben usar el MTU que es el denominador común mas bajo de todas las redes locales. Estos requerimientos tienen una repercusión en el rendimiento ("throughput") que compromete seriamente el desempeño de una red FDDI por ejemplo. La mayoría de los protocolos de la capa tres pueden fragmentar y reensamblar mensajes que son demasiado grandes para la subred en particular.

El segundo factor importante a considerar es debido a que los "switches" operan en la capa 2 y deben utilizar una función de traducción para intercambiar medios no semejantes; la función de traducción puede resultar un problema serio tal como una conversión no canónica contra canónica en el formato MAC de una red Token Ring a una red Ethernet.

### **1.1.1 Mecanismo de control de la congestión.**

El control de la congestión comprende la habilidad del "switch" para permitir que varios puertos de entrada requieran una conexión simultánea con un mismo puerto de salida. El control de la congestión pasivo está basado en "buffers" que almacenan los datos de transmisión, hasta que el puerto de salida esta libre. La técnica de control de congestión activa detecta el tráfico de los puertos fuente usando las señales de tráfico Ethernet para especificar la ocurrencia de colisión. En los "switches" con bus compartido, los datos conmutados pasan a través de un bus común; por lo tanto, la capacidad del bus determina el máximo rendimiento del "switch". La tecnología actual puede soportar un bus de 1 Gbps.

Eventualmente, esto provoca embotellamiento. Porque un puerto simple 100Base-T provee un ancho de banda de 100 Mbps, un bus de 1 Gbps en teoría soporta puertos de 10-100 Mbps. En la actualidad solo se logra un 70 por ciento de la transmisión de datos con éste ancho de banda. Esto significa que el rendimiento del "switch", comenzará a degradarse rápidamente cuando más de siete u ocho puertos 100Base-TX estén operando a su máxima capacidad, porque a ese nivel, el bus de 1Gbps estará muy cerca de la saturación. La respuesta a esa limitación es un "switch" de matriz, llamado así, porque proporciona a cada puerto de entrada su propia trayectoria de salida, lo cual teóricamente significa que es infinitamente escalable la capacidad de transferencia de datos. Desafortunadamente, los "switches" de matriz son mucho más caros que los "switches" con bus compartido. Además se complica el monitoreo de red porque no existe una trayectoria de tráfico común, y sólo se puede observar una conversación a la vez.

## 1.2 Puentes.

Un puente se usa para:

- Conectar dos LAN's ó Extender una LAN.
- Hacer más eficiente el manejo del tráfico.
- Por seguridad.

Un puente es un dispositivo que conecta dos redes locales separadas para crear lo que aparenta ser una sola LAN<sup>9</sup>. Son dispositivos de hardware cuya función principal es extender las redes. Cuando se habla de una conexión de alto orden se hace referencia a un enlace hacia los niveles superiores y decimos que se trata de una conexión de bajo orden, cuando desde nuestra posición miramos hacia los niveles inferiores. Los puentes son dispositivos que copian la señal sólo en bajo orden.

Existen puentes digitales y analógicos, según el tipo de señal que puedan manejar y esto por supuesto depende del lugar donde van a ser instalados en el enlace. En general los puentes no son elementos inteligentes, son muy sencillos y de costos razonables, pueden conectarse en cascada, ampliando de esta forma su capacidad de ramificación. En una red Token Ring, son muy sencillos y de costos razonables, pueden conectarse cuando ya son muchas las estaciones conectadas en la red, ésta se segmenta porque la competencia por el "Token" es demasiada.

Todo mensaje que circule por el segmento, llega al puente que se considera en este caso como una estación (el mensaje se da de la forma "Broadcast"). El puente anota las direcciones de cada PC contenida en un segmento. Se dice entonces que un puente filtra, ya que si un mensaje llega a él, este se encarga de ver si la máquina de destino se encuentra en el mismo segmento (de tal forma que lo ignora) o si tiene que dirigirlo para otro segmento.

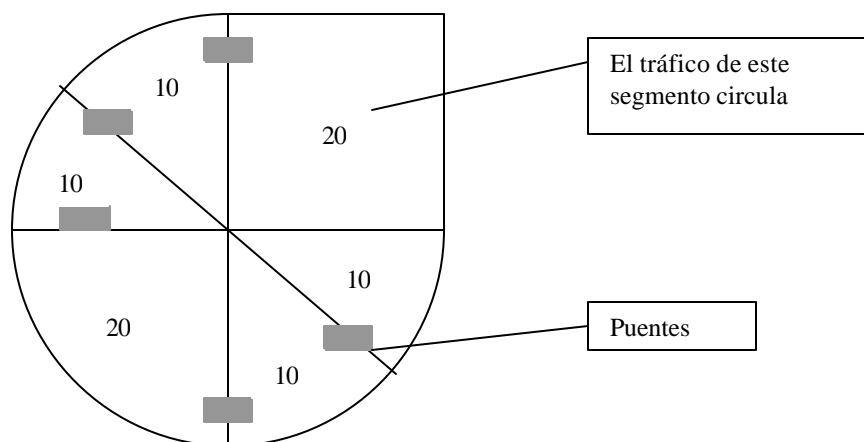


Figura 1.5  
Red Token Ring segmentada mediante el uso de puentes.

La mayoría de las redes de área local operan dentro del contexto de redes de mayor tamaño, Y como es indispensable que en estas grandes organizaciones se comuniquen las diversas redes departamentales es necesario incluir pasarelas y puentes. Una pasarela consta del hardware y software necesarios para que dos redes tecnológicamente diferentes se comuniquen entre sí. En contraste, un puente se utiliza para enlazar dos redes tecnológicamente similares. Una red debe respetar el nivel a la cual se le segmenta. Por ejemplo, en el caso de una pasarela, como ésta se ubica por arriba del nivel MAC, tiene que hacer forzosamente una traducción de protocolos para que se puedan entender las diferentes tecnologías. El enlace por puente a nivel eléctrico/mecánico requiere equivalencia eléctrica/mecánica; el enlace en el nivel MAC requiere equivalencia MAC, es decir que un puente se utiliza cuando el enlace entre las LAN no requiere la conversión de protocolos.

### **1.2.1 Puentes a nivel de MAC: 802.1 – interconexión en redes.**

Un puente en el nivel MAC es un dispositivo que procesa protocolos en el nivel MAC y es transparente para LLC y para los protocolos de niveles superiores. La información del nivel MAC controla cualquier decisión de envío. Una unidad de interconexión (IWU) es un dispositivo que procesa protocolos del nivel de red de OSI que operan directamente sobre el nivel LLC. En este esquema las decisiones de envío están basadas en direcciones del nivel de red. Con puentes a nivel MAC se pueden conectar en particular todo tipo de redes de área local con estándar 802 del IEEE. La red de área local enlazada por puente hace posible la interconexión de estaciones enlazadas a LAN separadas como si estuvieran conectadas a una misma red local. Una red de área local enlazada por puentes puede ofrecer:

- La interconexión de estaciones conectadas a redes de área local 802 de diversos tipos.
- Un incremento efectivo en el enlace físico, en el número permisible de conexiones o en el desempeño total de una red local.
- Partición del soporte de la red local física por razones administrativas de mantenimiento, de tráfico o de seguridad.

Las redes de área local 802 del IEEE autónomas se interconectan a través de puentes a nivel MAC como se muestra en la figura 1.6.

Un puente que conecta dos redes locales recibe una trama de datos de una red local y le da un nuevo formato para ajustarla a los requisitos de la otra red local. Por lo tanto, un usuario final de una red Token Ring podría comunicarse directamente con un usuario final en un bus que opera con el método de acceso al medio CSMA/CD. Un puente a nivel MAC sólo realiza conversiones de protocolo dentro del alcance de 802.

Los elementos primarios de la operación del puente son el reenvío y la filtración de tramas. El puente almacena la información para tomar decisiones de filtración. El puente reenvía las tramas sólo a nivel MAC de las redes locales enlazadas que están conectadas a los puertos del puente. Se conserva el orden de las tramas que se reciben en un puerto y que se transmiten en otro.

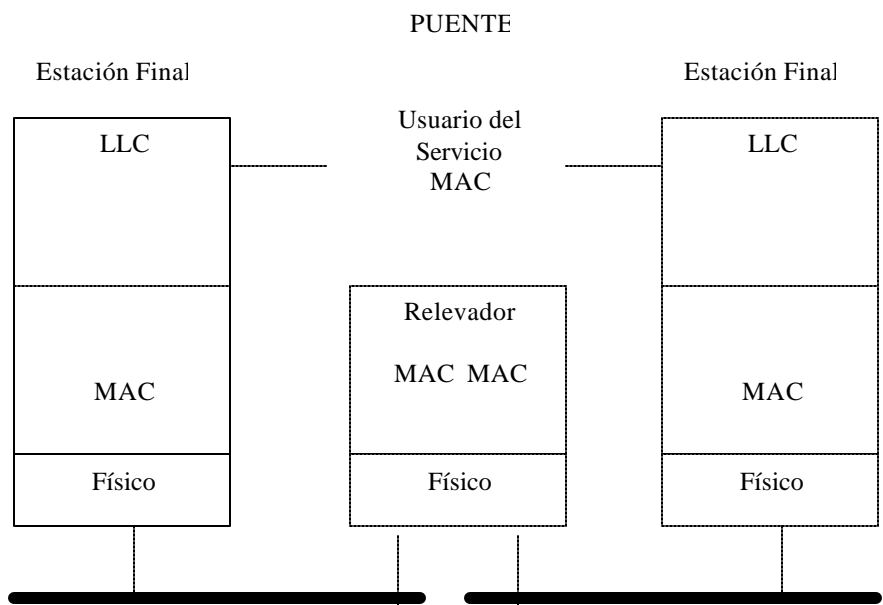


Figura 1.6  
Enlace por puentes a nivel de MAC.

La función de filtración opera para impedir la duplicación de tramas y para suprimir tráfico innecesario en las redes de área local. El puente debe tener los recursos suficientes para utilizar y mantener una configuración explícita de la información de filtración estática, de la adquisición automática de información de filtración dinámica a través de la observación del tráfico, de la caducidad de información de filtración que se ha obtenido en forma automática y del cálculo y configuración de la topología de la red de área local enlazada por puente.

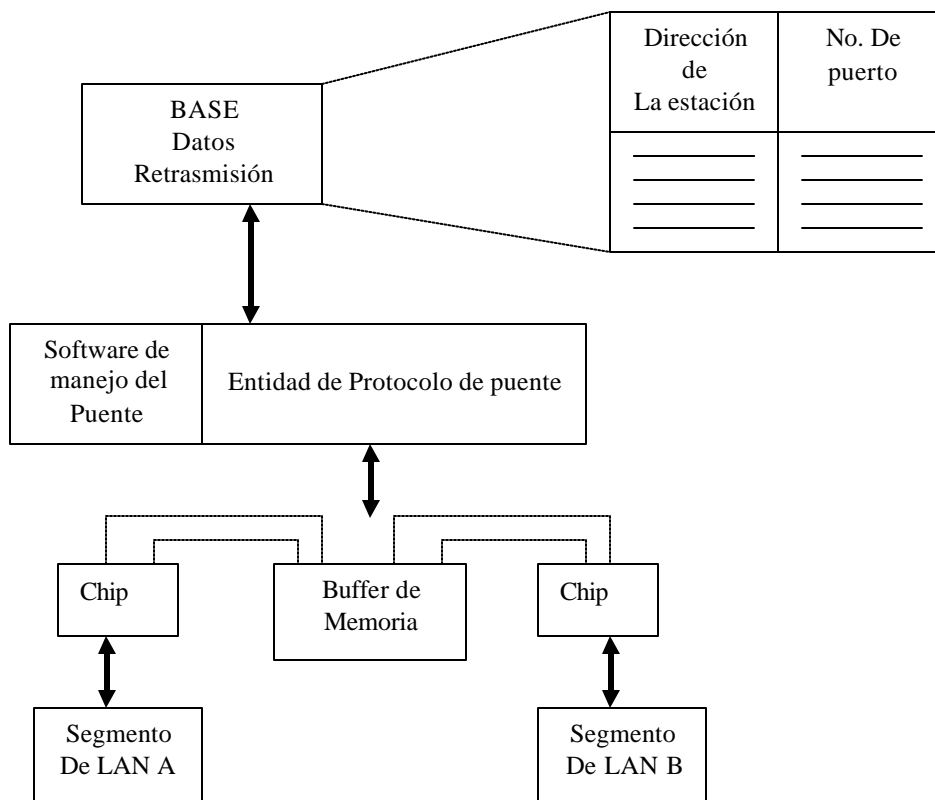
Un puente añade un nivel de jerárquico a una conexión entre redes porque conecta dos segmentos de una red iguales o distintos. Podemos ver un puente como un clasificador de correo que inspecciona las direcciones de los mensajes y los coloca en la red adecuada. Cada segmento de red puede ser distinto (Ethernet Token Ring, Arcnet, etc.) Las funciones del puente y el enrutador son distribuir el tráfico de una red entre los segmentos de una red local.

Con el puente podemos conectar dispositivos que utilicen protocolos diferentes, pero el nivel de enlace de datos no sabe nada sobre el mejor camino hacia un cierto destino; no existe ninguna forma de enviar mensajes a un segmento de red de modo que alcancen su destino de la forma más rápida o eficiente. Esa es la función de un Enrutador. No obstante los puentes ofrecen filtrado.

## 1.2.2 Estructura de un puente.

Un puente crea una tabla o directorio de ruteo<sup>11</sup>. En esta tabla se encuentra la dirección de la estación y el número de puerto al cual el puente tiene conectada dicha estación.

Con los puentes se incluyen nuevas ventajas en las redes, pero también tiene inconvenientes, uno de ellos por ejemplo es la generación de LOOPS. Tenemos 2 Segmentos conectados a través de 2 puentes; Se utiliza esta configuración por si falla un Puente, pero tiene el inconveniente que se puede generar un LOOP (esto sucede cuando el puente no sabe a que puerto pertenece la información y entonces la trama queda circulando al infinito). Para evitar esto la arquitectura es reconfigurada analógicamente para formar un árbol de expansión (Tree Spanning).



( A )

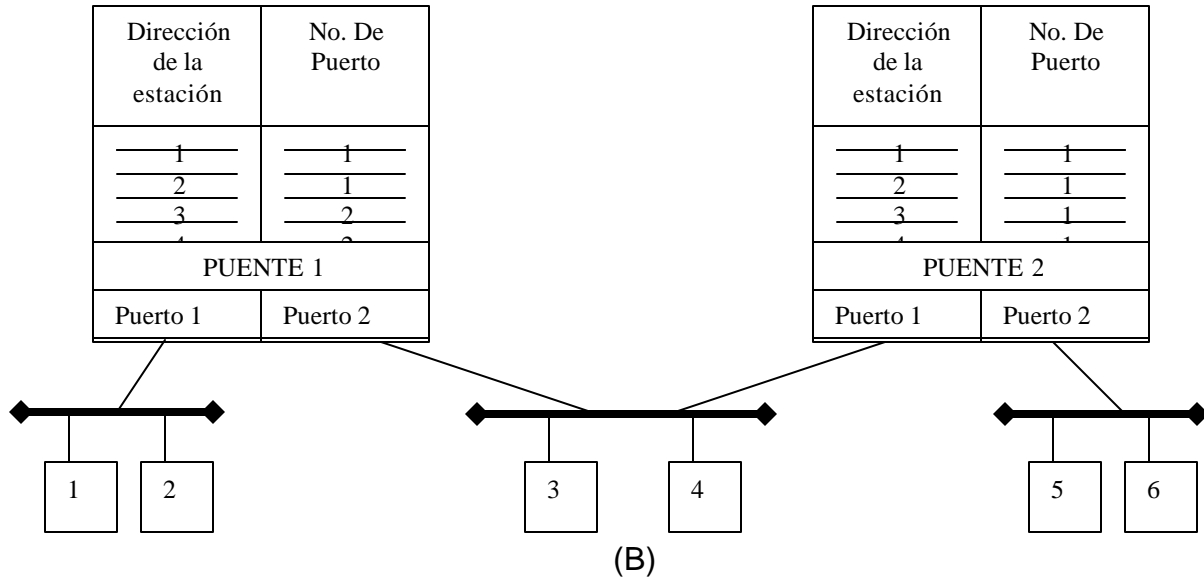


Figura 1.7  
 A) Comunicación entre dos segmentos. B) Cada puente crea una tabla de ruteo donde sabe a que puertos tiene conectadas las estaciones.

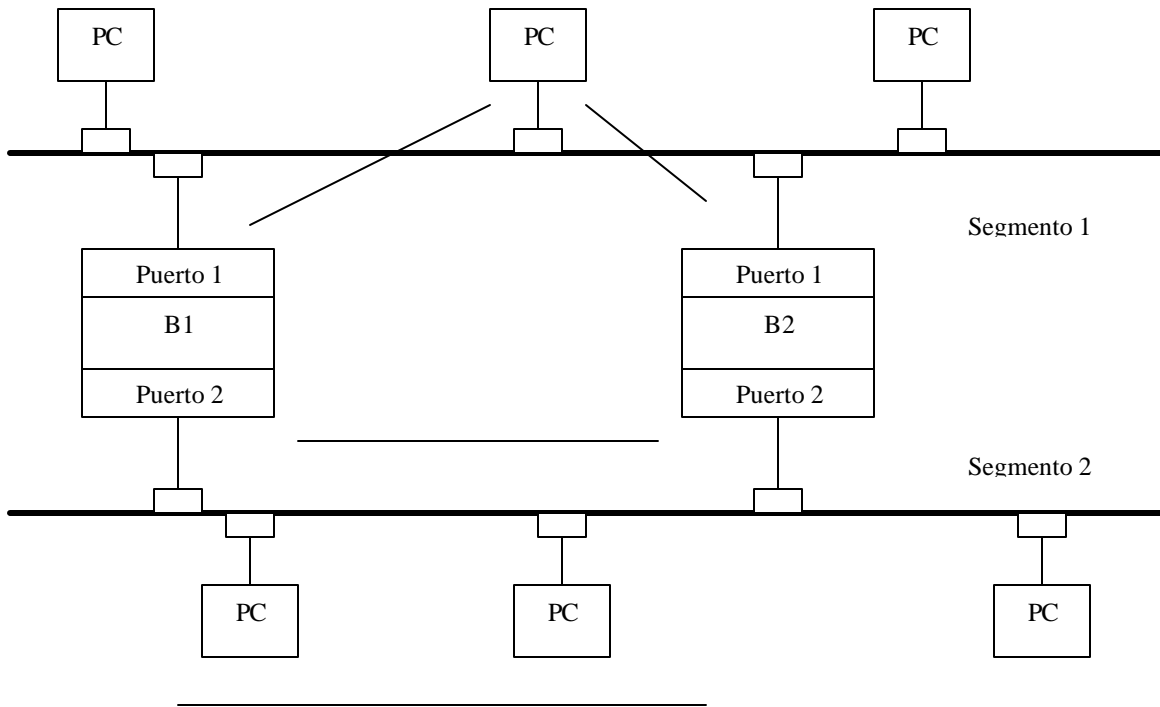


Figura 1.8  
 Formación de "LOOP's" en los puentes.

Para eliminar este inconveniente se utilizan los algoritmos de puenteo. Existen dos algoritmos de puenteo: Spanning Tree (utilizado en las redes Ethernet) y Source Routing (utilizado en las redes Token Ring).

### **1.2.3 Árbol de expansión (SPANNING TREE).**

El árbol de expansión es un protocolo usado por los puentes para quitar ligas redundantes de las redes Ethernet. Todos los dispositivos del protocolo de expansión envían tramas a cada uno de los puertos y esperan tramas que se envían desde otros dispositivos. El protocolo permite a los dispositivos del árbol de expansión comunicar información de identificación para que de esta forma cada dispositivo pueda eliminar la liga que es redundante.

La creación de ligas redundantes en su mayoría, resulta útil en situaciones de emergencia; por ejemplo, si la liga principal de un archivo del servidor es eliminada por la falta de cuidado del administrador del sistema, la liga redundante puede introducirse al servidor mientras que la liga principal se encuentra en reparación.

El árbol de expansión resuelve esta situación de manera automática. En la red se verifica la validez de todas las ligas y si alguna de estas ligas falla, la siguiente liga en el segmento reemplaza a la liga que no está o que ya no es válida.

El árbol de expansión está diseñado para un conjunto de puentes que tienen un "loop" en la red. Las ligas que se toman como reemplazos son siempre de prioridad más baja. Se debe ser muy cuidadoso cuando se coloca el árbol de expansión en un dispositivo multipuerto (tal como un "switch" Ethernet) que es usado en las redes locales virtuales. La mayoría de los vendedores no permiten que el árbol de expansión coexista con redes virtuales.

### **1.3 Enrutamiento de origen.**

Las redes Token Ring utilizan una rutina especial de enrutamiento de origen que no sólo le indica al puente a dónde debe ir el mensaje, sino el cómo llegar ahí. Esto puede ser muy útil para quitar carga de procesamiento a los nodos intermedios. En el enrutamiento de origen, los mensajes guardan la información de envío. El puente no mantiene tablas de enrutamiento; envía los mensajes a los segmentos de red local basándose en la dirección del mensaje.

Los puentes que realizan enrutamiento de origen utilizan un método de exploración para determinar la ruta que debe seguir un mensaje para llegar a un dispositivo. El puente con enrutamiento de origen es simplemente un dispositivo de reenvío que conoce las direcciones de los puentes restantes. La información sobre el mejor camino de envío va dentro del mensaje.

Una estación que necesite enviar un mensaje a una estación o servidor en otro segmento de red envía un mensaje explorador por toda la red. Si hay varios puentes en la red, llegarán varias copias del mensaje al dispositivo de destino, ya que cada puente envía su propia copia del mensaje explorador. La estación de destino recibe todas las copias y determina la mejor ruta en la interconexión de

redes basándose en la información almacenada en los mensajes sobre el camino a la estación de envío.

Entonces comunica la información sobre el mejor camino a la estación de envío, que lo almacena para su uso futuro y lo pondrá en los mensajes que envía a ese destino.

En el caso de las redes Token Ring como se mencionó anteriormente, el algoritmo de puenteo que se usa es el "Source Routing" e incluye en la trama de Token Ring una serie de campos para este fin.

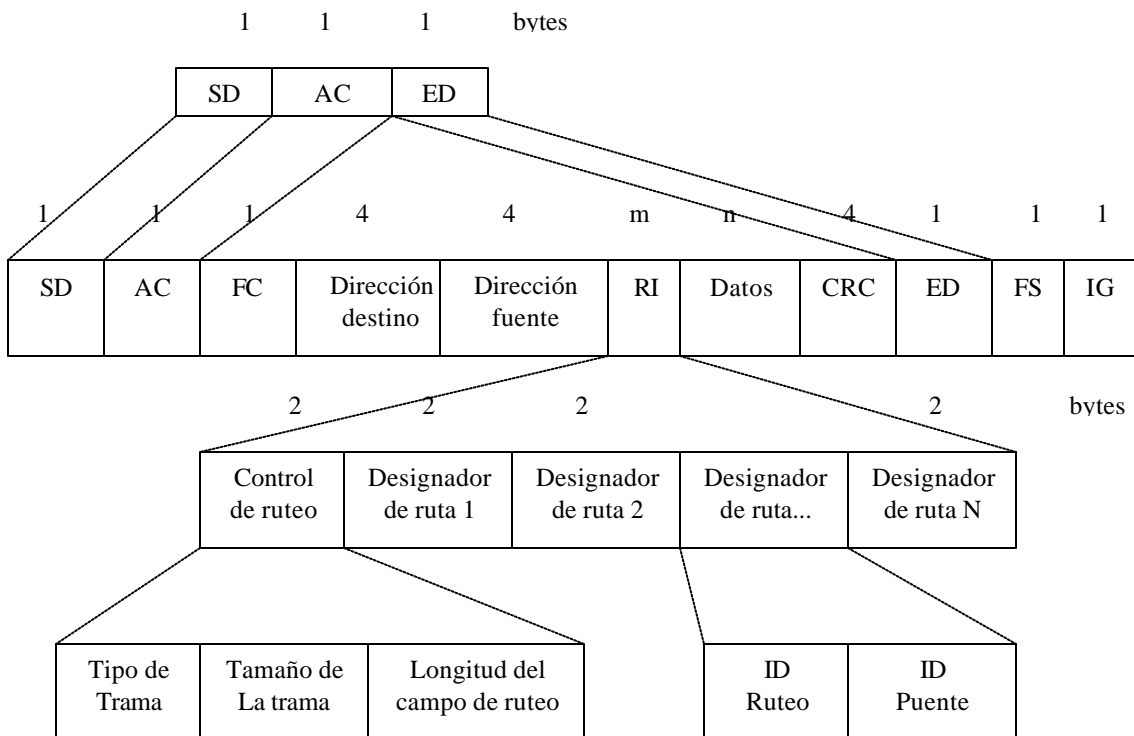


Figura 1.9  
Trama de Token Ring, incluido el campo de información de ruteo.

Los dos tipos de algoritmos de puenteo, se basan en las tablas que poseen los puentes. Estas tablas les permiten conocer a que estación se lleva la información.



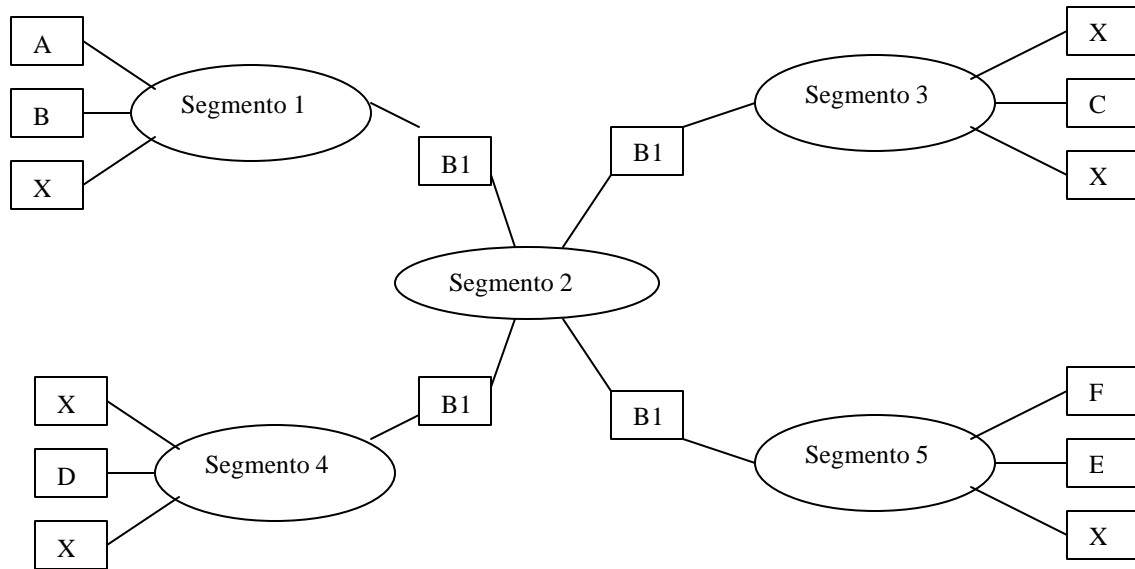


Figura 1.10  
Caminos posibles de un mensaje con enrutamiento de origen.

#### TABLA DE RUTEO EN LA ESTACIÓN B

Destino    A = Segmento 1  
 .  
 .  
 .  
 D = Seg1, B1, Seg2, B3, Seg4

#### TABLA DE RUTEO EN LA ESTACIÓN A

Destino    B = Segmento 1  
 C = Seg1, B1, Seg2, B2, Seg3  
 .  
 .  
 E = Seg 1, B 1, Seg2, B4, Seg5

#### TABLA DE RUTEO EN LA ESTACIÓN E

Destino    A = Seg5, B4, Seg2, B 1, Seg 1  
 .  
 .  
 C = Seg5, B4, Seg2, B2, Seg3

Podemos definir al encaramamiento como un proceso mediante el cual tratamos de encontrar un camino entre dos puntos de la red: el nodo origen y el nodo destino. Debemos tener también en cuenta la capilaridad de las redes (conexión

de los nodos con los terminales de usuario, por lo que no se trataría de buscar un camino entre dos nodos, sino entre dos terminales. En realidad, resolviendo lo Primero tenemos resuelto lo segundo, por lo que enunciaremos nuestro problema como la búsqueda de un camino de conexión entre dos nodos de la red.

El objetivo que se persigue es encontrar las mejores rutas entre pares de nodos  $j-k$ . Para ello tendremos que establecer lo que se entiende por mejor ruta y la métrica que se emplea para determinarla:

a) Mejor Ruta. Por mejor ruta se entiende aquella que cumple alguna de estas condiciones.

- presenta el menor retardo medio de tránsito,
- consigue mantener acotado el retardo entre pares de nodos de la red ( $T_{jk} < T_0$ ),
- ofrece transferencias efectivas independientemente del retardo medio de tránsito,
- ofrece el menor costo.

b) Métrica de la Red. Citaremos dos de ellas:

- Número de saltos necesarios para ir de un nodo a otro. Esta métrica no se comporta de forma óptima, pero sí ofrece buenos resultados y es empleada con bastante frecuencia. La distancia es igual a 1 para todos los nodos.
- Retardo de Tránsito entre nodos vecinos. En este caso la distancia se expresa en unidades de tiempo (ms), y no es constante a lo largo del tiempo sino que depende del tráfico que soporta el nodo.

Nos centraremos en redes de conmutación de mensajes, tanto en modo datagrama como en modo de circuito virtual.

a) Red en modo circuito virtual.

Si la red funciona en modo circuito virtual generalmente se establece una ruta que no cambia durante el tiempo de vida de ese circuito virtual ya que esto es lo más sencillo para preservar el orden de los mensajes. El encaminamiento se decide por sesión y no se cambia a menos que sea imprescindible (ej. caída de un enlace). Cuando esto ocurre se busca inmediatamente otra ruta, pero este cambio al propagarse por la red ocasiona que los nodos tarden en enterarse.

b) Red en modo datagrama.

En una red funcionando en modo datagrama no se garantiza el ordenamiento final de los mensajes, se puede cambiar el criterio de encaminamiento por cada mensaje que se envía, esto da origen a un menor número de Problemas.

El problema no es trivial, como puede parecer a simple vista para redes reales, puesto que hay que tener en cuenta los cambios de carga de los enlaces y los cambios de configuración (debidos a altas y bajas, a averías en los nodos, etc.). En este punto conviene mencionar que el encaminamiento busca el camino óptimo, pero como el tráfico varía con el tiempo, el camino óptimo también dependerá del instante en que se observa la red. Los protocolos serán los encargados de ocultar la red y comprobar que las condiciones impuestas se verifican siempre.

Por esta razón, el encaminamiento debe proveer a la red de los mecanismos para que ésta pueda reaccionar ante situaciones como:

- Variabilidad del tráfico: se tienen que evitar las congestiones de la red.
- Variaciones topológicas: como las mencionadas arriba: caídas de enlaces, caídas de nodos, altas y bajas...
- Cambios en la QoS (Quality of Service): a veces se pide un servicio que no puede ser soportado.

En las redes reales se tiene un sistema distribuido, donde cada nodo debe tener una cierta capacidad y donde se produce un procesamiento independiente de cada nodo (no hay proceso central). También las redes operan muchas ocasiones como un sistema asíncrono, en el sentido de que no hay un momento determinado para que ocurran las cosas (un nodo transmite cuando le llega información, y esto sucede a su vez cuando el usuario decide mandarla).

### **1.3.1 Propiedades de un algoritmo de encaminamiento.**

Para que un algoritmo de encaminamiento real funcione de forma aceptable debe cumplir con las siguientes propiedades:

- Corrección. Queremos que el mensaje llegue precisamente al nodo al que lo mandamos.
- Simplicidad. Debe aportar soluciones sencillas. Esto es útil para red reales (grandes). Los protocolos más simples son los más utilizados (RIP).
- Robustez. Debe comportarse correctamente frente a posibles problemas (Caída de enlaces) que deben haber sido previstos de antemano.
- Estabilidad. Es muy importante que se cumpla. El procedimiento debe converger antes de que la red cambie de estado (caída de nodo, alta de usuario, etc.). Cuando esto ocurre, se recalculan de nuevo las rutas, donde los nodos deben llevar a cabo acciones coherentes que conduzcan a situaciones estables.

- Equidad o Justicia. Debe tratar a todos los usuarios de la misma manera, aunque de hecho sabemos que no todas las aplicaciones son iguales.
- Administración. Supone tener información de lo que ha hecho la red para que en el caso de que ocurran "cosas raras" sea posible corregirlas.
- Escalabilidad. Debe tener un comportamiento óptimo sea cual sea el número de nodos (incluso si este aumenta mucho).

### 1.3.2 Métodos de enrutamiento.

En este apartado abordaremos primero la estructura de un nodo de conmutación de mensajes, lo cual nos valdrá para hacer una clasificación posterior de los métodos de encaminamiento atendiendo a la forma en la que los nodos recogen y distribuyen la información que les llega de la red y a otros factores.

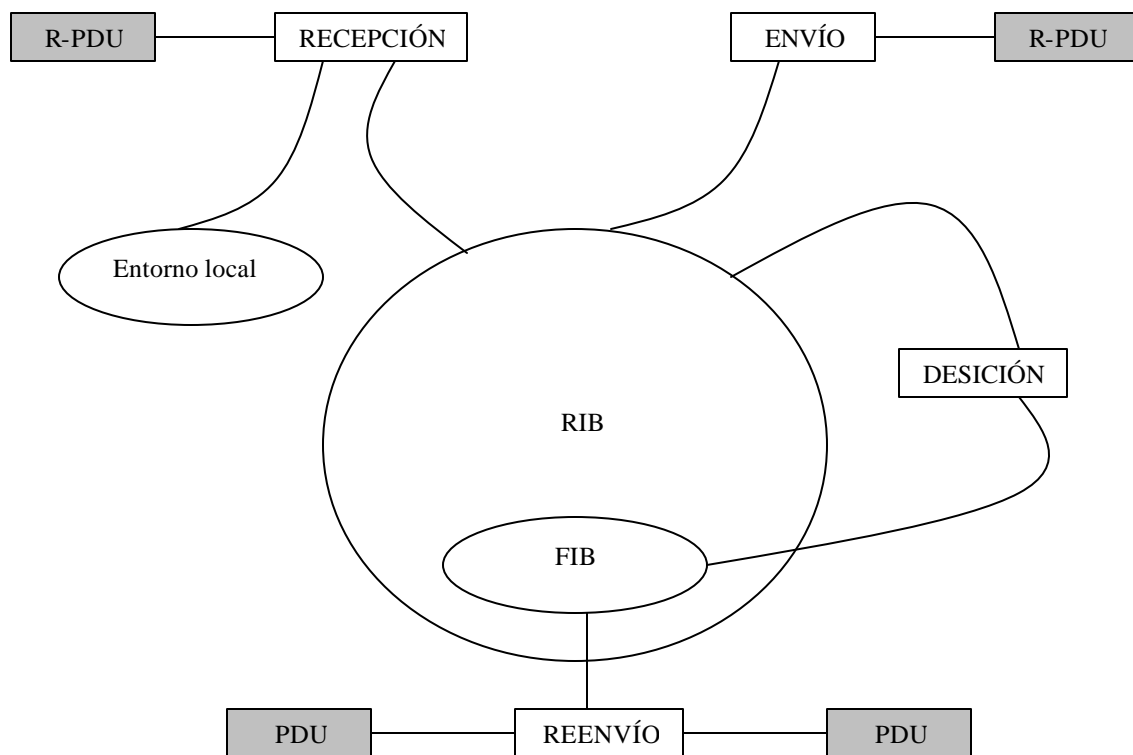


Figura 1.11  
Esquema de un nodo.

Podemos tomar como representación de un nodo la figura 1.11.

- FIB: Base de Información de Reenvío (Forward Information Base) Es la tabla de encaminamiento que se consulta para hacer el reenvío de los mensajes generados por los usuarios (los PDU representan estos mensajes).

- Entorno local: Contiene la información de lo que el nodo ve (memoria disponible, enlaces locales, etc.), más la que hay que proporcionarle.
- PDU (Protocol Data Unit): Unidad fundamental de intercambio de información para un nivel determinado (a veces se indica explícitamente el nivel poniendo N-PDU, o PDU de nivel N), como nivel de enlace de red, etc. Son llamadas también tramas.
- R-PDU (Routing-PDU): Mensaje de control. Estos mensajes los mandan otros nodos con información sobre la red (no son datos de usuario). Por ejemplo, se manda información de que el nodo sigue activo, y también las distancias a otros nodos (vector de distancias).
- Decisión: Contiene la lógica necesaria para decidir si la tabla de encaminamiento debe ser cambiada.
- RIB: Base de Información de Ruteo (Routing Information Base): Es la base de información de encaminamiento que se consulta para decidir y formar la FIB. La información de la RIB se consigue mediante interacción con el entorno local de cada nodo (cada nodo observa sus enlaces) y mediante la recepción de los R-PDU de información de control procedentes de otros nodos vecinos que informan del conocimiento que estos nodos tienen sobre el estado de la red. A su vez, con la información obtenida por la RIB, ésta manda sus PDU de control para informar del conocimiento del estado de la red que el nodo tiene a los demás nodos.

### 1.3.3 Clasificación de los métodos de enrutamiento.

Los métodos de encaminamiento los podemos clasificar en función de: El procedimiento de encaminamiento o por las tablas de encaminamiento empleadas.

Los procedimientos de encaminamiento pueden ser:

- **Estático**

Las tablas de encaminamiento de los nodos se configuran de forma manual y permanecen inalterables hasta que no se vuelve a actuar sobre ellas. La adaptación a cambios es nula. Tanto la recogida como la distribución de información se realiza por gestión (se realiza de manera externa a la red), sin ocupar capacidad de red. El cálculo de ruta se realiza fuera de línea (en una máquina específica) y las rutas pueden ser las óptimas al no estar sometido al requisito de tiempo real. Este tipo de encaminamiento es el óptimo para topologías en las que solo hay una posibilidad de encaminamiento (topología en estrella).

- **Cuasi estático**

Este encaminamiento, es igual que el estático pero en vez de dar una sola ruta fija, se dan además varias alternativas en caso de que la principal no funcione, de ahí que tenga una adaptabilidad reducida.

Este tipo de encaminamiento puede dar lugar a situaciones incoherentes, ya que no se enteran todos los nodos de los problemas de la red, sino sólo los adyacentes a los problemas.

- **Centralizado**

En este tipo de encaminamiento, todos los nodos son iguales salvo el nodo central, que recoge la información de control de todos los nodos y calcula la FIB (tabla de encaminamiento) para cada nodo, es decir, el nodo central decide la tabla de encaminamiento de cada nodo en función de la información de control que éstos le mandan. El inconveniente de este método es que consumimos recursos de la red, y se harían necesarias rutas alternativas para comunicarse con el nodo central. La adaptación a cambios es perfecta siempre y cuando las notificaciones de los cambios lleguen antes de iniciar los cálculos de las rutas.

- **Aislado**

Se basa en que cada vez que un nodo recibe un mensaje que tiene que reenviar (porque no es para él) lo reenvía, por todos los enlaces salvo por el que le llegó.

- **Distribuido**

En este tipo de encaminamiento todos los nodos son iguales, todos envían y reciben información de control y todos calculan, a partir de su RIB (base de información de encaramamiento) sus tablas de encaramamiento. La adaptación a cambios es óptima siempre y cuando estos sean notificados. Hay dos familias de procedimientos distribuidos:

**a) Vector de distancias:** Cada nodo informa a sus nodos vecinos de todas las distancias conocidas por él, mediante vectores de distancias (de longitud variable según los nodos conocidos). El vector de distancias es un vector de longitud variable que contiene un par (nodo: distancia al nodo) por cada nodo conocido por el nodo que lo envía, por ejemplo (A:0;B:1;D:1) que dice que el nodo que lo manda dista "0" de A, " 1 " de B y " 1 " de D, y de los demás no sabe nada (ésta es la forma en la que un nodo dice lo que sabe en cada momento). El nodo solo conoce la distancia a los distintos nodos de la red pero no conoce la topología. Con todos los vectores recibidos, cada nodo monta su tabla de encaramamiento ya que al final conoce qué nodo vecino tiene la menor distancia al destino del mensaje, pues se lo han dicho con el vector de distancias.

Estado de enlaces: Cada nodo difunde a todos los demás nodos de la red sus distancias con sus enlaces vecinos, es decir, cada nodo comunica su entorno local a todos los nodos. Así cada nodo es capaz de conocer la topología de la red. La clave y dificultad de este método es la difusión.

En los encaminamientos estático y cuasi-estático la información necesaria se recoge y envía mediante gestión (al crear la red y en operaciones de mantenimiento), y los cálculos de la FIB se realizan fuera de línea (mediante administración, es decir en una máquina a parte).

En función de las tablas de encaramamiento empleadas. Los nodos manejan tablas de encaramamiento, en las que aparece la ruta que deben seguir los mensajes con destino a un nodo determinado de la red. Podemos distinguir entre encaramamiento salto a salto y encaramamiento fijado en el origen.

- **Encaminamiento alto a salto**

En la literatura inglesa, este tipo de encaminamiento se denomina como salto por salto. Se basa en que cada nodo no tiene que conocer la ruta completa hasta el destino, sino que sólo debe saber cuál es el siguiente nodo al que tiene que mandar el mensaje; las tablas proporcionan el nodo siguiente en función del destino.

- **Encaminamiento fijado en origen**

En ingles este encaminamiento se llama source routing. En él, son los sistemas finales los que fijan la ruta que ha de seguir cada paquete. Para él, cada paquete lleva un campo que especifica su ruta (campo RI: Routing Information), y los nodos sólo se dedican a reenviar los paquetes por esas rutas ya especificadas. Así pues, son los sistemas finales los que tienen las tablas de encaminamiento y no se hace necesario la consulta o existencia de tablas de encaminamiento en los nodos intermedios. Este tipo de encaminamiento suele ser típico de las redes IBM.

## Capítulo 2

# CONTROL DE CONGESTIÓN DE RED

### INTRODUCCIÓN

El fenómeno de la congestión es complejo, al igual que los mecanismos para su control. En términos muy generales la congestión se produce cuando el número de paquetes que están transmitiéndose a través de la red comienza a aproximarse a la capacidad de manejo de paquetes de la red, el tráfico es muy alto. El objetivo del control de la congestión es mantener el número de paquetes dentro de la red por debajo del nivel a partir del cual el desempeño cae dramáticamente.

El tráfico en las redes de computadoras es uno de los principales problemas contra los que se tiene que estar preparados de manera preventiva, para evitar el congestionamiento. Conceda por un momento la similitud que existe entre una red de computadoras y las carreteras que existen en el mundo. En este caso las computadoras serían algo así como las ciudades a las que arriban los paquetes de datos (automóviles). Es sencillo suponer que mientras más autos quieren transportarse entre las ciudades los problemas debidos a la falta de espacios carreteros o a obstáculos en el camino, ocasionan congestionamiento, entonces tendríamos que estudiar las limitantes para poder encontrar un buen desempeño de estos elementos.

Comenzaremos viendo lo que es la congestión, luego analizaremos sus causas y finalmente distinguiremos entre control de flujo y control de congestión, ya que se suelen confundir.



## 2.1 Fenómeno, Definición.

El caudal depende del tipo de red y tiene un valor nominal máximo, que no podremos superar en ningún caso<sup>5</sup>. Pero además, la red no ofrece el mismo caudal real si se le ofrece poco tráfico o si se le ofrece mucho. Veamos la siguiente figura:

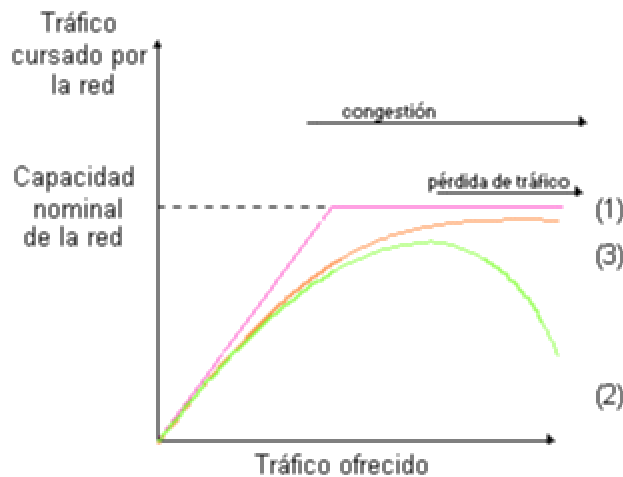


Figura 2.1  
Caudal en función del tráfico ofrecido.

La curva **(1)** representa el comportamiento ideal de la red: hay linealidad hasta llegar a la capacidad nominal de la red, momento en el que el tráfico cursado se satura. La curva **(2)** representa el comportamiento real típico de una red. Como puede observarse, al llegar a la zona de saturación, cuanto más tráfico se ofrece menos tráfico se cursa. Esto es debido, por ejemplo, a que los paquetes tardarán mucho tiempo en llegar a su destino, y mientras tanto serán retransmitidos por la fuente, pensando que se han perdido por el camino. Esto, a su vez, origina una explosión de tráfico, ya que cada paquete es retransmitido varias veces, hasta que consigue llegar a tiempo al destino.

Para evitar esa degradación, se introduce el control de congestión que trata de aproximar el comportamiento de la red al dado por la curva **(3)**, evitando así entrar en una zona de degradación.

El retardo de tránsito en la red sigue la siguiente curva:

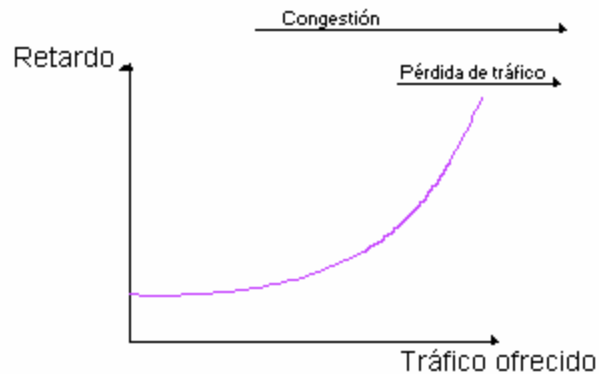


Figura 2.2  
Retardo de tránsito en función del tráfico ofrecido.

Vemos que el retardo no aumenta linealmente, sino que el aumento de éste es mayor que el aumento de tráfico ofrecido.

### 2.1.1 Causas de la congestión.

Hay varias causas de congestión. Enumeraremos aquí las más importantes:

- **Memoria insuficiente de los conmutadores.** Por ejemplo, veamos la siguiente figura:

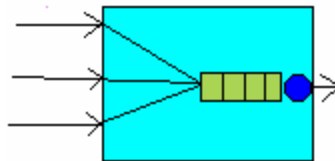


Figura 2.3  
Saturación del buffer de un nodo.

En ella se tiene un conmutador en el que tres líneas de entrada mandan paquetes a una de salida. Así puede llenarse el “buffer” (cola) de la línea de salida. Además, si hay congestión en otros nodos, las colas no liberan la información de los paquetes transmitidos (que se guarda por si hay que retransmitir), con lo que la situación empeora aún más.

- **Insuficiente CPU en los nodos.** Puede que el nodo sea incapaz de procesar toda la información que le llega, con lo que hará que se saturen las colas.

- **Velocidad insuficiente de las líneas.** Se tiene el mismo problema que en el caso anterior.

## 2.1.2 Diferencia entre control de flujo y control de congestión

### Control de flujo.

Es una técnica que permite sincronizar el envío de información entre dos entidades que producen/procesan la misma a distintas velocidades. Por ejemplo, supongamos el caso representado en la siguiente figura:

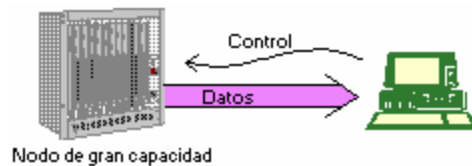


Figura 2.4  
Conexión entre nodo de alta capacidad y PC.

En este caso, dada la gran velocidad a la que produce y envía información, el nodo desborda al PC, por lo que éste debe enviar información de control (control de flujo) para que el nodo reduzca su tasa de envío de datos. De esta forma, parando a la fuente cada cierto tiempo, el PC puede procesar el tráfico que le envía el nodo.

### Control de congestión.

Es un concepto más amplio que el control de flujo<sup>7</sup>. Comprende todo un conjunto de técnicas para detectar y corregir los problemas que surgen cuando no todo el tráfico ofrecido a una red puede ser cursado, con los requerimientos de retardo, u otros, necesarios desde el punto de vista de la calidad del servicio. Por tanto, es un concepto global, que involucra a toda la red, y no sólo a un remitente y un destinatario de información, como es el caso del control de flujo.

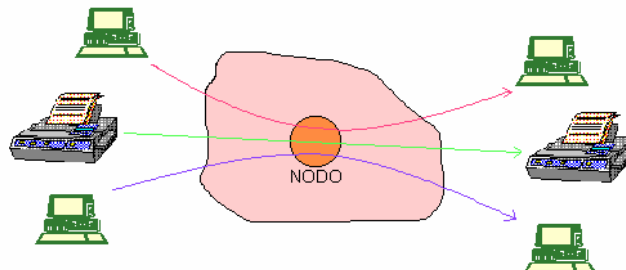


Figura 2.5  
Congestión en un nodo.

El control de flujo es una más de las técnicas para combatir la congestión. Se consigue con ella parar a aquellas fuentes que vierten a la red un tráfico excesivo. Sin embargo, como veremos, hay otros mecanismos.

## 2.2 Soluciones, Mecanismos de control de congestión.

El problema del control de congestión puede enfocarse matemáticamente desde el punto de vista de la teoría de control de procesos, y según esto pueden proponerse soluciones en bucle abierto y en bucle cerrado.

### 2.2.1 Soluciones en bucle abierto.

También llamadas soluciones pasivas. Combaten la congestión de las redes mediante un adecuado diseño de las mismas. Existen múltiples variables con las que el diseñador puede jugar a la hora de diseñar la red. Estas variables influirán en el comportamiento de la red frente a la congestión. Las resumiremos en función del nivel del modelo OSI al que hacen referencia:

Nivel Enlace

Variable de diseño.

- **Diseño de temporizadores y política de retransmisiones:** Cuando los temporizadores agotan su cuenta, los paquetes afectados serán retransmitidos por la fuente. Si este tiempo es muy pequeño, habrá gran cantidad de retransmisiones. Por el contrario, si es grande, habrá menos congestión, pero el retardo medio aumentará. Además, podemos controlar lo que se retransmite cuando el temporizador se agota (por ejemplo, cuando un paquete es sólo una porción de un mensaje, puede retransmitirse el mensaje entero o sólo el paquete afectado).
- **Política de descartes y almacenamiento de paquetes que llegan fuera de orden:** El rechazo puede ser simple, que origina más retransmisiones, o bien selectivo, obligando a un almacenamiento temporal de los paquetes que llegan fuera de orden y mejorando la congestión.
- **Política de asentimientos:** El “piggybacking”, o utilización de parte de un paquete de datos para enviar asentimientos de paquetes anteriormente recibidos, reduce, en principio, el tráfico, pero puede dar lugar a retransmisiones que contribuyan a la congestión.
- **Política de control de flujo:** Parando a una fuente que vierte mucho tráfico podemos reducir el riesgo de congestión.

Nivel Red

Variable de diseño.

- **Circuitos Virtuales frente a datagramas:** Muchos algoritmos de control de congestión funcionan sólo en modo circuito virtual.
- **Política de colas y de servicio:** Los enrutadores pueden diseñarse con una cola por línea de entrada, una cola por línea de salida, o ambos. Además, puede jugarse con el orden en que los paquetes son procesados, dando más prioridad a los paquetes de control, que contienen información útil desde el punto de vista de la congestión.
- **Política de descarte de paquetes:** De nuevo, la correcta elección de los paquetes que se descartan puede disminuir el riesgo de congestión.
- **Algoritmo de enrutamiento:** Es bueno desde el punto de vista de la congestión el balanceo del tráfico entre todas las líneas de la red.
- **Tiempo de vida de los paquetes:** La correcta elección de esta variable permite reducir el número de retransmisiones, mejorando así el comportamiento de la red desde el punto de vista de la congestión.

Nivel Transporte

Variable de diseño.

- Análogo al nivel de enlace, pero entre sistemas finales.

## 2.2.2 Soluciones en bucle cerrado.

También llamadas soluciones activas. Actúan cuando se detectan problemas. Tienen tres fases:

### 2.2.2.1 Monitorización de parámetros.

Se vigilan los siguientes parámetros:

- Ocupación de los enlaces y de los "buffers" (colas de espera en los nodos)
- Porcentaje de descartes
- Número de retransmisiones
- Retardos y "jitters". Jitter: oscilaciones de la separación temporal entre paquetes. En aplicaciones que requieren sincronización (videoconferencia, sincronizar audio con video), es muy importante que esas oscilaciones sean pequeñas.

### **2.2.2.2 Reacción: envío de información a los puntos necesarios.**

La comunicación se realiza gracias a:

- Paquetes especiales. NO están sometidos a control de congestión y se saltan las colas de espera en los nodos. Los envía el nodo que, gracias a la monitorización, ha detectado la congestión.
- Bits de cabecera. En los paquetes enviados, indico en la cabecera que empieza a haber congestión. (Ejemplo "Frame Relay").
- Información específica. Si se recibe una alerta de congestión (mediante bits de cabecera de paquetes que circulan por la red), se solicita más información.

### **2.2.2.3 Ajuste del sistema.**

Hay varias medidas:

- Reducir la velocidad de envío
- Control de acceso. No se permiten más conexiones.
- Tirar paquetes. Controlar ráfagas de paquetes que llegan.



## **Capítulo 3**

# **CONCEPTOS DE MÉTRICAS Y MODELOS OPERACIONALES**

### **INTRODUCCIÓN**

Cuando un usuario desea enviar un paquete de datos a un usuario remoto, el paquete viaja a través de la red desde la fuente hasta el destino. El paquete dentro de la red puede hacer uso de uno o varios medios de comunicación, generalmente en un envío a larga distancia (WAN) se hacen uso de varios de estos medios. Según la arquitectura de la red y la aplicación, el tamaño del paquete varía de manera estadística, así como el acceso de paquetes dentro de la red que han sido generados por los diversos usuarios que no mantienen una ocupación de la red de manera determinística, lo cual convierte al envío de los paquetes en un proceso estocástico. Una buena medida para encontrar los parámetros de desempeño de las redes de computadoras se da a través del estudio de la teoría de colas, que provee un medio probabilística para estudiar las incertidumbres generadas en los nodos de una red.

### 3.1 Modelo operacional.

Las redes de computadoras se representaron como centros de colas al momento de estudiarlos bajo el enfoque de modelado y de simulación<sup>2</sup>. Los recursos (CPU, discos, enlaces, etc.) se representan como servidores y las cargas de trabajos como los clientes, tal como sucede en un sistema discreto. El trabajo arriba al recurso (círculo), espera en cola (rectángulo) si éste está ocupado y sale del recurso cuando termina el servicio solicitado. El tiempo de permanencia o de residencia en el recurso es la suma de los tiempos de espera en cola (W) y en servicio (S), tal como se puede apreciar en la figura:

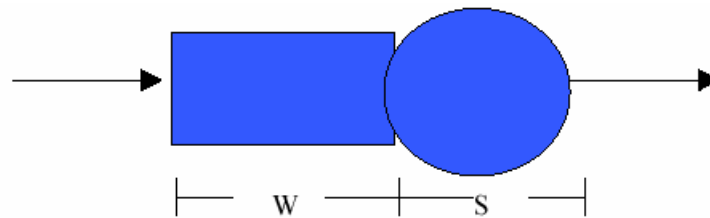


Figura 3.1  
Representación de una cola.

$R = S + W =$  tiempo de residencia en el recurso o tiempo de respuesta.

Los recursos o centros de servicio pueden estar en serie o en paralelo.

En el caso que esté en serie se denominan centros de cola o “queuing”.

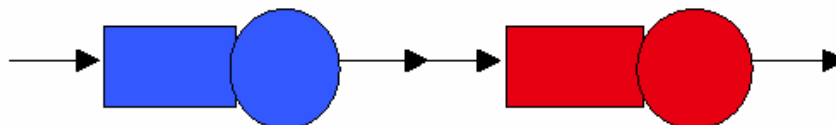


Figura 3.2  
Representación de colas en serie.

En el caso en que estén en paralelo se denominan centros de demora (muchos servidores, sin cola) o centros de retardo (“delay”).

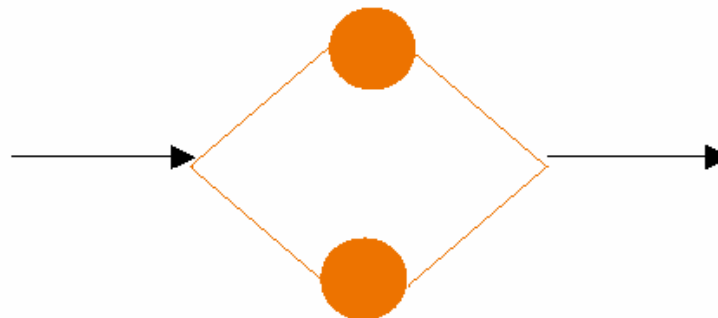


Figura 3.3  
Representación de colas en paralelo.



Para el caso de redes, los sistemas se representan mediante un diagrama de demoras de procesamiento de comunicación. Básicamente consiste en la representación del tiempo que transcurre desde que el cliente hace un requerimiento al servidor y recibe la respuesta. Los tiempos incluyen los clientes, servidores, segmentos de LAN's y WAN's. Son secuencias de ejes de tiempos verticales paralelos, donde los ejes completos significan los tiempos en los clientes y/o los servidores y los ejes a trazas son tiempos en redes (LAN y WAN).

Los mensajes se representan como  $[id, m]$ , donde  $id$  es el identificador del mensaje de demanda y respuesta y  $m$  es el tamaño en bytes. El "overhead" del protocolo se considera incluido en el tamaño del mensaje. La figura 3.4 es un diagrama de demoras de un sistema cliente-servidor. En las métricas de desempeño de una red se darán más detalles del sistema.

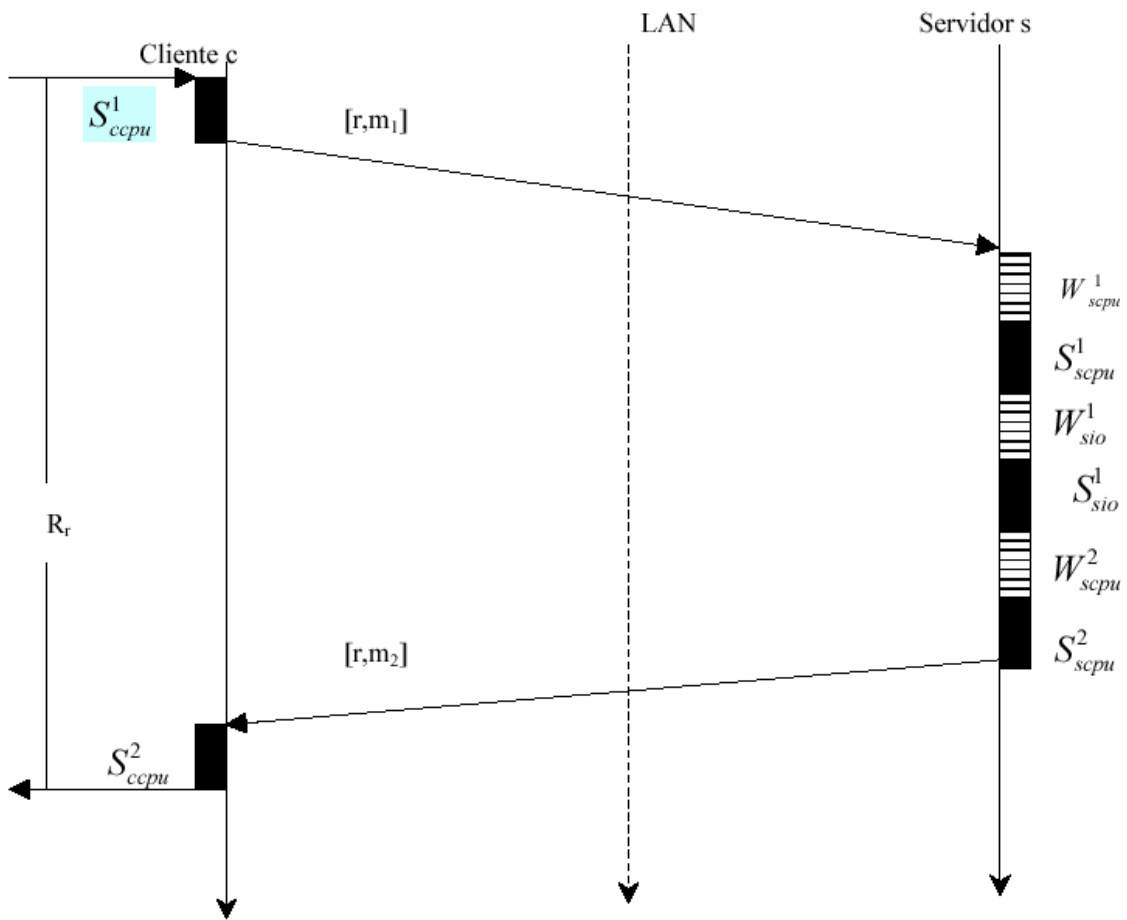


Figura 3.4  
Diagrama de demora de un proceso de comunicación de datos.

### 3.2 Definición de variables de entrada o parámetros.

T: Es el i intervalo de tiempo de una observación / monitorización.

Ej.: 30 minutos, 1 hora, etc.

A<sub>k</sub>: Es el número de arribos observados a un centro en el intervalo T (sino tiene subíndice k, representa el número de arribos al sistema). Ej.: A<sub>CPU</sub> = 203 arribos a una CPU en el intervalo T.

C: Número de trabajos completos observados en el sistema en el intervalo T.

Ej.: C = 50 trabajos procesados completamente en el intervalo T.

C<sub>k</sub>: Número de accesos o visitas observadas en un centro en el intervalo T.

Ej.: C<sub>Disco</sub> = 10000 accesos observados a un disco en el tiempo T.

B<sub>k</sub>: Tiempo de ocupación de un centro. Ej.: B<sub>CPU</sub> = 15 minutos de ocupación de la CPU en el intervalo T.

S<sub>k</sub>: Tiempo de servicio de un centro por visita o acceso.

Ej.: S<sub>Disco</sub> = 2 milisegundos (mseg) de atención por cada acceso al disco.

V<sub>k</sub>: Número de visitas a un centro. Ej.: V<sub>Disco</sub> = 30 visitas por cada trabajo.

N: Número de trabajos o población observada en el sistema.

Ej.: 10 trabajos "batch", 50 trabajos interactivos.

Z: Tiempo de pensado o de reflexión. Tiempo en que los trabajos residen en terminales sin requerimiento de CPU y/o discos. Ej.: Z = 30 segundos de trabajo en terminales (proceso de edición, etc.).

### 3.3 Estimación de parámetros con datos observados.

S<sub>k</sub>: Tiempo de servicio del recurso k: (segundos/acceso, mseg/accesos): es el tiempo de servicio brindado por cada recurso a un acceso. Se puede calcular como el tiempo de ocupación de un recurso k dividido por el total de accesos:

$$S_k \equiv \frac{B_k}{C_k} \equiv \frac{U_k T}{C_k}$$

Ej.: si se observó que un disco estuvo 22 minutos (B<sub>Disco</sub>) ocupado durante los 30 minutos de observación, y se registraron 10000 accesos, entonces S<sub>Disco</sub> = (22 minutos/10000 accesos)(60 segundos/minuto) = 0.132 segundos/acceso.

$V_k$ : Número de visitas al centro  $k$  para el procesamiento de un trabajo. Es la cantidad de veces que se accede a un recurso  $k$  para ejecutar un trabajo.

$$V_k \equiv \frac{C_k}{C}$$

Ej.: si se monitorearon 10000 accesos al disco durante 30 minutos y se procesaron un total de 50 trabajos,

$$V_{\text{Disco}} = 10000 \text{ accesos}/50 \text{ trabajos} = 200 \text{ accesos (o visitas)}/ \text{trabajo}$$

$D_k$ : Demanda de servicio del centro  $k$ . Es el tiempo demandado por parte de un trabajo a un recurso para su procesamiento. Se puede calcular como el tiempo de ocupación de un recurso  $k$  durante el intervalo de observación  $T$  dividido el número de trabajos procesados por el sistema en el tiempo  $T$ . O como la cantidad de visitas por el tiempo de servicio del recurso para cada visita:

$$D_k \equiv V_k S_k \equiv \frac{U_k T}{C} \equiv \frac{B_k}{C}$$

$I_k$ : Velocidad de arribos a un centro  $k$ . Es la relación entre los arribos y el tiempo  $T$ :

$$I_k = \frac{A_k}{T}$$

Si se trata del sistema, se denomina simplemente velocidad de arribos:

$$I = \frac{A}{T}$$

Ej.: Si al sistema se produjeron 250 arribos de trabajos o negociaciones en 3 minutos,  $I = (250 \text{ trabajos}/ 3 \text{ minutos})(1 \text{ minuto}/ 60 \text{ segundos}) = 1.389 \text{ trabajos} / \text{segundos}$  (ó tps).

### **3.4 Definición de variables de salida, de respuesta del modelo o medidas / índices de desempeño.**

$X$ : Velocidad de procesamiento del sistema (productividad, "throughput"): Es la cantidad de trabajos que procesa el S.I. por unidad de tiempo.

$$X \equiv \frac{C}{T}$$

Si se monitorearon 50 trabajos procesados (C) en 30 minutos (T) en el sistema, entonces la velocidad de procesamiento del sistema X es (50 trabajos / 30 minutos)(1minutos/60 segundos) = 0.0278 trabajos/ segundos

X<sub>k</sub>: Velocidad de procesamiento de un centro k. Es la cantidad de accesos que procesa un centro por unidad de tiempo.

$$X_k \equiv \frac{C_k}{T}$$

Si se monitorearon 10000 accesos (C<sub>k</sub>) en 30 minutos (T) en el disco, entonces la velocidad de procesamiento del disco X<sub>Disco</sub> es (10000 accesos / 30 minutos)(1minutos/60 segundos) = 5.556 accesos / segundos.

U<sub>k</sub>: Utilización de un centro k: es el tiempo en que estuvo ocupado un centro k durante el tiempo de observación T. La utilización siempre se refiere a un solo servidor o recurso.

$$U_k \equiv \frac{B_k}{T}$$

Si la CPU estuvo ocupada durante 15 minutos en el tiempo de observación de 30 minutos (T), la utilización de la CPU es U<sub>CPU</sub> = 15/30 = 0.5 o 50 %.

### 3.5 Leyes operacionales.

**Son las relaciones entre los parámetros (entradas) del sistema y las variables de respuesta (salidas)**

1. **Ley de utilización:** como se vio anteriormente, la utilización es la relación entre tiempos:

$$U_k \equiv \frac{B_k}{T}$$

Si en esta expresión, multiplicamos y dividimos por la cantidad de trabajos procesados C en el tiempo de monitorización T, se tendrá:

$$U_k \equiv \frac{B_k}{T} \frac{C}{C} = \frac{B_k}{C} \frac{C}{T} = D_k X$$

Si a la misma expresión de U<sub>k</sub>, ahora multiplicamos y dividimos por la cantidad de accesos registrados en un centro k (C<sub>k</sub>) en el tiempo de monitorización T, se tendrá:

$$U_k \equiv \frac{B_k}{T} \frac{C_k}{C_k} = \frac{B_k}{C_k} \frac{C_k}{T} = S_k X_k$$

En síntesis, la utilización se puede relacionar con la velocidad de procesamiento del sistema o la velocidad de procesamiento de un centro:

$$U_k = X_k S_k = X D_k$$

Un segmento de red transmite 1,000 paq/seg. Cada paquete tiene un tiempo promedio de transmisión igual a 0.15 mseg. ¿Cuál es la utilización del segmento de LAN ?

$$U_{LAN} = X_{LAN} * S_{LAN} = 1\ 000 * 0.00015 = 0.15 = 15\%$$

## 2. Ley del flujo forzado:

Sea  $X_k$  la velocidad de procesamiento de un centro  $k$ . Si multiplicamos y dividimos el segundo miembro de la ecuación por la cantidad de trabajos procesados  $C$ , se tiene:

$$X_k \equiv \frac{C_k}{T} \frac{C}{C} = \frac{C_k}{C} \frac{C}{T} = V_k X$$

Esto significa que la velocidad de procesamiento de un centro  $k$  es proporcional a la velocidad de procesamiento del sistema.

$$X_k = V_k X$$

Las negociaciones de una base de datos ejecutan un promedio de 4.5 operaciones de I/O sobre el servidor de base de datos. Durante un período de monitoreo de una hora, se ejecutaron 7 200 negociaciones. ¿Cuál es la velocidad promedio del disco? Sí cada I/O insume 20 mseg en promedio; ¿cuál es la utilización del disco?

$$X_{server} = 7\ 200 / 3\ 600 = 2\ tps$$

$$X_{disk} = V_{disk} * X_{server} = 4.5 * 2 = 9\ tps$$

$$U_{disk} = X_{disk} * S_{disk} = 9 * 0.02 = 0.18 = 18\%$$

## 3. Ley de Little:

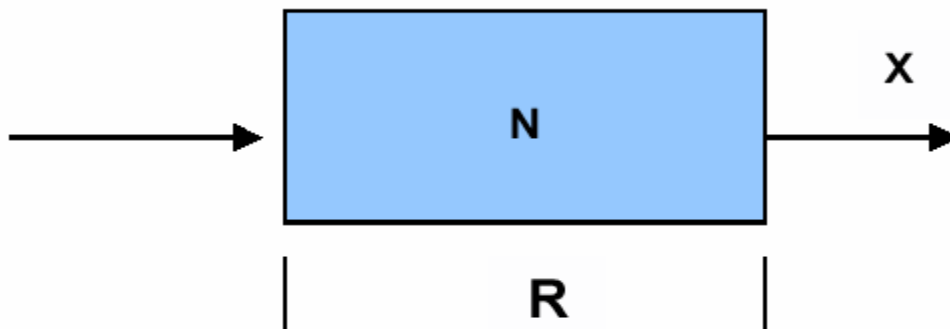


Figura 3.5  
Representación de la ley de Little.

El número promedio de clientes en una “caja” es igual al tiempo promedio que cada cliente insume en la “caja” por la velocidad de la “caja”.

$S_{CPU}$   
 $W_{red}$

### Ejemplo (1)

Un servidor NFS fue monitoreado durante 30 min y el número de operaciones de I/O ejecutadas durante este período fue de 10,800. El número promedio de requerimientos activos ( $N_{req}$ ) fue de 3.

¿Cuál fue el tiempo de respuesta promedio por requerimiento NFS en el servidor?

“caja” = servidor NFS

$$X_{server} = 10,800 / 1,800 = 6 \text{ req/seg}$$

$$R_{req} = N_{req} / X_{server} = 3 / 6 = 0.5 \text{ seg}$$

### Ejemplo (2)

La demora promedio experimentada por un paquete cuando recorre un segmento de red es de 50 mseg.

El número promedio de paquetes que recorren la red por segundo es de 512 paquetes/seg (velocidad de la red).

¿Cuál es número promedio de paquetes en tránsito en la red?

“caja” = segmento de red

$$N_{packets} = R_{packet} * X_{network}$$

$$N_{packets} = 0.05 * 512 = 25.6 \text{ paquetes.}$$

#### 4. Ley del tiempo de respuesta:

Si se considera un usuario que está en una terminal y requiere el procesamiento por parte de la CPU y del o de los discos, el tiempo del trabajo en el sistema es el tiempo de respuesta de procesamiento más el tiempo de residencia en las terminales (tiempo de reflexión o “pensado”  $Z$ ). Entonces la ley de Little se convierte en:

$$N = X(R+Z)$$

**La Ley de Little tiene mucha versatilidad porque se puede aplicar a un centro, a un centro y su longitud de cola o a un conjunto de centros (sistema completo).**

### 3.6 Métricas de desempeño en un sistema de redes de computadoras.

Las métricas más importantes usadas son el tiempo de respuesta, la velocidad de procesamiento y el costo.

El tiempo de respuesta se compone de los tiempos transcurridos desde que el cliente inicia un requerimiento a un servidor y recibe la respuesta.

Tiempos involucrados:

- tiempo de reflexión, donde el cliente prepara su requerimiento ( $t_1 - t_0$ )
- tiempo de envío de la respuesta de la transacción del servidor al cliente ( $t_3 - t_2$ )
- Tiempo de reacción ( $t_2 - t_1$ ) que es el tiempo transcurrido entre el envío del requerimiento del cliente hasta la información desde el servidor.

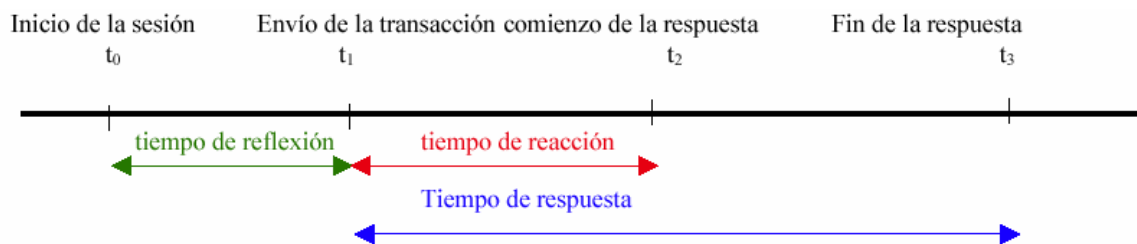


Figura 3.6  
Tiempos en el procesamiento de la comunicación.

Las unidades de la velocidad de procesamiento dependen del tiempo de transacción de la red en cuestión:

- Si el servidor es NFS, las unidades son Operaciones I/O NFS por segundo.
- Si se trata de un servidor de Web, las unidades son operaciones HTTP por segundo.
- Si se trata de servidores de base de datos, las unidades son negociaciones por segundo.

Los costos están generalmente asociados con alguna medida de desempeño, por ejemplo la respuesta o la velocidad de procesamiento como una relación precio-desempeño. Por ejemplo el "benchmark" TPC de la "Transaction Processing Performance Council" (TPC) mide las negociaciones en línea y da una métrica de dólar por tps, que indica cuando se necesita para gastar por unidad de velocidad de procesamiento.

### 3.6.1 Métricas de desempeño en una red.

Las demoras en un sistema de redes se representan mediante el diagrama de demora de procesamiento de la comunicación<sup>5</sup>, como se pudo apreciar en la figura 3.4.

Estos diagramas tienen la siguiente información:

- [id,m] : id identifica la demanda y sus respuestas y m indica el tamaño promedio en bytes del mensaje de la demanda o la respuesta.
- m1: tamaño de mensaje en bytes de la demanda
- m2: tamaño de mensaje en bytes de la respuesta
- B: ancho de banda de la red (bits por segundo: bps)
- Tiempo de transmisión: tamaño del mensaje en bits dividido B.
- "Overhead": incluido en el tamaño del mensaje
- Tiempo de servicio de un recurso S
- Tiempo de espera por algún recurso W
- Demanda de servicio: D
- Demanda de servicio del segmento LAN: tamaños de respuesta totales (demanda y respuesta) divididos ancho de banda:  $D_{LAN} = 8(m1 + m2)/B$ .

El tiempo de residencia de un mensaje en el sistema siempre es la suma del tiempo en cola y del tiempo en servicio.

Ejemplo: una transacción t en un sistema de redes requiere 5 mseg de CPU en el cliente, 10 mseg de CPU en el servidor y lee 10 bloques de 2048 bytes del disco del servidor, para lo cual el tiempo de servicio por bloque es de 0.01333 segundos.

De modo que las demandas en el sistema son las siguientes:

$D_{cliente,CPU} = 0.005$  segundos/transacción

$D_{servidor,CPU} = 0.01$  segundos/transacción

$D_{servidor,Disco} = 0.0133$  segundos  $\times$  10 = 0.1333 segundos/transacción

El cliente y el servidor están conectados por una red Ethernet con ancho de banda  $B=10$  Mbps y el requerimiento del cliente al servidor toma un paquete Ethernet completo de 1.518 bytes, mientras que la respuesta del servidor toma 7 paquetes.

$m1 + m2 = 1.518 + (7 \times 1.518) = 12.144$  bytes

El tiempo de transmisión de la LAN es entonces:

$D_{LAN} = 8(12.144)/10,000,000 = 0.0097$  segundos

El tiempo de respuesta mínimo posible para la transacción t es:

$R_t = D_{cliente,CPU} + D_{servidor,CPU} + D_{cliente,CPU} + D_{LAN} = 0.005 + 0.010 + 0.1333 + 0.0097 = 0.158$  seg.



El tiempo de espera por los recursos no se tuvo en cuenta en este ejemplo.

**Tiempos de servicio en redes.**

Sea la figura 3.7 donde se muestra un sistema C/S de tres capas:

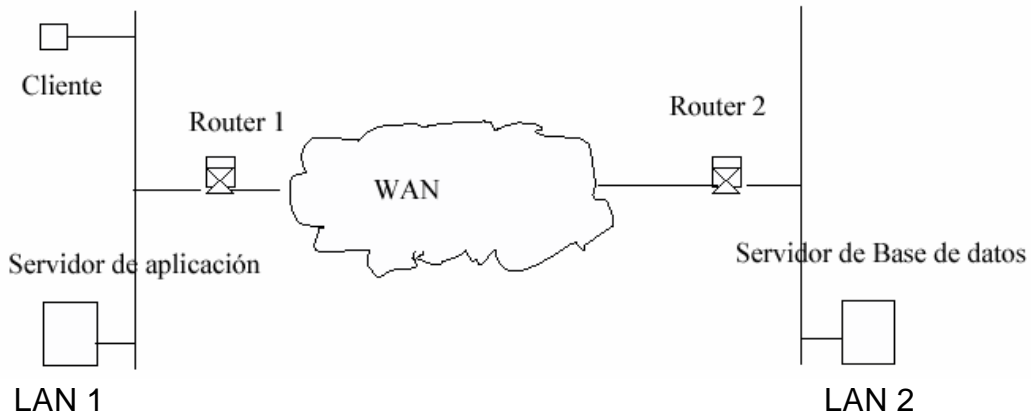


Figura 3.7 Sistema C/S de tres capas.

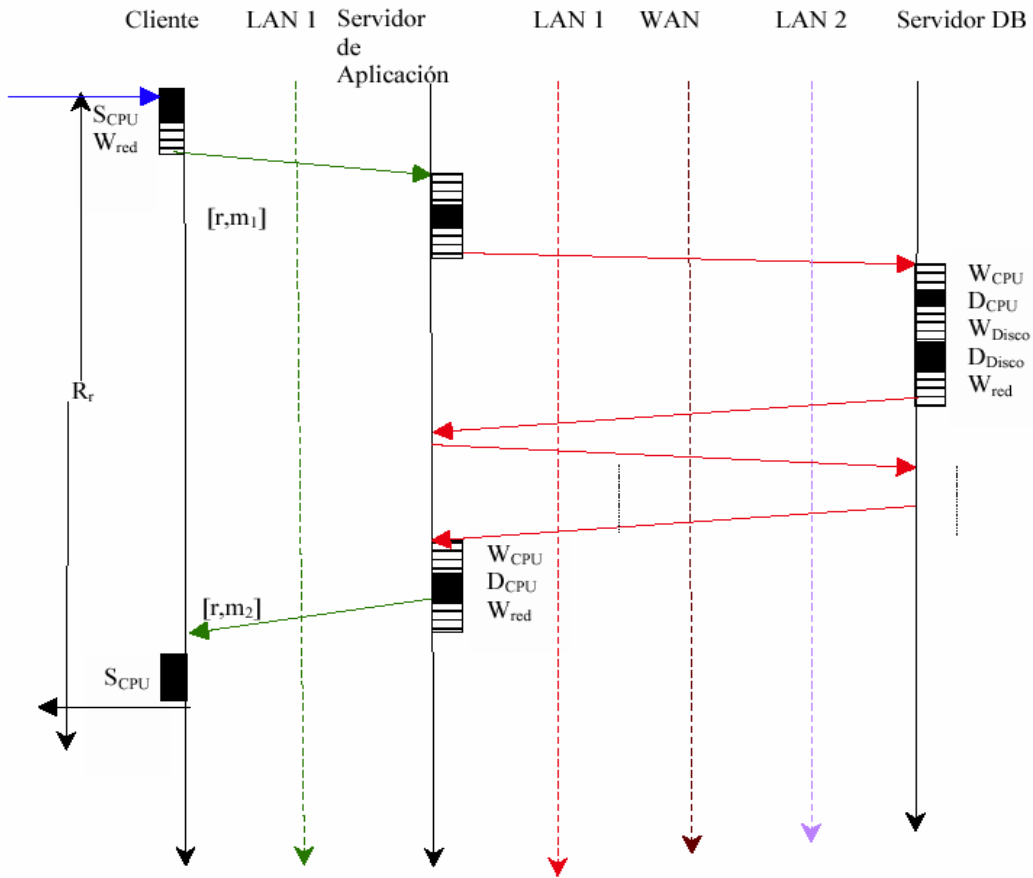


Figura 3.8 Diagrama de demora del procesamiento de comunicación de un sistema C/S tres capas.

El cliente envía el mensaje al servidor de aplicación local en la misma LAN (LAN1). La lógica de la aplicación lógica se ejecuta en el servidor pero requiere de un servidor de base de datos ubicado en la LAN 2, para lo cual alcanza el enrutador1, atraviesa la WAN y a través del enrutador 2 accede al servidor de base de datos. La figura 3.8 muestra el diagrama de demoras del procesamiento de la comunicación, sin incluir por ahora a los enrutadores. Los tiempos de espera por la red se definen como  $W_{red}$ .

Sea el siguiente caso de conectividad entre un cliente y un servidor:

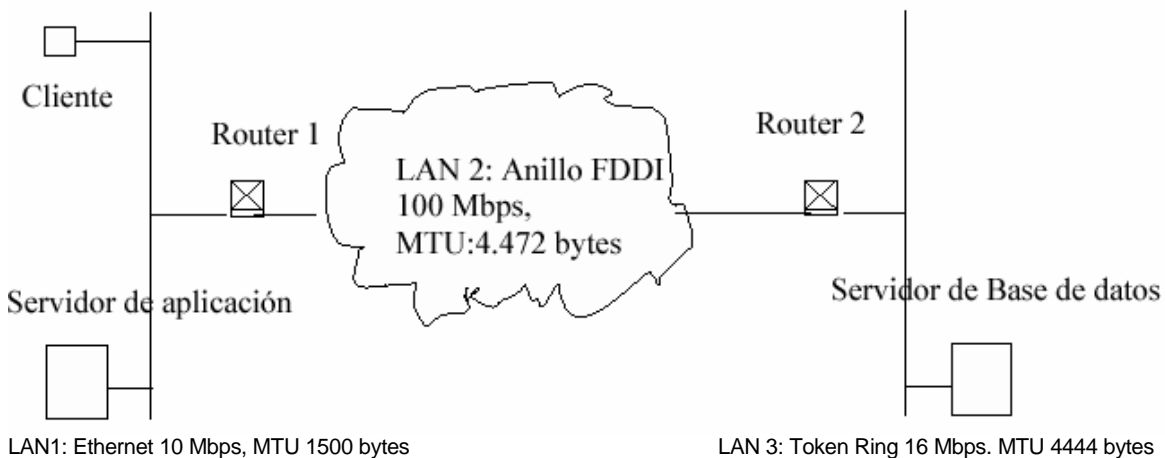


Figura 3.9  
Conectividad entre un cliente y un servidor.

Ejemplo: Un mensaje que viaja desde un cliente hasta un servidor tiene que pasar por varias capas de protocolos y puede ser transmitido a través de una o más redes. En la figura 3.10 un mensaje pasa desde un cliente hasta el servidor por una Ethernet 10 Mbps, un “backbone” FDDI de 100 Mbps y una LAN Token Ring de 16 Mbps.

Los mensajes generados tienen que pasar por una serie de protocolos que involucran al menos un protocolo de capa de transporte (TCP o UDP) un protocolo de Internet (IP o IPX) y un protocolo de red (Ethernet o Token Ring).

Las entidades de protocolo en cada capa se comunican entre sí intercambiando PDU’s compuestos por un encabezado y un área de datos. Los PDU’s reciben distintos nombres según el protocolo y tienen un tamaño máximo para el área de los datos.

En la capa red, el máximo tamaño de área de datos se llama unidad de transmisión máxima (MTU).

MTU para Ethernet 1500 bytes  
 MTU para Anillo FDDI MTU: 4472 bytes  
 MTU Token Ring: 4444 bytes

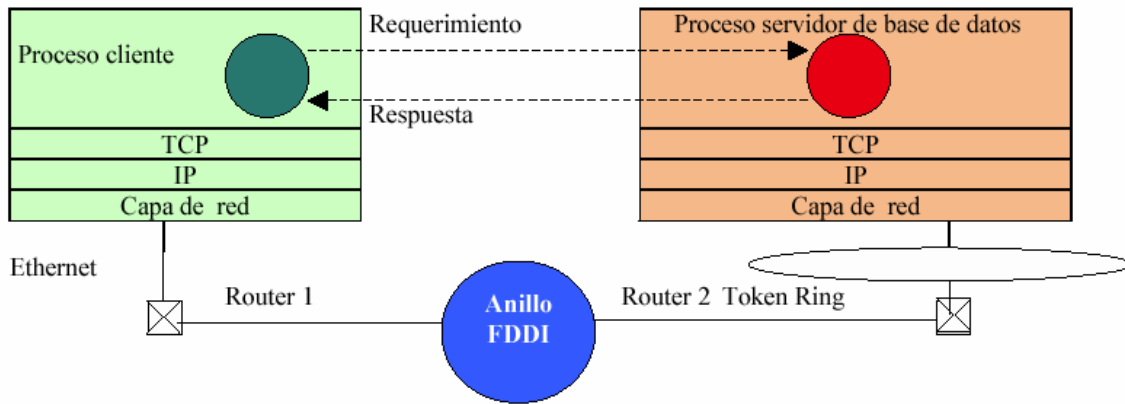


Figura 3.10 Interacción cliente-servidor.

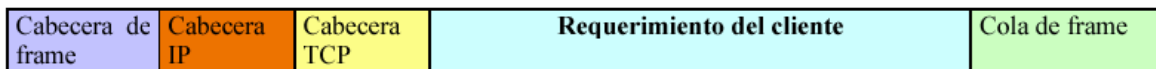


Figura 3.11 Formato de la estructura de un mensaje.

Los enrutadores tienen que ser capaces de fragmentar los datagramas cuando van a través de redes con MTU's decrecientes. Los fragmentos son armados en el nivel IP por el host de destino. Si un paquete de 2500 bytes cruza desde una red FDDI a la Ethernet, tiene que ser fragmentado en dos fragmentos por el enrutador1.

Cada capa de protocolo agrega su propia cabecera y a veces una cola. Cada protocolo tiene su nombre de PDU, el tamaño máximo de PDU, el "overhead" de la cabecera y de la cola y el tamaño máximo de área de datos, el cual es igual al tamaño máximo de PDU menos el "overhead". A continuación, en la tabla 3.1 se dan estos valores para varios protocolos importantes<sup>10</sup>.

Protocolo	Nombre de PDU	Tamaño de PDU (bytes)	Overhead (bytes)	Área de datos máxima (bytes)
TCP	Segmento	65535	20	65515
UDP	Datagrama		8	
IP versión 4	Datagrama	65535	20	65515
IP versión 6	Datagrama	65535	40	65515
ATM	Celda	53	5	48
Ethernet	Frame	1518	18	1500
IEEE 802.3	Frame	1518	21	1497
IEEE 802.5 Token Ring	Frame	4472	28	4444
FDDI (RCF 1390)	Frame	4500	28	4472

Tabla 3.1 Parámetro de protocolos.

**El tiempo de servicio de un mensaje en una red es el tiempo que toma transmitirlo sobre la red, igual al número de bytes necesarios para transmitirlo dividido por el ancho de banda.**

El “overhead” del protocolo depende de los protocolos involucrados en la fragmentación.

Ej. :Un mensaje de demanda de 300 bytes tiene una respuesta de 10000 bytes.

La interacción cliente y servidor es con conexión TCP. La demanda se sitúa en un área de datos de un segmento de TCP, y viaja en el área de datos de un datagrama IP, que es encapsulado por un trama Ethernet, luego por un trama FDDI y finalmente por un trama Token Ring que viaja por las LAN’s 1, 2 y 3 respectivamente.

La demanda o el requerimiento de 300 bytes reciben 20 bytes de TCP y 20 bytes de IP de cabecera y 18 bytes de “overhead” en la LAN1, 28 bytes del “overhead” de la trama en la LAN2 y 28 bytes de “overhead” en la LAN3

LAN1:  $300+20+20+18 =$  trama de 358 bytes

LAN2 y 3:  $300+20+20+28 =$  tramas de 368 bytes

Tiempo de transmisión del requerimiento en las LAN’s:

LAN 1 :  $(358 \times 8)/10000000 = 0.000286$  seg = 0.286 mseg.

LAN 2 :  $(368 \times 8)/10000000 = 0.0002944$  seg = 0.02944 mseg.

LAN 3 :  $(368 \times 8)/16000000 = 0.000184$  seg = 0.184 mseg.

La respuesta del servidor al cliente:

- El segmento TCP tiene 10000 bytes + 20 bytes de cabecera.
- En el nivel IP en el servidor de base de datos el mensaje se divide en 3 datagramas con una cabecera de 20 bytes porque los datagramas no deben superar los MTU. El primer datagrama tiene el “overhead” de TCP.
- Como el MTU de datagrama de Token Ring es de 4444 bytes, el área de datos del primer datagrama de Token Ring es de  $4444 \text{ bytes} - 20 - 20 = 4404$  bytes.
- El segundo datagrama tiene  $4444 - 20 = 4424$  bytes.
- El tercer datagrama tiene 1172 bytes.
- La cabecera del Token Ring tiene 28 bytes por lo que los tamaños de las tres tramas del Token Ring son:  $4444+28 = 4472$  bytes (para el primero y el segundo) y 1220 para el tercero (con  $28+20$ ).

El tiempo de servicio de la respuesta para la LAN 3 es:

$[(4472+4472+1220) \times 8]/16000000 = 0.005082$  seg = 5.082 mseg.

- Los tres datagramas que alcanzan el enrutador2 vienen de la LAN3 a la LAN 2 y no necesitan fragmentarse porque el MTU de la LAN2 es mayor que la LAN3.

- Los nuevos tamaños de los datagramas en la LAN2 se obtienen restando el “overhead” de las tramas en la LAN3 y sumando el “overhead” de las tramas en la LAN2. Como el “overhead” en las LAN’s 2 y 3 son los mismos, los tamaños de las tramas son los mismos.
- El ancho de banda de LAN2 es diferente al de la LAN3, entonces, el tiempo de servicio en la LAN 2 es:

$$[2x(1518+1518+1502)+1210]8 / 10,000,000 = 0.00823 \text{ seg} = 8.23 \text{ mseg.}$$

Los tres fragmentos que contienen la respuesta alcanzan el router1 y de allí la LAN1.

Los tamaños de datagramas vistos por el enrutador 1 después que los 28 bytes de “overhead” son

Eliminados son 4444 bytes, 4444 bytes y 1192 bytes.

Los dos primeros se tienen que fragmentar porque el límite de MTU de la LAN 1 es 1500 bytes.

Cada datagrama de 4444 bytes tienen que generar 3 datagramas cada uno con sus propios 20 bytes de cabecera IP.

El primero es de 1500 bytes de longitud. Y el último de 1484 bytes.

En total dan 4484 = 4444+40. Los 40 bytes se agregan porque los fragmentos 2 y 3 necesitan 20 bytes de cabecera IP.

Agregando el “overhead” de Ethernet, se tiene:

$$[2x(1518+1518+1502)+1210]8 / 10,000,000 = 0.00823 \text{ seg} = 8.23 \text{ mseg.}$$

Sean los siguientes parámetros:

- Tamaño de mensajes en bytes
- $MTU_n$  de una red n
- TPC “overhead” del protocolo TCP
- IP “overhead”
- Trama “overhead”
- “overhead”<sub>n</sub> : total de “overhead” (TCP+IP+trama)
- Ancho de banda en mega bits por segundo
- Ndatagramas
- N número de redes

El número de datagramas es:

$$N_{\text{datagramas}} = \left[ \frac{\text{tamaño.mensaje} + \text{Overhead.TCP}}{\min_{n=1}^N MTU_n - \text{Overhead.IP}} \right]$$

El “overhead” total de protocolo involucrado en la transmisión de un mensaje está dado por:

"overhead"<sub>n</sub> = "overhead" TCP + Ndatagramas x ("overhead" IP + "overhead" Trama<sub>n</sub>)

El tiempo total de servicio en una red n para un mensaje es igual al total de bits necesarios para transmitir el mensaje (incluyendo el "overhead") dividido por el ancho de banda en bits por segundo:

$$(Tiempo.Servicio)_n = \left[ \frac{8 \times tamaño.mensaje + Overhead_n}{10^6 \times ancho.banda_n} \right]$$

**La utilización de la red n esta dada por la velocidad de arribos del mensaje en la red multiplicada por el tiempo promedio de servicio de un mensaje en la red.**

Para entender la teoría de colas, desde un punto de vista más matemático se anexa los fundamentos en el apéndice A, con el cual podremos entender y analizar mejor este problema.

# Capítulo 4

## TEORÍA DEL MODELADO Y LA SIMULACIÓN

### INTRODUCCIÓN

En este capítulo se estudian las técnicas que son usadas en computadoras que simulan las operaciones de varios tipos de procesos del mundo real.

El conjunto de elementos dispuestos en un arreglo que ordenadamente contribuyen a determinado objetivo, es usualmente llamado un sistema y dado nuestro interés, se hará un análisis científico de cómo trabajan. Los elementos que conforman un sistema están o pueden representarse con relaciones de matemática o lógica, ya que el mundo real de donde estamos extrayendo este sistema puede modelarse mediante expresiones matemáticas que describen el comportamiento de los elementos que lo conforman.

Si la relación que compone un modelo es simple, entonces este tal vez podrá ser resuelto por métodos matemáticos (Como álgebra, cálculo, ó teoría de la probabilidad, etc.) y se obtendría información exacta sobre las preguntas de interés; este es llamado una solución analítica. Muchas de las veces, los sistemas del mundo real son demasiados complejos y obtener un modelo para ser evaluados por métodos analíticos son muy difíciles y estos modelos podrían ser estudiados por técnicas de simulación. En una simulación nosotros usamos una computadora para evaluar un modelo cual sea, los datos son generados en orden de estimación con características deseadas del modelo.

Entonces el modelado y la simulación de los modelos se utilizan con fines de análisis y planificación, por ejemplo los modelos de datos que indican la forma en que éstos son considerados por los diferentes departamentos, pueden utilizarse para predecir posibles congestionamientos cuando los usuarios solicitan cierto tipo de información,

La simulación es una de las técnicas más extensas usadas en la administración y búsqueda de operaciones. Un indicativo de este es el winter Simulation Conferece, que atrae de 600 a 700 personas todos los años, donde hay muchos vendedores a usuarios, de bastantes simuladores, también se encuentran mas de 100 conferencistas por año.

## 4.1 Sistemas, Modelos y Simulación.

Un sistema esta definido como el conjunto de componentes y eventos relacionados que interactúan unos con otros para alcanzar un objetivo específico<sup>1</sup>. Nosotros definimos el estado de un sistema, como la colección de variables necesaria para describir al sistema en un tiempo particular, relativo a los objetos de un estudio.

Clasificamos a los sistemas según su naturaleza en (Law and Kelton, 1991):

Continuo: Es aquel en que las variables de estados varían suavemente en el tiempo sin discontinuidades. Ejemplo la voz.

Discreto: Es aquel en que los estados de las variables mantienen un valor constante durante un cierto tiempo tras el cual pasan a otro valor de forma discontinua. Ejemplo: señal binaria (que sólo puede tomar los valores 0 y 1).

Determinístico: Si el sistema no contiene ningún elemento aleatorio es un sistema. En este tipo de sistema, las variables de salidas e internas quedan perfectamente determinadas al especificar las variables de entrada.

Estocástico: En este caso algún elemento del sistema tiene una conducta aleatoria. Entonces, para entradas conocidas no es posible asegurar los valores de salida.

Cuando estamos interesados en algunos puntos importantes de cualquier sistema, hay la necesidad de estudiarlo y tratar de obtener algunas características importantes de sus componentes que permitan predecir el desempeño bajo algunas condiciones de inicio consideradas. En el siguiente diagrama se muestran diferentes caminos para conocer y ser estudiado un sistema y en especial un sistema de redes de computadoras.

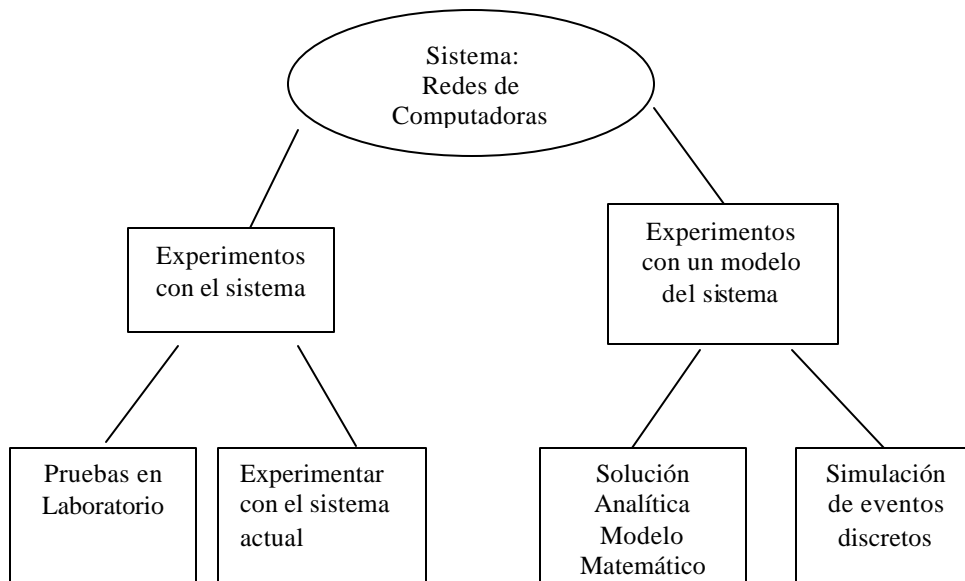


Figura 4.1  
Caminos de estudio en un sistema.



## 4.1.1 Simulación frente a otras técnicas de evaluación.

### Experimentar con el sistema actual.

Sí es posible alterar el sistema físico, operar bajo nuevas condiciones y no afecta el costo-efectivo, probablemente este podría ser un buen camino, realmente no es común seguir este tipo de evaluación, por que los sistemas que se estudian no son tan sencillos, por ejemplo; en un sistema de redes de computadoras, la evaluación nos llevaría a hacer las siguientes acciones:

- Evaluar el diseño lógico de la red existente y hacer recomendaciones sobre los cambios necesarios para el desempeño y/o confiabilidad, con base en las necesidades actuales y futuras.
- Evaluar toda la infraestructura de cableado para determinar la fuente de los problemas de desempeño en una red existente o probar si la infraestructura actual puede utilizarse con una nueva tecnología de red.
- Evaluar el desempeño de la red y hacer recomendaciones de cómo mejorarlo. Incluye un análisis del tráfico de la red, balanceo de servidores, conflictos de direcciones, colisiones de paquetes, cuellos de botella en la comunicación y desempeño de los dispositivos de interconexión.

La evaluación por medio de la experimentación del sistema actual, nos llevaría mucha inversión, bastante tiempo y perdidas en otros factores que no podamos identificar como problemas del análisis del sistema.

### Experimentar con un modelo del sistema.

Un modelo es la representación matemática o lógica de las características dinámicas de un sistema, si pensamos en un modelo que maneja la computadora<sup>13</sup>, son conjuntos de ecuaciones que representan alguna condición o conjunto de operaciones en el sistema real o hipotético.

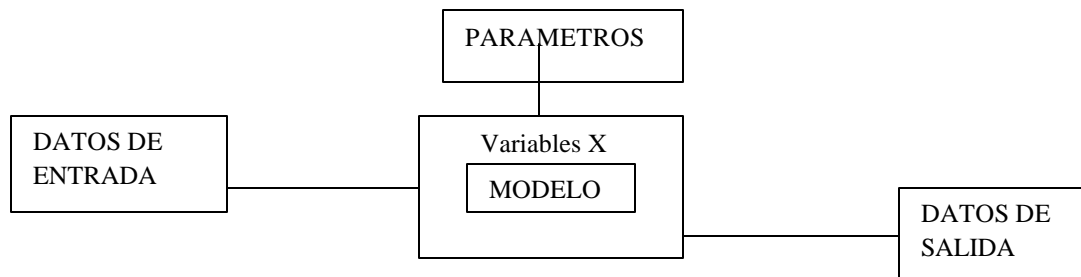


Figura 4.2  
Funcionamiento de un modelo.

De esta manera si ocupamos un modelo del sistema, no tendríamos problemas de tiempo, inversión y sobre todo, que podemos analizar cualquier parte de nuestro sistema, con el propósito que estamos buscando.

## **La simulación como alternativa a la solución analítica de los modelos matemáticos.**

En la solución analítica por lo regular es adecuado acotar el problema, es decir, es necesario hacer algunas suposiciones de simplificación. Aunque con estas simplificaciones el modelo se hace menos válido para varias situaciones reales.

Por ejemplo, pensemos en el sistema de un enlace de salida para un conmutador de paquetes, el problema es la tarea básica de un análisis de colas. Dado una tasa de llegadas y un tiempo de servicio como entrada, suministrar, como salida información referente a lo siguiente:

- Elemento en espera
- Tiempo en espera
- Elementos en el sistema de colas
- Tiempo en el sistema de cola

En específico de estas salidas podría necesitarse conocer sus valores promedios. Además, pudiera ser útil conocer algo a cerca de su variabilidad. Así, podría ser útil la desviación estándar de cada una de ellas. También pudieran ser útiles otras medidas. Por ejemplo, para diseñar el buffer asociado de un enlace de salida de un nodo de conmutación, que tamaño de buffer se necesita para la probabilidad de desbordamiento.

Para responder tales interrogantes se necesita tener un conocimiento completo de las distribuciones de probabilidad de la tasa de llegada y del tiempo de servicio y las fórmulas que resultarían serían demasiadas complejas.

La simulación por lo contrario, permite abordar modelos más complejos, la simulación produce estimaciones, no soluciones exactas y por ultimo permite validar estudios analíticos y viceversa.

La simulación también suple la experimentación con modelos físicos, prototipos o con el sistema real cuando ésta es imposible o muy costosa.

### **4.1.2 Simulación en un sistema de eventos discretos.**

El concepto de la simulación nos permite hacer estudio cuantitativo de sistemas reales que consisten en:

Definir un modelo matemático del sistema real, (incluyendo su entorno).

Representar numéricamente la evolución del modelo, durante un cierto periodo de tiempo.

Estimar las características de interés del sistema a partir de los datos recogidos en el paso anterior.

Entonces la simulación de un evento discreto, comprende el modelado de un sistema representado en cada estado de las variables cambiando en instantes separados de tiempo, (Matemáticamente el sistema toma valores discretos, descritas por diferentes ecuaciones). Estos instantes en tiempo son únicos en cada evento ocurrido, un evento es definido como un punto ocurrido que tal vez cambie el sistema.

### **4.1.3 Componentes y organización de una simulación de un modelo de eventos discretos.**

En la simulación de los modelos de sistemas de eventos discretos, comparten un número común de componentes y hay una organización lógica para estos que permiten la programación, el debugging y los futuros cambios del modelo en un programa de computadora<sup>14</sup>. En particular, el siguiente proceso de simulación de eventos discretos es usado en la programación de lenguajes para propósito general:

- Estado del sistema: La colección de estados de las variables necesarias para describir el sistema en un tiempo particular
- Reloj de simulación: Una variable dando el valor actualizado del tiempo de simulación
- Lista de eventos: Una lista conteniendo el próximo tiempo, cuando cada tipo de evento ocurrirá
- Estadística de los contadores: Variables usadas para obtener información acerca del desempeño del sistema
- Iniciación de rutina: Un subprograma que inicializa la simulación en el tiempo 0
- Tiempo de rutina: Un subprograma que determina el siguiente evento de la lista de eventos
- Rutina del evento: Un subprograma que actualiza el estado del sistema, cuando un tipo de evento en particular ocurre
- Librerías de rutina: Un programa puesto para ser usado en la generación aleatoria de observaciones de la distribución de probabilidad que esta determinada como parte de la simulación del modelo
- Generación de reporte: Un subprograma que estima y calcula el performance cuando la simulación termina

- Programa maestro: Un subprograma que invoca las rutinas del próximo evento, entonces transfiere el control a la rutina correspondiente, actualizando el estado del sistema correctamente, el programa maestro también checa la terminación del sistema e invoca el generador de reportes.

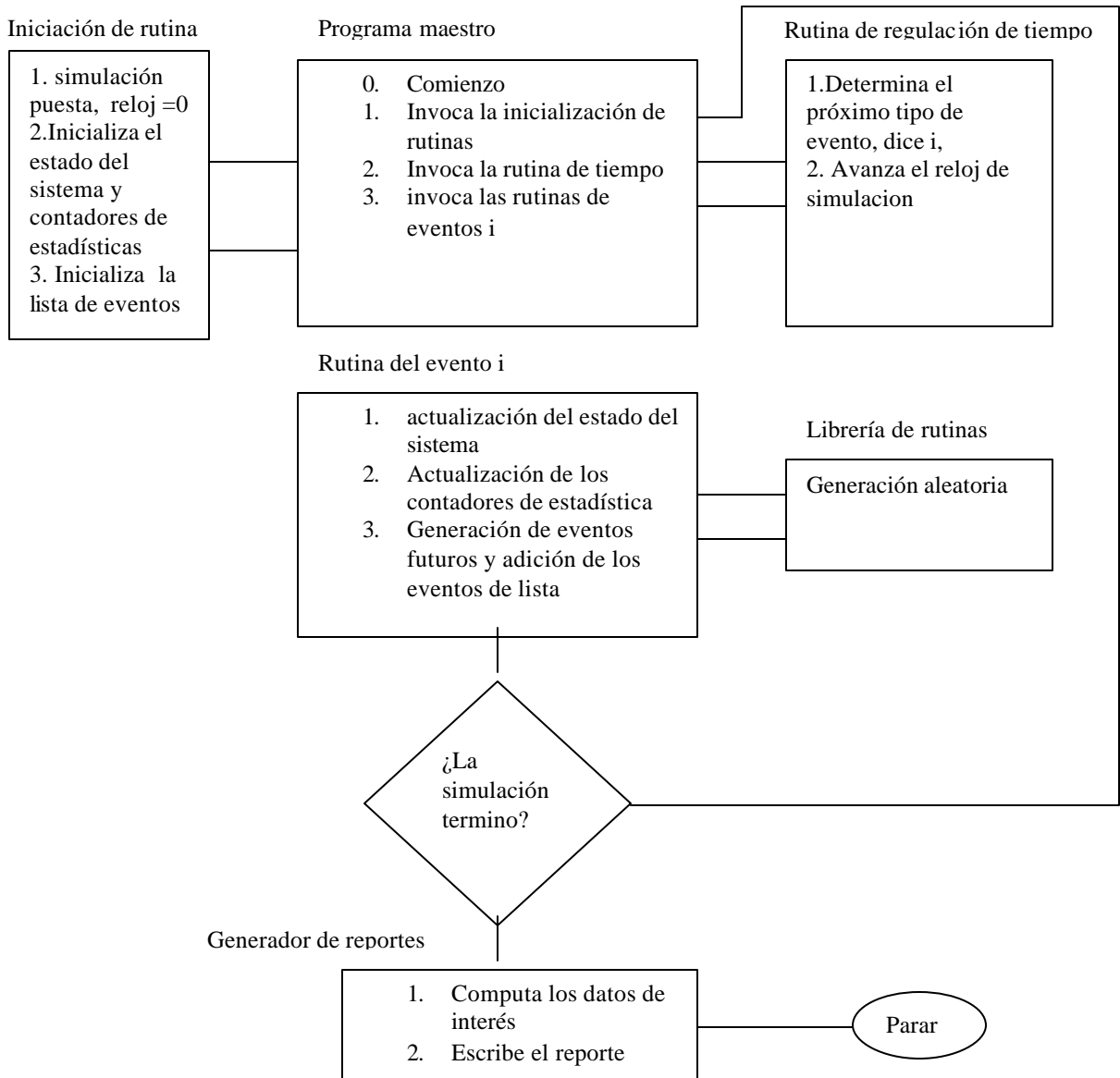


Figura 4.3  
Diagrama de control de flujo para la simulación de modelos de sistemas discretos.

## 4.2 Modelado en sistemas de eventos discretos.

El modelado es el proceso de construcción de un modelo. Un modelo es una representación de un objeto, sistema, o idea. Usualmente, su propósito es ayudar explicar, entender o mejorar un sistema (Shannon, 1988).

El arte de modelar consiste en la habilidad para analizar un problema, resumir sus características esenciales, seleccionar y modificar las suposiciones básicas que caracterizan al sistema, y luego enriquecer y elaborar el modelo hasta obtener una aproximación útil.

Entonces los modelos de simulación nos representan:

- El estado del sistema; que lo integra bs elementos que lo comprenden, como son los procesos, recursos, colas etc. Además los atributos de cada elemento, conocidos o a estimar y las relaciones entre estos.
- Cambios de estado o transición; estos son eventos o sucesos que pueden alterar el estado del sistema a lo largo del tiempo, que identifican nuevas estructuras (ejemplo nuevo elemento) y de valores de atributos.

Según el tipo de evolución temporal, se clasifican en modelos continuos y discretos.

**Los modelos continuos**, son aquellos que cambian en forma continua en el tiempo, por ejemplo:

Sistema: automóvil por una vía con poco tráfico

Modelo: (para calcular su posición)

- estado: posición X, inicialmente  $X_i$
- Transiciones:  $dX/dt = V$  constante (en el modelo)

En la simulación se calcularía X en una serie de instantes separados un  $\Delta t$  Suficientemente pequeño.

**Los modelos discretos** son aquellos que cambian solo en ciertos instantes de tiempo, cuando ocurren en determinados sucesos, no de forma continua. Entre suceso y suceso, el estado no cambia. Para la simulación sólo son importantes los instantes en que ocurren los cambios.

Por ejemplo:

Sistema: procesador

Modelo: (para calcular su porcentaje de ocupación)

- estado: ocupado (sí, no)
- transiciones:

- Llegada de tarea: no -> sí
- Fin de tarea: sí -> no

Modelo: (para calcular el número de tareas procesados)

- estado: número de tareas  $N: 0, 1, 2, \dots$
- transiciones:
  - Fin de tarea:  $N \rightarrow N + 1$

En ambos casos, falta especificar cuándo llegan las tareas y cuánto dura su ejecución a partir de información sobre el sistema (datos del procesador y de las tareas que recibe).

### Ejemplo 2

Sistema: cola de un conmutador de paquetes

Modelo:

- estado: longitud de la cola  $L: 0, 1, 2, \dots$
- transiciones:
  - Llegada de paquete:  $L \rightarrow L + 1$
  - Salida de paquete:  $L \rightarrow L - 1$

Falta especificar cuándo llegan y salen los paquetes (como antes a partir de datos del sistema) y qué queremos estimar: longitud media, máxima, porcentajes,... En el sistema real,  $L$  tendrá un valor máximo.

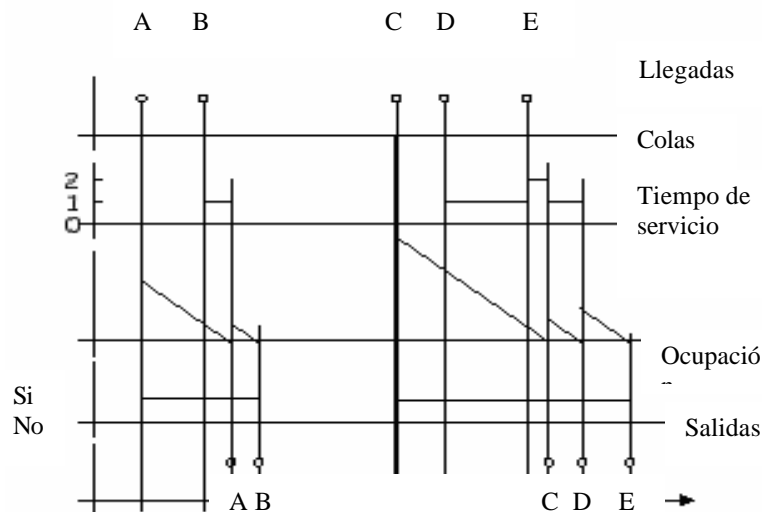


Figura 4.4  
Evolución del ejemplo 2.

La mayoría de los modelos de interés para simular redes de comunicaciones son estocásticos, dinámicos, discretos y algunas veces continuos para el nivel físico.

## **4.3 Modelos de simulación.**

### **Modelos de simulación discretos.**

Independientemente del lenguaje en que luego se programen, los modelos de simulación se pueden caracterizar por:

- Los objetos o elementos básicos que intervienen
- Las primitivas u operaciones, consideradas indivisibles, que permiten manipular los objetos anteriores.

Cada lenguaje ofrecerá un conjunto particular de objetos y primitivas.

### **Los modelos de eventos discretos los podemos clasificar en:**

- Orientados a sucesos; estos tienen un enfoque de bajo nivel, son un único elemento básico, el evento o suceso, el conjunto de primitivas son mínimas y por su estructura son más flexibles y eficientes.
- Orientados a procesos; tiene mayor nivel de abstracción, sus elementos básicos son procesos y recursos, el conjunto de primitivas es más rico, y por su naturaleza son más fáciles de desarrollar.

### **4.3.1 Simulación orientada a sucesos.**

El procedimiento de simulación orientada a sucesos tiene los siguientes pasos de análisis:

- Cada suceso elemental que pueda cambiar el estado del sistema se considera por separado.
- A cada suceso se le asocia un procedimiento de tratamiento, que agrupa todas las acciones a que dé lugar (incluyendo la programación de sucesos futuros)
- El programa que llama a los procedimientos anteriores mantienen una lista de sucesos futuros en un orden cronológico, controla el avance del tiempo simulado y va llamando a los procedimientos de tratamiento cuando corresponda.

### **Conceptos a distinguir:**

- Tipos genéricos de sucesos
- Sucesos concretos
- Información particular asociada a cada suceso concreto de un cierto tipo

- Procedimiento de tratamiento de todos los sucesos concretos de un cierto tipo

Ejemplo: Conmutador

- Tipos de sucesos: llegadas y salidas
- Sucesos concretos:
  - Llegada de paquete 124 en el instante ...
- Información asociada:
  - Tiempo de servicio de cada paquete que llega
- Procedimientos de tratamiento:
  - De llegadas: si el procesador esta libre, ocuparlo durante el tiempo de servicio del paquete que llega; si no, encolar paquete
  - De salidas: ...

### 4.3.2 Simulación orientada a procesos.

El procedimiento de simulación orientada a procesos tiene los siguientes pasos de análisis:

- Los sucesos elementales no se tratan individualmente, si no que se consideran dentro del flujo de control de los componentes activos del sistema
- El programa principal coordina el avance de los diferentes procesos que intervienen en el modelo

Ejemplo:

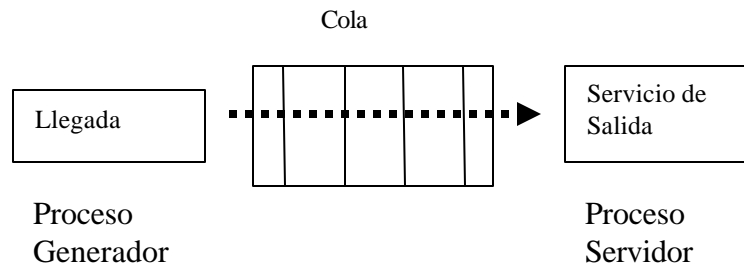


Figura 4.5  
Procesamiento de un paquete en cola.

### Conceptos a distinguir:

- Tipos genéricos de procesos
- Parámetros y datos locales de cada ejemplar de un cierto tipo



- Procedimientos que indica las acciones que realizan todos los procesos de un cierto tipo

### Estados posibles de un proceso:

ACTIVO: tiene algo que hacer en un instante conocido

- Activo y listo para ejecutar: tiene algo que hacer en el instante actual

EJECUTANDO: proceso que está ejecutando sus acciones (elegido entre los activos 'y listos)

PASIVO: no tiene nada que hacer (hasta que otro lo active)

ESPERANDO: necesita algún recurso temporalmente no disponible

TERMINADO: no hace nada más

### Avance de la simulación.

En cada instante de tiempo simulado se consideran todos los procesos activos y listos para ejecutar.

Estos procesos se van ejecutando uno a uno sucesivamente desde el punto de vista de tiempo real (pero concurrentemente desde el punto de vista del tiempo simulado, ya que el tiempo simulado no avanza).

Un Proceso está en ejecución hasta que decida pasar a estado PASIVO llamando a DESACTIVÁR, TERMINAR o a ESPERAR (o bien pida un recurso no disponible, como se verá).

Cuando no queda ningún proceso listo para ejecutar se avanza el tiempo simulado hasta el próximo instante en que algún proceso tenga algo que hacer

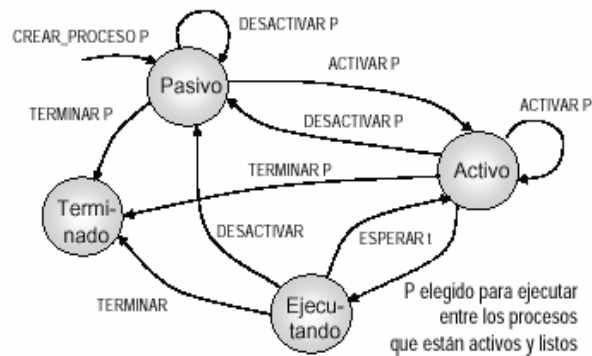


Figura 4.6  
Estados de un proceso P.

## Recursos.

Son elementos pasivos por los que compiten los procesos. Se denomina estación a un bloque de varias unidades de un cierto recurso.

Si un proceso pide más unidades de las que hay disponibles, queda en estado ESPERANDO en una cola de procesos asociada a la estación hasta que otro proceso libere un número suficiente de unidades.

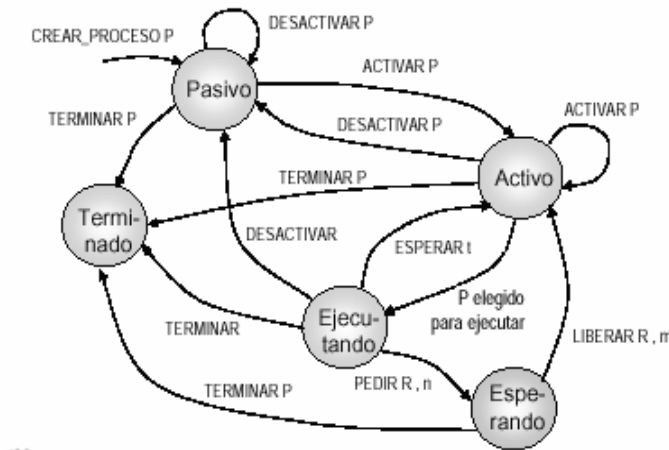


Figura 4.7  
Estados de un proceso P incluyendo recursos.

## En Resumen

Los modelos de simulación orientados a sucesos son el enfoque de más bajo nivel y más general. Los modelos orientados a procesos y recursos ofrecen abstracciones de más alto nivel para representar los elementos del sistema a modelar.

## 4.4 El proceso de la simulación.

La eficiencia de un proyecto de simulación depende en mucho, de la experiencia y habilidad de las personas que intervienen en la planeación, ejecución y evaluación del proyecto. “La simulación es un proceso el cual se desarrolla en varias etapas” las cuales se pueden resumir en tres fases:

1. Fase de definición,
2. Fase de preparación,
3. Fase de producción,

La fase de definición incluye:

- a) El establecimiento preciso del problema y la definición de objetivos.

b) Colección de datos e información.

La fase de preparación incluye:

- a) Formulación y diseño del modelo
- b) Prueba de la consistencia y veracidad del modelo
- c) Diseño de experimentos y plan de simulación.

La fase de producción incluye:

- a) Desarrollo de experimentos
- b) Análisis y evaluación de resultados
- c) Remodelación y depuración del programa de simulación.

No todos los proyectos de simulación necesariamente deben pasar por estas fases, esto más bien depende del modelo y el objetivo de la simulación. La etapa crítica en todo proceso de simulación

Para el desarrollo de la simulación se requiere:

1. Obtener y analizar datos del sistema.
2. Desarrollar un diagrama de flujo que refleje el comportamiento del modelo.
3. Codificar el diagrama de flujo a un lenguaje de computadora.
4. Experimentar con el programa para diferentes datos y analizar los resultados.

## 4.5 Ventajas, desventajas y artificios de la simulación.

Ahora en esta etapa listaremos algunas características buenas y malas de la simulación (como oponerse a otros métodos de estudio de sistemas), y anotando algunos errores comunes hechos en estudios de simulación que pueden empeorar o arruinar un proyecto por simulación<sup>4</sup>.

Como mencionamos la simulación es ampliamente usado y un popular método, incrementándose para el estudio de sistemas complejos. Algunas ventajas posibles de la simulación que podrían informar de su extendido interés son los siguientes:

- Más complejo, sistemas de mundo real con elementos estocásticos que no podría ser exactamente descritos por un modelo matemático que pueda ser evaluado analíticamente. Así, una simulación es comúnmente el único tipo de investigación posible.
- La simulación permite a uno mismo estimar las ejecuciones de un sistema existente debajo de algunos proyectos determinados con condiciones de operación.
- Proponer alternativos sistemas de diseño (o políticas alternativas de operación para un sistema único) pueden ser comparados vía simulación para ver cual conoce lo mejor de un requerimiento específico.
- En una simulación podemos mantener mucho mejor control superior a condiciones experimentales que pueden generalmente ser posibles cuando experimentan con el sistema mismo.
- La simulación nos permite estudiar un sistema con un marco de tiempo largo, -un sistema económico-, en tiempo comprimido, o alternativamente para estudiar el trabajo detallado de un sistema en tiempo expandido.

La simulación no esta sin sus inconvenientes. Algunas desventajas son las siguientes:

- Cada corrida de un modelo de simulación estocástico produce solo estimaciones de características validas de un modelo para un particular parámetro determinado de entrada. Así, diversas corridas independientes del modelo probablemente será requerido para cada parámetro determinado de entrada para ser estudiado. Por otro lado, un modelo analítico, si es apropiado puede usualmente producir fácilmente las características reales de aquel modelo para una variedad de determinados parámetros de entrada. Así, si un válido modelo analítico esta disponible o puede fácilmente ser diseñado, será generalmente preferible para un modelo de simulación.

- Los modelos de simulación son usualmente caros y consumen tiempo de diseño.
- El largo volumen de números producidos por un estudio de simulación o el impacto persuasivo de una animación realista usualmente crea una tendencia en excelente confianza en unos resultados de estudio que son justificados. Si un modelo no es una representación valida de un sistema bajo un estudio, la simulación resultante, no importa que tan impresionantes aparezcan, proveerá una pequeña información útil acerca del sistema actual.

Cuando decida en todo caso un estudio en simulación es apropiado en una situación dada, podemos solo advertir que esas desventajas e inconvenientes que abrimos en mente y que todas las facetas relevantes de una situación particular propia son compradas para hacerlo mejor. Finalmente, nótese que en algunos estudios las dos, simulación y modelos analíticos podrían ser útiles. En particular, la simulación puede ser usada para revisar la validez de suposiciones necesitadas de un modelo analítico. Por otro lado, un modelo analítico puede sugerir alternativas razonables para investigar un estudio en simulación.

Asumiendo que una decisión ha sido hecha para usar la simulación, hemos encontrado los siguientes artificios para completar satisfactoriamente un estudio de simulación:

- Falla al tener un determinado objetivo mal definido al inicio del estudio en simulación.
- Inapropiado nivel de detalle.
- Tratar un estudio de simulación como si fuera primariamente un ejercicio en un programa de computadora.
- Falla al tener gente sin un conocimiento de metodología de simulación y estadísticas sobre el equipo de modelo.
- Inapropiado software de simulación.
- Obviamente usando productos de software de simulación quienes con complicadas sentencias de macro podría no ser bien documentada y podría no ser implementada la lógica de modelo deseada.
- Creencia que es fácil de usar paquetes de simulación, el cual requiere poca o no programación, requiere un significativo nivel bajo de competencia técnica.
- Abuso de animación.
- Usar distribuciones arbitrarias (normal, uniforme, triangular) como entrada a la simulación.
- Haciendo una replica única de un particular sistema de diseño y tratando las estadísticas de salida como las verdaderas respuestas.
- Usando las malas medidas de ejecución.

## 4.6 Software de simulación.

En el estudio de los ejemplos de simulación en secciones anteriores, probablemente notaste diversos rasgos necesarios en una programación más discreta en eventos de modelos de simulación, incluyendo:

- Generación de números aleatorios, que son, observaciones desde una probabilidad de distribución  $U(0,1)$ .
- Generación de variaciones aleatorias desde una probabilidad de distribución específica, (exponencial).
- Tiempo avanzado de simulación.
- Determinar el evento siguiente desde la lista de eventos y pasando el control al apropiado bloque de código.
- Añadir registros a, o borrar registros desde, una lista.
- Acumular estadísticas de salida y reportar los resultados.
- Detectar condiciones de error.

Estos y otros rasgos de los programas de simulación que guían al desarrollo de un software de propósito especial y de los paquetes de simulación. Además, creemos que el mejoramiento y gran facilidad de uso de estos paquetes han sido factores mayores en el incremento de la popularidad en años recientes.

Discutimos a continuación los meritos relativos de usar un paquete de simulación antes que un lenguaje de programación tales como C o C++ para construir modelos de simulación. En la sección presentamos una clasificación del software de simulación, incluyendo una discusión del propósito general y las aplicaciones orientadas a los paquetes de simulación. Rasgos deseables para los paquetes de simulación, incluyendo animación.

La publicación IIE Solutions and ORIMS tienen hoy estudios de software de simulación sobre una base regular justa<sup>12</sup>.

### 4.6.1 Clasificación del software de simulación.

Propósito general contra paquetes de simulación orientados a la aplicación.

Históricamente, los paquetes de simulación se agruparon para ser de dos tipos mayores, nombrados, lenguajes de simulación y simuladores orientados a la aplicación (o paquetes). Los lenguajes de simulación fueron generados al natural, y el desarrollo de modelos fue hecho escribiendo el código. Los lenguajes de simulación proveen, en general, un gran trato de flexibilidad al modelar, pero fueron usualmente difíciles de usar. Por el otro lado, los simuladores fueron orientados hacia una aplicación en particular, y un modelo fue desarrollado usando gráficos, cuadros de dialogo, y menús hacia abajo. Los simuladores fueron algunas veces fáciles de aprender y usar, pero podrían no ser lo suficientemente flexibles para algunos problemas.

Sin embargo, en años recientes vendedores de lenguajes de simulación tienen la tentativa de hacer su software fácil de usar empleando un abordado modelo de construcción gráfico. Un escenario típico podría ser, el tener una paleta de iconos de construcción de modelos localizada en un lado de la pantalla de la computadora. Los iconos son seleccionados desde la paleta con el ratón y están establecidos el área de trabajo. Los iconos entonces son conectados para indicar flujo de entidades directas al sistema de interés. Finalmente, un doble click sobre un icono traerá un cuadro de diálogo donde los detalles son añadidos. Por ejemplo, suponga que un icono representa un servidor, en algún sistema. Entonces el cuadro de diálogo podría permitir al usuario especificar la información tal como el número de servidores paralelos, el tiempo de servicio de distribución para cada servidor, y si los servidores pueden caerse (y, en que manera). Por el otro lado, los vendedores de simuladores tienen la tentativa de hacer su software más flexible permitiendo la programación en ciertas localidades de modelos usando un pseudo lenguaje interno. A lo más un simulador, es ahora posible modificar existiendo un constructor de modelos y al crear unos nuevos. Así, la distinción entre los lenguajes de simulación y simuladores están realmente llegando a ser empañada.

Basado en la antedicha discusión, diremos ahora que hay dos tipos de paquetes de simulación. Un paquete de simulación general puede ser usado por cualquier aplicación, pero podría tener rasgos especiales para cada uno (para manufactura, comunicaciones o procesos de reingeniería). Por el otro lado, un paquete de simulación orientado a la aplicación está diseñado para ser usado para una determinada clase de aplicación tal como manufactura, cuidado a la salud o call centers.

#### **4.6.2 Elementos comunes de simuladores.**

Los paquetes de simulación típicamente incluyen entidades, atributos, recursos y colas como parte de su estructura de modelado. Una entidad (ver tabla 1.3 para ejemplos) es creada, viaja directo a algunas partes del sistema de simulación, y entonces es usualmente destruida. Entidades son distinguidas desde cada otro por sus atributos, los cuales son piezas de información establecidas con las entidades. Como una entidad se mueve directo al sistema simulado, este solicita el uso de recursos. Si un recurso solicitado no está disponible, entonces la entidad junta una cola. Las entidades en una cola en particular podrían ser servidas en una manera tipo FIFO (primero dentro, primero fuera), servido en una manera tipo LIFO (último dentro, primero fuera), o se encuentra sobre algunos atributos en orden de incremento o decremento.

Tipo de sistema	Entidades	Atributos	Recursos	Colas
Manufactura	Parte	Número de parte, fecha de vencido	Maquinas, trabajadores	Colas o amortiguador
Comunicaciones	Mensaje	Destino, longitud de mensaje	Nodos, conexión	Colas
Aeropuerto	Avión	Número de vuelo, peso	Pista de aterrizaje, puentes	Colas
Agencia de seguros	Aplicación, demanda	Nombre, número de policía, cantidad	Agentes, oficinista	Colas

Tabla 4.1  
Atributos de sistemas.

### 4.6.3 Características deseables de software.

Hay características a considerar cuando seleccionamos software de simulación. Categorizamos estas características que están en uno de los siguientes grupos:

- Capacidades generales (incluyendo flexibilidad de modelo y fácil de usar).
- Requisitos de hardware y software.
- Animación.
- Características estadísticas.
- Soporte y documentación de cliente.
- Reportes de salida y planos.

Discutimos ahora cada grupo de características en turno.

#### Capacidades generales.

En nuestra opinión, la característica más importante para un producto software de simulación es tener flexibilidad de modelo o, en otras palabras, la habilidad de modelar un sistema a quien los procedimientos de operación pueden tener cualquier cantidad de complejidad. Nótese que los dos sistemas no son exactamente lo mismo. Así, un paquete de simulación que depende de un número compuesto de constructores de modelo sin capacidad de hacer algún tipo de programación en cualquier manera es ligado a ser inadecuado para ciertos sistemas encontrados en la práctica. Idealmente, debe ser posible modelar cualquier sistema usando solo los constructores proporcionados en el software, no debe ser necesario usar rutinas escritas en un lenguaje de programación tales como C. Lo siguiente son algunas capacidades específicas que hacen un producto de simulación flexible:

- Habilidad para definir y cambiar atributos para las entidades y también variables globales, y usar las dos decisiones lógicamente (constructores if-then-else).
- Habilidad de usar expresiones matemáticas y funciones matemáticas (logaritmos, exponenciación, etc).



**Ejemplo.** Algunas veces es deseado cambiar una distribución de probabilidad tal como una distribución gama  $c$  unidades a la derecha (esto empieza a  $c$ ) y usarlo por el tiempo de hacer algunas tareas (un tiempo de servicio). Supóngase que el símbolo para la distribución gama en un paquete particular de simulación es `gama` (alfa, beta), donde alfa y beta son los parámetros de forma y escala, respectivamente. Si el software es todavía flexible, entonces debe ser posible usar `c+ gama` (alfa, beta) como el deseado cambio en tiempo de tarea. Desafortunadamente, este no podría ser posible en algunos paquetes de simulación.

- Habilidad para crear nuevos constructores de modelos y modificar algunos existentes, y ser capaz de usar el nuevo o modificado constructor de modelo actual y modelos futuros.

La segunda característica más importante de un producto de simulación es facilidad de usar (y facilidad de aprender), y muchos paquetes contemporáneos de simulación tienen una interfaz gráfica con el usuario para facilitarlos. El producto software debe tener constructores de modelo (iconos o bloques) que no son ni primitivos ni macros. En el caso formal, un número largo de constructores serán requeridos para modelar una relativa situación simple, en otro caso, cada constructor de cuadro de diálogo contendrá un número excesivo de opciones si esto se permite para una adecuada flexibilidad. En general, el uso de apéndices en cuadros de diálogo pueden ayudar a manejar un largo número de posiciones.

La habilidad para rehusar piezas de un modelo lógico incrementa en uno la eficiencia de modelado. Jerarquía es un importante concepto en un número de paquetes de simulación. Es también un camino útil para manejar cubrir la pantalla para un modelo orientado gráficamente que consiste de muchos iconos o bloques.

El software debe tener buena ayuda de depuración tales como un depurador interactivo. Un depurador poderoso permite al usuario hacer cosas tales como:

- Seguir una entidad única directo al modelo para ver si es procesado correctamente.
- Ver el estado del modelo a cada vez que un evento ocurre (una maquina que se cae).
- Configurar el valor de algunos atributos o variables para forzar a una entidad caer un trayecto lógico que ocurre con una probabilidad pequeña.

Alta velocidad de ejecución del modelo es importante para determinadas aplicaciones tales como modelos militares largos y modelos en los cuales un número largo de entidades debe ser procesado (una red de comunicaciones alta y rápida). Programamos un simple sistema de manufactura en seis productos de simulación y encontramos que, para este modelo, un producto fue más de 11 veces más rápido que otro.

Es deseable el ser capaz de desarrollar un modelo de uso amigable “inicio termina” cuando el modelo de simulación ha sido usado por alguien mas que el desarrollador del modelo. Esta capacidad permite al desarrollador el crear una interfase por la cual el usuario inexperto pueda fácilmente introducir parámetros de modelo tales como el medio tiempo del servicio o por cuanto tiempo correr la simulación.

Muchos vendedores de software de simulación ofrecen una versión de tiempo de corrida (ver Abstraction Softwares (1999)) dentro de su software, la cual, hablando aproximadamente, permite al usuario el cambiar un modelo de datos pero no lógico empleando un usuario amigable “inicio termina”. Aplicaciones para una versión de corrida en tiempo incluye:

- Correr escenarios nuevos o existentes para un modelo construido por un consultante interno o externo.
- Vender herramientas para repuestos de equipo o integradores de sistema.
- Entrenamiento.

Nótese que una licencia de corrida en tiempo generalmente tiene un considerable costo bajo que una licencia normal desarrollada.

Una característica que es concretamente de interés considerable es la habilidad para importar datos desde (una exportación de datos hacia) otras aplicaciones (hoja de calculo Excel o bases de datos).

Es deseable para un paquete de simulación que automáticamente simule diferentes escenarios que interaccione sobre algún modelo de parámetro, tales como el número de maquinas en una estación de trabajo para una fabrica. Esto debe entonces ser posible al mapear alguna medida desarrollada (tiempo común en sistema para una fabrica) como una función del parámetro que esta interaccionando.

En algunos eventos discretos de simulación (steelmaking), puede ser necesario tener ciertas capacidades disponibles desde la simulación continua. Llamamos tal simulación como una simulación combinada discreta continua.

Ocasionalmente, uno podría tener un complejo determinado de lógica escrita en un lenguaje de programación que necesita ser integrado hacia un modelo de simulación. Así, es deseable para un paquete de simulación el estar disponible para invocar rutinas externas.

Es muy útil para el paquete de simulación el ser fácilmente inicializado en un estado de no vacío y desocupado. Por ejemplo, en una simulación de un sistema de manufactura, podría ser deseable el inicializar el modelo con todas las maquinas ocupadas y todos los buffers de en medio llenos, en orden de reducir el tiempo requerido para el modelo para alcanzar un estado constante.

Otra característica útil es que el estado de simulación puede ser salvado al final de la corrida y usado para reiniciar fácilmente la simulación al poco rato.

Finalmente, el costo es usualmente una consideración importante para el propósito del software de simulación. Concretamente, el costo del software de simulación para una PC tiene un rango desde \$500 hasta aproximadamente \$50 000 dólares o tal vez un poco mas. Sin embargo, existen otros costos que pueden ser considerados, tales como cuotas de mantenimiento, cuotas de actualización, y el costo por algún hardware y software adicional que podría ser requerido.

### **Requisitos de hardware y software.**

Seleccionando un software de simulación, uno debe considerar que software esta disponible para plataformas de computadoras. Casi todo el software esta disponible para PC's basadas en Windows, y algunos productos también están disponibles para trabajar en estaciones UNIX y computadoras Apple. Si un paquete de software esta disponible para distintas plataformas, entonces debe ser compatible con las distintas plataformas. La cantidad de RAM requerida para ejecutar el software debe también ser considerado, desde que algunos productos requieren 64 Mb o los 128 Mb. Uno debe también considerar que sistemas operativos son soportados, tales como Windows 98, Windows NT y UNIX.

### **Animación y gráficas dinámicas.**

La disponibilidad de construir la animación dentro de los programas es una de las razones del uso incrementado en la simulación. En una animación, los elementos clave de los sistemas son representados en la pantalla por iconos que dinámicamente cambian posición, color y forma así como el modelo de simulación se desarrolla a través del tiempo. Por ejemplo, en un sistema de manufactura, un icono representa carretilla elevadora de horquillas que cambiara de posición cuando exista un cambio correspondiente en el modelo, y un icono representa una maquina podría cambiar el color cuando el estado de la maquina cambie (desocupado a ocupado) en el modelo.

Lo siguiente son algunos usos de la animación:

- La comunicación es la esencia de un modelo de simulación (o la simulación misma) a un desarrollador a las personas que no estén enteradas de los detalles técnicos del modelo.
- Comprobando el programa de computadora de simulación.
- Mostrando que un modelo de simulación no es valido.
- Sugiriendo procesos operacionales mejorados para un sistema (algunas cosas pueden no ser aparentemente desde una vista a los justos resultados numéricos).
- Entrenando personal de operación.

- Fomentando la comunicación en medio del proyecto de equipo.

Existen dos tipos fundamentales de animación: concurrente y post procesado (también llamado playback). En la animación concurrente la animación comienza desplegada al mismo tiempo que la simulación esta ejecutándose. Nótese, no obstante, que la animación es normalmente apagada cuando se producen las ejecuciones de producción, porque la animación reduce la ejecución de la simulación. En la animación post procesada, los estados de cambio en la simulación son guardados a un archivo en disco y usa para dirigir las graficas después de que la simulación termino. Algunos productos software de simulación tienen los dos tipos de animación.

Ahora discutiremos las características deseables para la animación. Primero, el software de simulación debe proporcionar la animación por naturaleza como una parte del proceso de construcción de modelo. Desde que la animación es primeramente un aparato en las comunicaciones, debe ser posible el crear iconos de alta resolución y guardarlos para rehusarlos mas tarde. El software debe venir con una librería estándar de iconos. El software debe proporcionar un suave movimiento de iconos; los iconos no deben relampaguear o saltar. Ahí debe haber un control para darse prisa o reducir la actividad de la animación. Debe ser posible acercarse hacia adentro y hacia fuera y hacer toma panorámica para ver las diferentes partes de un sistema que sea muy largo para ajustarse sobre una pantalla. Algunos productos software tienen vistas de animación nombradas, para que uno pueda construir un menú de las vistas correspondientes a las diferentes partes del sistema de simulación. Es deseable si la animación usa vectores basados en gráficos (pintar y dibujar con líneas, arcos, y figuras rellenas) antes que gráficos basados en píxel (pintar y dibujar con píxeles individuales que se encienden y apagan). El anterior tipo de gráficos permite la rotación de un objeto (un rotor de helicóptero) tan bien como un vehículo para mantener sus propiedades de orientación así como va alrededor de una esquina.

Algunos productos de simulación con animación concurrente permite al usuario detener la simulación en el vuelo mientras observa la animación, hacer cambios para determinados parámetros del modelo (el número de maquinas en una estación de trabajo) y entonces instantáneamente reiniciar la simulación. Sin embargo, esto puede ser estadísticamente peligroso si el estado del sistema y los contadores estáticos no son reseteados.

Ciertos paquetes de simulación permiten la animación tridimensional (el punto ventajoso desde el cual se ve la animación puede ser rotado alrededor de los tres ejes), la cual puede ser importante para el manejo de presentaciones y para simulaciones en la cual los espacios verticales son importantes. En estos productos esto puede ser también posible de proporcionar la vista de la animación con una perspectiva de pasear directamente al sistema sobre la parte trasera de una entidad.

Debe ser posible el importar dibujos CAD y clip Art. hacia la animación.

Usualmente es deseable el desplegar gráficas dinámicas y estáticas sobre la pantalla así como se ejecuta la simulación. Ejemplos de gráficas dinámicas son los bloques, cuadrantes, metros de nivel (quizá representando una cola), y dinámicamente actualizar histogramas y tiempo de mapeo. Un ejemplo de lo último puede ser el actualizar un mapa de un número en algunas colas así como la simulación se mueve a través del tiempo.

# Capítulo 5

## **METODOLOGÍA DE DISEÑO Y PLANEACIÓN DE REDES, MEDIANTE ANÁLISIS DE DESEMPEÑO POR TÉCNICAS DE SIMULACIÓN**

### **INTRODUCCIÓN**

Hoy en día los ambientes de red se caracterizan por una creciente mezcla de protocolos, medios físicos y tecnologías, todos ellos orientados a satisfacer las necesidades de negocio de aplicaciones críticas. Diseñadas para operar bajo el modelo de cliente servidor.

Dada su popularidad las LAN están creciendo en tamaño y cobertura y están siendo conectadas, no solo entre ellas si no también a bases de datos y aplicaciones distribuidas a través de WAN's esta complejidad convierte a las actividades de diseño de redes una necesidad proactiva.

Si para la habilidad de diseñar, los especialistas en redes, aplican metodologías tradicionalistas, pensando que son la única opción de implementar lo que ellos creen que es el mejor diseño, toman el riesgo de tomar una actitud reactiva, basado en técnicas, como es la medición del desempeño a través de herramientas de monitores, ó el aumento en gastos de diseño, con técnicas de análisis, en laboratorios con ambiente controlado ó técnicas donde se pierden tiempo diseñando modelos matemáticos.

La metodología que se presenta a continuación, con técnicas de simulación y modelado, es una excelente opción para diseñar redes, puesto que permite a los ingenieros estimar; tiempos, trabajo y dinero en la planeación de esta.

## 5.1 Introducción al diseño y planeación de redes.

Hoy en día las metodologías de diseño, reúnen ciertas características, como son<sup>7</sup>:

Diseño raramente repetitivos.

Algo de Arte.

Combinación de reglas.

Evaluación y selección de tecnologías.

Conocimiento de servicios, protocolos y otras experiencias.

Tradicionalmente tiene el enfoque, sobre la planificación de la capacidad y ante problemas en la red, como, incremento en el ancho de banda.

Las metodologías tradicionales, tienen el siguiente proceso en general:

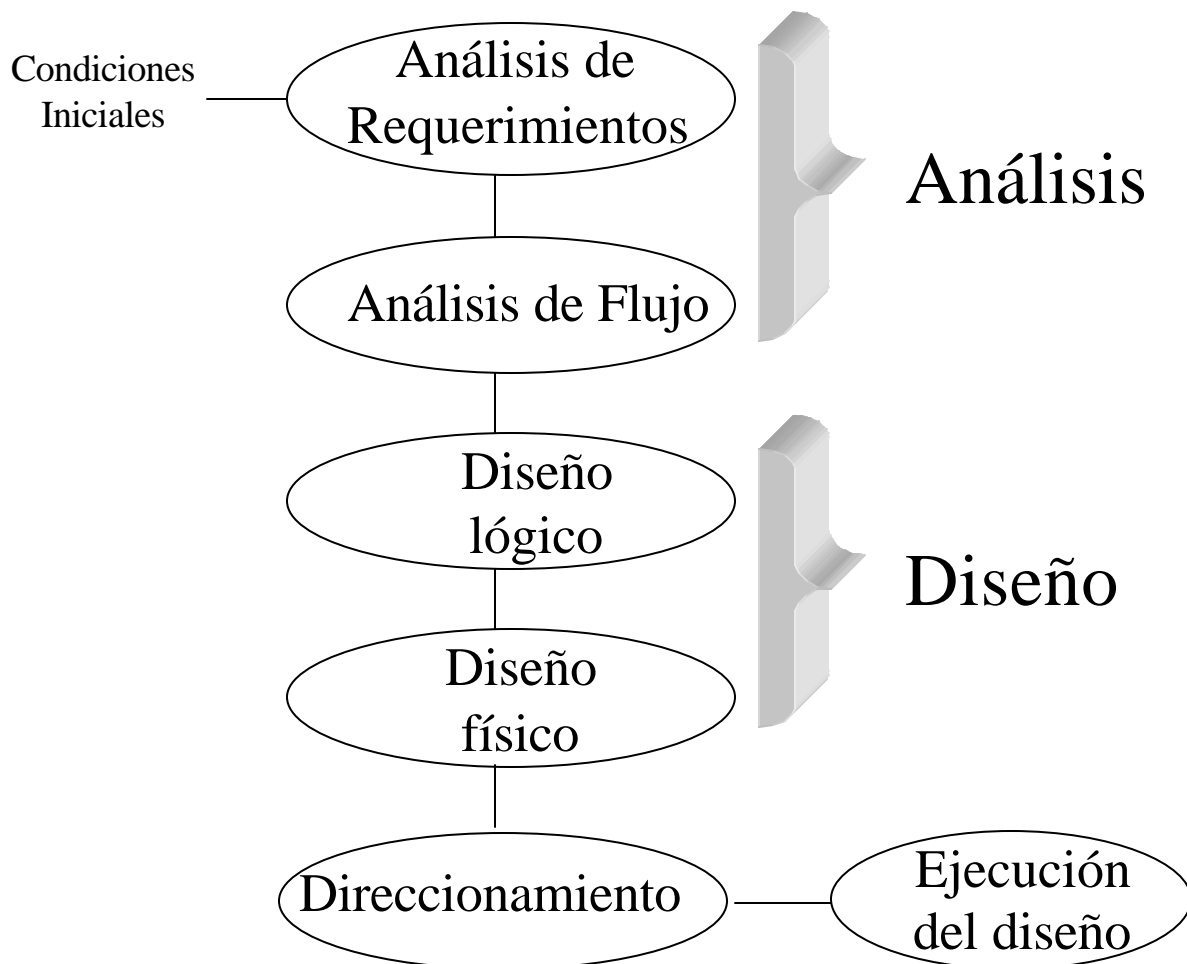


Figura 5.1  
Proceso de la metodología tradicional en diseño y planeación de redes.

## **Determinación de requerimientos.**

Consiste en :

- Identificar, capturar y comprender los requerimientos del sistema y sus características
- Desarrollar umbrales de rendimiento para distinguir entre servicios de bajo y alto rendimiento
- Determinar servicios específicos

Requerimientos de

- Usuarios
- Aplicaciones
- Servidores
- Red

Para obtener requerimientos desarrollar:

- Perfiles de usuario
- Entrevistas
- Grupos de análisis, tests en laboratorio

## **Análisis de flujo.**

Un flujo:

- Es transmitido durante una sola sesión de una aplicación/host (end-to-end)
- Es información sobre un conjunto de protocolos y aplicaciones que:

Tienen atributos comunes, tales como

- Direcciones destino y fuente > Fuente/Sumidero
- Tipo de información
- Enrutamiento
- Información extremo a extremo

Existen modelos de flujo

- Peer-to-peer
- Cliente/servidor Computación cooperativa
- Computación distribuida.

Hay tres tipos de flujos

- Individual
- Compuesto
- Backbone



## **Diseño lógico.**

- Sube tapas:
  - Establecer objetivos de diseño
  - Evaluar y seleccionar tecnología
  - Desarrollar un plan de ínter conectividad
  - Considerar Administración y seguridad de la red
  
  - Ubicar el equipo de la red:
    - Existen reglas par asegurar las ubicaciones más adecuadas
    - Esquemas
      - Centralizado
      - Distribuido
  - Construir diagramas físicos de la red (nodos. puertas, direcciones...)

## **Direccionamiento.**

- Sube tapas
- Establecer flujos de ruteo
- Manipular flujos de ruteo
- Desarrollar una estrategia de direccionamiento
- Desarrollar una estrategia de ruteo
- Considerar construir, según situación:  
Subredes, Superredes

## **Ejecución.**

Consta de:

- Instalación eléctrica
- Tendido de cables: rack, canaletas, patch panel, etc.
- Instalación de equipos
- Respalda y restaurar en caso de equipos existentes
- Configuración
  - Protocolos
  - Cuentas de usuario
  - Aplicaciones
  - Servicios, ...
  - Pruebas y análisis de sensibilidad

## 5.2 Metodología de diseño y planeación de redes mediante simulación y modelado.

¿Recuerdas cuando aprendiste a manejar? Acaso el instructor recito una letanía de reglas: como y cuando cambiar velocidades, las señales en el camino, como evitar a la policía. Ahora, años después manejes en el camino que manejas sin recordar aquellas reglas. ¿Revisas el retrovisor antes de cambiar carriles? Seguro lo haces. Solo lo haces por intuición (o lo piensas). La aprobación mecánica para manejar esta ahora muy dentro de tu experiencia. Esta sección contiene una lista metódica de reglas y pasos para seguir un proceso de simulación. Así como ganas experiencia con el proceso, muchas de aquellas reglas llegaran a ser grabadas en tus hábitos de simulación, y conocerás cuando es apropiado desviar algunas de estas reglas, o cuando añadir mas reglas a las tuyas.

En esta sección veremos una lista de reglas y pasos a seguir en un proceso de simulación, este no es parecido aun método científico. En general este es el siguiente proceso.

<b>DEFINIR EL PROBLEMA</b>	Definir los objetivos. Definir el sistema a simular.
<b>FORMULAR LA HIPÓTESIS</b>	Reunir los datos del problema (modelo). Desarrollo de la topología. Diseñar los flujos del trabajo. Validar las proposiciones del modelo con base lógica.
<b>DEFINIR LAS PRUEBAS DEL AMBIENTE</b>	Definiendo los escenarios de prueba. Construir el modelo. Verificar el modelo.
<b>DESEMPEÑO DE LOS EXPERIMENTOS</b>	Operación del modelo. Reunir resultados. Validar el modelo.
<b>ESTABLECER CONCLUSIONES</b>	Análisis de salida con estadística. Determinando el número de corridas. Reportando los resultados de la simulación.

Tabla 5.1  
Proceso de planeación y diseño de redes mediante simulación y modelado de redes.

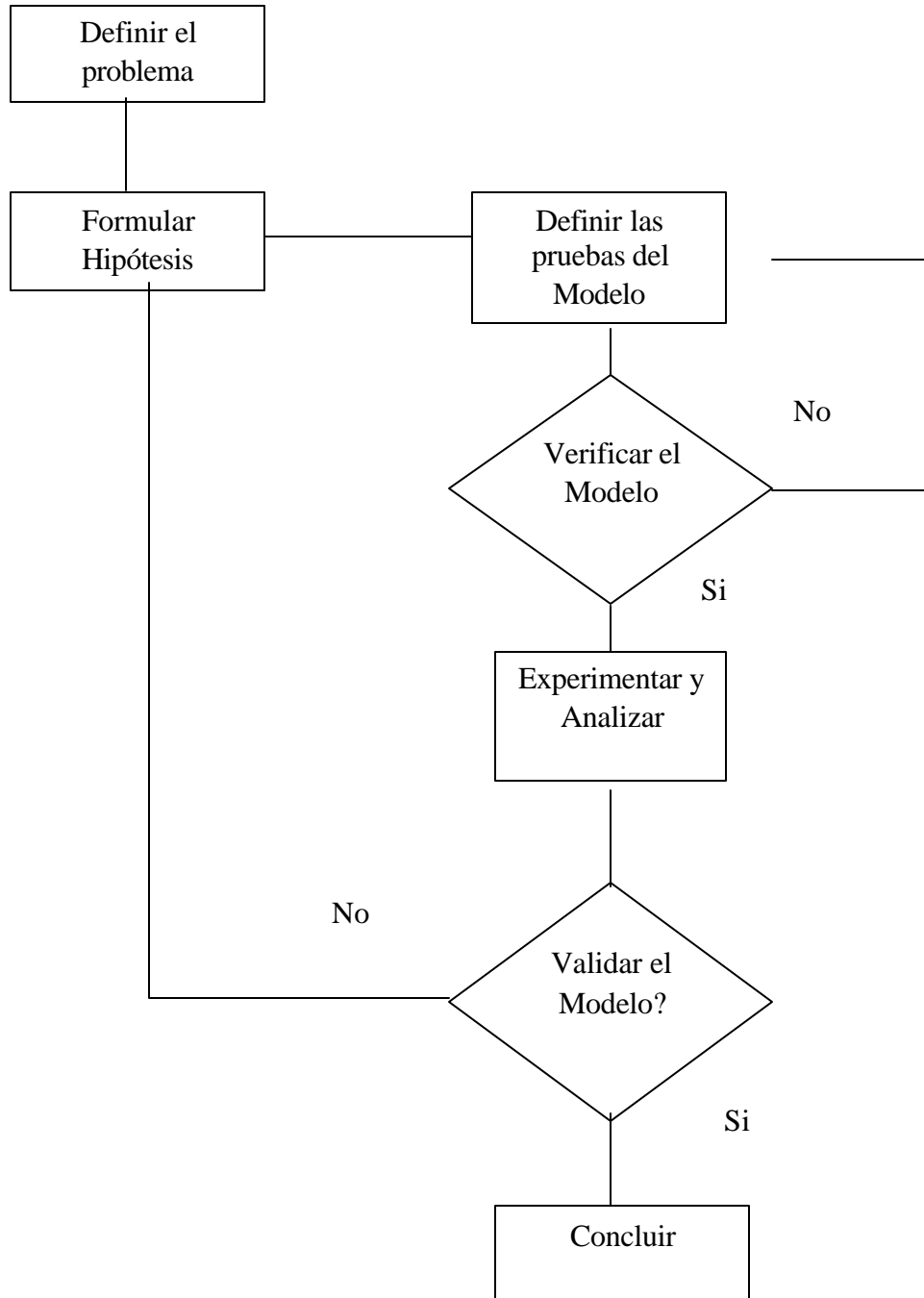


Figura 5.2  
Pasos en el estudio de la simulación.

## 5.2.1 Definiendo el problema.

Esta etapa se refiere a la fase de pensar y planear. Aquí, se reflexiona sobre el problema, y se toma la decisión de como representarlo. También se usa esta fase para determinar y definir los parámetros requeridos y obtener o registrar los datos. Las siguientes etapas son recomendadas para este paso:

**I. Definir los objetivos:** Una declaración del problema definido, debe indicar las siguientes características acerca del problema:

- reconocimiento de la existencia del problema.
- preguntas que serán respondidas.
- alcance del problema.
- mal éxito del sub-problema.

### II. Definir el sistema a simular.

¿El sistema que se va ha diseñar, tiene datos de entrada estocásticos, y salidas también estocásticas?, ¿Tienen entradas y salidas deterministas? ó ¿Una combinación de las dos?

## 5.2.2 Formulando la hipótesis.

### I. Reunir los datos del problema (modelo).

- ¿Qué información y datos serán necesarios?
- ¿Dónde será obtenida la información y los datos?
- ¿Cómo será manipulada la información y los datos?

Es importante para construir un modelo, saber que información hay en este, para poder evaluar el desempeño, de los elementos de red que se estén analizando, este es un criterio para poder empezar a obtener esta información.

Saber de los elementos del modelo a diseñar:

Los perfiles de ejecución  
Capacidades de procesamiento  
Características de desempeño  
Ambiente de operación

Por ejemplo, para obtener un alto nivel de modelado de una típica conexión LAN a LAN por medio de un enlace WAN, la siguiente información es requerida:

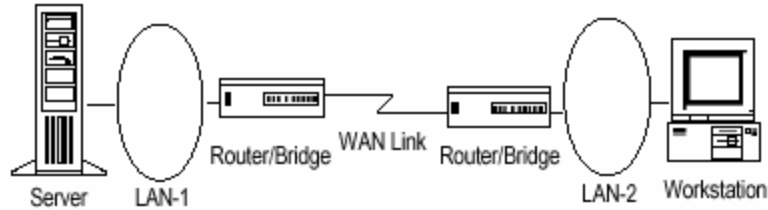


Figura 5.3  
Conexión LAN a LAN por medio de WAN.

### De los Servidores.

- ¿Cuántas transacciones por segundo será capaces de hacer el servidor?
- ¿Cuál es el Mip del servidor y el promedio de ejecución para cada tipo de transacción?
- ¿Cuál es el promedio en retardo de transacciones en el servidor?
- Si el acceso al disco esta echo para cada transacción, ¿Cuál es el número promedio de acceso al disco por transacción?
- ¿Cuales son las características de desempeño del uso del disco?

### De la LAN-1 y LAN-2.

- ¿Cuales son los anchos de banda de las LAN`s?
- ¿Si la información del Snifer es correcta y la LAN es Ethernet, cual es el promedio de la taza de colisiones de paquetes?
- ¿Cuál es el protocolo usado en la LAN?
- ¿Cuántas tramas por segundo soportara el adaptador de red de la LAN?

### De los Enrutadores y puentes.

- ¿Cuál es la taza de paquetes por segundo para los enrutadores?, Si es un puente, ¿Cuántas tramas por segundo puede el puente pasar por sus puertos?
- Si el enrutador o puente son un PC, ¿Cuál es la taza de Mip?, ¿Cuál es el modelo de PC?
- ¿Cuál es el tamaño de paquete utilizado?

### De los enlaces WAN.

- ¿Cuál es el ancho de banda del enlace?
- ¿Cuál es el protocolo usado en la WAN?
- ¿Cuáles son las retransmisiones con este enlace?

### De las estaciones de trabajo.

- ¿Cuál es el Mip para las estaciones de trabajo?
- ¿Cuánto tiempo tarda la estación de trabajo en ejecutar el software?
- ¿Cuáles son las características de desempeño del disco duro local?

## En general.

- a. ¿Cuántas transacciones por segundo/minuto se realizaron en las estaciones de trabajo, durante el tiempo de saturación?
- b. ¿Son las estaciones de trabajo Tontas?
- c. ¿Qué formato usaran VT 100, 3270, telnet?
- d. ¿Qué aplicación se correrá, SQL, RPC, etc.?
- e. ¿Que sistema operativo de red se utilizara?

Un componente del análisis de los sistemas de trabajo es la identificación de las apropiadas distribuciones entre llegadas. Si es posible se debe estudiar el camino por el cual el trabajo llega al sistema real. En particular, uno esta interesado en descubrir la frecuencia de llegadas d Antes de empezar a construir tu modelo de red, contesta la siguiente pregunta:

¿Que información tratas de extraer del modelo de red?

- ¿Estas tratando de encontrar una respuesta típica en tiempo de alguna transacción?
- ¿Estas tratando de deducir el número de enrutadores que se necesita para mantener un nivel de utilización del recurso?
- ¿Son suficientes los anchos de banda en el enlace de la WAN?
- ¿Queremos saber la latencia en el proceso del puerto del servidor?

Tu modelo tomara diferentes características dependiendo de las preguntas que quieras contestarte.

Por ejemplo, Si estas interesado solo en la utilización del servidor, no tiene sentido modelar cada elemento de la red, entonces tu modelo se desarrollara en el recurso que te interesa modelar, en este caso el servidor.

De esta manera solo especificaras las proporciones de trafico de la red, que llegara al puerto del servidor y por lo tanto obtendrás la información que necesitas, y tu modelo corredera más rápido.

Si en cambio quieres averiguar el tiempo de la contestación por una transacción, entonces tendrás que involucrar otras características que te contesten tus preguntas primitivas.

Predecir sustenta las distribuciones más comunes. En medio de estas están la Exponencial Negativa, Poisson, Erlang, Normal, Uniforme e Hiper-exponencial (que también es distribución de Poisson). Una importante virtud del proceso Poisson es que esta en poca memoria, la probabilidad de una llegada en Inter dependencias  $t$ , depende solo de la duración de  $t$ , y no sobre cuando el arribó previo ocurrió. En otras palabras, las distribuciones de Poisson son un atractivo aleatorio, y se asemejan a la distribución más natural, no controlados por

humanos, eventos: el camino de los mensajes llegan a un sistema de computo; la distribución de tráfico sobre una autopista, etc.

Trata de aplicar una distribución teórica para describir una distribución real, al final, ninguna de las distribuciones existentes serán perfectas; así escoge la que más cerradamente se acerca al proceso de simulación.

Sí no te queda claro, este tema será profundizado en uno de los capítulos posteriores.

Una vez que se ha definido el tipo de la información requerida, se obtiene información tanta como sea posible. Los siguientes mecanismos son recomendados:

- Búsqueda literaria.
- Recolección de los materiales fuente, documentos y reportes.
- Generación de datos artificiales.
- Recolectando datos por experimentación.
- Consultando con especialistas y expertos.

## **II. Desarrollo de la topología.**

Describiendo cada parámetro y variable en términos de la información obtenida:

- Que es su definición y símbolo (nombre de recurso, transacción y su división en mensajes, procedimientos, etc).
- Unidades de parámetros de medidas.
- Rango de valores.
- Características: simple, multivalor, aleatorio, controlado, etc. Los valores dinámicos cambian de acuerdo a los valores de otros parámetros. Los parámetros estadísticos son aquellos de quien afecta el valor a los procesos estocásticos y cambios en el sistema.

## **III. Diseñar los flujos del trabajo.**

Desarrollar un modelo que involucra dos tareas:

- Desarrollar una representación del sistema, y
- Desarrollar una representación del trabajo hecho por el sistema.

Estos están interrelacionados: el nivel de abstracción de las dos debe corresponder porque un modelo es una representación de un sistema real, y porque modelar es un sistema de abstracción. La clave para el proceso de modelar es abstrayendo un diseño del sistema hacia un modelo designado. Así, un modelo es una abstracción de un sistema, y representa una vista particular de ese sistema.

El modelo no es el sistema real; lo mejor que puede ser esperado es que el modelo despliegue una conducta análoga al sistema siendo modelado para el dominio del problema que es simulado.

Todo lo que puedes hacer es establecer un nivel de confianza razonable que resulta comparar la realidad dentro de límites predefinidos, y esperar a conocerlos.

En general, existen dos aprobaciones para modelar:

**Síntesis.** En esta aprobación (bottom-down), primero representas elementos individuales, y entonces haces suposiciones así como simplificar aquellos elementos.

**Descomposición.** La reversa de la síntesis. En esta aprobación (top-down), primero ves al sistema como un todo, y entonces descompones el sistema hacia sus elementos constituyentes, progresivamente añadiendo detalles como sea necesario.

En adición, puedes definir el modelo, como, el trabajo opera sobre el sistema, o donde el sistema opera sobre el trabajo. En un alto nivel de abstracción recomendamos que tomes una vista posterior. La simulación trabaja con la mejor vista que el sistema (los recursos) opera sobre el trabajo (los mensajes). De esta manera, animas a representar las entidades dinámicas en el modelo en una vía explícita. Porque los sistemas largos son mejor aprovechados vía descomposición, se recomienda que sigas la aprobación (top-down) para el modelo de sistemas.

Se recomienda una aprobación (top-down) para decidir si la partición del modelo es verticalmente u horizontalmente. La partición vertical involucra definir el sistema entero como un pequeño grupo de cajas negras. Después de que descompones cada una de esas cajas negras hacia mas subcomponentes detallados. Con la partición horizontal, divides al sistema hacia una serie de modelos diferentes, cada modelo representa una escena del completo flujo de trabajo. Estas dos aprobaciones no son exclusivas de cada otra. Verdaderamente, en algunos casos donde el sistema bajo la simulación consiste de elementos operando con diferentes escalas de tiempo, tendrás que cambiar un sistema hacia modelos separados.

#### **IV. Validar las proposiciones del modelo con base lógica.**

Volver a visitar, las suposiciones iniciales acerca del modo en que opera el modelo y acerca de cómo representa la realidad. Por ejemplo, tal vez, el mejor camino para obtener aquel retardo estimado es usando el retardo de la cola antes que el recurso del ciclo ocupado. Otras preguntas que puedes querer contestar en esta escena son:

- ¿Que funciones son determinísticas, cuales son no determinísticas?
- ¿Cómo aproximamos aquellas funciones en el modelo?



- ¿Qué factores ambientales afectan el modelo? (los empleados hacen pausas para el café).
- ¿Cuáles son las interacciones entre el hombre y el sistema? ¿Hombre y ambiente? ¿Sistema y ambiente?

### **5.2.3 Definir las pruebas del ambiente.**

#### **I. Definiendo los escenarios de prueba.**

Es importante el desarrollar en avanzado un modelo de medida del sistema. La medida del modelo es derivado de la descripción de análisis del modelo, y define las medidas y los puntos de medidas requeridos para una completa descripción de la ejecución del sistema, en un sentido operacional, al deseado nivel de detalle.

Nunca sabrás que tan bien o mal esta tu análisis hasta que valides tus resultados en comparación con el sistema de medidas.

Basado en los objetivos de la simulación, se identifican que variables de simulación en particular quieres recoger, y bajo que condiciones. El software de simulación te deja observar las dinámicas del sistema simulado, pero para eso se pierde tiempo de ejecución. Debes decidir si el interés es observar las dinámicas del sistema u obtener el valor significado de un parámetro específico con un alto grado de confianza. Después de tomar estas decisiones serás capaz de determinar que características específicas debes ejercitar.

#### **II. Construir el modelo.**

Por rutina, un modelo que se esta usando, debe ser verificado, corregido, y verificado nuevamente, hasta que un nivel razonable de confianza en los resultados estén archivados. Cuando se modela, se procura aplicar el principio de Parsimony: comienza el modelo de construcción con el mejor entendimiento y elementos obvios del sistema y así como vayas ganando confianza, ya que el modelo este trabajando como se esperaba, añade más partes a este. Quizá la primera vez que se introduzca parámetros de entrada, se introducirán con carencia de mejor información, suposiciones razonables. Pero así como el proceso de construcción del modelo continuo, debes siempre ir atrás a los parámetros de entrada, y corregirlos como sea necesario. Muchas veces, también se encontraran resultados que claramente están fuera de las expectativas, y las cuales pueden ser llevadas a equivocados parámetros de entrada. Por ejemplo, puedes tener introducido un perfil con una impropia escala de tiempo de trabajo (tictac). Que se, ha indicado que los valores se refieren a, decir, minutos, cuando en realidad se refieren a segundos. En un software flexible, se trata de detectar muchos de estas condiciones como sea posible.

### III. Verificar el modelo.

Verificación es el proceso de revisar que el modelo fue construido de la forma que intentaste.

El software hace un extenso chequeo de la sintaxis, rango de parámetros, tan bien como un chequeo transversal de referencias. Tranquilamente, hay ciertos elementos semánticos que no se pueden verificar automáticamente. Es todavía tranquilo para ti verificar que las cantidades y los valores que has introducido son exactos.

En esta escena todavía no has corrido la simulación y no puedes decir si los resultados tienen sentido, pero puedes manualmente revisar que tus suposiciones y que el modo en que el modelo ha sido construido profundamente representa el sistema bajo estudio. Para este proceso usa:

- Observación. Imprime el modelo y compáralo con el sistema.
- Opiniones de expertos. Igual si eres un experto sobre el sistema siendo simulado, no serás lastimado al recibir una segunda opinión.
- Intuición y experiencia.

Recuerda, tú verificas al identificar errores conceptuales y de abstracción. Estadísticamente se definen tres clases semejantes de errores que se presentan frecuentemente:

**Tipo de errores I.** Una hipótesis puede ser malamente rechazada igual si esta correcta. Esto puede ocurrir porque hay siempre una probabilidad finita de que un error puede ocurrir. Para decrementar esta probabilidad de error, debes incrementar las muestras usadas para probar la hipótesis, consecuentemente incrementando las corridas de la simulación. Después, en la parte de experimentación y análisis del proceso de simulación aprenderás como hacerlo.

**Tipo de errores II.** Una falsa hipótesis que es aceptada como verdadera. Nuevamente, después cubriremos como puedes aplicar nivel de confianza de análisis para maximizar la exposición a este tipo de error, pero la siguiente máxima aplica: la mejor medida contra errores tipo I y tipo II es incrementar el número de medidas.

**Tipo error cero.** Has preguntado, la pregunta errónea y el modelo no representa la realidad. El resultado es un modelo que hace las cosas mal, solo de una manera rápida y exacta gracias a la computadora. Los resultados pueden ser validos sobre un terreno estadístico, pero son sin sentido en la práctica. Este es el más grande error para evitar y este es el único error que mas debes tratar de prevenir a este punto del proceso de simulación.

Una común tendencia que debes evitar es incluir muchos detalles acerca de la realidad hacia tu modelo. Recuerda que modelar es un proceso de abstracción; no de duplicación, y esos detalles triviales de simulación que no añaden información importante a tus resultados solo causan degradación en la corrida de la simulación. Por el otro extremo, evita sobre simplificar. Lo posterior es el camino más expeditivo para incurrir en un error tipo cero.

## **5.2.4 Desempeño de los experimentos.**

### **I. Operación del modelo.**

En esta fase sabremos como obtener calidad en los resultados, puedes probar diversas situaciones “y si”, o estudiar condiciones dinámicas en base a un tiempo real.

Siempre se debe recordar que la simulación puede ser y usualmente es sensible para las condiciones iniciales. Por ejemplo, inicialmente todas las colas están libres, y si tú calculas el tiempo medio de las estadísticas de las colas libres iniciales con los valores que obtienes una vez que las colas han alcanzado su longitud verdadera de tiempo medio, encontraras que el completo significado será cambiado porque el valor inicial es cero. Condiciones de comienzo pueden cambiar las estadísticas y llevarte a realizar conclusiones falsas. Haz correr tu simulación un tiempo más largo para que los efectos de los valores iniciales sean diluidos, bajando su impacto sobre el resultado final.

Antes de preguntar cuanto tiempo debe de correr la simulación, debes definir la exactitud deseada. Cuanto tiempo y cuantas veces debe correr tu modelo depende del sistema siendo simulado. Muchas de las simulaciones en computadora o sistemas de diseño de redes son del tipo estado constante, solo en cualquier otro ambiente interesado, con comportamiento equilibrado. Que es, convergen a un valor limitado (número promedio de mensajes en una cola; promedio de retardo de mensajes, etc). Entonces hay simulaciones de terminación o transitorias que negocian con los procesos perecederos. Esto se completa después de que un evento o secuencia de eventos predeterminados ocurran (un número específico de mensajes ha ocurrido, o un cierto tiempo ha expirado). Para este posterior tipo de simulación la pregunta no es cuanto debe correr la simulación; pero cuantas duplicaciones o corridas debe ser para alcanzar una precisión específica. Este punto será desarrollado, cuando se establezcan conclusiones, porque se hacen análisis en la precisión de los resultados.

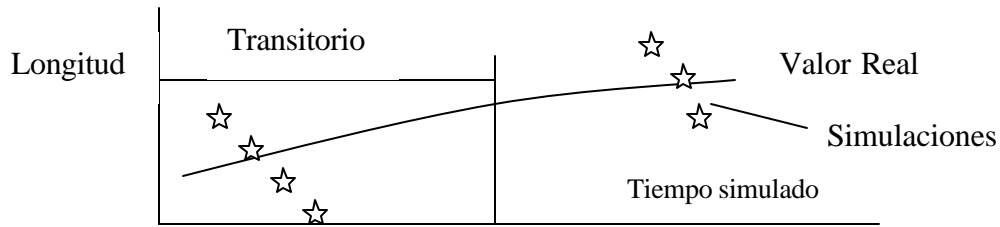


Figura 5.4  
Estados de ejecución de la simulación.

## II. Reunir resultados.

Los resultados de parámetros y variables seleccionadas son automáticamente presentados y almacenados para un futuro análisis. Puedes también imprimir estos resultados, o exportarlos a otras aplicaciones para un análisis detallado.

## III. Validar el modelo.

Validación es el proceso de determinar que el sistema real siendo estudiado es exactamente representado por el modelo, esto establece un aceptable nivel de confianza que los resultados corresponden con la realidad.

Esta énfasis se basada sobre el hecho de que no hay nada peor que un creíble resultado malo. La diferencia entre esta validación y las validaciones que ejecutaste en las fases previas es que ahora tienes una muestra de resultados contra los cuales validar el modelo. Puedes ahora aplicar las siguientes técnicas para identificar trampas potenciales:

- Comenzar con un modelo conceptual (simulación de alto nivel) para obtener un resultado general que puede ser usado como un punto de referencia genérico. Este modelo conceptual puede ser cualquiera de los dos un paso intermedio o un fin el mismo.
- Compara los resultados con medidas parciales de un sistema real o con otros sistemas de simulación, o con medias históricas.
- Ejecutar pruebas estadísticas (análisis de varianza).
- Ejecutar análisis sensible. Después de los datos por pequeños grupos, mas extremos máximos y mínimos.

En esta etapa debes ver los síntomas típicos de una simulación inestable:

**Síntoma:** mas mensajes de los esperados son generados. El retardo no parece congruente con los recursos, capacidad, etc.

**Enfermedad:** introdujiste datos inválidos, o hiciste suposiciones malas. Revisar si hay un tipo en el número de mensajes que generas, o en los valores definiendo las capacidades de procesamiento de recursos.

Debes revisar también si cada transacción fluye al camino que has intentado, y si no esta configurando un estado de transacción o un recurso inapropiadamente.

**Síntoma:** los mensajes parecen no alcanzar el recurso deseado.

**Enfermedad:** verifica que estas configurando las destinaciones apropiadas cuando regresas los mensajes. Tal vez el tamaño de la cola que especificaste y el número de mensajes descartados debido a las condiciones de la cola llena.

**Síntoma:** los resultados varían ampliamente de corrida a corrida.

**Enfermedad:** esto no debe ser un problema con la simulación. Esto solo podría ser que el sistema de simulación es muy caótico (muy sensible para las condiciones iniciales, o muy pequeño para variaciones aleatorias). También parecidamente es que las veces de la impropia simulación están siendo aplicadas. Define si tu simulación debe finalizar basado sobre una terminación o sobre un estado de condición segura. Una condición de terminación es detenida por un evento natural (5:00 PM es la hora para hacer salida); o el número total de solicitudes esta siendo servida (finalizar después de que 1000 mensajes han sido procesados).

**Síntoma:** salida inconsciente. Si usas más recursos usualmente esperas retardos bajos.

**Enfermedad:** podrías estar introduciendo elementos los cuales se comportan en el sistema traspasan las estadísticas ampliamente. Se debe validar la simulación o asegurar que no se estén introduciendo elementos los cuales reaccionan contrariamente a lo esperado. Por ejemplo, una gran carga laboral en una línea de proceso podría estar estorbando a un recurso. Así reduciendo el grupo de trabajo otro recurso baja la línea obtenida. El resultado esta en que mientras para ver una gran visualización en un recurso debido a la alta razón de trafico, actualmente ves que el recurso es menos usado. También ten cuidado de ciertos intereses en el fenómeno de la cola tales como la paradoja Braess. Una paradoja descubierta por la búsqueda de operaciones Dietrich Braess, quien describió como ciertas topologías de redes, incrementan la capacidad en redes electrónicas gestionadas actualmente reduce la actividad de comunicación.

**Síntoma:** corrida de la simulación fuera de memoria.

**Enfermedad:** podrías estas exigiendo un gran esfuerzo a los recursos de tu computadora porque la propia simulación requiere de almacenar un largo número de mensajes o porque has causado inadvertidamente un circuito cerrado en el mensaje con una condición correr.

Estas proposiciones nunca son redundantes para propósitos de verificación y validación:

- Documento, documento, documento.
- Seguir un sabio paso de modelo de construcción.
- Depurar: usa técnicas estándares de depuración: usa la facilidad de despliegue de mensaje para revisar el estado de la simulación. Usa la facilidad de grabado de simulación para una revisión posterior pasó por paso; o usa la función por dentro de rastro de mensaje.

Con el software, se puede observar los efectos de la simulación en tiempo real, una característica útil para identificar recursos de secuencias de saturación, o cuando los resultados parciales son sospechosos, permitiéndote suspender la simulación antes que termine. Claro, también dará un resultado total resumido para una posterior evaluación, vía herramientas gráficas disponibles, o para análisis adicional vía otras aplicaciones Windows (los resultados pueden ser exportados a Excel o Lotus 1-2-3, por ejemplo). Esto habilita el mostrar los resultados del modelo, y el ejecutar adicionalmente revisiones de líneas estadísticas distanciadas deberá proporcionar una confianza que la simulación esta verdaderamente prediciendo la realidad.

Debes ejecutar pruebas sensibles para validar que mas mensajes frecuentes tendrán consecuencias en tiempos de respuesta (para muchos casos imaginamos, tu simulación no debe comportarse de una manera caótica). Debes también correr pruebas máximas y mínimas. Cuando especificas las distribuciones, prueba la simulación con mínimas distribuciones de desviación. Finalmente, si estas tratando de encontrar una óptima solución, debes buscar los parámetros apropiados y ejecutar mas corridas, hasta que coloques un modelo satisfactorio. Entonces tendrás que hacer las conclusiones, y documentar tus recomendaciones. Tal vez es hora de comprar otro servidor para tu LAN.

## **5.2.5 Establecer conclusiones.**

### **I. Análisis de salida con estadística.**

El objetivo del análisis estadístico es determinar la varianza, una medida del error, de la estimación; o bien determinar la cantidad de observaciones requeridas para lograr una precisión deseada.

Cuando una simulación tiene cualquier tipo de comportamiento estocástico incorporado en su estructura, llega a ser necesario ver el resultado de una corrida de producción como una prueba experimental. Es decir sí:

Los resultados de una simulación, son considerados como experimentos y cumplen con las siguientes características<sup>6</sup>:

- 1.- Un conjunto grande de causas ( $y_1, \dots, y_2$ ),
- 2.- Independientes unas de otras,
- 3.- Que actúan sumando sus efectos,
- 4.- Siendo cada una de poca importancia.

Los resultados de una simulación estocástica son mas difíciles de interpretar que aquellas de modelos determinísticos. Muestras de procesos las cuales necesitan ser controladas para reducir los efectos de los errores, son:

- Viendo una desviación estándar.
- Estableciendo la longitud de las corridas.
- Importando datos a una hoja de cálculo y ensayando análisis de varianza y métodos de análisis de confianza proporcionados por una herramienta.
- La precisión de los resultados esta en proporción directa de la exactitud de los parámetros de entrada, pero las simulaciones tienen diferentes niveles de sensibilidad para estos valores de entrada.

Una muestra de prueba larga te permitirá explotar las ventajas definidas por el Teorema de Limite Central. Este teorema establece que si tomas una muestra larga de  $N$  valores de cualquier distribución, calcular sus media ( $X_m$ ), y repetir este procedimiento muchas veces, la distribución de estos diversos medios serán distribuciones normalmente.

Considere que para una variable de salida o estado dada (por ejemplo: clientes servidos, tiempos de espera de cada cliente en la fila, etc.), el simulador genera la siguiente secuencia de valores  $\{Y_1, Y_2, \dots, Y_n\}$ . Antes de tomar una decisión, es necesario procesar estos valores para obtener una estimación del valor medio y una medida del error de esta estimación.

La estimación puntual del valor medio  $Y_m$  se hace calculando la media muestral  $Y_m$ ; ésta última, como se planteó en un capítulo anterior, puede ser basada en observaciones o basada en tiempo. Para una muestra compuesta por  $n$  elementos, el promedio basado en observaciones se calcula como sigue:

$$Y_m = \frac{1}{n} \sum_{i=1}^n Y_i \quad (1)$$

El promedio basado en tiempo, considerando que cada valor  $Y_i$  tiene una duración  $t_i$ , se calcula de la siguiente manera:

$$T = \sum_{i=1}^n t_i \quad (2)$$

$$Y_m = \frac{1}{T} \sum_{i=1}^n Y_i t_i \quad (3)$$

En el caso ideal, el estimador  $Y_m$  será no sesgado (*unbiased*); es decir, se cumplirá:

$$E(Y_m) = \mu_Y \quad (4)$$

Sin embargo, por lo general el estimador tendrá cierto sesgo (*bias*) b:

$$E(Y_m) = \mu_Y + b \quad (5)$$

Lo deseable es contar con un estimador sin sesgo, o por lo menos con un sesgo pequeño. Para la estimación del intervalo de validez se requiere estimar la varianza del estimador puntual. La varianza real del estimador puntual  $S^2_{Y_m}$  es estimada con el estimador  $S^2_{Y_m}$ . En el caso ideal, se cumplirá:

$$E(S^2_{Y_m}) = \sigma_{Y_m}^2 \quad (6)$$

Sin embargo, por lo general el estimador tendrá cierto sesgo B:

$$E(S^2_{Y_m}) = B\sigma_{Y_m}^2 \quad (7)$$

Lo deseable es que  $B$  sea igual a 1. Si ambos estimadores ( $Y_m, S^2_{Y_m}$ ) son no sesgados, el siguiente parámetro cumple con la distribución  $t$  para cierto número de grados de libertad  $f$ .

$$t_f = \frac{Y_m - \mu_Y}{S_{Y_m}} \quad (8)$$

Por lo tanto, el siguiente intervalo contendrá a  $\mu_Y$  con un nivel de confianza  $100(1-a)\%$ :

$$Y_m - t_{\alpha/2, f} S_{Y_m} \leq \mu_Y \leq Y_m + t_{\alpha/2, f} S_{Y_m} \quad (9)$$

Cuando el tamaño de la muestra  $n$  es mayor que 30, el siguiente parámetro cumple con la distribución normal:

$$z = \frac{Y_m - \mu_Y}{S_{Y_m}} \quad (10)$$

En este caso, el intervalo de confianza es:

$$Y_m - z_{\alpha/2} S_{Y_m} \leq \mu_Y \leq Y_m + z_{\alpha/2} S_{Y_m} \quad (11)$$

Tanto los valores de  $t$  y  $z$  están disponibles en tablas en función de  $f$  y  $a$ . El mayor problema es obtener un estimador  $S^2_{Y_m}$  no sesgado ya que los valores  $Y$  están correlacionados.





$$S_{Y_m}^2 = \frac{1}{R} S_{Y_{mr}}^2 = \frac{1}{R(R-1)} \sum_{r=1}^R (Y_{m_r} - Y_m)^2 \quad (13)$$

y el grado de libertad  $f$  es igual a  $R-1$ .

Como ejemplo, suponga que se hacen 4 simulaciones del servicio de mantenimiento de aviones. Para el caso de dos talleres, la utilización media del taller 1 (promedio basado en tiempo) resultó ser 0.808, 0.875, 0.708 y 0.842. Entonces, los estimadores son:

$$Y_m = \frac{0.808 + 0.875 + 0.708 + 0.842}{4} = 0.808 \quad (14)$$

$$S_{Y_m}^2 = \frac{(0.808 - 0.808)^2 + (0.875 - 0.808)^2 + (0.708 - 0.808)^2 + (0.842 - 0.808)^2}{4 \times 3} = 0.036^2 \quad (15)$$

Para un nivel de confianza de 95% y grado de libertad 3,  $t_{0.025,3} = 3.18$  (obtenido de tabla anexada  $t_{\alpha/2}$ ); entonces, el correspondiente intervalo de confianza es:

$$\mu_y = 0.808 \pm 3.18 \times 0.036 \quad (16)$$

Esto es, la utilización media real pertenecerá al siguiente intervalo con un 95% de certeza:

$$0.694 \leq \mu_y \leq 0.922 \quad (17)$$

## II. Determinando el número de corridas.

Frecuentemente se especifican el error tolerable  $\epsilon$  y el nivel de confianza  $100(1-\alpha)\%$  deseado para la estimación; entonces, se deberá estimar la cantidad de repeticiones que serán necesarias para alcanzar dicho objetivo. Suponga que inicialmente se realizan  $R_0$  repeticiones (no menos de 4). Con estos resultados se estima  $S^2_{Y_{m0}}$ . Utilizando la definición del intervalo de confianza, ecuación (9), y la definición de  $S^2_{Y_m}$ , ecuación (13), se obtiene:

$$\frac{t_{\alpha/2, R-1} S_{Y_{m0}}}{\sqrt{R}} \leq \epsilon \quad (18)$$

Si se resuelve esta desigualdad, se tiene que  $R$  es el menor número entero que cumple con  $R \geq R_0$  y con:

$$R \geq \left( \frac{t_{\alpha/2, R-1} S_{Y_{mr0}}}{\varepsilon} \right)^2 \quad (19)$$

Debido a que,  $t_{\alpha/2, R-1} \geq z_{\alpha/2}$ , una estimación inicial de  $R$  se obtiene con:

$$R \geq \left( \frac{z_{\alpha/2} S_{Y_{mr0}}}{\varepsilon} \right)^2 \quad (20)$$

Determinado  $R$ , se procederá a realizar  $R - R_0$  simulaciones adicionales. Si el nuevo intervalo de confianza es mayor que el especificado, se deberán repetir los cálculos tomando  $R$  como el nuevo  $R_0$ .

El método de análisis de salida, también como otros métodos disponibles en otros software de estadística puede ayudarte para obtener una estimación exacta de una medida de ejecución para un modelo. El software te permite observar cuando el modelo llega a ser un estado constante y permite restablecer las estadísticas al tiempo de tu elección o programáticamente. Siempre recuerda que, al final, que tan exactos sean los resultados de una simulación representa la ejecución del sistema dependiendo de más de la validez de las suposiciones usadas para construir el modelo y caracterizar su modo de trabajo.

### III. Reportando los resultados de la simulación.

Una vez que has completado el proceso de simulación satisfactoriamente, es tiempo de publicar los resultados. Un formato típico para reportar los resultados de tu simulación es el siguiente:

1. Título de la página.
2. Tabla de contenidos.
3. Declaración de objetivos.
4. Metodología y suposiciones.
5. Conclusiones y recomendaciones.
6. Descubrimientos.
7. Apéndice:
  - Modelo.
  - Resultados.

# Capítulo 6

## EVALUACIÓN DEL CASO DE ESTUDIO

### INTRODUCCIÓN

Realizaremos la evaluación de un sistema de redes seleccionando un caso de estudio local, mediante la metodología y fundamentos teóricos propuestos en este trabajo, para realizar análisis de los resultados.

El propósito del análisis de los resultados de un sistema es predecir el desempeño del sistema o comparar entre dos o más alternativas. Como lo comentamos en el capítulo anterior, debido a la naturaleza estocástica del sistema, las salidas serán aleatorias; esto requiere que los resultados sean tratados estadísticamente. Si el desempeño del sistema se mide en base un parámetro  $q$ , el procesamiento de los resultados de varias simulaciones producirá una estimación  $qm$  de dicho parámetro.

Con el objetivo, de corroborar las mediciones del simulador, se construye el caso de estudio en un sistema real, mediante metodología tradicionalista.

Finalmente, realizaremos conclusiones de los resultados obtenidos del caso de estudio y también realizaremos conclusiones del error en el parecido de los resultados de los métodos, para determinar, que la simulación es realmente una alternativa para el diseño y planeación de redes.

## 6.1 Caso de estudio.

### 6.1.1 Definiendo el problema.

El caso de estudio consiste en la planeación y diseño de un sistema informático, con aplicación cliente / servidor, para la empresa de AXTEL Telecomunicaciones, donde el objetivo es, encontrar el desempeño en las redes LAN y WAN, que soporten el trabajo requerido y de esta manera saber el comportamiento de la operación de la misma para cambios futuros.

Para efectos de corroborar los resultados del simulador, se realizaron pruebas sobre un sistema real (red de computadoras), que se construyo en el laboratorio de Cabletron, instalado en E.S.I.M.E. Zacatenco.

### 6.1.2 Formulando la hipótesis.

#### 6.1.2.1 Reunir los datos del problema (modelo).

La red que se creo y que a partir de este, se generara el modelo para el simulador, tiene el siguiente esquema.

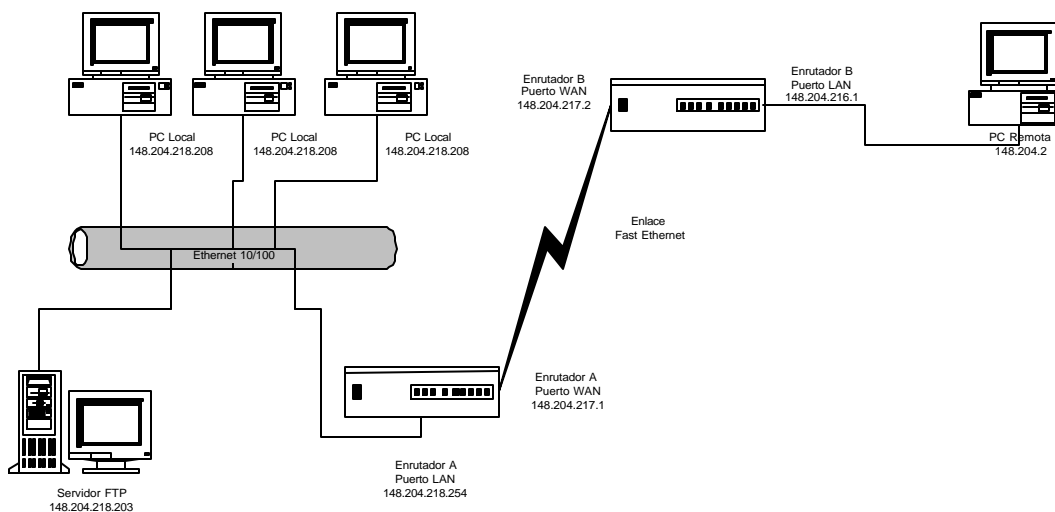


Figura 6.1  
Red a modelar c/s.

**El equipo y el software utilizado fue el siguiente:**

El servidor FTP y PC's:  
Marca: IBM  
Modelo: is 300  
Procesador: Pentium II  
Tarjeta de red 10/100 Mbps

RAM: 256 Bytes

Disco Duro: La velocidad de transferencia del disco es aproximadamente 16 Megabytes por segundo, hay una media de del buffer de 10 E/S con un tiempo de búsqueda de 10 ms. Por cada E/S para reparar el dato dinámico.

La aplicación FTP:

IIS (information internet service), Versión 4.0 Microsoft

Enrutadores (Routers):

Marca: Enterasys

Modelo: SSR 8000, Rev.1

Tarjeta de control: FAST ETHERNET de 16 puertos, con RJ-45 a 10Mbps

CPU: 83.33 MHZ

Procesador: R7000

Procesamiento de paquetes, 15 de millones por segundo.

Concentrador (Hub):

Marca: Allied Telesys International

Modelo: AT-3016sl

Puertos: 16 tipo RJ-45

Trabaja de acuerdo a la norma IEEE 802.3 10BaseT.

Monitores y Analizador de protocolos:

SPYNET Observer Software California LL.

Link Monitor Performance Houston LL.

Siguiendo la metodología propuesta en este trabajo se desarrolla los siguientes temas.

### **6.1.2.2 Desarrollo de la topología.**

Construimos un modelo que nos ayudará a estimar la utilización esperada para redes LAN's y WAN's con un servidor FTP hipotético con las siguientes características:

- El servidor soporta 100 peticiones concurrentes, esta conectado a una red local (LAN) Fast Ethernet a 10 Mbps, junto con un grupo de usuarios y de ahí a una línea enlace (WAN) Fast Ethernet que permiten transmitir con una velocidad de 10 Mbps y conecta a un usuario remoto.
- El usuario remoto esta conectado a la red local Fast Ethernet con un enrutador razonado a 640,000 paquetes por segundo para el tamaño del paquete que se ha definido (1500 bytes por paquete).

Basado sobre las proyecciones, esperas las siguientes cargas de trabajo:

- 100 golpes por media hora.
- 70% de las peticiones ocurrirán por los usuarios locales a, el servidor FTP con un tamaño de 100 +/- 50Kbytes.
- 30% de las peticiones desde el usuario remoto con un tamaño promedio de 100 +/- 50Kbytes cada uno.
- Todas tus peticiones son menos de 200 Kbytes.

Basado en esta información, deseas predecir lo siguiente:

- a) ¿Cuál será el promedio de utilización de la red Fast Ethernet Local ?
- b) ¿Cuál será el promedio de utilización de la red Fast Ethernet Enlace?

Si realizamos una aprobación para modelar por síntesis, podremos observar que no es necesario, representar en el simulador, toda la red, puesto que el objetivo es saber el desempeño de las redes LAN y WAN, entonces podremos representar al enrutador remoto y al usuario remoto, como un solo recurso realizando las peticiones.

Es inmediatamente obvio que necesitaremos un icono para representar lo siguiente: Servidor FTP, enrutador de la red local que conecta a la red enlace para Transmitir, Ancho de banda del enlace, la red local y maquinas de Usuarios.

Se usaran los siguientes nombres para cada recurso:

Recurso.	Descripción de recurso.
SERVIDOR FTP	El actual Servidor FTP.
RUTEADOR A	El enrutador IP conectando a la red local a la red enlace remota.
RED LOCAL (LAN) FAST ETHERNET	La red local que conecta a los usuarios locales y el servidor, junto con el enrutador.
ENLACE FAST ETHERNET	Los circuitos de la red de Fast Ethernet para transmitir.
TODOS LOS USUARIOS	Los iconos representando la carga total de los usuarios.

Tabla 6.1  
Nombrando los recursos del modelo.

### 6.1.2.3 Diseñando los flujos de trabajo.

La definición del flujo de trabajo es bastante simple:

1. Los usuarios provocan una petición tanto de la red Local como del enlace (WAN) que conecta al usuario remoto al servidor.
2. La petición de los usuarios locales pasan por la red Local (LAN) y alcanza al servidor FTP, la del usuario remoto pasa por la red de enlace (WAN), alcanza al Servidor FTP vía el enrutador IP.
3. Todas las peticiones generarán una respuesta de regreso a los usuarios de 5Mbytes +/- 500Kbytes.

La carga de trabajo esta para ser generada junto con el recurso USUARIOS. Sin embargo, nótese que el problema describe dos diferentes cargas de perfiles de trabajo:

- Archivos que regresará el servidor del FTP a los usuarios de la red local (70%).
- Archivos que regresará del servidor FTP al usuario remoto (30%).

Si la carga de trabajo total es de 100 golpes por media hora, y asumiendo que decides normalizar la "Marca" junto con nuestro modelo a un Segundo para razones explicadas abajo, necesitaras generar las siguientes transacciones:

Nombre de la transacción.	Descripción.	Frecuencia ínter arribo que será generada.
GOLPE DEBIL (cached)	Archivos que regresarán al usuario remoto (30%).	$100 * 0.3 = 30$ "Archivos" por media hora. 1800 segundos en media hora. $\rightarrow 1800 / 30 = 60$ golpes cada segundo.
GOLPE DURO (cgi)	Archivos que regresarán a los usuarios locales (70%).	$100 * 0.7 = 70$ "Archivos" por media hora, el trabajo se divide entre 3 usuarios. $70 / 3 =$ lo dejamos en 23 1800 segundos en una media hora. $\rightarrow 1800 / 23 = 78.26$ golpes principales cada segundo.

Tabla 6.2  
Diseño de la carga de trabajo.



### 6.1.3 Definir las pruebas del ambiente.

#### 6.1.3.1 Definiendo el escenario de prueba.

Podemos notar que los componentes del modelo son diferentes, son estimados con diferentes métricas. La velocidad de los enrutadores IP es dada en paquetes por segundo (pps); el software del servidor esta descrito en golpes por segundo (hps); los accesos de los usuarios y el enlace son dados en bits por segundo (bps). Puedes usar las siguientes métricas para cada recurso:

Recurso.	Métricas.	Descripción de recurso.
SERVIDOR FTP	Golpes por segundo	El Servidor FTP actual.
RUTEADOR	Paquetes por segundo	El enrutador IP conectando al servidor vía la red LAN y WAN.
RED LOCAL Y DE ENLACE	Bits por segundo	Los circuitos a transmitir.
TODOS LOS USUARIOS	Ninguno necesitado	Los iconos representando la carga total de los usuarios.

Tabla 6.3  
Homogeneización de métricas.

El corolario de la tabla previa es que debe ser mejor normalizando todo el tiempo las métricas por segundo. La única métrica en la descripción del problema la cual no fue dada en segundos fue la frecuencia de carga de trabajo (100 golpes por media hora), y es porque durante el diseño de la carga de trabajo normalizamos estas figuras para transacciones por segundo para cada tipo de transmisión.

Las cargas de trabajo sobre el sistema real se llevaron acabo de la siguiente manera:

Como lo indica la tabla del diseño de flujo, para la red local, aproximadamente 23 FTP's, por usuario, quedando el disparo por cada usuario como lo muestra la siguiente tabla:

Tiempo	Trabajo
0.0.0	1
1.18.26	2
2.36.52	3
3.54.78	4
5.13.54	5
6.31.80	6
7.50.06	7
9.08.36	8
10..26.62	9
11.44.88	10
13.03.18	11

Tiempo	Trabajo
14.21.40	12
15.39.66	13
16.57.92	14
18.16.18	15
19.34.44	16
20.52.70	17
22.10.96	18
23.19.22	19
24.37.48	20
25.55.74	21
27.14.00	22
28.32.26	23

Tabla 6.4  
Tiempos de disparo carga de trabajo golpe duro.

Para el flujo del usuario remoto, que se diseñaron 30 FTP's:

Tiempo	Trabajo
00.00.00	1
00.60.00	2
1.20.00	3
2.20.00	4
3.20.00	5
4.20.00	6
5.20.00	7
6.20.00	8
7.20.00	9
8.20.00	10
.	11
.	12
.	13
.	14
.	15

Tiempo	Trabajo
.	16
.	17
.	18
.	19
.	20
.	21
.	22
.	23
.	24
.	25
.	26
.	27
26.20.00	28
27.20.00	29
28.20.00	30

Tabla 6.5  
Tiempos de disparo carga de trabajo golpe débil.

Para realizar las mediciones del desempeño de las redes, Local y Enlace FAST Ethernet, se insertaron observadores de tráfico<sup>3</sup>, uno en la red local y otro en un puerto espejo del enrutador del puerto WAN, como lo muestra el siguiente esquema.

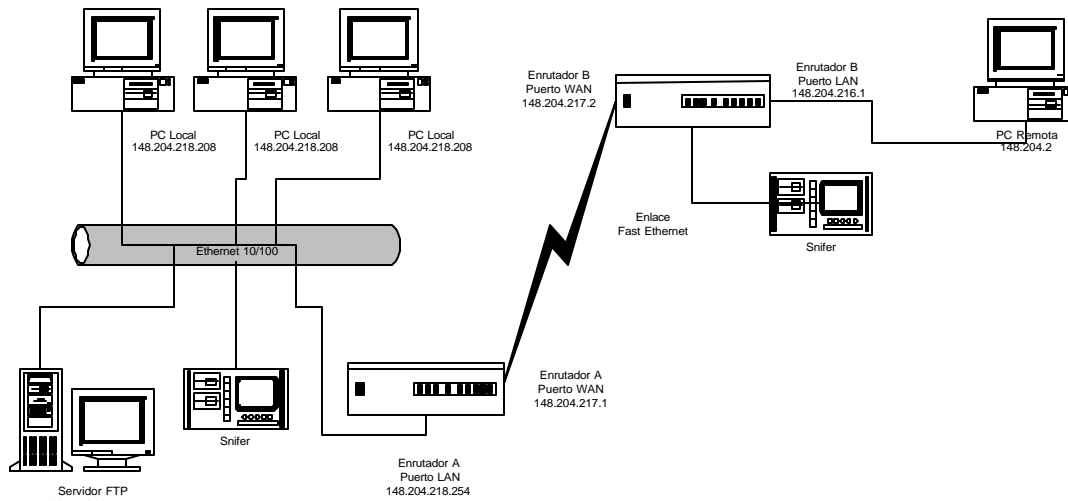


Figura 6.2  
Monitor del sistema real.

### 6.1.3.2 Construir el modelo.

El modelo que se creó en el simulador, tiene el siguiente esquema.

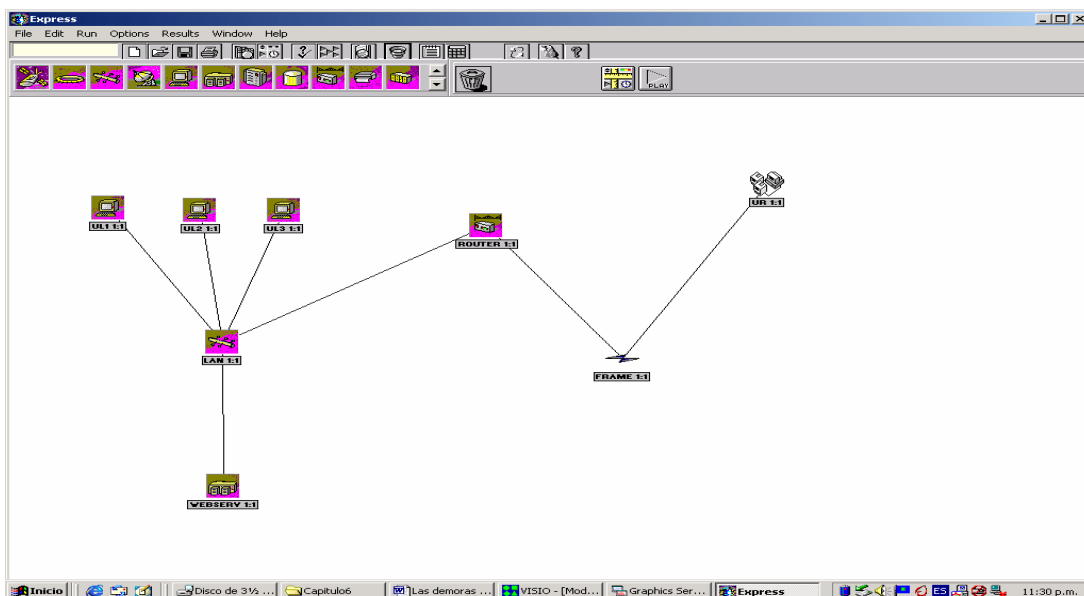


Figura 6.3  
Esquema del modelo en el simulador.

## 6.1.4 Desempeño de los experimentos.

### 6.1.4.1 Medida del desempeño en el sistema real.

Las observaciones del monitor de tráfico, durante este tiempo de trabajo, arrojaron los siguientes resultados del desempeño de las redes local (LAN) y del enlace (WAN) FAST Ethernet:

% Utilización Enlace (WAN) FAST Ethernet	Índices de medición
Utilización del ancho de banda, actual:	0.00%
Utilización del ancho de banda, promedio:	4.60%
Utilización del ancho de banda, Pico:	100.00%
Tiempo de observación:	1765 seg. (28.35)

Tabla 6.6  
Resultados de desempeño del sistema real en la WAN.

% Utilización red local (LAN) FAST Ethernet	Índices de medición
Utilización del ancho de banda, actual:	0.00%
Utilización del ancho de banda, promedio:	20.15%
Utilización del ancho de banda, Pico:	100%
Tiempo de observación:	1750 seg. (27.35 )

Tabla 6.7  
Resultados de desempeño del sistema real en la LAN

### 6.1.4.2 Medidas de desempeño de la simulación.

Los resultados que se generaron, se presentan en las siguientes tablas.

Corridas	Recurso	% utilización
1	FRAME	5.56
2	FRAME	6.12
3	FRAME	3.34
4	FRAME	2.5
5	FRAME	2.78
6	FRAME	8.62
7	FRAME	4.17
8	FRAME	2.78
9	FRAME	5.84
10	FRAME	6.4
11	FRAME	4.45
12	FRAME	2.5
13	FRAME	5.84
14	FRAME	4.17
15	FRAME	3.61
16	FRAME	6.95
17	FRAME	5.56

Corridas	Recurso	% utilización
18	FRAME	3.34
19	FRAME	5.84
20	FRAME	5.56
21	FRAME	5.28
22	FRAME	3.61
23	FRAME	4.45
24	FRAME	5.84
25	FRAME	4.45
26	FRAME	8.06
27	FRAME	5.01
28	FRAME	5.56
29	FRAME	8.62
30	FRAME	4.17
31	FRAME	2.5
32	FRAME	3.61
33	FRAME	5.84
34	FRAME	3.06
35	FRAME	4.45

Tabla 6.8  
Resultados de la simulación en el recurso Frame (Enlace).

Corridas	Recurso	% utilización
1	LAN	16.96
2	LAN	22.8
3	LAN	10.57
4	LAN	21.13
5	LAN	20.3
6	LAN	14.74
7	LAN	15.29
8	LAN	16.13
9	LAN	15.57
10	LAN	18.07
11	LAN	18.41
12	LAN	17.24
13	LAN	13.9
14	LAN	16.96
15	LAN	11.68
16	LAN	21.13
17	LAN	10.29

Corridas	Recurso	% utilización
18	LAN	16.68
19	LAN	16.68
20	LAN	17.41
21	LAN	18.35
22	LAN	20.69
23	LAN	15.57
24	LAN	16.96
25	LAN	16.35
26	LAN	16.13
27	LAN	12.51
28	LAN	15.02
29	LAN	14.18
30	LAN	13.9
31	LAN	19.74
32	LAN	14.18
33	LAN	18.91
34	LAN	21.41
35	LAN	16.68

Tabla 6.9  
Resultados de la simulación en el recurso LAN.

### 6.1.5 Estableciendo conclusiones de los resultados de la simulación.

Siguiendo las ecuaciones (11) y (20) que se presentan en el capítulo 5, sobre número de repeticiones necesarias de simulación para alcanzar un porcentaje aceptable de precisión en los resultados, las aplica el simulador, obteniendo:

% Utilización red local (LAN) FAST Ethernet	Índices de simulación
Utilización del ancho de banda, promedio:	16.64%
Utilización del ancho de banda, Pico:	100%
Tiempo de simulación:	1800 seg. (30:00)
Simulaciones	35
Intervalo de confianza con un 90% de certeza	Entre 15.78 y 17.51 (16.64 +/- 0.87)

Tabla 6.10  
Resultados de desempeño del sistema simulado en la LAN.

% Utilización Enlace (WAN) FAST Ethernet	Índices de simulación
Utilización del ancho de banda, promedio:	4.87%
Utilización del ancho de banda, Pico:	100%
Tiempo de simulación:	1800 seg. (30:00)
Simulaciones	35
Intervalo de confianza con un 90% de certeza	Entre 4.39 y 5.34 (4.87 +/- 0.47)

Tabla 6.11  
Resultados de desempeño del sistema simulado en la WAN.

Se anexa el apéndice D, de características y funcionamiento del simulador de redes empleado en este caso de estudio, Prophecy.

También se anexa en el apéndice B y C las capturas de pantallas, mostrando los resultados de medición de los monitores, conectados en la red y del simulador respectivamente.

La comparación de los métodos, frente a una carga característica de una instalación concreta, es la medición del comportamiento sobre un prototipo.

Como podemos observar los resultados, generados por la simulación, en el recurso del enlace (WAN), son muy parecidos a los obtenidos por el monitor del sistema real, sin embargo en los resultados del simulador en la red local (LAN), hay algo de parecido, esto tal vez se deba a que, se tuvo un retraso en disparo de trabajo, se perdió el monitoreo en el puerto espejo del enrutador, etc. En el simulador, los datos del enrutador no son los correctos, y se deba nuevamente investigarlos. Otra razón, tal vez se necesiten mas simulaciones, o el tiempo de simulación debe de ser más grande.

El porcentaje de error es el siguiente<sup>6</sup>:

% Utilización red local (LAN) FAST Ethernet		
Simulación	Monitor	% de Error Relativo
16.64%	20.15%	17.41

Tabla 6.12  
Resultados, porcentaje error, sistema real y simulado en la LAN.

% Utilización Enlace (WAN) FAST Ethernet		
Simulación	Monitor	% de Error Relativo
4.87%	4.60%	5.86

Tabla 6.13  
Resultados, porcentaje error, sistema real y simulado en la WAN.

### 6.1.5.1 Conclusiones del caso de estudio simulado.

Como conclusión para este caso de estudio, podemos decir que, sin ningún problema, los elementos de red y la carga de trabajo, diseñados y planeados, soportaran la aplicación deseada por la redes WAN y LAN de AXTEL. Para solucionar sus actualizaciones de base de datos (Aplicación).

Para finalizar este capítulo, con el análisis de los porcentajes de parecidos entre los métodos, podemos decir que los resultados del simulador son suficientes para poder tomar una decisión en la implementación ó cambios en la red de computadoras, debemos de tomar en cuenta que en el sistema real, influyen otros factores, que engañan el valor de los resultados, como por ejemplo, la transmisión de paquetes ó tramas, de protocolos no considerados en el simulador, estos pueden ser del equipo monitor, de las aplicaciones en red de los usuarios, etc. Que representan por experiencia por lo menos el 5% de veracidad en los resultados.

## CONCLUSIONES Y RECOMENDACIONES

Uno de los problemas claves a los que se enfrentan las redes de computadoras es su uso creciente, prestaciones de servicios más complejas y la implementación o mejoras de nuevas tecnologías, que si bien proporciona múltiples beneficios, también trae consigo diversas complicaciones.

Por esto es importante planear y diseñar las redes de computadoras, estas acciones implican, el encontrar o predecir el desempeño del sistema de redes de computadoras con su hardware y su sistema operativo, cuando procesan las transacciones de los usuarios.

El predecir el desempeño de nuestro sistema, nos da información para controlar:

- Niveles de servicio
- Diseños alternativos
- Determinar el valor óptimo de algún parámetro del sistema
- Identificar congestiones (complicación del tráfico)
- Caracterizar la carga de trabajo
- Determinar el número y el tamaño de los componentes
- Estimar costos para mejoras del sistema
- Planificar futuras cargas.

Las mediciones que conseguimos al encontrar el desempeño de una red de computadoras, en términos generales son:

- Consumo de tiempo
- Utilización del recurso
- Tiempo de respuesta.

La comparación de los resultados obtenidos en el capítulo seis, nos permite decir que la simulación es una excelente decisión como herramienta de soporte para predecir el desempeño en un diseño específico de una red de comunicaciones y para probar el impacto de nuevas aplicaciones sobre una red ya existente.

En el campo de las comunicaciones de datos, es importante simular el sistema antes de realizar su implementación, prueba y operación real. Además, algunas veces no es posible probar nuevas soluciones sobre redes ya implementadas debido a la inconveniencia de disturbios sobre el sistema.

La simulación es una metodología usada en diferentes áreas de la ingeniería que permite analizar el comportamiento de un sistema antes de implementarlo ó de optimizar su funcionamiento al probar soluciones y cambios sobre el mismo. Para lograr lo anterior se requiere construir un modelo del sistema y experimentar con éste en el computador.



El trabajo busca generar conciencia de prevención de problemas mayores en la utilización de la red sin necesariamente caer en las soluciones típicas que cortan el desarrollo de una verdadera ingeniería.

También en este trabajo, se ha realizado una investigación que no se había hecho anteriormente, el objetivo es proporcionar una guía que permita a las personas interesadas en el área, aplicar los conocimientos adquiridos a través de su formación profesional para solucionar los problemas que se presentan en el uso cotidiano de las redes de las computadoras.

La prueba del monitoreo u observación práctico, fue necesario para demostrar de manera sencilla, que los resultados encontrados de utilización de las redes en el modelo del caso práctico, fueron predichos por el simulador, con un porcentaje muy cercano a los valores reales de los monitores y de esta manera expresar opiniones que no subyacieran en especulaciones teóricas.

La recomendación de este trabajo se enfoca en proporcionar un panorama completo, en integrar las técnicas de simulación aplicadas para adquirir una metodología de análisis de redes.

Nosotros podemos utilizar, en cualquier software de simulación, las siguientes etapas:

1.- Construcción del modelo; Aquí creas y defines recursos, la vía en como los unes a cada otro, y las actividades y procedimientos que ejecutan.

2.- Ejecución de la simulación del modelo; En esta parte depuras el modelo.

3.- Análisis; Mientras estas ejecutando la simulación, puedes inspeccionar y analizar los resultados.

4.- Documentando el modelo; Un formato sugerido para el tipo de documentación es:

Declaración del problema. Encontrar si un nuevo servidor reduce el tiempo de respuesta de una estación de trabajo.

Objetivos de simulación. Encontrar todo el tiempo de viaje de mensajes regresando a la estación de trabajo.

Suposiciones. El nuevo servidor ejecutara lo mismo que el servidor en cuestión.

5.- Resultados; Es suficiente con tres tipos de información estadística:

Uso de recursos. Contiene la utilización promedio del recurso.

Uso de colas. Consiste primeramente de la utilización de los recursos de la cola.

Estadísticas del mensaje. Contiene información de un tipo de mensaje específico (transacción / estado). Retardo del mensaje.

Es importante también recomendar que la información sobre la operación de los recursos, debe de ser lo más verdadero, por ejemplo, el de un enrutador, se debe

de consultar al proveedor de este, cual es el manejo máximo de paquetes por segundo, recuerda que no necesitas simular la red de trabajo entera junto con un modelo único, tantas suposiciones realices en tu modelo, menos confiables serán tus resultados, es importante que verifiques las veces de ejecución de la simulación (corridas), puesto que de este punto depende que los valores sean lo más precisos.

Si estas en los negocios de diseño de protocolos o ejecutando detalles de puesta a punto, no necesitas simular los protocolos específicos para obtener un buen sentido de congestionamiento y recursos necesitados. Si estas tratando de ajustar parámetros para ganar una décima de segundo de perfeccionamiento aquí y allá, no creo que la simulación sea la mejor opción.

Por ultimo, espero que este trabajo de una idea concreta sobre la proposición de simulación como herramienta para predecir el desempeño de redes de computadoras en el diseño y planeación de estas. Hay muchas funciones que no cubrimos. Por ejemplo, podrías ver una grafica de utilización así como la corrida de la simulación, o puedes reestablecer las estadísticas para eliminar las bases de la condición inicial. Sugiero que revises la bibliografía de esta tesis para informarte de los puntos de más interés que encuentres en el trabajo y descubras el poder total de la simulación.

## **BIBLIOGRAFIA Y SITIOS WEB**

---

### BIBLIOGRAFIA

[1]Fishman G.S. Conceptos y Métodos en la simulación digital de eventos discretos, 1978.

Editorial Limusa. México.

[2]Klein rock, L. Queuing System, Volume 2, computer Application.

J. Wiley and Sons, 1976.

[3]Link Monitor.

Course Notes:

Performance Houston LL.

[4]Modeling and simulation in Prophecy.

Course Notes:

Abstraction software LLC.

[5]Networks Analysis and troubleshooting.

J. Scoot Haugdahi.

Addison Wesley.

[6]Probability, Random Variables, and stochastic Processes.

Papoulis.

McGraw-Hill.

[7]Practical Computer Network Analysis and Design.

James McCabel.

Editorial Morgan Kaufmann,1998.

[8]Probabilidad y Estadísticas para ingeniería y Administración.

William W. Hines y Douglas c. Montgomery.

[9]Redes de Computadoras.

Cornelio Robledo Sosa.

I.P.N. México D.F. 1999.

[10]Redes Globales de Información con Internet y TCP/IP.

Douglas E. Comer.

Prentice Hall.

[11]Redes de Ordenadores.

Andrew Tanenbaum.

Prentice Hall.

[12]Redes de Telecomunicaciones (Protocolos, Modelado y Análisis).  
Mischa Schwartz.  
Addison – Wesley Iberoamericana.

[13]Simulación de sistemas, diseño desarrollo e implementación.  
Shannon R.E.  
Trillas, 1988, México.

[14]Simulation, Modeling and Analysis,  
Averill M. Law, W.David Kelton (Third Edition).  
McGraw-Hill, Boston 2000.

#### SITIOS WEB

<http://www.abstraction.com/abstraction.html>  
<http://www.raczynski.com/pn/encyk.htm>  
<http://www.topology.org/soft/sim.html>  
<http://www.acs.ilstu.edu/faculty/wjyurci/nsfteachsim/indexnew.html>  
<http://www.mixcoac.upmx.mx/mcleod/cfp5.htm>  
<http://www.idsia.ch/~andrea/simtools.html>  
[www.softonic.com](http://www.softonic.com)  
<http://whale.hit.bme.hu/omnetpp/>  
<http://www.cs.uwa.edu.au/cnet/>  
<http://www.cs.cornell.edu/skeshav/real/overview.html>  
<http://www.opnet.com/>  
<http://www.goldsim.com/>  
[www.linkview.com/](http://www.linkview.com/)  
<http://www.scalable-networks.com/products/>  
<http://www.isi.edu/nsnam/ns/>  
<http://irg.cs.ohiou.edu/one/>  
<http://www.shunra.com/>  
<http://www.pscs.umich.edu/education/websites.html#simulationSystems>

## **GLOSARIO**

---

100BASET : Una versión a alta velocidad de Ethernet (IEEE 802.3). También conocida como Fast Ethernet. La 100BASE-T transmite a 100 Mbps con cable UTP, con una distancia de conexión de 100 metros.

10BaseT: Una versión estándar Ethernet (IEEE 802.3). Transmite a 10 Mbps con cable UTP, con una distancia de conexión de 100 metros.

### **A**

Abstraction software LLC: Empresa dedicada a realizar software de simulación para redes.

Allied Telesyn International: Empresa dedicada a construir equipos para monitorear y analizar protocolos.

Ancho de banda: Capacidad de un medio para transmitir una señal, que en el caso de una red hace referencia a la cantidad de ficheros y mensajes que se pueden enviar sin degradar sus prestaciones.

Apple: AppleTalk. Un estándar para redes de ordenadores propietaria de Apple Computer para la utilización con ordenadores Macintosh y otros periféricos, particularmente impresoras LaserWriter; trabajando 230 Kbps.

Arcnet: Red disponible de Data Point que permite conectar PC's máximo de 255, usando el método de acceso al medio token passing.

ATM: (Asynchronous Transfer Mode). Estándar internacional para el relevo de celdas en el cual servicios de diversos tipos (voz, datos, video) están contenidos en celdas de extensión fija (53 bits). Estas celdas fijas permiten procesamiento mediante *hardware*, lo que reduce el tiempo de envío. ATM está diseñado para tomar ventajas de medios de transmisión de alta velocidad como E3, SONET y T3.

Atributos: Perfil que permite hacer o no hacer una actividad en un recurso de red.

AXTEL Telecomunicaciones: Empresa dedicada a proveer servicios de telecomunicaciones.

### **B**

Backbone: Conexión de alta velocidad dentro de una red que conecta a otros circuitos.

Batch: Archivos por lotes.

Benchmark: Método para compara el rendimiento entre sistemas.

Bits: palabra que significa símbolos o dígitos binarios; proviene de *binary digits*; es también una medida de la cantidad de información contenida en un mensaje, definida por C. E. Shannon.

Bps: (Bits por segundo). Se refiere a la velocidad de intercambio de información entre computadoras, modems o enlaces.

Byte: Octeto; mínima unidad direccionable, caracter. (8 bits).

Bottom-down: Técnica que utiliza Prophecy, para simplificar el análisis de una red.

Broadcast: Emisión, Enviar un mensaje a todos los usuarios.

Buffer: Un dispositivo de almacenamiento temporal utilizado para compensar las diferencias entre la velocidad de los datos y el flujo de datos entre dos dispositivos temporary storage device used to compensate (típicamente un ordenador y una impresora); también conocido como spooler.

Bus. Topología de red en la que las estaciones se conectan a un único cable.

## C

C o C++: Lenguaje de programación orientada a objetos.

Cliente/Servidor: (C/S). Arquitectura de red en la que cada computadora o proceso en la red es un cliente o un servidor. Los servidores son computadoras o procesos de gran capacidad dedicados a administrar los discos, las impresoras y el tráfico de trabajo en la red. Los clientes son computadoras personales en las que los usuarios corren las aplicaciones.

Cabecera: (header). Porción de un paquete, precediendo los datos, que contiene las direcciones fuente y destino y campos de detección de errores. También parte de un mensaje o artículo de news.

Cliente: Usuario de un servicio de la red. También se utiliza para designar aquel ordenador que depende de otro para alguno de (o todos) sus servicios).

Cabletron: Empresa dedicada a construir equipo de ínter conectividad de redes,.

CAD: (Computer Aided Design). Diseño asistido por computadora.

Call centers: Sitios que funcionan para atender llamadas simultáneamente.

CISC: (Common Instruction Set Computing). Arquitectura utilizada para medir procesamiento de un CPU.

Clip art: Formato de imágenes.

CPU (Central Process Unit): Parte del ordenador que se encarga de procesar los datos.

CSMA/CD: (Detección de Portadora para Acceso Múltiple/Detección de Colisión). Un método de acceso a una red de área local en la cual la contención entre dos o más estaciones se resuelve mediante la detección de colisión. Cuando dos estaciones transmiten al mismo tiempo, ambas se paran y avisan que está ocurriendo una colisión. Cada uno trata de nuevo, después de esperar un tiempo predeterminado, normalmente varios microsegundos.

## D

Datagrama: La unidad de información básica usada en Internet. Contiene direcciones de fuente y destino, conjuntamente con el dato. Aquellos mensajes que son muy grandes se dividen en una secuencia de datagramas IP.

Debugging: proceso de un programa antes de ser ejecutado para revisar si hay errores en la compilación.

Desbordamiento: Pérdida de datos por exceso de los mismos en el punto de recepción.

Discos duros: El disco duro es el sistema de almacenamiento más importante de su computador y en él se guardan los archivos de los programas

## E

E.S.I.M.E. Zacatenco: Escuela Superior de Ingeniería Mecánica y Eléctrica.

Emulación 3270: Emulación de una terminal de Main Frame IBM.

Encapsulado: Contención de un protocolo en otro protocolo de transporte.

Enlace: un canal de comunicaciones entre dos nodos o dos equipos.

Enrutador: (Router). Equipo de comunicaciones que sirve para enlazar LAN's en la capa de red de OSI.

Ethernet: Esquema de red de 10 Mbits/seg.\ desarrollado originalmente por Xerox Corporation. Está muy extendida en redes de área local, ya que está disponible para muchos tipos de ordenadores, no precisa de licencias y existen componentes para soportarla de diversos fabricantes.

## F

FDDI: (Fiber Distributed Data Interface). Estándar para tecnología de red basado en fibra óptica establecido por la ANSI que está siendo utilizado cada vez más.

Trama: Un proceso constituido por una serie de grupos de datos.

Frame Relay: Protocolo utilizado en la interfaz entre los dispositivos de usuario (routers, bridges, hosts) y equipos de red (nodos de conmutación) que posibilita la transmisión de los datos aplicando técnicas de conmutación de paquetes. Se caracteriza por confiar en la fiabilidad de los actuales medios transmisión, eliminando el nivel de red y reduciendo la funcionalidad del nivel de enlace.

FTP: (File Transfer Protocol). El protocolo estándar de alto nivel en la Internet utilizado para la transferencia de ficheros de un ordenador a otro.

## G

Gateway: Ordenador dedicado que conecta dos o más redes y encamina los paquetes de una red a otra. Los *gateways* encaminan los paquetes hacia otros gateways hasta pueden ser entregados al destino final directamente a través de una red física.

Gbps: Mil millones de bits por segundo.

## H

Hertz: hercio. Unidad de frecuencia electromagnética. Equivale a un ciclo por segundo.

HDLC: (High-Level Data Link Control) — Un protocolo de comunicaciones ISO utilizado en redes de conmutación de paquetes X.25. Proporciona corrección de errores al Nivel de Enlace de Datos. SDLC, LAP, y LAPB son elementos de HDLC.

Hps: Golpes por segundo, métrica usada por Prophesy para hacer referencia a la cantidad de trabajo recibida por un recurso.

HTTP: (HyperText Transport Protocol). El protocolo de comunicación empleado por los servidores de WWW.

## I

Icono: imagen que generalmente permite el acceso a determinado software instalado en el disco rígido de cada máquina.

ISO: (International Standards Organization). Una organización mundial cuyos miembros trabajan en diferentes proyectos. La mayoría de los países industrializados tienen afiliados locales: ANSI, IEEE, BSI, AFNOR, DIN, etc.

I/O: Siglas en los puertos de comunicaciones describiendo entrada y salida.



IBM: Empresa dedicada a crear software, PC's, Telecomunicaciones.

IEEE: Instituto de Ingenieros en electrónica y Electricidad.

IIE Solutions and ORIMS: Empresa dedicada a el análisis de software de simulación.

IIS: (Internet Information Services). Internet de servicios de información.

IP: (Internet Protocol). El protocolo utilizado en gateways para conectar redes a Nivel de Red OSI (Nivel 3) y superiores. IP enruta un mensaje a través de la red.

IPX: (Internet Packet Exchange). Un protocolo de comunicación en Novell NetWare que crea, mantiene y termina la conexión entre dispositivos de red, tales como estaciones de trabajo y servidores.

IWU: (Individual Work Unit). Unida de trabajo, PC.

## **J**

Jitter: Distorsión que se encuentra en líneas de comunicación.

## **L**

LAN: (Local Area Network). Término que significa "Red de Area Local," y que describe cualquier tecnología de red física que trabaja a gran velocidad en distancias cortas (de hasta unos cientos de metros).

Librerías de rutina: Parte de un programa con una actividad en especial.

LOOPS: Aros, pruebas que se realizan entre dos puntos de conexión.

Lotus 1-2-3: Software de hoja de calculo.

LLC: (Control de Enlace Lógico). Un protocolo desarrollado por el comité IEEE 802 para el control de una transmisión del nivel de enlace de datos; el nivel superior del Nivel 2 del protocolo IEEE (OSI) que completa el protocolo estándar MAC ; el estándar IEEE 802 .2; incluye direccionamiento del sistema y comprobación de errores.

## **M**

MAC: Son direcciones de 48 bits que vinculan directamente con el hardware en la capa de enlace de datos y que mantienen una relación biunívoca con las direcciones IP mediante los protocolos de resolución ARP y RARP.

MAC: (Control de Medios de Acceso) — Un tipo específico de protocolo para el control de acceso dentro de las especificaciones IEEE 802, normalmente incluye variaciones para Token Ring, Token bus, y CSMA/CD; el subnivel inferior de los niveles de enlace IEEE (OSI), complementando el Control de enlace Lógico (LLC).

Mbps: (Megabits por segundo). Velocidad de transmisión, millones de bits por segundo.

Mensaje: Término usado para identificar un paquete de información en envío o respuesta.

Megahertz: (MHz). Un millón de hertz o hercios.

Mips: (Million of instruction per seconds). Unidad utilizada para medir el proceso de un CPU.

MTU: Maximum Transfer Unit. Tamaño máximo de los datos de usuario que pueden viajar en un paquete de un protocolo determinado.

## **N**

NFS: (Network File System). Sistema de archivos de red.

Nodos: puntos en los cuales se ubican equipos de procesamiento en una red, y a los cuales están conectados los enlaces de la misma.

## **O**

Overhead: Parte de datos de control de los protocolos (Cabecera) en un mensaje.

## **P**

Paquete: Unidad de datos enviados en una red conmutada. También es posible referirse a aquellos datos enviados físicamente por la red o a los datagramas que utiliza IP.

PC: (Personal Computer). Computadora Personal.

PDU: (Protocol Data Unit). Porción de un paquete de un protocolo determinado que contiene intercambio de datos entre equipos.

Pentium: Marca registrada de Intel que indica la versión del CPU.

Prophesy: Software de simulación para redes hecho por Abstraction LLC.

Protocolo: Descripción formal del formato de los mensajes y de las reglas que dos ordenadores deben seguir para intercambiar mensajes.

Puente: (Bridge). Un dispositivo que conecta dos segmentos de red juntos, los cuales pueden ser de tipos similares o distintos, tales como Ethernet y Token Ring.

## **Q**

QoS: (Quality of Service). Calidad de Servicio.

## **R**

RAM: (Random Acces Memory) Memoria de lectura exclusiva.

RISC: (Reduce Instruction Set Computing). Reducción de instrucciones puesta en la computadora.

## **S**

S.I: Sistema Informático.

Servidor: Servidor. Computadora que presta servicios a muchos usuarios. Provee información y programas a Internet.

Software: Programas de Computación.

SQL: (System Quering Line). Sistema de colas en linea.

## **T**

TCP/IP: (Transmission Cotrol Protocol/Internet Protocol). Conjunto de protocolos usados en Internet para soportar servicios tales como acceso remoto telnet, transferencia de ficheros FTP y correo electrónico SMTP.

TCP: (Transmisión Control Protocol) Protocolo de control de transmisión.

Telnet.: Protocolo estándar en Internet que permite mantener una sesión en un sistema remoto. Existen manuales de Telnet.

Throughput: Capacidad de transmisión de un dispositivo. Se mide en bits por segundo.

Tick: Métrica de tiempo usado en Prophecy.

Token Ring: Arquitectura de red propuesta por IBM, utiliza el protocolo testigo de señal y transmite a velocidad de 16 Mbps máximo.

Top-down: Procedimiento dentro de la simulación para simplificar el análisis de redes de computadoras.

## **U**

UDP: (User Datagram Protocol). Otra posible capa superior del protocolo Internet.

UNIX: Sistema operativo de red, escrito en C.

## **V**

vt100: Terminal de uso muy generalizado, y reconocido por la mayoría de los Hosts.

## **W**

WAN: (Wide-Area Network). Red que cubre un territorio amplio.

WWW (World Wide Web): La "Telaraña Mundial" es un sistema de información que basa su funcionamiento en los documentos en formato hipertexto que forman Internet en todo el mundo.

## **FIGURAS Y TABLAS**

---

### FIGURAS

Figura 1.1 Interconexión en un "switch " de 4 puertos.	2
Figura 1.2 Mecanismos para el reenvío de mensajes en un "switch " .	3
Figura 1.3 Conexión de dos terminales mediante un "switch " en una LAN.	4
Figura 1.4 Red LAN switchheada con loops.	5
Figura 1.5 Red Token Ring segmentada mediante el uso de puentes.	8
Figura 1.6 Enlace por puentes a nivel de MAC.	10
Figura 1.7 A) Comunicación entre dos segmentos. B) Cada puente crea una tabla de ruteo donde sabe a que puertos tiene conectadas las estaciones.	12
Figura 1.8 Formación de LOOPS en los puentes.	12
Figura 1.9 Trama de Token Ring, incluido el campo de información de ruteo.	14
Figura 1.10 Caminos posibles de un mensaje con enrutamiento de origen.	15
Figura 1.11 Esquema de un nodo.	18
Figura 2.1 Caudal en función del tráfico ofrecido.	23
Figura 2.2 Retardo de tránsito en función del tráfico ofrecido.	24
Figura 2.3 Saturación del buffer de un nodo.	24
Figura 2.4 Conexión entre nodo de alta capacidad y PC.	25
Figura 2.5 Congestión en un nodo.	25
Figura 3.1 Representación de una cola.	30
Figura 3.2 Representación de colas en serie.	30
Figura 3.3 Representación de colas en paralelo.	30
Figura 3.4 Diagrama de demora de un proceso de comunicación de datos.	31
Figura 3.5 Representación de la ley de Little.	35
Figura 3.6 Tiempos en el procesamiento de la comunicación.	37
Figura 3.7 Sistema C/S de tres capas.	39
Figura 3.8 Diagrama de demora del procesamiento de comunicación de un sistema C/S tres capas.	39
Figura 3.9 Conectividad entre un cliente y un servidor.	40
Figura 3.10 Interacción cliente-servidor.	41
Figura 3.11 Formato de la estructura de un mensaje.	41
Figura 4.1 Caminos de estudio en un sistema.	46
Figura 4.2 Funcionamiento de un modelo.	47
Figura 4.3 Diagrama de control de flujo para la simulación de modelos de sistemas discretos.	50
Figura 4.4 Evolución del ejemplo 2.	53
Figura 4.5 Procesamiento de un paquete en cola.	55
Figura 4.6 Estados de un proceso P.	57

Figura 4.7 Estados de un proceso P incluyendo recursos.	57
Figura 5.1 Proceso de la metodología tradicional en diseño y planeación de redes.	69
Figura 5.2 Pasos en el estudio de la simulación.]	73
Figura 5.3 Conexión LAN a LAN por medio de WAN.	75
Figura 5.4 Estados de ejecución de la simulación.	82
Figura 6.1 Red a modelar c/s.	91
Figura 6.2 Monitor del sistema real.	97
Figura 6.3 Esquema del modelo en el simulador.	97

## TABLAS

Tabla 1.1 Tabla de retransmisión de un 'switch'.	2
Tabla 1.2 Tamaños mínimos y máximos de mensajes según el MTU.	6
Tabla 3.1 Parámetro de protocolos.	41
Tabla 4.1 Atributos de sistema.	62
Tabla 5.1 Proceso de planeación y diseño de redes mediante simulación y modelado de redes.	72
Tabla 5.2 Valores de $t_{\alpha/2}$ .	87
Tabla 6.1 Nombrando los recursos del modelo.	93
Tabla 6.2 Diseño de la carga de trabajo.	94
Tabla 6.3 Homogeneización de métricas.	95
Tabla 6.4 Tiempos de disparo carga de trabajo golpe duro.	96
Tabla 6.5 Tiempos de disparo carga de trabajo golpe débil.	96
Tabla 6.6 Resultados de desempeño del sistema real en la WAN.	98
Tabla 6.7 Resultados de desempeño del sistema real en la LAN.	98
Tabla 6.8 Resultados de la simulación en el recurso Frame (Enlace).	99
Tabla 6.9 Resultados de la simulación en el recurso LAN.	99
Tabla 6.10 Resultados de desempeño del sistema simulado en la LAN.	100
Tabla 6.11 Resultados de desempeño del sistema simulado en la WAN.	100
Tabla 6.12 Resultados, porcentaje error, sistema real y simulado en la LAN.	101
Tabla 6.13 Resultados, porcentaje error, sistema real y simulado en la WAN.	101

## **APENDICE A**

---

### **Introducción al Modelado Analítico y Teoría de Colas**

Algunas de las *técnicas* más conocidas de modelado analítico son:

- La teoría de colas.
- Los procesos de Markov.

Los *modelos analíticos*:

- Son las representaciones matemáticas de los sistemas.
- Permiten al evaluador del rendimiento sacar conclusiones acerca del comportamiento del sistema.

Las expresiones *teoría de colas* y *teoría de líneas (o filas) de espera* deben considerarse equivalentes.

El término matemático *cola* significa una *línea de espera*.

Si no hubiera líneas de espera se podría recibir servicio de inmediato:

- Sería lo deseable.
- El costo de disponer de la suficiente capacidad de servicio para no tener que esperar sería muy elevado.

Se consume cierta cantidad de tiempo en líneas de espera por servicio pero:

- El costo de ese servicio es menor debido a la mejor utilización de la instalación de servicio.

Si existe una población de clientes que demandan cierto servicio prestado por servidores:

- Algunos clientes ingresarán a la red de colas y esperarán que un servidor quede disponible.

Algunas *colas* son:

- *Ilimitadas*: pueden crecer tanto como sea necesario para contener a los clientes que esperan.
- *Limitadas*: solo pueden contener un número fijo de clientes en espera y quizás hasta ninguno.

Se deben tener en cuenta *variables aleatorias* que pueden ser descritas por *distribuciones probabilísticas*.

La variable aleatoria “q” representa el *tiempo que emplea un cliente esperando en la cola a ser servido*.

La variable aleatoria “s” representa la cantidad de *tiempo que emplea un cliente en ser servido*.

La variable aleatoria “w” representa el tiempo total que emplea un cliente en el sistema de colas: “w = q + s”.

## **Fuente, Llegadas y Llegadas de Poisson**

### **Fuente**

Los clientes son proporcionados a un sistema de colas desde una *f fuente que puede ser infinita o finita*.

Con una *f fuente infinita* la cola de servicio puede llegar a ser *arbitrariamente grande*.

Para una *f fuente finita* la cola de servicio es *limitada*, pero una fuente finita pero muy grande suele considerarse como infinita.

### **Llegadas**

Supondremos que los clientes llegan a un *sistema de colas* en los *tiempos*:

$$t_0 < t_1 < t_2 < \dots < t_n.$$

Los clientes llegan de uno en uno y nunca hay una colisión.

Las variables aleatorias “ $t_k$ ” miden los *tiempos entre las llegadas sucesivas* (arbitrario) y se denominan *tiempos entre llegadas*:

- Son variables aleatorias *independientes* y están *idénticamente distribuidas*:

$$t_k = t_k - t_{k-1}; \text{ con } (k \geq 1).$$

### **Llegadas de Poisson**

Las llegadas pueden seguir distintos patrones arbitrarios pero suele suponerse que forman un *proceso de llegadas de Poisson*:

- Los *tiempos entre llegadas* están distribuidos *exponencialmente*:

$$P(t \leq t) = 1 - e^{-\lambda t} :$$



- La probabilidad de que lleguen *exactamente* “ $n$ ” clientes en cualquier *intervalo de longitud* “ $t$ ” es:
  - $( e^{-\lambda t} (\lambda t)^n ) / n!$ ; con  $(n=0,1,2,.. .)$ .
- $\lambda$  es una *tasa promedio de llegadas constante* expresada en “*clientes por unidad de tiempo*”.
- El *número de llegadas por unidad de tiempo* se dice que tiene *distribución de Poisson con una media*  $\lambda$ .

## **Tiempos de Servicio, Capacidad de la Cola y Número de Servidores en el Sistema**

### **Tiempos de servicio**

Se supone que *los tiempos de servicio son aleatorios*.

“ $s_k$ ” es el tiempo de servicio que el *k-ésimo cliente* requiere del sistema.

Un tiempo de servicio arbitrario se designa por “ $s$ ”.

La *distribución de los tiempos de servicio* es:

$$W_s(t) = P(s \leq t).$$

Para un *servicio aleatorio con una tasa promedio de servicio* “ $\lambda$ ”:

$$W_s(t) = P(s \leq t) = 1 - e^{-\lambda t}, \text{ con } (t \geq 0).$$

### **Capacidad de la cola**

Las colas deben tener:

- *Capacidad infinita:*
  - Cada cliente que llegue puede entrar en el sistema de colas y esperar, independientemente de cuántos clientes hay en espera.
- *Capacidad cero (o sistemas de pérdidas):*
  - Los clientes que llegan cuando la instalación de servicio está ocupada no podrán ser admitidos al sistema.
- *Capacidad positiva:*
  - Los clientes que llegan solo esperan si hay lugar en la cola.

## Número de servidores en el sistema

Los sistemas de colas se pueden categorizar según el número de servidores en:

- *Sistemas de un solo servidor:*
  - Tienen un solo servidor y nada más pueden darle servicio a un solo cliente a la vez.
- *Sistemas de servidores múltiples:*
  - Tienen “c” servidores con idéntica capacidad y pueden dar servicio a “c” clientes a la vez.

## Disciplinas de Colas

*Son las reglas usadas para elegir al siguiente cliente de cola que va a ser servido.*

La más conocida es la “FCFS” o primero en llegar, primero en ser servido.

Generalmente se utilizan las siguientes *notaciones*:

- *Notación Kendall.*
- *Notación Kendall abreviada.*

*Notación Kendall (A/B/c/K/m/Z):*

- A: distribución de tiempos entre llegadas.
- B: distribución de tiempos de servicio.
- c: número de servidores.
- K: capacidad de cola del sistema.
- m: número de clientes en la fuente.
- Z: disciplina de cola.

*Notación Kendall abreviada (A/B/c):*

- No hay límite en la longitud de la cola.
- La fuente es infinita.
- La disciplina de cola es “FCFS”.
- “A” y “B” pueden ser:
  - GI: para tiempo entre llegadas *general independiente*.
  - G: para tiempo de servicio *general*.
  - $E_k$ : para las distribuciones de tiempos entre llegadas o de servicio *Erlang-k*.
  - M: para las distribuciones de tiempos entre llegadas o de servicio *exponenciales*.
  - D: para las distribuciones de tiempos entre llegadas o de servicio *determinísticos*.

- $H_k$ : para las distribuciones de tiempos entre llegadas o de servicio hiperexponenciales (con “k” estados).

## Intensidad de Tráfico y Utilización del Servidor

### Intensidad de tráfico

Es una *medida de la capacidad del sistema* para dar servicio efectivo a sus clientes.

Se define como la razón de la *media del tiempo de servicio “E(s)”* y la *media del tiempo entre llegadas E( t)”*..

La *intensidad de tráfico “u”* es:

$$u = [E(s)] / [E( t)] = I E(s) = ( I / m):$$

*I*: tasa de llegadas.

*m* tasa de servicio.

Es útil para *determinar el número mínimo de servidores idénticos* que necesitará un sistema para dar servicio a sus clientes:

- Sin que las colas se hagan indefinidamente largas.
- Sin tener que rechazar clientes.
- Ej.:  $E(s) = 17$  segundos y  $E( t) = 5$  segundos,  $u = 17 / 5 = 3,4$ : si  $E(s) = 17$ 
  - El sistema deberá tener un mínimo de 4 servidores.

Se debe tener en cuenta que:

- La *tasa de llegadas* de los clientes es un *promedio*:
  - Es posible que no llegue ningún cliente durante un largo tiempo.
  - Es posible que los clientes lleguen en rápida sucesión, excediendo la capacidad física de la cola y ocasionando el rechazo de clientes.
  - Si se utilizan colas de tamaño fijo debe haber capacidad suficiente para soportar excesos ocasionales de la tasa de llegadas.
  - Se podrían utilizar colas de longitud variable junto con lista encadenada.

## Utilización del servidor

Se define como la *intensidad de tráfico por servidor*:

$$r = u / c = l / m.$$

Es la *probabilidad de que un servidor determinado se encuentre ocupado*. Según la *ley de los grandes números* esta probabilidad es aproximadamente *la fracción de tiempo que cada servidor está en uso*.

En sistemas de un solo servidor es igual a la intensidad de tráfico.

## Estado Estable en Función de Soluciones Transitorias

Los sistemas de colas que se han “*asentado*” se dice que están operando en *estado estable*.

Generalmente la operación inicial de un sistema no es indicativa de su comportamiento periódico.

Los sistemas de colas deben pasar por algún período inicial de operación antes de tener un funcionamiento:

- Uniforme.
- Predecible.

La solución y estudio de un sistema de colas se simplifica mucho si se sabe que se encuentra en *estado estable*:

- Ciertos parámetros importantes permanecen fijos.
- Resulta relativamente fácil categorizar la operación del sistema.

Las *soluciones transitorias o dependientes del tiempo*:

- Son mucho más complejas.
- Están fuera del alcance de este curso.

## Resultado de Little

Es una de las mediciones más sencillas y útiles del rendimiento de un sistema de colas.

Relaciona las siguientes cantidades:

- $W_q$ : *tiempo medio que emplea un cliente en una cola*.

- $I$ : tasa de llegadas.
- $L_q$ : número de clientes en la cola.
- $W$ : tiempo medio que emplea un cliente en el sistema.
- $L$ : número de clientes en el sistema.

El resultado de Little se expresa como:

$$L_q = I W_q$$

$$L = I W$$

## Resumen del Proceso de Poisson

Se define " $P(k,t)$ " como la probabilidad de exactamente " $k$ " llegadas en un intervalo de tiempo de longitud " $t$ ".

Un proceso es de Poisson si y solo si:

- Para intervalos apropiadamente pequeños  $\Delta t$ :
  - $P(k,t) =$ 
    - $\lambda \Delta t$  para  $k = 1$  ( $\lambda$  es la tasa promedio de llegadas).
    - $1 - \lambda \Delta t$  para  $k = 0$ .
    - $0$  para  $k > 1$ .
- Cualesquiera eventos definidos para tener lugar en intervalos de tiempo no superpuestos son *mutuamente independientes*.

Un proceso también es de Poisson si los tiempos entre llegadas sucesivas (tiempos entre llegadas de primer orden):

- Son variables aleatorias exponenciales.
- Idénticamente distribuidas.

Si la variable aleatoria " $k$ " indica el número de llegadas:

- La probabilidad de, exactamente, " $k$ " llegadas en un intervalo de longitud " $t$ " es:
  - $P(k; t) = [(I t)^k e^{-I t}] / k!; t \geq 0; k = 0; 1; 2; \dots$
- El valor esperado o valor medio de " $k$ " es:
  - $E(k) = I t$ .
- La varianza de " $k$ " es:

- $(s_k)^2 = \mathbf{1}t.$

La suma de dos variables de Poisson aleatorias independientes “x”e “y” también describen un proceso de Poisson:

- Los valores esperados son:
  - $E(y) = \mathbf{m}_2 = \mathbf{1}_2t.$   
 $E(x) = \mathbf{m}_1 = \mathbf{1}_1t.$
- La probabilidad de “k” llegadas en el tiempo “t” es:
  - $P(k; t) = [(\mathbf{1}_1 + \mathbf{1}_2)^k e^{-(\mathbf{1}_1t + \mathbf{1}_2t)} ]/k!; t \geq 0; k = 0; 1; 2; \dots$   
 $P(k; t) = [(\mathbf{m}_1 + \mathbf{m}_2)^k e^{-(\mathbf{m}_1 + \mathbf{m}_2)} ]/k!;$   
 $P(k; t) = [(\mathbf{m}_s)^k e^{-\mathbf{m}_s} ]/k!; \mathbf{m}_s = \mathbf{m}_1 + \mathbf{m}_2 :$   
 $P(k; t) = [(\mathbf{1}_s t)^k e^{-\mathbf{1}_s t} ]/k!; \mathbf{1}_s = \mathbf{1}_1 + \mathbf{1}_2$

La suma de “n” procesos de Poisson independientes resulta en un proceso de Poisson con una tasa de llegada:

$$\mathbf{1} = \sum_{i=1}^n \mathbf{1}_i$$

Para un proceso de Poisson con una tasa de llegada “ $\mathbf{1}$ ” se puede formar un nuevo proceso de Poisson utilizando borradas aleatorias independientes:

- Cada llegada al proceso original:
  - Se acepta al nuevo proceso con probabilidad “P”.
  - Se rechaza con probabilidad “1 - P”.
- La tasa de llegada del nuevo proceso derivado es “ $\mathbf{1} P$ ”.

La generalización para la descomposición de un proceso de Poisson en “n” procesos derivados independientes, cada uno con una probabilidad asociada “ $p_i$ ” resulta:

$$\begin{aligned} \mathbf{1}_n &= p_n \mathbf{1}; \\ \sum_{i=1}^n p_i &= 1. \\ \sum_{i=1}^n \mathbf{1}_i &= \sum_{i=1}^n p_i \mathbf{1} = \mathbf{1} \sum_{i=1}^n p_i = \mathbf{1}. \end{aligned}$$

En un proceso de Poisson:

- La probabilidad de que no haya llegadas en un intervalo de longitud “t” es:
  - $P(0; t) = [(\mathbf{1}t)^0 e^{-\mathbf{1}t} ]/0! = e^{-\mathbf{1}t} .$
- La probabilidad de una o más llegadas en un intervalo de longitud “t” es:

- $1 - P(0; t) = 1 - e^{-\lambda t}$ .

La función de densidad de probabilidad para el tiempo entre llegadas de primer orden (tiempo hasta la primer llegada) es:

$$f_t(t) = \lambda e^{-\lambda t}; (t \geq 0):$$

El valor esperado "t" es:

$$E(t) = 1/\lambda ?$$

La varianza es:

$$E(t) = 1/\lambda.$$

La función de densidad de probabilidad para el tiempo entre llegadas de orden r-ésimo (tiempo hasta la r-ésima llegada) es:

$$f_t(t) = (\lambda^r t^{r-1} e^{-\lambda t}) / (r-1)!; (t \geq 0; r = 1; 2; \dots).$$

El valor esperado "t" es:

$$E(t) = r/\lambda.$$

La desviación estándar es:

$$s_t^2 = r/\lambda^2.$$

Las instalaciones de servicio pueden proporcionar tiempos de servicio exponenciales:

- La probabilidad de que el tiempo de servicio sea menor o igual a "t" es:

- $P(S \leq t) = 1 - e^{-\mu t}; (t \geq 0).$

- La tasa promedio de servicio es " $\mu$ ".
- El tiempo promedio de servicio es " $1/\mu$ ".
- La función de densidad de probabilidad para el tiempo de servicio "t" es:

- $f_t(t) = \mu e^{-\mu t}; (t \geq 0).$

- La media del tiempo de servicio es:

- $E(s) = 1/\mu$

- La varianza es " $1/\mu^2$ ".

Un servidor que opera de esta manera se denomina *servidor exponencial*.

### **Análisis de un Sistema de Colas M/M/1**

Las fórmulas de estado para el sistema de colas M/M/c son las siguientes:

- *Intensidad de tráfico:*

$$u = \lambda/m = \lambda E(s).$$

- *Utilización del servidor:*

$$r = u/c.$$

- *Probabilidad de que todos los servidores estén en uso, por lo que un cliente que llega debe esperar:*

$$C(c,u) = [(u^c)/c!] / [(u^c)/c! + (1-r) \sum_{n=0}^{c-1} [u^n/n!]].$$

- *Tiempo promedio en la cola:*

$$W_q = [C(c,u)E(s)] / [c(1-r)].$$

- *Tiempo promedio en el sistema:*

$$W = W_q + E(s).$$

- *Porcentil 90 de tiempo de espera en la cola:*

$$p_q(90) = \{[E(s)] / [c(c-r)]\} \{ \ln[10C(c,u)] \}.$$

Las fórmulas de estado para el sistema de colas M/M/1 son las siguientes:

- Se deducen de las anteriores:

$$\begin{aligned} C(c,u) &= r = \lambda E(s). \\ W_q &= [\lambda E(s)] / (1-r). \\ W &= E(s) / (1-r). \\ p_q(90) &= W[\ln(10r)]. \end{aligned}$$

Seguidamente se detalla un *ejemplo* para el análisis:

- Los operadores de una empresa precisan usar un equipo especial.
- La empresa opera las 24 hs. del día.
- Los 48 operadores (como promedio) necesitan usar el equipo una vez al día.



- Los operadores llegan al equipo en forma aleatoria (llegadas de Poisson).
- El tiempo que cada operador utiliza el equipo es exponencial y como promedio es de 20 minutos.

Utilizando un sistema de colas  $m/m/1$  para modelar el uso del equipo especial del ejemplo se obtiene:

- *Utilización del equipo:*

$$u = \lambda E(s) = (48/24) \cdot (1/3) = 2/3; \quad r = 2/3; \quad E(s) = 20 \text{ minutos.}$$

- *Tiempo promedio de espera de un operador antes de usar el equipo:*

$$W_q = [rE(s)] / (1 - r) = [(2/3) \cdot 20] / (1/3) = 40 \text{ minutos. Tiempo total que un operador utiliza el equipo:}$$

$$W = W_q + E(s) = 40 \text{ min.} + 20 \text{ min.} = 60 \text{ minutos.}$$

- *Porcentil 90% de tiempo de espera en la cola:*

$$p_q(90) = W[\ln(10r)] = 60 \ln(6,667) = 113,826 \text{ minutos:}$$

Un 10 % de los operadores (unos 5 por día) sufre prolongadas esperas de casi 2 horas.

Según el resultado de Little:

- *Tasa de llegada de operadores al equipo:*

$$\lambda = 48 / 24 (60) = 1 / 30 \text{ operadores por minuto.}$$

- *Operadores en espera:*

$$L_q = (1 / 30) \cdot 40 = 1,33 \text{ operadores en espera.}$$

- *Operadores en el sistema:*

$$L = (1 / 30) \cdot 60 = 2 \text{ operadores en el cuarto del equipo.}$$

### **Conclusión:**

- *Un solo equipo no es suficiente para hacer frente a las necesidades de los operadores sin provocar esperas excesivas.*

## Análisis de un Sistema de Colas M/M/c

Seguidamente se detalla un *ejemplo* para el análisis, que es el mismo del tema anterior:

- Si la decisión es adquirir más equipos, los interrogantes son los siguientes:
  - ¿Cuántos equipos adicionales deberán adquirirse para mantener el percentil 90 de tiempo en espera por debajo de 10 minutos?.
  - ¿Deberán mantenerse todos los equipos en un lugar central, o deberán distribuirse por todo el edificio?:
    - (Nota: se debe ignorar el tiempo que les lleva a los operadores llegar hasta los equipos).

*Colocando los equipos en diferentes lugares de la empresa:*

- Cada uno de los equipos se debe analizar como un *sistema de colas M/M/1*.
- La carga de trabajo debe dividirse en partes iguales entre los equipos.
- Colocando 1, 2, 3, 4 o 5 equipos en localidades separadas obtenemos los siguientes valores:
  - Utilización del servidor:  $\rho$ : 2/3; 1/3; 2/9; 1/6 y 2/15.
  - Tiempo de espera de servicio:  $E(s)$ : 20 minutos en todos los casos.
  - Tiempo de espera en la cola:  $W_q$ : 40; 10; 5,7; 4 y 3,1 minutos.
  - Tiempo de espera en el sistema:  $W$ : 60; 30; 25,7; 24 y 23 minutos.
  - Percentil 90 de tiempo de espera en la cola:  $p_q(90)$ : 113,8; 36,1; 20,5; 12,3 y 6,6 minutos.
- *Conclusiones:*
  - Los tiempos de espera en la cola bajan muy rápido tan pronto como se añade el segundo equipo M/M/1.
  - El percentil 90 de tiempo de espera en la cola es superior a 10 minutos hasta que se añade el quinto equipo.

*Colocando todos los equipos en un lugar central:*

- Se considera un *sistema de colas M/M/2 sencillo*.
- Utilizando las fórmulas de los *sistemas M/M/c* se obtienen los siguientes valores:
  - Intensidad de tráfico:  $u$ : 2/3.
  - Utilización del servidor:  $\rho$  : 1/3:
  - Probabilidad de que todos los servidores se encuentren en este momento en uso, por lo que un cliente que llega debe esperar:  $C(c,u)$ : 1/6.
  - Tiempo promedio en la cola:  $W_q$ : 2,5 minutos.
  - Tiempo promedio en el sistema:  $W$ : 22,5 minutos.
  - Percentil 90 de tiempo de espera en la cola:  $p_q(90)$  : 7,66 minutos.

### Conclusiones:

- El percentil 90 de tiempo de espera en la cola del sistema M/M/2 es inferior al criterio de 10 minutos.
- Con solo 2 equipos centralizados en una posición se pueden eliminar los problemas de espera del sistema de un solo equipo.
- Para asegurar un percentil 90 de tiempo de espera en la cola inferior a 10 minutos serán necesarios:
  - 5 equipos M/M/1 distribuidos, o.
  - 2 equipos en una configuración M/M/2 central.

### Procesos de Markov

Un *proceso de Markov* es un modelo adecuado para describir el comportamiento de sistemas donde el sistema está situado en uno de un conjunto de estados discretos mutuamente excluyentes y colectivamente exhaustivos  $S_0, S_1, S_2, \dots, S_n$ .

El estado presente del sistema y las probabilidades de transición entre varios estados del sistema, caracterizan el comportamiento futuro del sistema.

Dado que un proceso de Markov se encuentra en un estado determinado, su comportamiento futuro *no depende de su historia anterior* a su entrada a ese estado.

Muchos procesos de Markov exhiben un *comportamiento de estado estable*, es decir que las probabilidades de que el proceso se encuentre en un estado determinado son *constantes en el tiempo*.

Se dice que un estado " $S_j$ " es *transitorio* si desde un estado " $S_k$ " que puede ser alcanzado desde " $S_j$ ", el sistema no puede regresar a " $S_j$ ".

Se dice que un estado " $S_j$ " es *recurrente* si desde cada estado " $S_k$ " alcanzable desde " $S_j$ ", el sistema puede regresar a " $S_k$ ".

Una *cadena sencilla* es una serie de estados recurrentes tal que el sistema puede llegar a cualquier estado de la cadena desde cualquier otro estado de esta.

Un *cambio de estado* en un *proceso de Markov de transición continua* puede producir cambios de estado en cualquier instante de una escala de tiempo continua.

## Procesos de Nacimiento y Muerte

Son un caso importante de los *procesos de Markov*.

Son particularmente aplicables al *modelado* de sistemas de computación.

Un *proceso de Markov de nacimiento y muerte continuo* tiene la propiedad de que:

- $\lambda_{ij} = 0$  si  $j \neq i + 1$  y  $j \neq i - 1$ .
- $\lambda_{ij}$  es la tasa a la cual ocurren las *transiciones* del estado " $S_i$ " al estado " $S_j$ ".
- $\lambda_{i(i+1)} = b_i$  es la *tasa promedio de nacimiento* desde el estado " $S_i$ ".
- $\lambda_{i(i-1)} = d_i$  es la *tasa promedio de muerte* desde el estado " $S_i$ ".
- " $P_i$ " es la *probabilidad de estado estable* de que el proceso se encuentre en el estado " $S_i$ ".

En estado estable, en cualquier intervalo de tiempo aleatorio " $t$ ", el proceso puede realizar las siguientes *transiciones* con la misma probabilidad:

$S_i \rightarrow S_{i+1}$  con una probabilidad  $P_i b_i$ .

$S_{i+1} \rightarrow S_i$  con una probabilidad  $P_{i+1} d_{i+1}$ :

$$P_i b_i = P_{i+1} d_{i+1}.$$

La *resolución de un proceso de nacimiento y muerte continuo* significa determinar los diferentes " $P_i$ " usando las relaciones:

$$P_{i+1} = (b_i / d_{i+1}) P_i.$$

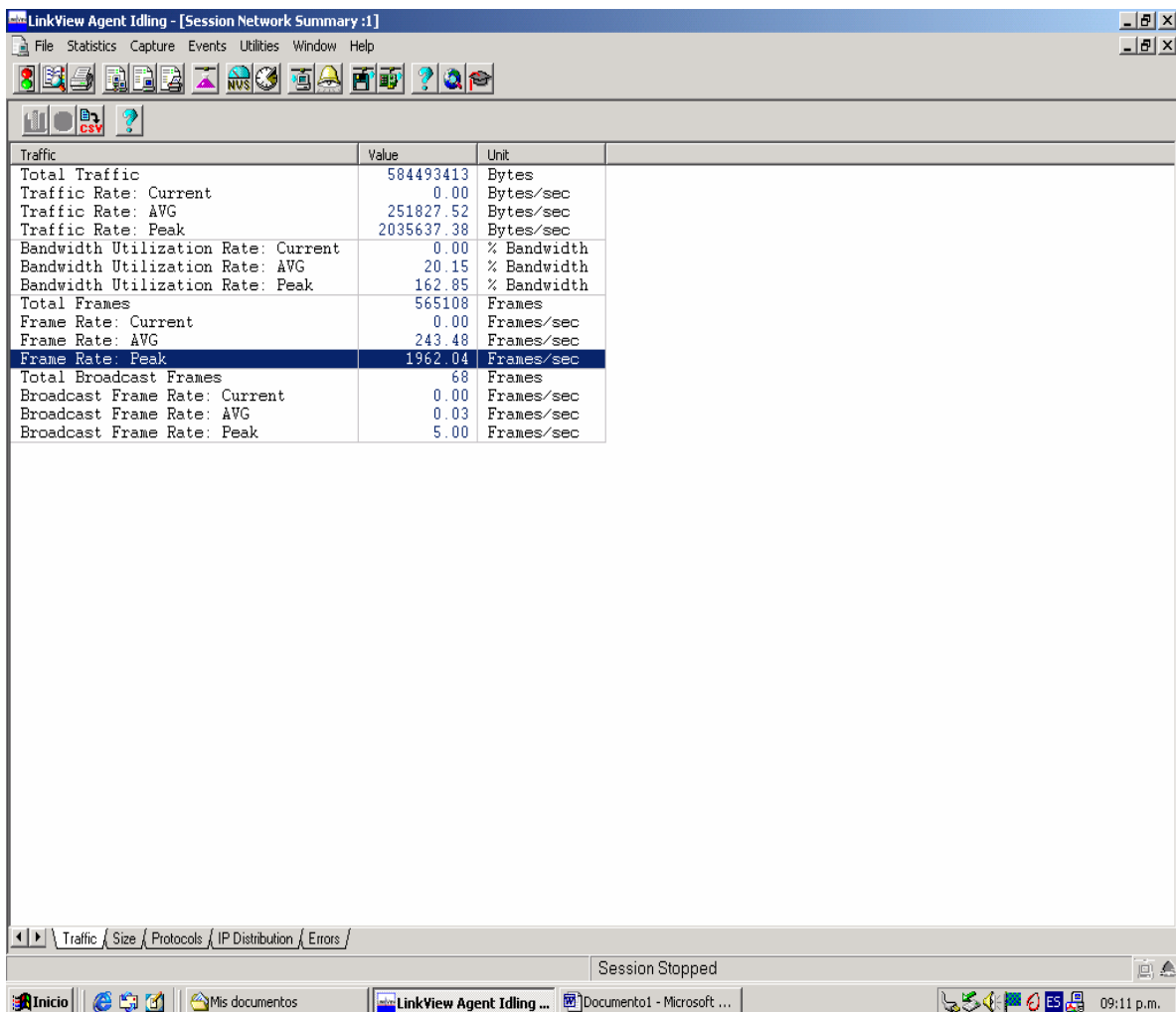
$$\sum_{i=1}^n P_i = 1.$$

## APENDICE B

Captura de pantallas del monitor de los resultados del porcentaje de uso.

Red local LAN

Estadísticas de la red.

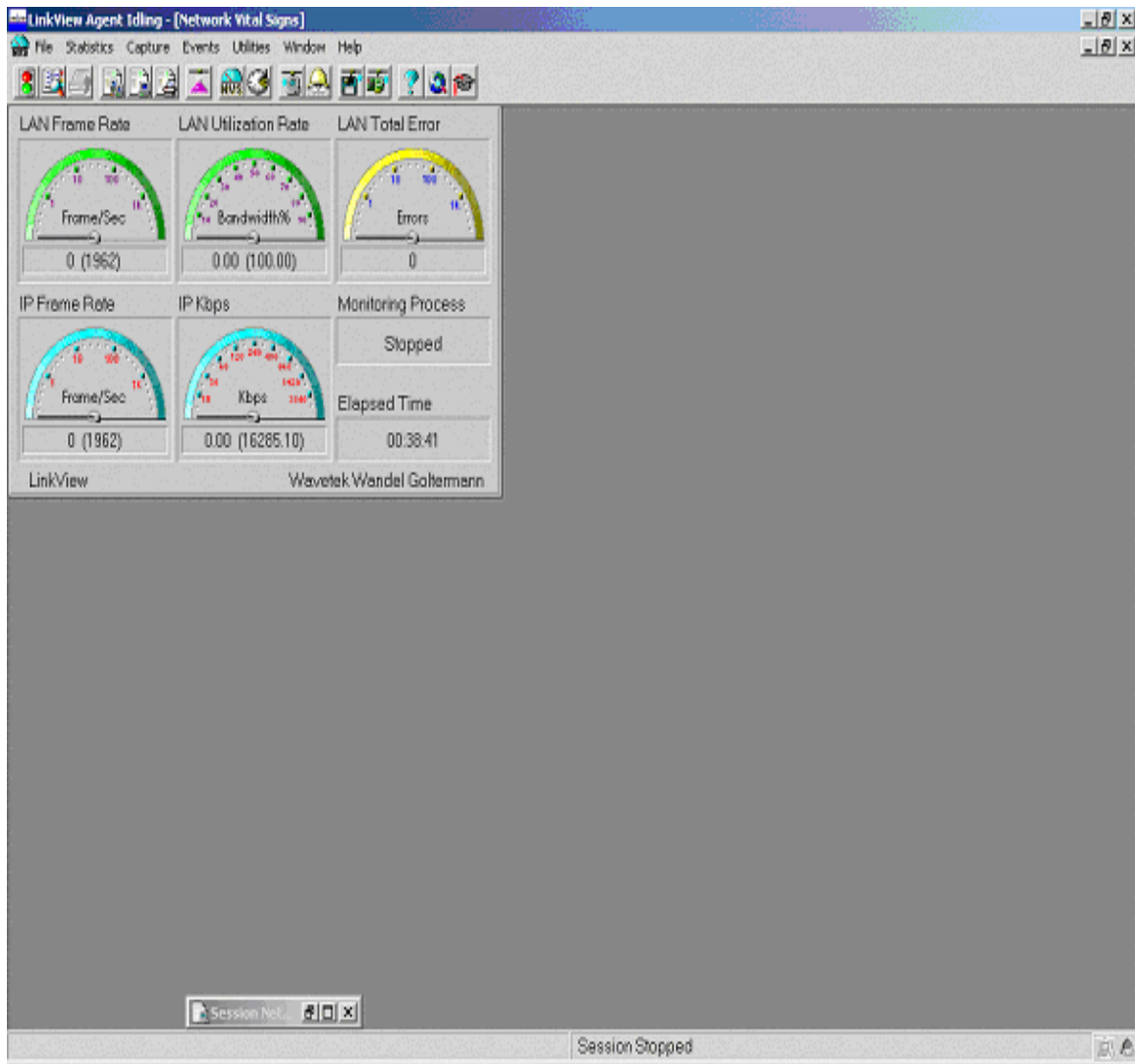


The screenshot displays the LinkView Agent Idling interface, showing a session network summary. The main window contains a table with the following data:

Traffic	Value	Unit
Total Traffic	584493413	Bytes
Traffic Rate: Current	0.00	Bytes/sec
Traffic Rate: AVG	251827.52	Bytes/sec
Traffic Rate: Peak	2035637.38	Bytes/sec
Bandwidth Utilization Rate: Current	0.00	% Bandwidth
Bandwidth Utilization Rate: AVG	20.15	% Bandwidth
Bandwidth Utilization Rate: Peak	162.85	% Bandwidth
Total Frames	565108	Frames
Frame Rate: Current	0.00	Frames/sec
Frame Rate: AVG	243.48	Frames/sec
Frame Rate: Peak	1962.04	Frames/sec
Total Broadcast Frames	68	Frames
Broadcast Frame Rate: Current	0.00	Frames/sec
Broadcast Frame Rate: AVG	0.03	Frames/sec
Broadcast Frame Rate: Peak	5.00	Frames/sec

The interface also shows a menu bar (File, Statistics, Capture, Events, Utilities, Window, Help), a toolbar with various icons, and a status bar at the bottom indicating "Session Stopped". The Windows taskbar at the bottom shows the time as 09:11 p.m. and several open applications.

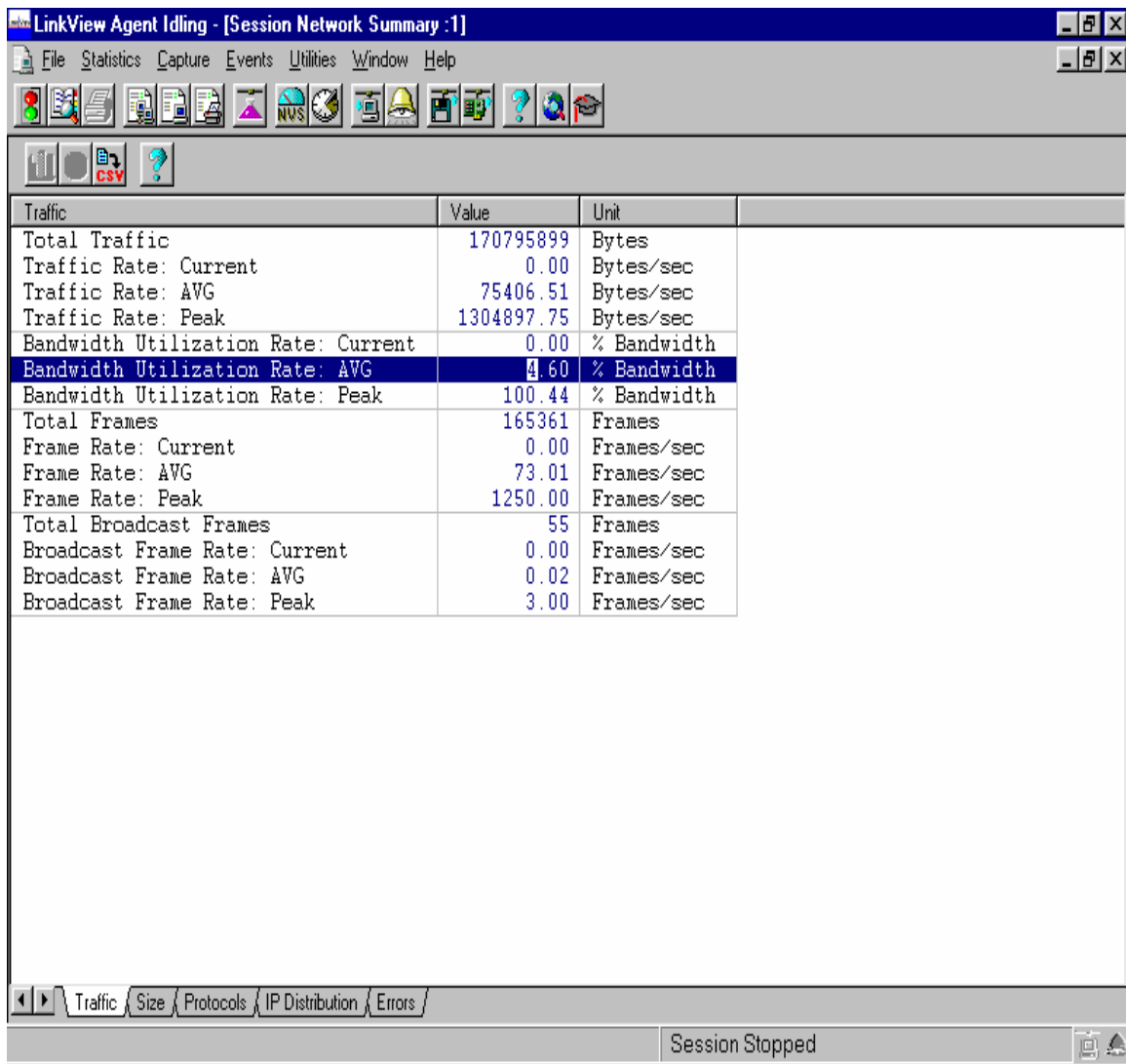
## Signos vitales de la red.



Captura de pantallas del monitor de los resultados del porcentaje de uso.

Red enlace (WAN)

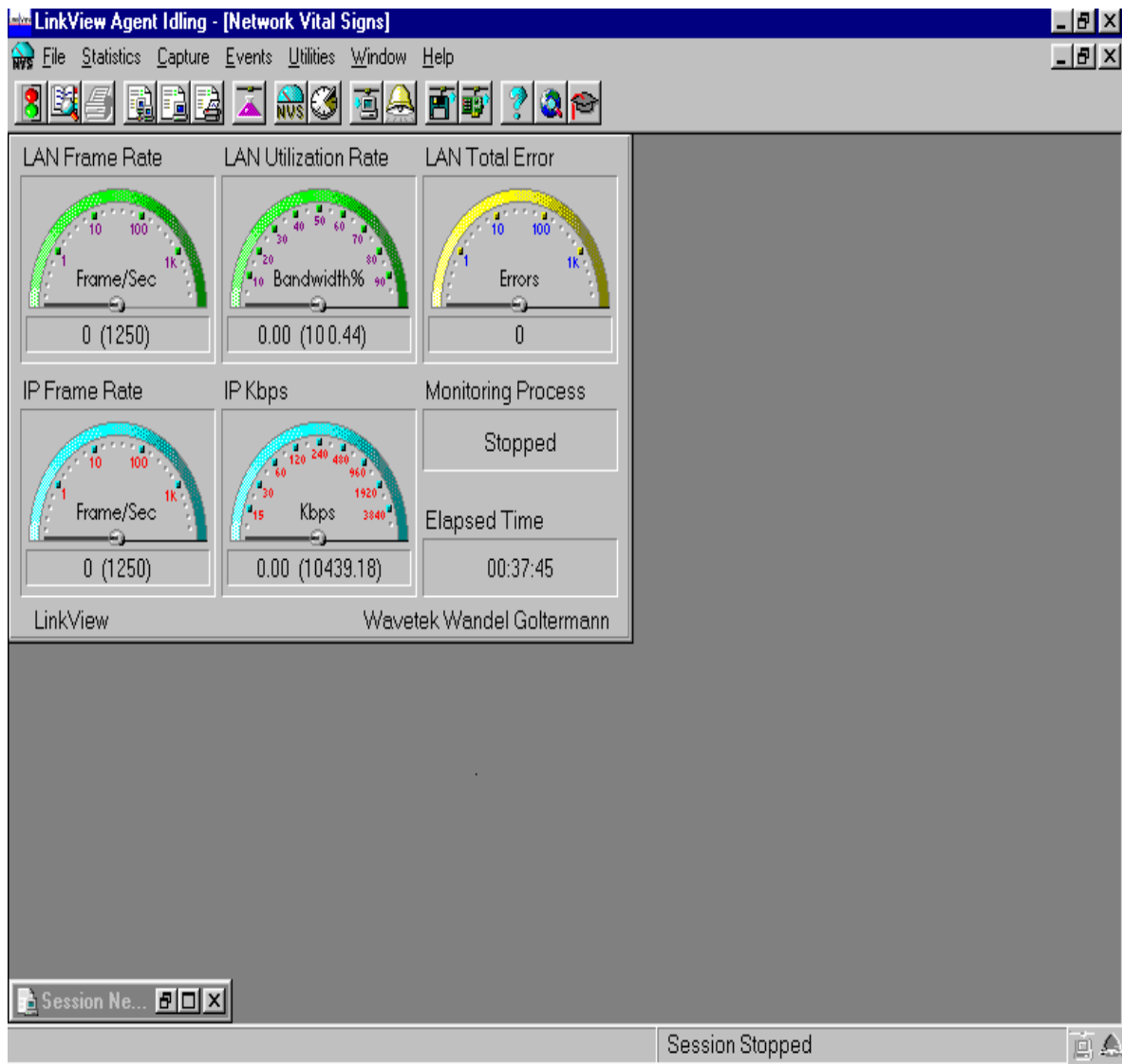
Estadísticas de la red.



The screenshot shows a software window titled "LinkView Agent Idling - [Session Network Summary :1]". The window has a menu bar with "File", "Statistics", "Capture", "Events", "Utilities", "Window", and "Help". Below the menu bar is a toolbar with various icons. The main area of the window displays a table of network statistics. The table has three columns: "Traffic", "Value", and "Unit". The "Bandwidth Utilization Rate: AVG" row is highlighted in blue. At the bottom of the window, there is a navigation bar with tabs for "Traffic", "Size", "Protocols", "IP Distribution", and "Errors". The status bar at the bottom right indicates "Session Stopped".

Traffic	Value	Unit
Total Traffic	170795899	Bytes
Traffic Rate: Current	0.00	Bytes/sec
Traffic Rate: AVG	75406.51	Bytes/sec
Traffic Rate: Peak	1304897.75	Bytes/sec
Bandwidth Utilization Rate: Current	0.00	% Bandwidth
<b>Bandwidth Utilization Rate: AVG</b>	<b>4.60</b>	<b>% Bandwidth</b>
Bandwidth Utilization Rate: Peak	100.44	% Bandwidth
Total Frames	165361	Frames
Frame Rate: Current	0.00	Frames/sec
Frame Rate: AVG	73.01	Frames/sec
Frame Rate: Peak	1250.00	Frames/sec
Total Broadcast Frames	55	Frames
Broadcast Frame Rate: Current	0.00	Frames/sec
Broadcast Frame Rate: AVG	0.02	Frames/sec
Broadcast Frame Rate: Peak	3.00	Frames/sec

## Signos vitales de la red.





## APENDICE C

Captura de pantallas del simulador de los resultados del porcentaje de uso.

Red local LAN

Intervalo de Confianza.

The screenshot shows the Express simulation software interface. On the left, a network diagram features four nodes: UL1 1:1, UL2 1:1, LAN 1:1, and WEBSERV 1:1. The LAN 1:1 node is highlighted. A 'Result File' window is open, displaying a table with data for LAN usage and a 'CONFIDENCE INTERVAL' dialog box.

	A	B	C	D	E	F	G
92	19	LAN	16.68	.03			
93	20	LAN	17.41	.04			
94	21						
95	22						
96	23						
97	24						
98	25						
99	26						
100	27						
101	28						
102	29						
103	30						
104	31						
105	32						
106	33						
107	34						
108	35						
109	R1						
110	1						
111	2						

**CONFIDENCE INTERVAL**

Sample Size: 35      Sample Mean: 16.64      Sample Variance: 9.17

There is a 90% Confidence that the True Mean is:

**Between 15.78 and 17.51 (16.64 +/- 0.87)**

Within 5.20% of the Sample Mean

Half-Width Within Acceptable Range:  OK      ? Help

Erase      \sims\MODEL.OUT Loaded

Time Last Run Ended:

Captura de pantallas del simulador de los resultados del porcentaje de uso.

Red enlace (WAN)

Intervalo de Confianza.

The screenshot shows the Express software interface. The main window displays a network diagram with nodes labeled UL1 1:1, UL2 1:1, LAN 1:1, and WEBSERV 1:1. A 'Result File' window is open, showing a table of data and a 'CONFIDENCE INTERVAL' dialog box. The dialog box displays the following information:

Sample Size	Sample Mean	Sample Variance
35	4.87	2.76

There is a 90% Confidence that the True Mean is:

**Between 4.39 and 5.34 (4.87 +/- 0.47)**

Within 9.75% of the Sample Mean

Half-Width Within Acceptable Range

Buttons: OK, Help

Buttons: Erase, \sims\MODEL.OUT Loaded

Time Last Run Ended:

## APENDICE D

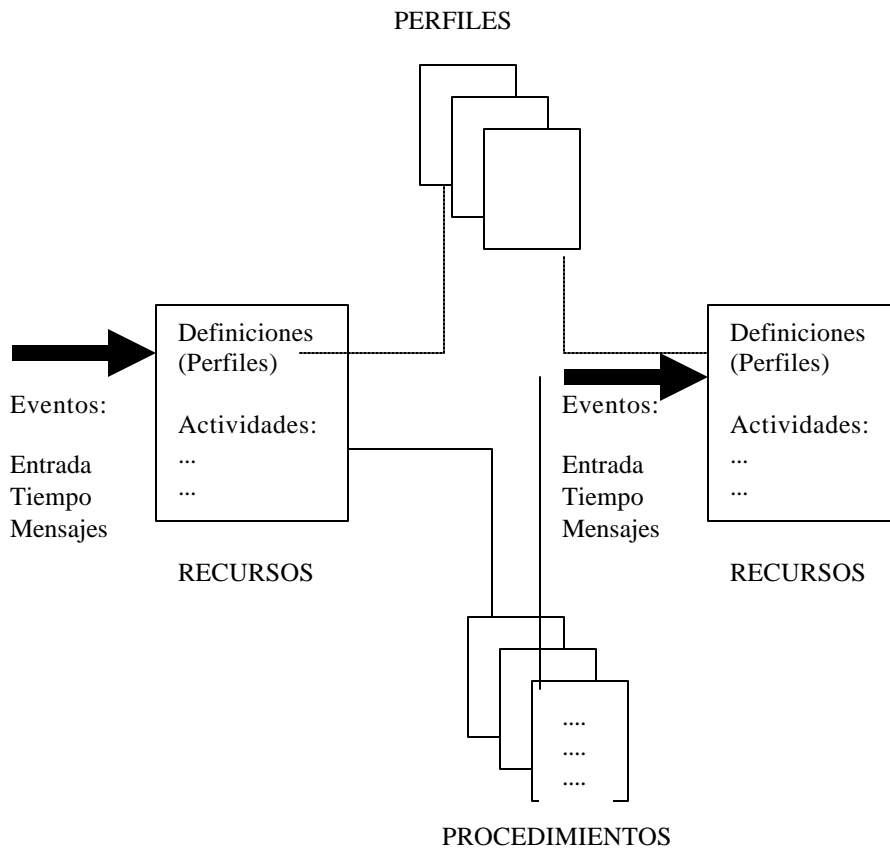
---

### USO DE PROPHECY

#### Funcionamiento lógico de Prophecy

Prophecy permite definir una red compleja en términos lógicos. Los elementos de red ya sean enrutadores, puertas de enlace (gateway) o puentes se convierten en recursos, los recursos son entidades con capacidad finita desempeñando un trabajo. Se define la capacidad de recurso a través del perfil y el tipo de trabajo a través de procedimientos, los procedimientos contienen la secuencias de eventos (procesos que representan el flujo del trabajo de un recurso en orden cronológico).

Las transacciones que fluyen a través de la red son representadas como mensajes. Los mensajes son creados como objetos dinámicos mientras la simulación esta ejecutando. Los mensajes se crean de acuerdo a las especificaciones del modelo y son borrados cuando ya no se requieren.



Figura, Operación lógica de Prophecy.

Prophesy corre con una interacción de los objetos: recursos, perfiles, procedimientos y mensajes. Cuando los mensajes llegan a un recurso el procedimiento asociado para ese evento es ejecutado. Dentro del procedimiento se especifica el retraso en procesamiento que se desea simular, todas las actividades se pueden introducir mediante paneles con línea de comando.

Un tipo de mensaje dado, puede implicar la ejecución de una aplicación de base de datos o la ejecución de un mapeo de direccionamiento de un enrutador y su lógica de enrutamiento. Es posible avanzar dicho mensaje a otro recurso sin cambiarlo o bien es posible modificar el estado del mensaje y sus demás características (tamaño, prioridad, etc.) antes de enviarlo. También es posible simplemente ignorarlo.

Prophesy proporciona un rango de opciones de como pueden ser calendarizados los eventos que dependan de los mensajes. Por ejemplo mensajes con alta prioridad son atendidos primero. También es posible especificar un índice de "paciencia" para cada mensaje en donde se indica cuanto tiempo debe esperar el mensaje en la cola de espera antes de retirarse de la misma. Esta característica permite simular situaciones de congestión.

A medida que los mensajes se crean, se entregan a colas de espera, se procesan y avanzan, Prophesy mantiene etiquetas apropiadas para mas tarde emitir estadísticas de retraso de los mensajes. Correr la simulación involucra que un flujo de mensaje viaje a través de los recursos requeridos disparando las acciones apropiadas que en respuesta ejecutan los procedimientos específicos. Los mensajes son entonces procesados, los recursos cesados para simulaciones específicas.

Algunos de los retrasos en el procesamiento se especifican en términos absolutos, pero otros son función de las características definidas en el perfil del recurso. El perfil de recurso es un archivo separado que contiene las características de desempeño asociadas al recurso, por ejemplo en el es posible especificar el número de paquetes por segundo que el recurso es capaz de atender, incluyendo retransmisiones de mensajes y periodos de termino de espera.

Prophesy calcula la utilización de recurso con base a la cantidad de trabajo necesaria para procesar el flujo de los mensajes, los mensajes descartados debido a la congestión y la utilización promedio de la espera de cola.

Cuando se modela con Prophesy existe la libertad de representar o abstraer un sistema en la forma que mejor se ajuste a tus objetivos de simulación.

Lo que Propesy hace, es generar los mensajes que tu especificas asegurándose de que estos mensajes activan los procedimientos apropiados. Propesy entonces aplica el procedimiento para calcular y acumular los retrasos y la

utilización correspondientes al recurso en el cual el procedimiento tiene lugar. También Propesy se asegura de los mensajes fluyan a través de unos recursos en el modelo de acuerdo a las trayectoria y conexiones que indiques. Las actividades que tu haces en Propesy son:

- a) Arrastrar y soltar para crear un icono para la representación de un recurso.
- b) Hacer doble click en el icono o pasar, definir el recurso y para indicar lo que el recurso debe hacer cuando recibe un mensaje de determinadas características (transacción designado / estado) típicamente querrás que el recurso ejecute un procedimiento cuando reciba un mensaje.
- c) Dar click en el botón de procedimiento para crear y definir un procedimiento, es en este procedimiento en donde tu especificas el trabajo / retraso en el que el recurso va a incidir como resultado de haber recibido un mensaje.
- d) Opcionalmente, darás click en el botón de perfiles para abrir un panel de configuración, donde crearas y definirás un perfil para cada recurso. También especificaras que perfil corresponde al uso, en el paso b.
- e) Obviamente demostraras tener un recurso generando carga de trabajo.

### **Como opera Propesy**

Hay tres pasos básicos para definir un modelo en Propesy:

1. Crear y nombrar los recursos del modelo.
2. Conectar los recursos.
3. Definir los recursos y sus transacciones.

Junto con Propesy, defines las transacciones con un nombre alfa-numérico. Propesy soporta tres tipos de transacciones:

- Generar peticiones (GEN).
- Manejar peticiones (REQ).
- Manejar respuestas (RSP).

Obviamente, al menos uno de los recursos de tu sistema debe generar transacciones (GEN), o no debe ser cualquier trabajo simularlo. Una transacción generada es, por definición, una transacción de petición. Ejemplos de transacciones generadas son:

- a) Una base de datos interrogante originándose desde una usuario de estación de trabajo.
- b) Un proceso de trabajo iniciando un flujo de trabajo específico. Por ejemplo, una encuesta de compras o de clientes.
- c) Una operación de datos de entrada.

Una vez generada, una transacción se representa una entidad de trabajo que viajará a través del sistema, visitando uno o mas recursos y sufriendo retardos

de variables, antes de que sea servido y completado. Es ascendente para ti especificar que ruta y que retardos serán incurridos por cualquier transacción junto con cada recurso.

### **Creando y nombrando recursos.**

Para crear y nombrar recursos bajo Prophesy, simplemente arrastra el icono deseado de la barra de iconos y déjalo sobre el modelo de campaña. Un panel apuntándote para el nombre del recurso será abierto. Puedes nombrar el recurso de cualquier forma que quieras, pero las siguientes reglas aplican:

- No se permite los recursos duplicados.
- El nombre del recurso en Prophesy debe ser único para el séptimo carácter.
- No hay caso de sensibilidad. Que es: recurso = RECURSO.

### **Conectar los recursos.**

Después de que has establecido los recursos sobre el trabajo de campaña, los conectas dando click izquierdo en el ancho de los recursos (la conexión “original”) y entonces da click derecho para conectar los recursos.

### **Definir los recursos y sus manejadores de transacciones.**

Después de crear y nombrar tus recursos del modelo, entonces procederás al definir las características del recurso y los flujos de transacción y retardos a través de tu modelo. Doble click sobre el recurso inicial (el recurso donde muchas de las transacciones serán generadas), y definir la transacción inicial para ser generada.

Una vez que los recursos del panel abren, puedes especificar el número de unidades de trabajo por marca que puede generar (el termino “instrucciones por marca” es usado). Aquí, también puedes especificar el número de recursos representados por el icono. Prophesy aumenta la capacidad de “instrucciones por marca” en conformidad. Puedes también indicar si el recurso es capaz de procesar diversas transacciones a la vez (caja de diálogo “Concurrente”) hablando ventajas de tiempos ideales.

Desde el panel de definición de recursos, especificas a las transacciones este recurso con manejo (transacción de manejo). Para definir una nueva transacción, click sobre el botón NEW para abrir el panel de manejador de transacciones.

El panel de transacciones contiene tres secciones:

En Entrada: usado para nombrar la transacción ajustando el manejador y, opcionalmente para especificar transacciones generadas y sus atributos.

En Proceso: usado para especificar los retardos de las transacciones que ocurrirán.

En Envío: usado para especificar cual recurso para entregar la siguiente transacción, si cualquier; y opcionalmente especificar los atributos de transacción de salida.

### **En Entrada.**

La sección en Entrada requiere el nombre de la transacción que será manejada. Elige o escribe el nombre de la transacción. Los nombres de transacción son casos intensivos y solo significantes a los primeros 8 caracteres.

Si la transacción esta para ser generada en este recurso, click sobre el botón GENERATE. Entonces esperaras introducir el número de transacciones por marca para generar; el tamaño del mensaje, y prioridad. Todos los mensajes en Prophecy son generados con una distribución Poisson (exponencial negativa). Los mensajes generados son, por definición, transacciones tipo petición.

### **En Proceso.**

La porción de en Proceso del panel te permite especificar uno o más tiempos de recursos, esperar / proceso:

Ocupado: aquí puedes especificar en número medio de marcas para hacer este recurso ocupado para una transacción posterior de este tipo. El tiempo ocupado es exponencial negativo.

Desocupado: aquí especificas el número medio de marcas que el recurso estará desocupado para cada transacción dentro. Nótese que si el recurso no fuera definido como Concurrente, los tiempos de desocupado bloquearan el nombre de la misma forma que el tiempo ocupado lo hace. Los recursos concurrentes tienen la habilidad de acordar recursos a mensajes adicionales mientras se desocupa para otra transacción.

“Ejecutar” n instrucciones. Si especificaste un valor en el número de instrucciones por campo de marca del panel de recursos, Prophecy aplicara el retardo requerido para simular el número de instrucciones que especificaste.

“Ejecutar” n instrucciones por x mensaje de byte. Prophecy simulara la ejecución del número específico de instrucciones, veces del tamaño de mensaje.

## **En Envío.**

En la sección de en Envío, especificas los nombres del recurso donde la transacción será entregada. Puedes definir a cuatro destinos diferentes con el porcentaje respectivo de veces que el mensaje debe ser entregado a cada recurso. Los recursos de destino deben ser diferentes desde el recurso fuente. Debes regresar la transacción a los recursos de destino como una transacción de petición. Pero si un recurso en concreto esta actualmente hiendo a regresar la transacción a un recurso de petición, debes determinar la transacción de banda de salida como una respuesta. En envío puedes también modificar el tamaño del mensaje y darle prioridad revisando la caja de dialogo Modify.

## **Corriendo una simulación con Prophesy .**

Una vez que has definido tu modelo, debes determinar las opciones de corrida de simulación apropiada.

Dale click al botón PLAY para correr la simulación. Una vez mas, las operaciones y estadísticas son equivalentes para aquellas disponibles en Prophesy.

## **Especificando las características de desempeño.**

En propesy todos los nodos de red son recursos. Existen tres formas de especificar el retraso de trabajo en cada recurso

- a) Indicando la capacidad de instrucciones por tick (golpe) por recurso y la longitud de trayectoria de ejecución involucrada al simular la llegada de un mensaje. Por ejemplo especificando el número de instrucciones por segundo (o paquetes por segundo) que el recurso puede manejar y luego cuantas instrucciones se ejecutan por mensaje.
- b) explícitamente introduciendo el retraso real por mensaje.
- c) combinado los dos métodos anteriores.

¿Cómo sabes cuales características especificas de desempeño incluir? Vas a necesitar aplicar una mezcla de ingenio, investigación bibliográfica (revistas o hojas de especificaciones de productos) y comparaciones.

Identificar las características de desempeño de una computadora un software y una red no es únicamente un arte: Es un arte negro, con la gran cantidad de configuraciones de hardware, protocolos de red, variables de configuración, sin mencionar sistemas operativos y accesorios de hardware disponibles en el mercado, es imposible para nosotros (o para cualquiera) formular respuestas predictivas.



En la medida de lo posible deberían existir puntos de comparación de componentes de sistemas para obtener los parámetros de desempeño para ser introducidos en los perfiles de recurso.

En muchos casos los mismos fabricantes proveeran información del desempeño de sus equipos. Mientras que se deben ver estos datos con saludable escepticismo, a veces no hay alternativas disponibles entonces se deben de utilizar los datos del fabricante como punto de partida.

Después de obtener la información deseada para un recurso se puede introducir en un perfil de Prophesy que será utilizado durante la simulación. Prophesy incluye ciertos ejemplos de perfiles que se pueden ajustar como se deseen.

### **Especificando las características de desempeño de los elementos de red en Propshy**

En Prophesy, todos los nodos de la red son recursos. Hay tres maneras de especificar retrasos de trabajo en cada recurso:

- a. Especificando el número de instrucciones por segundo (o paquetes por segundo)
- b. Entrando en el retraso real explícitamente por el mensaje
- c. Combinando los dos métodos anteriores.

¿Cómo sabes qué características del desempeño específicas debes incluir? Necesitarás aplicar una mezcla de ingeniosidad, investigación de la literatura (revistas o hojas de especificación de producto), y referencias, para obtener estas características.

Es importante mencionar que el desempeño de las computadoras, software y las redes, no solo es un arte, son un arte negro, las configuraciones del hardware, protocolos de la red, son configuraciones variables, sin mencionar sistemas operativos y hardware que se adiciona, es imposible para nosotros (ò para todos) venir a pre-cocinar respuestas.

Siempre debe haber referencias de los componentes del sistema para obtener los parámetros de consumo de recursos que serán configurados dentro de los perfiles del recurso.

En muchos de los casos, los vendedores proporcionan valuaciones del desempeño, y debemos considerar estas valuaciones con escepticismo, y como punto de arranque

Después de obtener la información deseada para un recurso, puedes introducirlo como perfil, para ser usado durante la ejecución de la simulación. Prophesy cuenta con algunos perfiles que puedes ajustar a tus necesidades.

## **Perfiles de las computadoras.**

Una de las formas de poder medir el desempeño de las computadoras es por medio del Mips (Million of instruction per seconds), este concepto es una consecuencia de la velocidad de procesamiento de la unidad de central de procesamiento (CPU's) de las maquinas, usualmente dado en frecuencia.

La antigua PC-XT tiene una tasa de velocidad de 0.8 Mips. La primera generación de las 286 corrían a 6 Mhz, con un estado de espera, pero la 286 empezó a tomar popularidad cuando corría a 12 Mhz sin espera para memoria, aproximadamente tenía una tasa de velocidad de 8 Mips, sucesivamente, fue creciendo los procesadores hasta llegar a la generación de los Pentium que llegan a tener velocidades hasta de 100 Mips.

Esta generación de procesadores está dentro de la arquitectura de CISC, (Common Instruction Set Computing), sin embargo hay otras arquitecturas que tienen mayor velocidad como la RISC (Reduce Instruction Set Computing) que están siendo más populares.

No hay una equivalencia general entre CISC y RISC, pero si un porcentaje de acercamiento del 60% de Mips de velocidad mas para RISC de CISC, por ejemplo, una maquina RISC tiene una tasa de 200 Mips, mientras una maquina CISC se aproxima a una taz de 120 Mips.

## **Perfiles de los enrutadores y puentes.**

Los enrutadores y puentes, hoy tienen capacidad para manejar el caudal mejor conocido como througput (medida de la velocidad de la transferencia de los datos através de comunicaciones complejas) igual o más rápidos que los circuitos que ellos conectan. Aunque los enrutadores son altamente optimizados, usualmente no podemos simular este como una extinción de la tubería del circuito, por que los enrutadores imponen un retraso de latencia sobre los paquetes que van arribando a él.

El tiempo de latencia del mensaje, entre otras cosas, esta dado en función del protocolo, la inter-llegada de los paquetes a el y la velocidad de procesamiento de este.

Los enrutadores usualmente consisten en una cadena de tarjetas con procesadores, capaz de recibir, inspeccionar y retransmitir paquetes, estas entidades son conocidas como puertos y ocupan un procesador RISC, su promedio de procesamiento de estos puertos esta en los rangos de los 30 Mips, frecuentemente los enrutadores miden su througput en tramas por segundos, para diferentes tamaños de trama.

La información que se necesita para simular la latencia de los enrutadores, lo podemos extraer de las mismos especificaciones de los enrutadores, como

recomendación si no tenemos alcance a esta información podemos sugerir un promedio de 20 CICS, por byte reenviado y 1000 instrucciones por paquete, estas estimaciones se aplican a cada puerto y no incluye el retardo de la latencia de llegadas a este.

Un tiempo razonable para un procesador RISC, asumiendo que es usado UNIX, es que cada procesador entregara 100 peticiones/respuestas cada segundo, Esto es, 100 transacciones, saturando el recurso, teniendo en cuenta que cada mensaje esta contenido en un paquete de 256 bytes.

### **Perfiles de Servidores.**

Cuando especificamos servidores, estaremos usando mas el concepto de mensajes que paquetes, estos mensajes usualmente están escritos en un lenguaje de alto nivel, como C, nosotros podemos asumir que el servidor tiene un procesamiento de 5000 instrucciones en el lenguaje C, un mensaje de 256 bytes tiene un camino de ejecución de 30 a 600 instrucciones de C.

Este valor representa el código típico para inspeccionar un mensaje llegando, procesado en memoria de la tabla de ruteo, mapeado y ser remitido el mensaje a la pila del protocolo destino.

Note, todas estas suposiciones, tal vez cambien el ambiente de la simulación, por el lenguaje maquina que se utilice y el compilador que trabaje con este.