

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA
Y ELECTRICA
DIVISION DE ESTUDIOS DE POSGRADO



DISEÑO DE UN CONTROLADOR PARA LA
ADMINISTRACION JERARQUICA DE RECURSOS
EN UN SISTEMA DE TELECOMUNICACIONES
CON ASIGNACION DE ANCHO DE BANDA

POR

ING. RICARDO ALONSO FLORES TORRES

TESIS

EN OPCION AL GRADO DE MAESTRO EN
CIENCIAS DE LA INGENIERIA CON
ESPECIALIDAD EN TELECOMUNICACIONES

SAN NICOLAS DE LOS GARZA, N. L.
DICIEMBRE DEL 2003

TM
Z5 153
.M2
FIME
2003
.F4

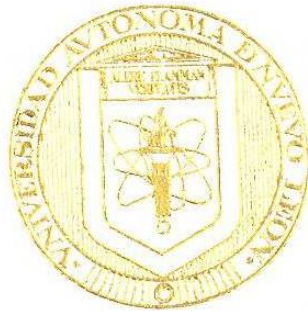
ING. RICARDO ALONSO FLORES TORRES
DICIEMBRE DEL 2003



1020131294

~~2~~

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA
Y ELECTRICA
DIVISION DE ESTUDIOS DE POSGRADO



DISEÑO DE UN CONTROLADOR PARA LA
ADMINISTRACION JERARQUICA DE RECURSOS
EN UN SISTEMA DE TELECOMUNICACIONES
CON ASIGNACION DE ANCHO DE BANDA

POR

ING. RICARDO ALONSO FLORES TORRES

TESIS

EN OPCION AL GRADO DE MAESTRO EN
CIENCIAS DE LA INGENIERIA CON
ESPECIALIDAD EN TELECOMUNICACIONES

SAN NICOLAS DE LOS GARZA, N. L.
DICIEMBRE DEL 2003



2003

TH

Z5853

.M2

FINE

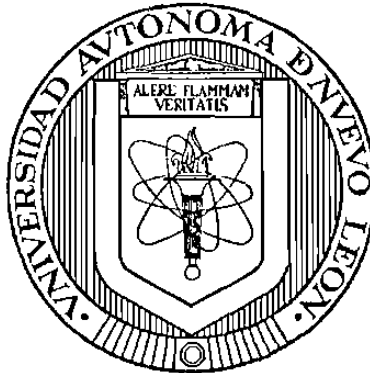
2003

.F4



FONDO
TESIS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POST-GRADO



**DISEÑO DE UN CONTROLADOR PARA LA ADMINISTRACIÓN
JERÁRQUICA DE RECURSOS EN UN SISTEMA DE
TELECOMUNICACIONES CON ASIGNACIÓN DE ANCHO DE
BANDA**

POR

ING. RICARDO ALONSO FLORES TORRES

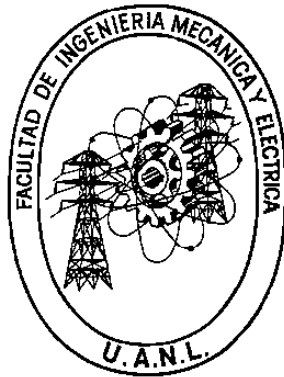
TESIS

**EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS DE LA
INGENIERÍA CON ESPECIALIDAD EN TELECOMUNICACIONES**

**SAN NICOLAS DE LOS GARZA, N.L.
DICIEMBRE DEL 2003**



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POST-GRADO



**DISEÑO DE UN CONTROLADOR PARA LA ADMINISTRACIÓN
JERÁRQUICA DE RECURSOS EN UN SISTEMA DE
TELECOMUNICACIONES CON ASIGNACIÓN DE ANCHO DE
BANDA**

POR

ING. RICARDO ALONSO FLORES TORRES

TESIS

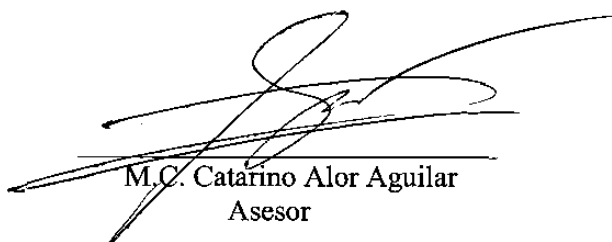
**EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS DE LA
INGENIERÍA CON ESPECIALIDAD EN TELECOMUNICACIONES**

**SAN NICOLAS DE LOS GARZA, N.L.
DICIEMBRE DEL 2003**

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Post-grado

Los miembros del comité de tesis recomendamos que la tesis “ **Diseño de un Controlador para la Administración Jerárquica de Recursos en un Sistema de Telecomunicaciones con Asignación de Ancho de Banda** ”, realizada por el alumno Ingeniero Ricardo Alonso Flores Torres, matrícula 0271352 sea aceptada para su defensa como opción al grado de Maestro en Ciencias de la Ingeniería con especialidad en Telecomunicaciones.

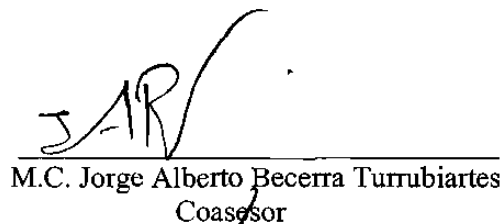
El Comité de Tesis



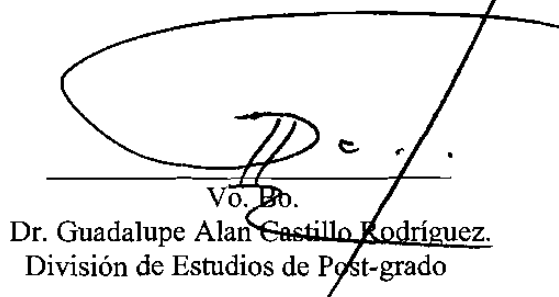
M.C. Catárino Alor Aguilar
Asesor



M.C. Fernando Estrada Salazar
Coasesor



M.C. Jorge Alberto Becerra Turrubiartes
Coasesor



Vó. Bo.
Dr. Guadalupe Alan Castillo Rodríguez.
División de Estudios de Post-grado

Ciudad Universitaria, a Diciembre del 2003.

DEDICATORIA.

A DIOS:

Te doy gracias por haberme permitido llegar felizmente a este día.
Por todo las cosas buenas y las cosas malas, por que, aunque por más difícil
que se ponga el camino no me abandonas.

A MI ESPOSA

ME L Y

Por apoyarme, por ser siempre mi inspiración para salir adelante, por
todo su amor y comprensión.

AGRADECIMIENTOS.

A mi esposa por estar conmigo en las buenas y en las malas.

A mi asesor, por transmitirme sus conocimientos y experiencias y ayudarme a realizar uno de mis sueños.

A mis coasesores de maestría, por todos los conocimientos transmitidos.

A mis maestros de maestría, por todos los conocimientos transmitidos.

A mis amigos y familiares por todo su apoyo, ellos me han acompañado en las buenas y las malas, a mis compañeros maestros y a todos los que de alguna u otra forma me apoyaron para la realización de mis estudios de maestría, que no los menciono por nombre no vaya a olvidarme de alguien, pero saben quienes son.

PRÓLOGO.

Por muchos años la Internet fue usada principalmente por científicos para intercambiar información entre ellos. El acceso remoto (TELNET), la transferencia de archivos (FTP) y el correo electrónico eran aplicaciones que no exigían muchos recursos por parte de la red, por lo que la Internet funcionaba muy bien.

Las aplicaciones en tiempo real, sin embargo, requieren de un nivel mínimo de recursos para operar efectivamente. Las aplicaciones interactivas tales como la telefonía por Internet son sensibles a retardos y a la pérdida de paquetes. Cuando se excede un cierto nivel en el retardo o la tasa de pérdidas, estas aplicaciones llegan a ser literalmente inservibles. A diferencia de la transferencia de archivos, que pueden tolerar una cierta cantidad de retardos y pérdidas sin degradar el rendimiento percibido. Los requerimientos del cliente también varían, dependiendo del uso que están dando a la Internet. Por ejemplo, las organizaciones que usan la Internet para transacciones bancarias o para el control de un equipo industrial, están dispuestas a pagar más para recibir un trato preferencial a su tráfico.

La manera tradicional de manejar el tráfico es utilizando el esquema de servicio sin garantías, el cual no permite que estas nuevas aplicaciones puedan ser transportadas dentro de la red IP de manera satisfactoria, debido a que la Internet actual no soporta algunas formas de asignación activa de recursos. La red trata a todos los paquetes individuales exactamente de la misma forma y atiende a los paquetes basados en el principio “**El primero que llega es el primero que sale**” (FIFO, first in, first out). Tampoco existe un control de admisión, pues los usuarios pueden enviar paquetes en la red tan rápido como sea posible, y no hace ninguna diferenciación en cuanto al tipo de paquete que se trate.

El objetivo del presente trabajo es el de diseñar un controlador para la administración jerárquica de recursos en un sistema de telecomunicaciones con asignación de ancho de banda, usando el algoritmo WFQ jerárquico y estando en contacto directo con los paquetes que provengan de todos los usuarios y con el enlace físico de transmisión, para evitar tiempos muertos cuando haya paquetes por transmitir.

CONTENIDO

	Página
Capítulo I Introducción	1
I.1 Antecedentes	1
I.2 Planteamiento del problema	3
I.3 Objetivos	5
I.4 Infraestructura	6
I.5 Organización de la tesis	7
Capítulo II. Arquitecturas de Asignación de Recursos	9
II.1 Introducción	9
II.2 Calidad de Servicio	10
II.3 Modelos de servicios	12
II.3.1 Mejor esfuerzo	12
II.3.2 Servicios Integrados (IntServ)	13
II.3.2.1 Introducción	13
II.3.2.2 Elementos de la Arquitectura	14
II.3.2.3 Clases de Calidad de servicio	15
II.3.2.4 Protocolo de Reservación de Recursos (RSVP)	16
II.3.2.5 Mensajes RSVP	18
II.3.2.6 Mecanismo de funcionamiento IntServ	19
II.3.3 Servicios Diferenciados (DIFFSERV)	20
II.3.3.1 Introducción	20
II.3.3.2 Dominio DiffServ	22
II.3.3.3 Comportamiento por salto (PHB)	23
II.3.3.4 Clasificación y acondicionamiento de tráfico	25
II.3.3.5 Mecanismos DiffServ	29
II.3.3.5.1 Clasificadores	29
II.3.3.5.2 Medidor y marcador	30
II.3.3.5.3 Descartador	32

CONTENIDO (Continuación)

	Página
II.3.3.5.3.1 Calendarizador de paquetes	32
II.3.3.5.3.2 Prioridad Simple	34
II.3.3.5.3.3 “El primero que entra, es el primero que sale” (FIFO)	35
II.3.3.5.3.4 Colas con prioridades (PQ)	36
II.3.3.5.3.5 Encolamiento justo (FQ)	37
II.3.3.5.3.6 Encolamiento justo ponderado (WFQ)	38
II.3.3.5.4 Acondicionamiento	39
II.3.3.5.4.1 Acondicionamiento de tráfico (TSWTCM)	40
II.3.3.5.4.2 Acondicionador leaky bucket (LB)	42
Capítulo III Controlador WFQ	44
III.1 Introducción	44
III.2 Modelo de Fluido	44
III.3 Mecanismo general para la repartición de un procesador (GPS)	45
III.3.1 Curvas de llegada – Salida	46
III.4 Encolamiento Justo Ponderado (WFQ)	50
III.4.1 Introducción	50
III.4.2 Mecanismo general para la repartición de un procesador de paquetes (PGPS)	51
III.4.3 Implementación del algoritmo WFQ	52
III.4.3.1 Diagrama de flujo del algoritmo WFQ	54
III.4.4 Posibles modalidades refinadas del algoritmo WFQ	56
III.4.4.1 El retardo más largo primero (LDF, Longest Delay First)	56

CONTENIDO (Continuación)

	Página
III.4.4.2 Menor tiempo para sobre flujo (LTO, Least Time to Overflow)	57
III.4.4.3 WFQ dinámico basado en predicción (DP, Dynamic Predictive)	57
III.4.5 Modelo típico de un enrutador que incluye un controlador WFQ	58
III.5 Modelo WFQ jerárquico	59
III.5.1 Debilidades de los sistemas jerárquicos actuales	61
III.6 Modelo jerárquico de dos niveles (CH2WFQ)	62
Capítulo IV. Implementación de los Modelos Jerárquicos WFQ y CH2WFQ	64
IV.1 Introducción	64
IV.2 Modelo de nodos jerárquico tradicional	67
IV.3 Modelo de nodos jerárquico CH2WFQ	70
IV.4 Modelo de Procesos para WFQ	75
IV.5 Modelo de procesos CH2WFQ	83
Capítulo V. Simulación y Análisis de Resultados	87
V.1 Simulación	87
V.2 Evaluación	87
V.2.1 Parámetros de simulación	88
V.2.2 Parámetros de desempeño	90
V.2.3 Resultados de la Simulación	91
V.2.3.1 Primer conjunto de experimentos	92
V.2.3.2 Segundo Conjunto de Experimentos	95

Capítulo VI Conclusiones y Trabajos Futuros	101
VI.1 Conclusiones	101
V1.2 Trabajos Futuros	103

LISTA DE FIGURAS

Figura		Página
1	Diferentes tipos de tráfico	11
2	Operación básica RSVP	20
3	Nodos internos y de frontera en un dominio DS	23
4	Clasificación y Acondicionamiento de tráfico	26
5	Clasificador de Paquetes	30
6	Elementos de la red usando el modelo de la cubeta con fichas	31
7	Calendarizador de paquetes	33
8	Modelo de Encolamiento FIFO	36
9	Modelo de encolamiento PQ	37
10	Encolamiento tradicional: Una cola por línea de salida	38
11	Encolamiento justo: Una cola por flujo	38
12	Modelo de Encolamiento WFQ	39
13	Diagrama a bloques del acondicionador TSWTCM	40
14	Acondicionador Leaky Bucket	43
15	Modelo de Fluido y de paquetes	45
16	Curva de llegada – salida para el flujo 1	48
17	Curva de llegada – salida para el flujo 2	48
18	Diagrama de Flujo del algoritmo WFQ	54
19	Estructura típica de un enrutador que soporta Servicios Diferenciados	58
20	Propuesta tradicional para WFQ jerárquico	61
21	Nueva Propuesta CH2WFQ	63
22	Modelo de nodos jerárquico WFQ	69
23	Modelo de nodos jerárquico de dos etapas CH2WFQ	71
24	Modelo de Procesos WFQ	73
25	Diagrama de flujo del estado de inicio	74
26	Diagrama de flujo del estado de arribo	76
27	Diagrama de flujo del estado de inicio de servicio	79

LISTA DE FIGURAS (Continuación)

Figura		Página
28	Diagrama de flujo del estado de terminación de servicio.	81
29	Diagrama de flujo del estado inactivo.	82
30	Bits transmitidos en el primer conjunto de simulaciones.	93
31	Bits destruidos en el primer conjunto de simulaciones.	94
32	Retardo promedio del paquete en el primer conjunto de simulaciones	95
33	Jitter del paquete en el primer conjunto de simulaciones.	95
34	Bits transmitidos en el segundo conjunto de simulaciones.	96
35	Bits destruidos en el segundo conjunto de simulaciones.	97
36	Retardo promedio del paquete en el segundo conjunto de simulaciones.	98
37	Jitter del paquete en el segundo conjunto de simulaciones.	98
38	Bits destruidos en el segundo conjunto de simulaciones.	99

LISTA DE TABLAS

Tabla		Página
1	Asignación de color	70
2	Parámetros utilizados en la simulación	90

CAPÍTULO I INTRODUCCIÓN

I.1 ANTECEDENTES

Por muchos años la Internet fue usada principalmente por científicos para intercambiar información entre ellos. El acceso remoto (TELNET), la transferencia de archivos (FTP) y el correo electrónico eran aplicaciones que no exigían muchos recursos por parte de la red, por lo que la Internet funcionaba muy bien.

La aparición de las aplicaciones en tiempo real, como la transmisión de voz (VoIP), videoconferencia, telecontrol, transacciones comerciales y bancarias, música, imágenes de alta resolución entre otros, ha hecho que aumente la necesidad de brindar a los usuarios una gama más amplia de servicios, desarrollando algoritmos más sofisticados y a la vez más complejos para la administración de recursos de estas redes, incrementado a su vez, la capacidad de los medios de transmisión y de los enrutadores.

La manera tradicional de manejar el tráfico es utilizando el esquema de servicio sin garantías, el cual no permite que estas nuevas aplicaciones puedan ser transportadas dentro de la red IP de manera satisfactoria, debido a que la Internet actual no soporta algunas formas de asignación activa de recursos. La red trata a todos los paquetes individuales exactamente de la misma forma y atiende a los paquetes basados en el principio “el primero que llega es el primero que sale” (FIFO, first in, first out). Tampoco existe un control de admisión, pues los usuarios pueden enviar paquetes en la red tan rápido como sea posible, y no hace ninguna diferenciación en cuanto al tipo de paquete que se trate.

Las aplicaciones en tiempo real, sin embargo, requieren de un nivel mínimo de recursos para operar efectivamente. Las aplicaciones interactivas tales como la telefonía por Internet son sensibles a retardos y a la pérdida de paquetes. Cuando se excede un cierto nivel en el retardo o la tasa de pérdidas, estas aplicaciones llegan a ser literalmente inservibles. A diferencia de la transferencia de archivos, que pueden tolerar una cierta cantidad de retardos y pérdidas sin degradar el rendimiento percibido. Los requerimientos del cliente también varían, dependiendo del uso que están dando a la Internet. Por ejemplo, las organizaciones que usan la Internet para transacciones bancarias o para el control de un equipo industrial, están dispuestas a pagar más para recibir un trato preferencial a su tráfico.

Para resolver éstos problemas, la IETF (Internet Engineering Task Force) ha desarrollado nuevas tecnologías y estándares para proporcionar recursos garantizados y diferenciación de servicios en la Internet, bajo el término de Calidad de Servicio (QoS) (ver sección II.2). El objetivo de QoS es distribuir adecuadamente el ancho de banda disponible en los enlaces digitales. Por ejemplo, algunas aplicaciones como las videoconferencias, necesitan tener una tasa de arribo con muy poca variación (jitter pequeño). Para realizar esto, las arquitecturas de QoS se basan en los mecanismos de calendarización de paquetes en las colas (ver sección II.3.3.4.3).

Para soportar las nuevas aplicaciones, es necesario establecer diferentes niveles de servicio, los cuales garanticen al usuario un desempeño adecuado a la aplicación. Los proveedores de servicio de red se pueden auxiliar de los convenios de nivel de servicio (SLA, Service Level Agreements), los cuales se encargan de caracterizar los perfiles de

conducta que se le aplica al tráfico en los nodos y las políticas que se le deben aplicar a cada flujo agregado apoyándose en las arquitecturas de QoS para mantener los niveles de desempeño dentro de los límites permitidos.

Las tecnologías que han emergido para soportar QoS en la Internet son los servicios integrados (ver sección II.3.2) y servicios diferenciados (ver sección II.3.3), los cuales reservan y asignan recursos a varios tipos de flujos en una red.

I.2 PLANTEAMIENTO DEL PROBLEMA

Cuando se desea dar, a través de una misma red, diferentes servicios a un conjunto heterogéneo de usuarios que por naturaleza tienen diferentes requerimientos en cuanto a calidad de servicio, es necesario identificar los diferentes tipos de tráfico dentro de los equipos de comunicación y darle un tratamiento diferente a cada uno de ellos. Una de las propuestas para lograr este fin en la Internet de la nueva generación son los llamados Servicios Diferenciados (ver sección II.3.3), en los que se crea un número pequeño de diferentes tipos de servicio y a cada conexión de tráfico se le clasifica dentro de alguno de esos tipos de servicio. El tratamiento que se da a cada paquete depende del tipo de servicio al que corresponda, pero sin importar a que conexión pertenezca; es decir, la diferenciación es a nivel de agregados.

Los algoritmos que se usen para asignar recursos a los diferentes flujos de tráfico, deben satisfacer requerimientos mínimos de equidad y eficiencia. La propuesta más aceptada para distribuir los recursos, desde el punto de vista genérico aunque existen

variantes, es lo que se conoce como *Encolamiento Justo Ponderado* (WFQ, Weighted Fair Queueing). WFQ garantiza para cada tipo de tráfico, como mínimo, una fracción (denotado por ϕ_i) del ancho de banda, sin importar lo que pase con los otros flujos que compiten por el recurso (aislamiento y equidad); cuando uno de los flujos de tráfico está inactivo, su porción del ancho de banda se reparte entre los que se encuentran activos de manera proporcional a su propia porción (proporcional a los ϕ_i 's).

Además de soportar diferentes clases de servicio, los proveedores de servicio deberán tener la capacidad de repartir el ancho de banda, es decir, de asignar un cierto porcentaje de la capacidad de cada enlace a un grupo determinado de usuarios (posiblemente pertenecientes a una misma compañía) si están dispuestos a pagar el precio. Estos grupos de usuarios podrán incluir múltiples tipos de aplicaciones con diferentes requerimientos de calidad de servicio. Lo anterior implica que los flujos de tráfico tendrán que ser conceptualmente agrupados y atendidos de acuerdo a su filiación administrativa (compañía), tipo de aplicación, protocolo, etc., lo que se conoce como clasificación multicampo. Esto crea la necesidad de manejar estrategias jerárquicas de administración de recursos (ver sección III.5). Una de las metas de este trabajo es mostrar que la propuesta existente para la repartición jerárquica de recursos, que consiste en el uso de múltiples controladores de un solo nivel organizados en una estructura jerárquica, puede dar como resultado pérdidas y retardos innecesarios, incluso injusticia bajo ciertas condiciones. De igual manera, se propone mejorar el desempeño de los sistemas de telecomunicaciones con

asignación jerárquica de recursos, mediante la administración centralizada de estos recursos, en contraste a la propuesta tradicional.

I.3 OBJETIVOS

El objetivo del presente trabajo es el de diseñar un controlador para la administración jerárquica de recursos en un sistema de telecomunicaciones con asignación de ancho de banda, usando el algoritmo WFQ jerárquico y estando en contacto directo con los paquetes que provengan de todos los usuarios y con el enlace físico de transmisión, para evitar tiempos muertos cuando haya paquetes por transmitir.

Para lograr este objetivo se definieron las siguientes metas:

1. Analizar los requerimientos para la administración jerárquica de recursos en un solo paso, sin necesidad de recurrir a controladores simples en cascada.
2. Diseñar el algoritmo propuesto para la administración jerárquica de recursos en un solo paso.
3. Simular el algoritmo propuesto para analizar su desempeño.
 - 3.1. Familiarizarse con el programa de simulación Opnet.
 - 3.2. Crear el modelo para simular la nueva propuesta para la administración jerárquica de recursos
 - 3.3. Ejecutar los programas y recopilar resultados.

4. Comparar el desempeño del algoritmo propuesto con sistemas que utilizan controladores simples en cascada.

I.4 INFRAESTRUCTURA

Para la implementación, simulación y análisis del algoritmo que será presentando en los siguientes capítulos de este trabajo de tesis se contó con la siguiente infraestructura, que en su mayor parte fue proporcionada por libros que se encuentran en el departamento de comunicaciones de la FIME y que a continuación se enlista:

- Computadora Personal AMD Duron a 950Mhz, 256Mb en RAM y 40 GB en disco duro.
- Conexión a Internet.
- Biblioteca de la FIME.
- Biblioteca de la Universidad Aula Magna.
- Biblioteca Capilla Alfonsina.
- Herramienta de simulación OPNET Modeler, versión 7.0B de la compañía OPNET Technologies Inc..
- Programas de aplicación tales como: Matlab V5.0, Excel, Word, Powerpoint, Visio, Paint Brush, Explorer, Lenguaje de programación C, Reflection V9.0, Acrobat Reader V4, WS_FTP.

I.5 ORGANIZACIÓN DE LA TESIS

El contenido de este trabajo de tesis está organizado de la siguiente forma:

En el capítulo II se presenta información acerca de los modelos de servicio, correspondiente al primer modelo que se implementó para la entrega de paquetes, denominado “servicio de mejor esfuerzo”, y las nuevas arquitecturas de asignación de recursos, tales como los “Servicios Integrados” y posteriormente los “Servicios Diferenciados”, que garantizan los recursos y los diferentes niveles de servicio, en donde las dos arquitecturas introducen varios conceptos y primitivas que son importantes para soportar QoS en la Internet. Se hace mayor énfasis en la descripción de la arquitectura de Servicios Diferenciados ya que constituye la base del esquema que se analiza en este trabajo de tesis.

En el capítulo III se da una explicación de las disciplinas de servicio basado en el modelo de fluido, enfocándose principalmente a la disciplina de multicanalización llamado GPS (Generalized Processor Sharing), y al mecanismo PGPS (Packet – by - Packet GPS), comúnmente llamado WFQ (Weight Fair Queueing). El desarrollo del algoritmo correspondiente al calendarizador WFQ propuesto, es presentado en este capítulo, donde se describe el proceso de asignación de un tiempo virtual a cada evento (llegada o salida de paquete), lo que nos permite emular el progreso de GPS que conducirá a una implementación práctica de WFQ. También se presentan algunas posibles modalidades refinadas del algoritmo WFQ. Se hace una descripción en diagrama de bloques de las

arquitecturas jerárquicas WFQ, así como del modelo de dos niveles CH2WFQ, lo cual representa la principal contribución en este trabajo de tesis.

La implementación del algoritmo WFQ y de los modelos jerárquicos WFQ y CH2WFQ son mostradas en el capítulo IV, para lo cual fue utilizada la herramienta de simulación OPNET. Cabe mencionar que OPNET emplea modelos de red, modelos de nodo y modelos de procesos para la implementación y simulación de casi cualquier sistema de telecomunicaciones.

En el capítulo V se analizan, a través de gráficas generadas en base a la información captada, los resultados obtenidos mediante las diferentes pruebas realizadas en los modelos jerárquicos.

En el capítulo VI se presentan las principales conclusiones de este trabajo, así como extensiones que podrían realizarse en trabajos futuros de investigación.

CAPÍTULO II. ARQUITECTURAS DE ASIGNACIÓN DE RECURSOS

II.1 INTRODUCCIÓN

Muchos de los problemas que vemos en la Internet tienen su origen en la *asignación de recursos*. Una gran cantidad de los paquetes enviados son descartados o retardados debido a que los recursos en la red no pueden satisfacer todas las demandas del tráfico. Una red, en su forma más simple, consta de recursos compartidos tales como ancho de banda y colas, atendiendo al tráfico de los usuarios. Una red que soporta QoS necesita desempeñar un papel activo en el proceso de asignación de recursos y decidir quién debe obtener los recursos y en cuánto tiempo.

Las nuevas arquitecturas de asignación de recursos que ofrecen diferentes niveles de servicios con las garantías respectivas, son esenciales para que la Internet pueda evolucionar hacia una red de multiservicios.

Durante la década anterior, la comunidad de la Internet introdujo los Servicios Integrados y posteriormente los Servicios Diferenciados, que son dos nuevas arquitecturas para la asignación de recursos en la Internet. Las dos arquitecturas introdujeron varios conceptos que son importantes para soportar QoS en la Internet:

- Sistemas o estructuras para la asignación de recursos que garanticen la obtención de éstos recursos y la diferenciación de servicios.

- Nuevos modelos de servicio para la Internet además del ya existente servicio de Mejor Esfuerzo.
- Mecanismos para hacer valer la asignación de recursos.

II.2 CALIDAD DE SERVICIO

La Calidad de Servicio (QoS, Quality of Service) [Wang Z., 2001] consiste en ofrecer a las aplicaciones los recursos de red suficientes para cumplir con los requerimientos establecidos por el cliente. Todas las aplicaciones requieren de un ancho de banda determinado, y si éste no es suficiente, la red se verá en la necesidad de retardar o descartar paquetes, degradando el desempeño de las aplicaciones.

Algunos de los parámetros que afectan el rendimiento de las aplicaciones son:

- **Retardo:** Tiempo que transcurre desde que se genera el paquete en el emisor, hasta que es procesado por el nodo receptor.
- **Variación de retardo (Jitter):** Es la variación de retardo en la red, que es creado donde quiera que haya colas de espera; es una característica inevitable de los enrutadores y otros equipos de comunicación, ya que el número de paquetes que esperan para ser transmitidos es variable.
- **Ancho de banda:** Este término se usa para denotar la tasa o velocidad de transmisión de los enlaces que conectan los diferentes equipos de telecomunicaciones. Siendo estrictos, “ancho de banda” realmente significa una

porción del espectro de frecuencias, pero se usa con el significado anterior debido a que hay una relación de proporcionalidad entre ambos.

- **Tasa de pérdida de paquetes:** Mide la proporción de paquetes que son descartados. El descarte de paquetes se debe principalmente a las condiciones de congestión en los nodos.

Es importante mencionar que para que en una red se obtengan beneficios al aplicar mecanismos de QoS, es necesario que algunos flujos de tráfico sean tratados con baja prioridad y otros con alta, ya que si a todos los flujos se les brindara una prioridad alta, el beneficio resultante sería nulo.

Los sistemas de telecomunicaciones que pretendan ofrecer QoS, deben ser capaces de distinguir entre los diferentes tipos de tráfico (videoconferencia, voz sobre IP, web, correo electrónico, telnet, ftp, etc.) (ver figura 1), así como el usuario al que pertenecen dentro de los equipos de comunicación, para proporcionarle un tratamiento diferente a cada uno de ellos.

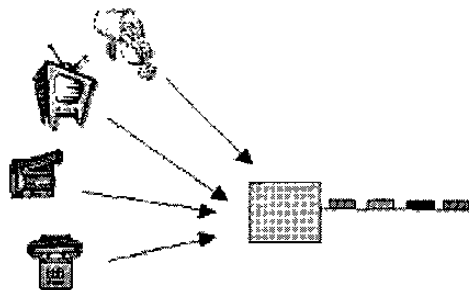


Figura 1. Diferentes tipos de tráfico

II.3 MODELOS DE SERVICIOS

Desde que se desarrolló la Internet, el objetivo primordial era la confiabilidad de la entrega de paquetes. El primer modelo que se implementó para la entrega de paquetes se denominó “servicio de mejor esfuerzo” [Bakiras S. y Li V., 2001]. Posteriormente, con la integración de las nuevas aplicaciones multimedia se desarrollaron otros modelos para permitir que estas nuevas aplicaciones utilicen las redes de IP. Éstos modelos son los Servicios Integrados y los Servicios Diferenciados. A continuación se describirán brevemente estos modelos de servicio.

II.3.1 MEJOR ESFUERZO

En este modelo la red hace su mejor esfuerzo para entregar los paquetes, sin considerar el tiempo que tardan los paquetes en llegar al nodo destino y el ancho de banda utilizado por el flujo. Los enrutadores o conmutadores pueden descartar paquetes en caso de que exista una congestión severa en la red sin notificar al nodo emisor del paquete. En los casos en que la aplicación utilice TCP, se provee de un mecanismo de retransmisión de paquetes, mientras que las aplicaciones basadas en UDP no cuentan con un mecanismo similar.

El uso de este esquema de servicio no ofrece garantías, ya que existe únicamente un nivel de servicio, lo que no permite garantizar un ancho de banda mínimo para cada flujo y, por consiguiente, tampoco garantiza límites en los parámetros de retardo, jitter y pérdida de paquetes. Cuando la aplicación está basada en TCP, los paquetes que son descartados son

retransmitidos, lo que causa una disminución del caudal eficaz y un aumento en el retardo para la entrega de paquetes. Las aplicaciones basadas en UDP no retransmiten los paquetes, por lo que una situación de congestión en la red afectará considerablemente la calidad de la aplicación.

Este esquema de servicio funciona muy bien cuando el tipo de tráfico que se transporta en la red no es sensible a los retardos y no requiere de un ancho de banda mínimo. Algunos de los tipos de tráfico que utilizan de manera satisfactoria este servicio son el correo electrónico, la transferencia de archivos (FTP), el acceso remoto (telnet), el tráfico http, entre otros.

II.3.2 SERVICIOS INTEGRADOS (INTSERV)

II.3.2.1 INTRODUCCIÓN

El modelo de Servicios Integrados (IntServ, Integrated Services) [Wang Z., 2001] fue el primer intento por darle a la Internet la capacidad de ofrecer QoS. La investigación y los esfuerzos desarrollados proporcionaron instrumentos valiosos en el complejo asunto de soportar QoS en la Internet. La arquitectura de asignación de recursos, los nuevos modelos de servicio, y el protocolo RSVP fueron estandarizados a finales de 1990.

IntServ requiere que las aplicaciones informen a la red de sus requerimientos de servicio a través de peticiones de reservación. Para ello emplea el Protocolo de Reservación de Recursos (RSVP, Resource Reservation Protocol) como protocolo de señalización de

extremo a extremo. Usando este protocolo de reservación, los paquetes que pasan a través de un enrutador pueden ser atendidos basándose en políticas y criterios de reservación previamente establecidos.

II.3.2.2 ELEMENTOS DE LA ARQUITECTURA

Los Servicios Integrados desarrollaron una nueva arquitectura de asignación de recursos para satisfacer los requerimientos de las aplicaciones en tiempo real basado en la *reservación de recursos por flujo*.

Hay todo un conjunto de mecanismos y protocolos que son usados para hacer explícita la reservación de recursos en la Internet, los cuales complementarán, más no reemplazarán, el servicio básico de IP. Se ha detectado la necesidad inminente de utilizar enrutadores con la habilidad de reservar recursos para proporcionar QoS para los paquetes o flujos de los usuarios.

En la Internet original, el servicio de IP es completamente equitativo; todos los paquetes reciben la misma calidad de servicio, y los paquetes se transmiten usando una disciplina de cola FIFO. Para los servicios integrados, un enrutador debe implementar una calidad de servicio apropiada para cada flujo.

II.3.2.3 CLASES DE CALIDAD DE SERVICIO

Las ideas, conceptos y mecanismos desarrollados en los Servicios Integrados también están dirigidos a la QoS. La IETF ha considerado tres clases de QoS, pero hasta el momento sólo se han especificado para usarse con RSVP el Servicio Garantizado y el Servicio de Carga Controlada.

Mejor esfuerzo: Dentro de esta clase de servicio se hallan todos aquellos flujos que no requieren realizar una reservación de recursos. Aunque la función primordial de IntServ es garantizar recursos a los flujos que lo soliciten, los flujos que utilizan la clase de *mejor esfuerzo* también deben obtener un servicio adecuado por parte de la red. A esta clase también se le debe garantizar al menos una porción del ancho de banda del enlace.

Servicio Garantizado: Este servicio proporciona un nivel garantizado de ancho de banda, con un retardo de extremo a extremo corto y sin pérdidas de paquetes. Está hecho para aplicaciones con estrictos requerimientos de entrega en tiempo real, tales como algunas aplicaciones de audio y video.

Servicio de Carga Controlada: El servicio de carga controlada no proporciona garantías cuantitativas. Se deben notificar las características del tráfico para los flujos en los que se requiere tener un servicio de carga controlada en el enrutador, aunque no es necesario incluir el parámetro de velocidad pico de transmisión. Si el flujo es aceptado para el servicio de carga controlada, el enrutador hace un compromiso para ofrecerle al

flujo un servicio equivalente al que daría bajo la técnica del mejor esfuerzo cuando la red tiene poca carga. Una diferencia importante es que el flujo de carga controlada no se deteriora si la carga en la red se incrementa con el tráfico de la clase de mejor esfuerzo.

El servicio de carga controlada se emplea para las aplicaciones que pueden tolerar una cierta cantidad de pérdidas y en las que el retardo se mantiene en un nivel razonable. Esta clase de servicio incluye algunas aplicaciones en tiempo real.

Los enrutadores que implementan el servicio de carga controlada deben revisar que los flujos de datos de carga controlada se adecúen a sus características de tráfico reservadas. Cualquier flujo que no se adecue, no debe afectar la QoS ofrecida a otros flujos con reservación ni el manejo del tráfico bajo la técnica del mejor esfuerzo.

Con estas limitaciones, el enrutador debe intentar transmitir tantos paquetes como pueda de los generados en exceso. Esto se puede realizar dividiendo los paquetes en dos grupos: aquellos que cumplen con las especificaciones de la reservación hecha y aquellos que no cumplen, transmitiendo los paquetes excesivos bajo la técnica del mejor esfuerzo. El enrutador también puede elegir degradar el nivel de QoS para todos los paquetes excesivos.

II.3.2.4 PROTOCOLO DE RESERVACIÓN DE RECURSOS (RSVP)

El protocolo de reservación de recursos (RSVP) [Braden R. *et al*, 1997] fue también estandarizado para informar a la red de los requerimientos de las aplicaciones y para la

reservación de recursos a lo largo del camino que seguirán los paquetes. RSVP hace una reservación en sólo una dirección. Cuando una aplicación desee actuar como un transmisor y como un receptor, deberá establecer una reservación en cada dirección.

Un aspecto interesante es que el establecimiento de la reservación lo hace el punto receptor y no el que envía, siendo sus ventajas las siguientes:

- Se simplifican las sesiones multicast pues pueden aparecer y desaparecer receptores sin que la aplicación transmisora se entere de esto.
- El receptor es quien padece de los efectos de tener que reservar recursos y diferentes receptores pueden estar conectados por diferentes medios.
- Finalmente, es muy probable que sea el receptor quien pague por los recursos.

RSVP identifica una sesión de comunicación por la combinación de la dirección destino, el tipo de protocolo de la capa de transporte, y el número de puerto destino. Es importante observar que cada operación RSVP sólo se aplica a los paquetes de una sesión en particular; por lo tanto, cada mensaje RSVP debe incluir los detalles de la sesión a la que se aplica.

Una de las suposiciones en RSVP es que no se debe mermar la robustez presente en las redes orientadas a no conexión (como TCP/IP). Puesto que tales redes se basan en poca o ninguna información almacenada en la red misma, es posible que los enrutadores y enlaces se “caigan” y se vuelvan a “levantar” sin que se pierda la conectividad de extremo a

extremo. RSVP trata de mantener esta robustez usando la idea de estado suave en los enrutadores. El estado de reservación en la red está asociado con un tiempo. Cuando expira este tiempo, el estado es borrado automáticamente. De esta manera RSVP refresca periódicamente el estado de reservación para mantener el estado a lo largo de la trayectoria de los paquetes.

La información que se le pasa a la red en RSVP se conoce como Flowspec (especificación de flujo). El flowspec se compone de dos partes: la parte que describe las características del tráfico (llamada Tspec) y la que describe el servicio solicitado a la red (llamada Rspec).

II.3.2.5 MENSAJES RSVP

Para entender mejor la mecánica de cómo hacer una reservación, se detallan algunos aspectos a continuación:

Los mensajes primarios usados por RSVP son el mensaje *Path* que tiene su origen en el emisor, y el mensaje *Resv*, que tiene su origen en el receptor:

- Mensaje Path: Sus objetivos son:
 - Establecer un estado de enrutamiento en cada nodo a través de la ruta, el cual es usado para recorrer los mensajes RESV en dirección inversa.

- Proporcionar a los receptores información sobre las características del tráfico que se desee enviar y de la ruta para que se puedan hacer las peticiones de reservación adecuadas.
- Mensaje Resv: Realiza las peticiones de reservación a los enrutadores a lo largo de la red de distribución entre receptores y emisores.

II.3.2.6 MECANISMO DE FUNCIONAMIENTO INTSERV

En la figura 2 se muestra la forma de operación del protocolo IntServ, el cual consiste en que cuando el emisor desea establecer una comunicación con otra terminal reservando recursos, envía al receptor un mensaje Path. Si éste llega correctamente al destino y la reservación es posible, se envía un mensaje Resv de vuelta al emisor. Este mensaje reservará de forma definitiva los recursos en los nodos por los que vaya pasando. Cuando el paquete llega a un nodo intermedio, éste puede aceptar o rechazar la solicitud. Esta decisión depende de si existen los recursos disponibles en el nodo para satisfacer la demanda de flujo. Si la solicitud es aceptada, el paquete RESV se pasa al siguiente nodo en la trayectoria. En caso de que el nodo rechace la petición de reservación, se enviará un mensaje de error a la terminal que solicitó la reservación y el proceso de señalización termina.

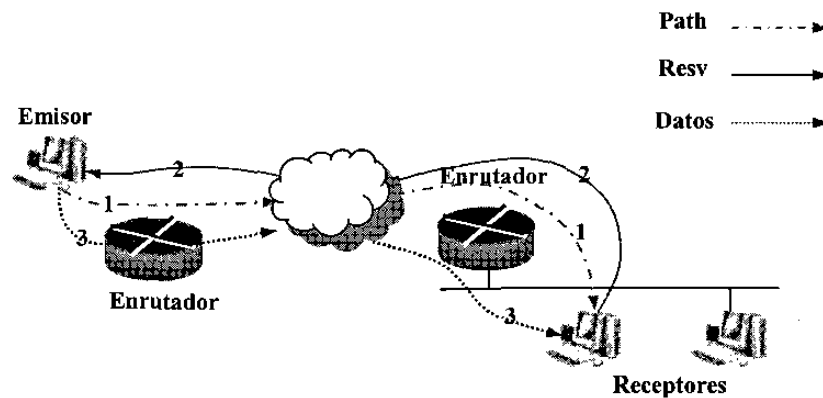


Figura 2. Operación básica RSVP

Una de las desventajas que presenta el protocolo IntServ, es que cada nodo debe mantener la información de cada uno de los flujos que lo atraviesan. Una red que cuente con demasiados flujos, tendría muchos problemas para poder mantener la información de cada uno de los flujos. Es por esta razón que este modelo de servicio no se utiliza en redes de gran cobertura. Por esta razón se dice que IntServ no es “escalable”.

II.3.3 SERVICIOS DIFERENCIADOS (DIFFSERV)

II.3.3.1 INTRODUCCIÓN

Las tecnologías de los enrutadores y conmutadores están avanzando de una manera tan impresionante que una opción eficiente y escalable es la de proporcionar diferenciación de servicio a través de un pequeño número de clases.

La IETF creó el grupo de servicios diferenciados (DiffServ, Differentiated Services) con el objeto proporcionar QoS en la Internet de manera que sea escalable [Blake S. *et al*,

1998]. A mediados de 1997 los proveedores de servicio se dieron cuenta de que los Servicios Integrados no estaban preparados para trabajar a gran escala, por lo cual sintieron la necesidad de mejorar el modelo de servicio.

El modelo DiffServ se basa en tres principios básicos:

Clasificación: Marca los paquetes en los nodos frontera de la red dependiendo de la clase de servicio que se requiera, la cual se encarga de dividir el tráfico en pequeños grupos. La clase de servicio al que pertenece el paquete es codificado en un campo en la cabecera del paquete IP. Cada clase de servicio representa un tratamiento de servicio predefinido en términos de prioridad de descarte y asignación de ancho de banda.

Diferenciación: Determina cómo deben ser tratados los paquetes marcados dentro de la red, obteniendo así un servicio diferenciado.

Conformador: La QoS controla la congestión de carga de la red y lo realiza limitando el número de paquetes que entran a la red a través de los nodos frontera de acuerdo a los requerimientos de la clase de servicio y suavizando el tráfico a través de la introducción de intervalos de tiempo en medio de la transmisión de paquetes consecutivos. El tráfico excesivo puede ser desechado, retardado o marcado con una diferente clase de servicio.

II.3.3.2 DOMINIO DIFFSERV

El Dominio de los Servicios Diferenciados (DS), es un conjunto de nodos DS que opera mediante un servicio común con la finalidad de otorgar políticas de encolamiento y un conjunto de grupos PHB (ver sección II.3.3.3) implementado en cada nodo. Un dominio DS consiste de nodos de frontera y nodos internos, los cuales tienen diferentes funciones, tal como se muestra en la figura 3. Los nodos de frontera interconectan el dominio DS con otros dominios que no poseen el comportamiento DS y los nodos internos son los que se encuentran en el interior del dominio DS, los cuales se encargan de la interconexión con otros nodos de núcleo o con los nodos de frontera, pertenecientes al mismo dominio.

Este modelo utiliza el campo de tipo de servicio (TOS, Type of Service) del encabezado IP para indicar el nivel de servicio que se le debe ofrecer al paquete. El código que se utiliza para definir la clase de servicio que el paquete debe recibir se denomina Código de Servicios Diferenciados (DSCP, Differentiated Services Code Point). Este código utiliza los primeros seis bits del campo TOS y cada valor del código representa una clase de servicio diferente.

Cuando un paquete entra a la red, los nodos internos realizarán un tratamiento diferenciado, el cual consiste en clasificar y marcar el paquete con el DSCP correspondiente al nivel de QoS que se le debe aplicar, pero sin importar a qué conexión pertenezca; es decir, la diferenciación de tráfico es a nivel de agregados. En el otro extremo

del dominio DS, los nodos frontera pueden enlazarse con otro segmento de red, además de desarrollar las funciones de condicionamiento de tráfico (ver sección II.3.3.4) dentro del mismo dominio DS. Los nodos frontera pueden actuar como nodos de entrada o nodos de salida.

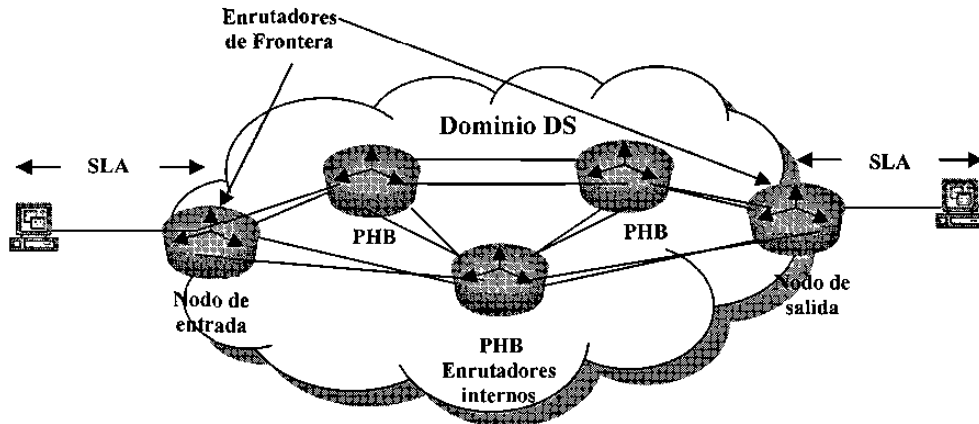


Figura 3. Nodos internos y de frontera en un dominio DS

A diferencia del modelo IntServ, en el que el proceso de reservación de recursos requiere que los nodos internos participen activamente y mantengan información del estado de cada uno de los flujos que tienen asignados recursos, DiffServ delimita el proceso de clasificación a los nodos de acceso a la red.

II.3.3.3 COMPORTAMIENTO POR SALTO (PHB)

DiffServ alcanza las propiedades de escalamiento definiendo un mecanismo para el descarte, envío y aceptación de paquetes dentro de una red, denominado comportamiento por salto (PHB, Per Hop Behaviour). Es una descripción de la transmisión de paquetes, los cuales se agrupan dentro de un tráfico agregado para ser tratados de la misma manera sobre

un determinado enlace dentro del dominio DS. Los PHB's pueden ser especificados en términos de sus recursos (por ejemplo, ancho de banda, memoria), prioridad relativa a otros PHB's, o en términos de sus características de tráfico (por ejemplo, pérdida de paquetes, retardo). Los PHB's pueden ser implementados en los nodos mediante algún manejo de memoria y mecanismos de calendarización de paquetes.

PHB describe las clases de servicio en términos relativos o absolutos.

1. **Envío Expedito** (EF, Expedited Forwarding): Es diseñado para soportar conexiones con reservación de ancho de banda, bajas pérdidas, niveles bajos de retardo y de variaciones en el retardo (jitter). Se usa en aplicaciones de tiempo real, tales como videoconferencia, y voz sobre IP, entre otros. Este servicio puede ser implementado por medio de un calendarizador con prioridades (PQ), de pesos (WFQ) o basado en clases (CBQ). EF maneja el tipo de *servicio Premium*, el cual no soporta datos en ráfagas, por lo que es conveniente acondicionar el tráfico por medio de un mecanismo llamado cubeta agujerada (leaky bucket) (ver sección II.3.3.5.4.2). Para mantener la escalabilidad, sólo una pequeña porción de ancho de banda podrá ser reservada para dar este servicio.
2. **Envío garantizado** (AF, Assured Forwarding): Esta clase de servicio permite el manejo de los recursos necesarios para el envío de paquetes en los nodos del dominio DS, tales como el tamaño de la memoria y ancho de banda. Los niveles AF definen cuatro clases de servicio que dependen de la asignación de recursos, la carga real de tráfico y los niveles de congestión, donde las clases se envían

independientemente y cada nodo se encarga de asignarle los recursos necesarios. En este servicio hay garantías para los flujos de tráfico agregado en cada nodo, pero no para conexiones individuales. Hay tres prioridades de descarte para los paquetes, dependiendo de si llegan dentro del contrato, ligeramente fuera, o representan un tráfico notoriamente excesivo. Estas prioridades de descarte se describen mediante “colores” asignados a los paquetes (verde, amarillo o rojo). Dentro de un dominio DS, cualquier nodo debe aceptar las tres prioridades de descarte y debe implementar al menos dos de ellas. El tipo de servicio que soporta es el *servicio garantizado*, el cual es adecuado para aplicaciones que no son efectuadas en tiempo real, tales como el acceso remoto (TELNET), transferencia de archivos (FTP), buscadores de web, entre otros.

II.3.3.4 CLASIFICACIÓN Y ACONDICIONAMIENTO DE TRÁFICO

En DiffServ, los nodos de frontera son responsables de etiquetar los paquetes para cada una de las clases de servicio en la red y deben asegurar que el tráfico se ajuste al SLA del cliente específico. Una vez que los paquetes pasan del nodo de frontera al nodo interior de la red, la asignación de recursos se realiza, basado únicamente en las clases de servicio. Estas funciones que los nodos de frontera realizan son referidos como *clasificación de tráfico* y *ajuste de tráfico*, tal como se muestra en la figura 4. El módulo de *clasificación* contiene un clasificador y un marcador de clase. El clasificador selecciona los paquetes basado en reglas predefinidas. Estas reglas de clasificación pueden especificar que

un paquete seleccionado sea marcado con un DSCP particular. La salida del módulo de clasificación es conectado al módulo de *acondicionamiento* de tráfico.

Los nodos de frontera traducen el convenio de acondicionamiento de tráfico (TCA, Traffic Conditioning Agreement) en un *perfil de tráfico* para cada cliente que se encuentra conectado. El perfil de tráfico especifica las propiedades temporales del flujo de tráfico de un usuario en términos de los parámetros de acondicionamiento de tráfico y proporciona un conjunto de reglas para determinar si un paquete está considerado dentro del perfil especificado o no. Cuando el flujo de tráfico está dentro del perfil, los paquetes serán admitidos para entrar a la red. El módulo de acondicionamiento de tráfico mide los flujos de tráfico del módulo de clasificación y lo compara con los perfiles de tráfico del usuario. Consta de cuatro elementos básicos: medidor, marcador/ remarcador de perfil, suavizador de tráfico y descartador.

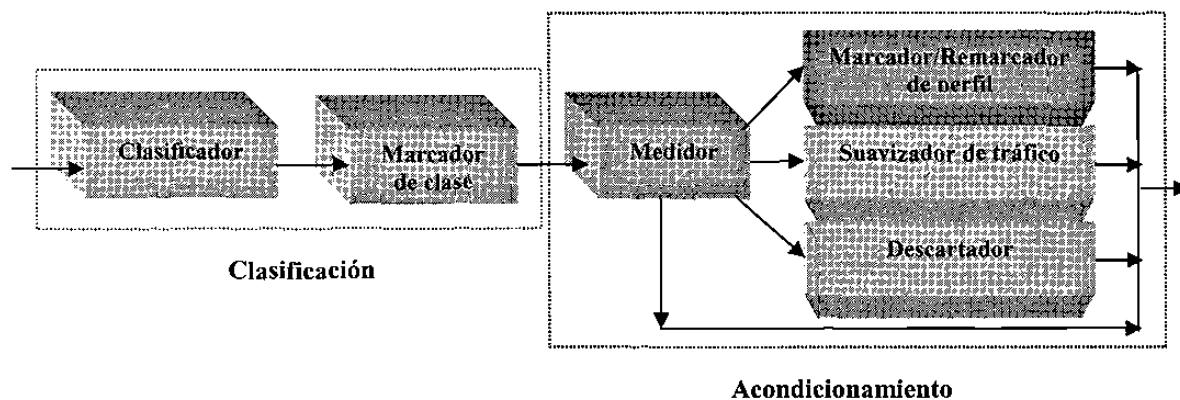


Figura 4. Clasificación y Acondicionamiento de tráfico

A continuación se describen los elementos de los módulos de clasificación de tráfico y de acondicionamiento de tráfico.

Elementos del módulo de Clasificación de tráfico:

Clasificador.- Selecciona y conduce a los paquetes en un flujo de tráfico basados en el contenido del encabezado del paquete. Existen dos tipos de clasificadores:

1. **Comportamiento agregado** (BA, Behavior Aggregate). Este clasificador selecciona los paquetes basado únicamente en los valores del DSCP en el encabezado del paquete.
2. **Multicampo** (MF, MultiField). Este tipo de clasificador selecciona los paquetes basado en la combinación de uno o más campos (dirección fuente, dirección destino, puertos fuente /destino, y el identificador de protocolo) en el encabezado del paquete para la clasificación. Generalmente es usado para soportar políticas de asignación de recursos más complicadas. Un ejemplo sería la diferenciación necesaria para realizar una repartición jerárquica de recursos de la red, que es precisamente el objeto de estudio de esta tesis.

Marcador de clase.- Estos marcadores escriben en el campo DS de un paquete un DSCP particular, añadiendo el paquete marcado a la clase de servicio. El marcador puede ser configurado para marcar todos los paquetes usando un solo DSCP o puede ser configurado para marcar un paquete usando un conjunto de códigos DS para seleccionar un PHB dentro de un grupo de PHB's.

Elementos del módulo de Acondicionamiento de tráfico:

Medidor.- Para cada clase de servicio este elemento mide las propiedades temporales del flujo de paquetes seleccionados por el clasificador y los compara con el perfil de tráfico especificado en el TCA. Los perfiles de tráfico están descritos en términos de los parámetros que realizan la medición (token bucket) (ver sección II. 3.3.5.2)

Marcador / remarcador de perfil.- Este elemento marca (o vuelve a marcar) los paquetes para indicar si se encuentran o no dentro del perfil esperado. La remarcación se daría cuando un paquete pasa de un dominio DS a otro.

Suavizador de tráfico.- Este elemento retarda los paquetes que llegan en exceso para hacer que el flujo cumpla con el perfil especificado. La diferencia entre un suavizador y un marcador es que el marcador simplemente etiqueta el paquete y lo deja en la red, mientras que un suavizador evita que el paquete entre a la red hasta que el flujo de tráfico cumpla con el perfil especificado.

Descartador.- Además de suavizar el tráfico, en ocasiones es necesario descartar los paquetes que llegan fuera del perfil. La función de este bloque es adicional al posible descarte de paquetes por el suavizador cuando no haya espacio en sus colas.

II.3.3.5 MECANISMOS DIFFSERV

En esta sección se dará una explicación más extensa de los elementos de clasificación y acondicionamiento de tráfico para la implementación de enrutadores que soportan DiffServ, los cuales requieren de varios mecanismos que serán descritos a continuación.

II.3.3.5.1 CLASIFICADORES Y MARCADORES DE CLASE

Con el modelo de servicios sin garantías, se necesita una sola cola por cada uno de los enlaces que tenga conectado el enrutador. Cuando se tiene más de una cola, tal como se muestra en la figura 5; es necesario clasificar los paquetes para determinar a que cola pertenecen. Esta clasificación se basa en la información de los encabezados del paquete.

El clasificador debe ser configurado con un conjunto de reglas y acciones que se manifiestan a la hora de procesar una lista de campos en la cabecera del paquete (dirección fuente/ destino, puerto fuente /destino, identificador de protocolo y el campo DS) e información adicional tal como la interfaz lógica de llegada. Las acciones incluyen el marcado del paquete con un DSCP particular, lo que implicará también la selección de un acondicionador de tráfico para el paquete, y la asignación de ancho de banda para la clase de servicio. De esta manera se sabe a qué cliente pertenece y cuáles son las características establecidas en el SLA.

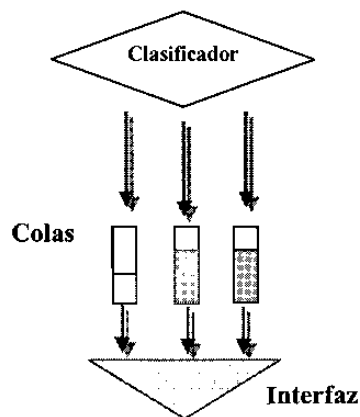


Figura 5. Clasificador de Paquetes

II.3.3.5.2 MEDIDOR Y MARCADOR DE PERFIL

Las funciones que tiene el acondicionador de tráfico en DiffServ es medir el tráfico para compararlo con el perfil especificado y después marcar los paquetes correspondientes.

Estas funciones son implementadas con el algoritmo de la cubeta de fichas (TB, token bucket), el cual responde a las señales de la red que indica la capacidad disponible en cualquier tiempo dado. Cuando un paquete llega a un TB, se examina el número de fichas (representados en unidades de bytes) en la cubeta. Si no hay suficientes fichas, para que el paquete pase, el paquete será considerado fuera del perfil. Por lo tanto, un TB puede ser usado para determinar si un flujo de tráfico está dentro o fuera del perfil de la cubeta de fichas. Sin embargo, para soportar DiffServ, se necesita dividir el tráfico en más de dos grupos, siendo necesario usar más de un TB.

En la figura 6 se puede ver un modelo de red con TB. El tráfico de entrada está definido por la función $E(t)$. El tráfico es introducido a la red y dependiendo de la carga de los nodos llegará al receptor con un comportamiento definido por la función $R(t)$. Si no hay carga en la red, esto implicará que el tráfico no se almacena en los nodos con lo que el comportamiento del tráfico en el receptor es equivalente al del emisor. Si hay mucha carga en los nodos, entonces los paquetes se almacenarán en la red, lo que implica un retraso y un cambio en general en el comportamiento.

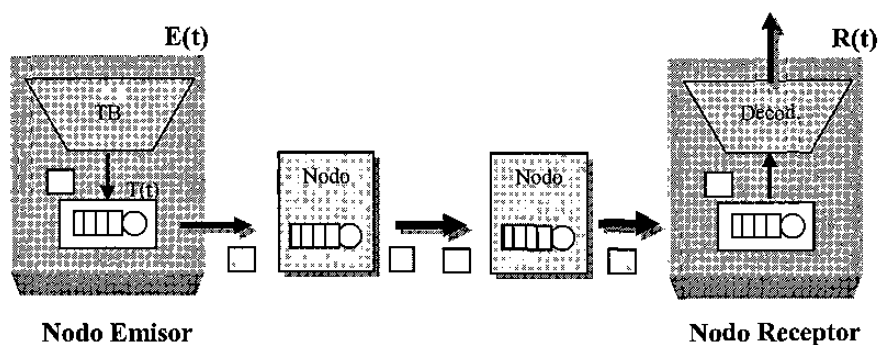


Figura 6. Elementos de la red usando el modelo de la cubeta con fichas

Las propuestas más aceptadas para el servicio de envío garantizado (AF) plantean dividir el tráfico en tres prioridades de descarte. El trTCM por ejemplo [Wang Z., 2001] mide el flujo de tráfico y marca sus paquetes con alguno de los tres colores: verde, amarillo o rojo. El trTCM usa dos TB en cascada y, por consiguiente, tiene cuatro parámetros configurables: una tasa pico con su respectivo tamaño de ráfaga y una tasa asegurada con su respectivo tamaño de ráfaga asignada. El paquete es marcado de color rojo si excede la

tasa pico. De otra forma es marcado con otro color, ya sea amarillo o verde, dependiendo de si éste excede la tasa asegurada o no.

II.3.3.5.3 DESCARTADOR

La intención de descartar paquetes, aún cuando hubiera todavía espacio en las colas para almacenarlos, es la de conseguir que las fuentes de tráfico se enteren anticipadamente de que hay posibilidades de entrar a una situación severa de congestión. Algunas de las propuestas más aceptadas son RED (Random Early Detection) [Floyd Sally y Jacobson Van, 1993], DRED (Dinamic -Random Early Detection) [Aweya J. *et al*, 2002], BLUE [Feng Wu – Chang *et al*, 2001], etc.

II.3.3.5.3.1 CALENDARIZADOR DE PAQUETES

Los mecanismos que se encargan de decidir qué paquete se transmite primero son los calendarizadores de paquetes. El calendarizador de paquetes es responsable de implementar la asignación de recursos para flujos individuales. Cuando los recursos de la red no pueden acomodar todos los flujos de tráfico, las colas empezarán a llenarse en los enrutadores. El propósito del calendarizador es decidir cuáles son los paquetes que deben obtener los recursos. Un calendarizador de paquetes trabaja como un repartidor, ya que conserva la información de cuántos paquetes de cada flujo se han enviado y lo compara con la cantidad de recursos que el flujo tiene reservado. El orden en que los paquetes van siendo extraídos de las colas, depende del algoritmo del calendarizador y de los SLA presentes en la red, tal como se muestra en la figura 7.

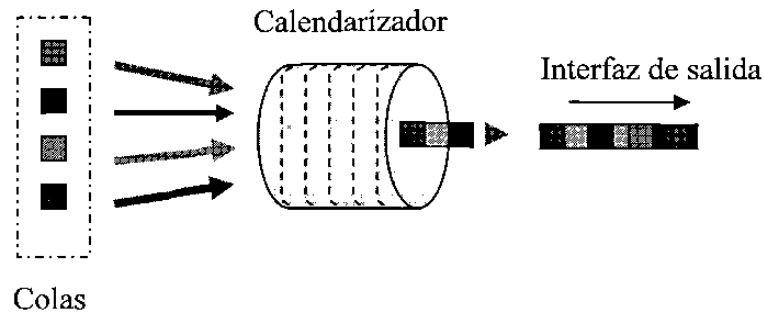


Figura 7. Calendarizador de paquetes

Los calendarizadores asignan tres tipos de recursos: ancho de banda (qué paquete es transmitido), tiempo (cuándo es transmitido el paquete) y memoria (qué paquetes son guardados y cuáles descartados), que se repartirán de acuerdo con los siguientes parámetros de la aplicación involucrada: velocidad de transmisión del flujo, retardo y tasa de pérdida de paquetes.

Algunas de las opciones fundamentales para calendarizar los paquetes son los calendarizadores *conservativos* y los calendarizadores *no conservativos*.

Un calendarizador conservativo es aquel que nunca está ocioso mientras haya paquetes esperando servicio. A este grupo pertenecen el mecanismo general para la repartición de un procesador (GPS, General Processor Sharing) [Parekh A. y Gallager R., 1993], el Reloj Virtual (VC, Virtual Clock) [Zhang H., 1995], Encolamiento equitativo (FQ, Fair Queueing) [J. C. R. Bennett y H. Zhang., 1997] y su versión de pesos (WFQ, Weighted Fair Queueing) [Parekh A. y Gallager R., 1993] que no es otra cosa que una aproximación a GPS, por lo que también se denota con las siglas PGPS. Otra disciplina de servicio que

gestiona la repartición del enlace de una forma controlada, permitiendo una estructura jerárquica, es el calendarizador basado en clases (CBQ, Class-Based Queueing) [Floyd Sally, y Jacobson Van., 1995,]. Sin embargo, hay algunos algoritmos *no conservativos* que han sido propuestos con el fin de reducir las variaciones de retardo de los paquetes y los requerimientos de búferes en los elementos de la red. Un ejemplo es la técnica conocida como “WFQ justo aún en el peor caso” (WF2Q, Worst – Case Fair WFQ) [Bennet J. C. R., Zhang H., 1996].

II.3.3.5.3.2 PRIORIDAD SIMPLE

En un sistema de prioridad simple, existe un número de niveles de prioridad, y la prioridad más alta siempre tiene precedencia sobre los de más baja prioridad. Cuando el enlace de salida llega a estar disponible para transmitir, el calendarizador siempre seleccionará los paquetes de la cola con la más alta prioridad. Los paquetes de más baja prioridad serán transmitidos sólo cuando no haya paquetes en las colas de más alta prioridad. Sin embargo esta prioridad debe tratarse con cuidado, porque pudiera ser que un gran volumen de paquetes de más alta prioridad no dejen transmitir paquetes de más baja prioridad si estos están siempre en la cola. Por lo tanto, la prioridad simple está reservada para tráfico crítico o es usado con un control de admisión estricto para asegurar que la cantidad de tráfico de más alta prioridad está limitada a una pequeña porción del total de tráfico. Este es el caso del tráfico EF, que se transmite con prioridad más alta que el tráfico AF.

A continuación se examinarán algunos métodos de encolamiento y disciplinas de servicio conservativas, los cuales se consideran importantes en la elaboración de esta tesis: el calendarizador “el primero que entra es el primero que sale” (FIFO, First In First Out), colas con prioridades (PQ, Priority Queueing), encolamiento justo (FQ, Fair Queueing) y su versión de pesos (WFQ, Weighted Fair Queueing).

II.3.3.5.3.3 “EL PRIMERO QUE ENTRA, ES EL PRIMERO QUE SALE” (FIFO)

Este es el modelo más común de encolamiento de tráfico de llegada. Almacena y envía el tráfico de una interfaz de llegada a una interfaz de salida. Cuando un paquete llega a un puerto de salida y ya está ocupado por otros paquetes, entonces los nuevos paquetes que lleguen son colocados en una cola de espera. Existe sólo una cola para cada puerto de salida, y el orden en que los paquetes son almacenados en la cola se hace conforme éstos vayan llegando. Este mecanismo trabaja bien cuando los paquetes son de la misma prioridad o cuando el servicio no tiene requerimientos estrictos de QoS. Debido a esta situación FIFO introduce un jitter considerable en el tráfico de datos. Esto no causa mayor problema en la transferencia de archivos y otras transacciones, pero sí causa problemas en las aplicaciones en tiempo real como videoconferencias o VoIP. Por ejemplo, un paquete muy grande puede retardar una docena de paquetes cortos, los cuales no requieren mucho ancho de banda, pudiendo de alguna manera ser movidos a la cabeza del paquete grande.

En la figura 8 se muestra la forma en que los paquetes llegan por la interfaz de entrada, los cuales son ordenados conforme vayan siendo recibidos en la interfaz de salida. Esto explica el nombre de este modelo: “el primero que entra, es el primero que sale”.

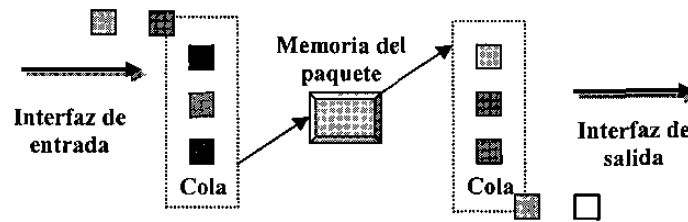


Figura 8. Modelo de Encolamiento FIFO

II.3.3.5.3.4 COLAS CON PRIORIDADES (PQ)

En este modelo, cada puerto de salida tiene un cierto número de colas tipo FIFO, una para cada nivel de prioridad (alto, medio, normal y bajo). Conforme los paquetes vayan llegando para ser transmitidos, el puerto de salida verifica la prioridad de los paquetes y los coloca en la cola apropiada. Cuando el puerto de salida envía un nuevo paquete, escoge el paquete que está disponible en la cola con prioridad más alta.

En la figura 9 se muestra cómo son recibidos los paquetes en la interfaz de entrada, los cuales son reordenados de acuerdo al criterio del usuario en la cola de salida. En este ejemplo, los paquetes de más alta prioridad son colocados en la cola de salida antes que los paquetes normales, los cuales esperan en la cola de salida hasta que no haya más paquetes de alta prioridad que esperen a ser transmitidos.

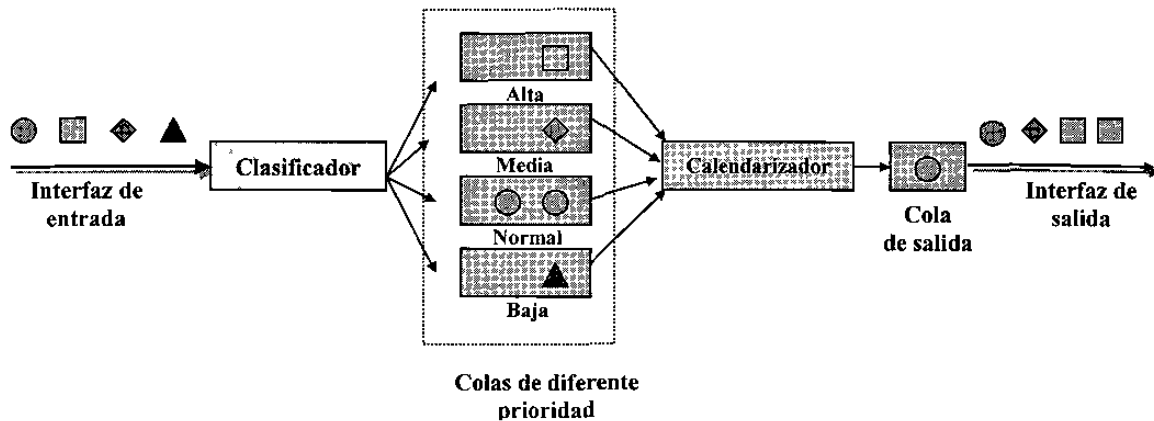


Figura 9. Modelo de encolamiento PQ

II.3.3.5.3.5 ENCOLAMIENTO JUSTO (FQ)

El modelo de encolamiento justo trata de resolver el problema de la incapacidad que tiene FIFO para diferenciar entre flujos para asegurar un acceso equitativo de servicio. Introduce un estado en los enrutadores, separando el tráfico de llegada en “flujos” bien identificados y garantizando a cada flujo una porción equitativa de la capacidad de transmisión.

El dispositivo de red da seguimiento a los flujos individuales, examinando las direcciones de los paquetes, los número del puerto y los tipos de protocolo. El puerto de salida toma a los paquetes de cada flujo de acuerdo con las garantías ofrecidas y asegurando que ningún flujo monopolice el enlace de salida.

Para poder hacer la diferenciación, FQ necesita colas separadas para los diferentes flujos de paquetes.

En la figura 10 se muestra el modelo de un encolamiento tradicional y en la figura 11 el modelo de encolamiento justo.

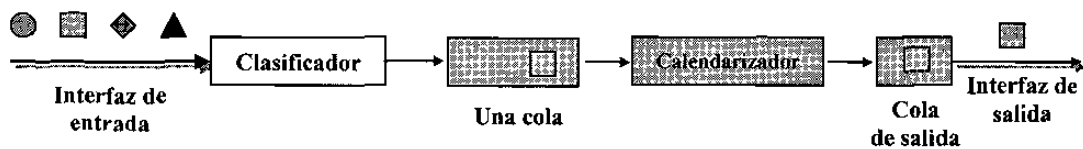


Figura 10. Encolamiento tradicional: Una cola por línea de salida

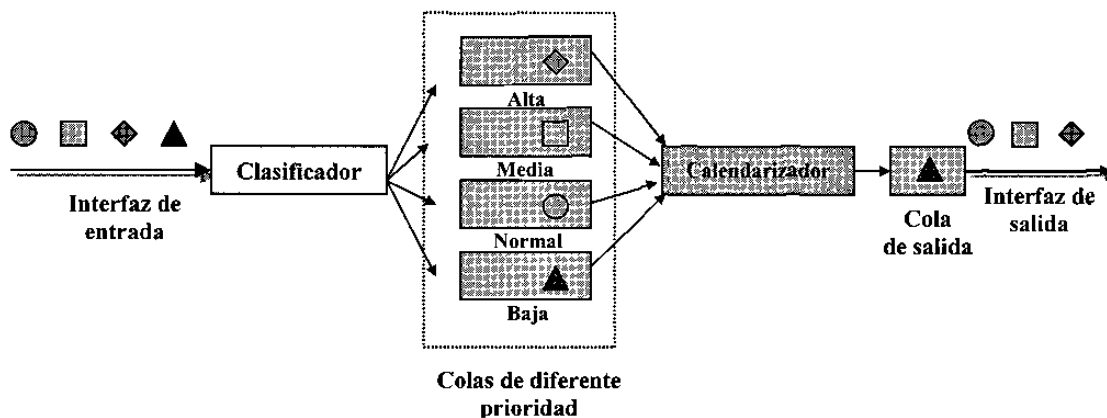


Figura 11. Encolamiento justo: Una cola por flujo

Algunos de los problemas que presenta éste modelo es que se necesita relativamente mucha potencia en el procesador para poder clasificar los paquetes en flujos y para manejar todas las colas individuales.

II.3.3.5.3.6 ENCOLAMIENTO JUSTO PONDERADO (WFQ)

El modelo WFQ es una forma de encolamiento más sofisticado, el cual es diseñado para mejorar la función de los flujos de paquetes. Da un servicio justo a cada una de las colas de acuerdo a un peso determinado. El peso esta en función del nivel de prioridad del flujo. Es una modificación del mecanismo FQ, el cual distribuye el ancho de banda

disponible en el enlace de una manera equitativa entre los diferentes flujos. De esta forma, WFQ es una implementación de FQ pero asignando un porcentaje de pesos a cada una de las colas para determinar la fracción de ancho de banda que le corresponde a cada flujo.

La operación de WFQ requiere una mayor potencia del procesador. A la hora de transmitir un paquete sobre un enlace de salida, el controlador examina todos los paquetes que están esperando para transmitirse al frente de cada una de las colas y escoger el paquete que será el próximo por transmitir.

En la figura 12 se muestra el esquema que representa el funcionamiento de WFQ.

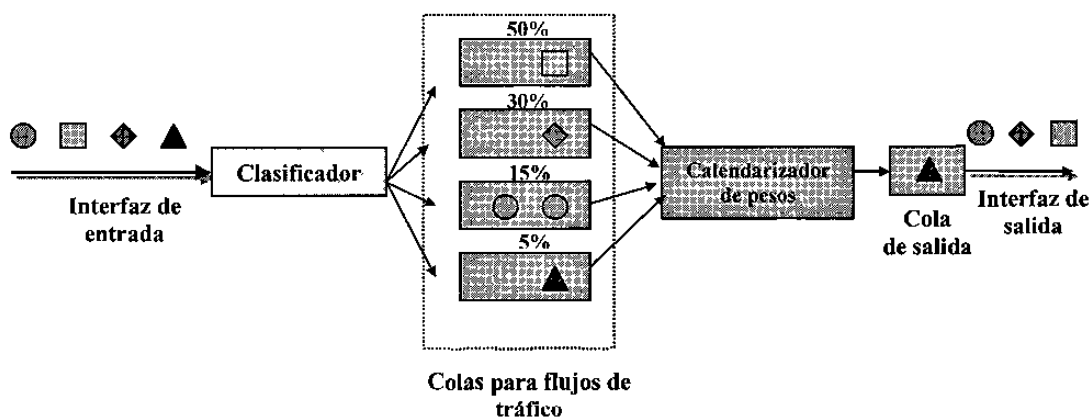


Figura 12. Modelo de Encolamiento WFQ

II.3.3.5.4 ACONDICIONAMIENTO

El acondicionamiento de tráfico controla la carga de tráfico del enlace de salida. Los mecanismos mencionados de forma general de acondicionamiento de tráfico son el de cubeta agujerada (LB, Leaky Bucket) y el marcador de tres colores con ventana de tiempo deslizante (TSWTCM, Time Sliding Window Three Colour Marker).

II.3.3.5.4.1 ACONDICIONAMIENTO DE TRÁFICO (TSWTCM)

El marcador de tres colores con ventana de tiempo deslizante (TSWTCM) puede ser usado como un componente de acondicionamiento de tráfico DiffServ. El marcador está diseñado para marcar los paquetes que serán tratados por la clase de servicio AF PHB [II.3.3.2]. El TSWTCM mide un flujo de tráfico IP y marca los paquetes con colores rojo, amarillo o verde. La marcación la realiza basándose en la medición del caudal eficaz del flujo de tráfico, comparándolo con la tasa garantizada (CTR, Committed Target Rate) y la tasa pico (PTR, Peak Target Rate). La función de TSWTCM se basa en los principios de teoría de control simple de un control retroalimentado proporcionalmente regulado. Se fundamenta por dos componentes independientes (Figura 13): un estimador de tasa y un esquema para asociar un color (precedencia de descarte) a cada paquete.

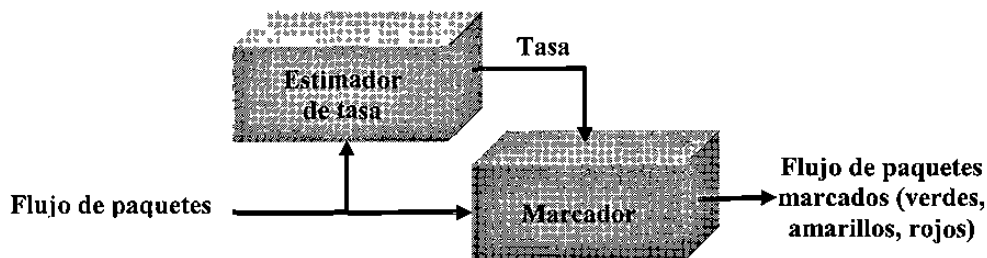


Figura 13. Diagrama a bloques del acondicionador TSWTCM

El estimador de tasa, proporciona una estimación de la tasa de llegada del flujo del tráfico. Esta tasa es un promedio del ancho de banda usado por el flujo de tráfico dentro de una ventana de tiempo.

El marcador usa la tasa estimada para asociar cada paquete estadísticamente con uno de los tres colores. El uso de las herramientas probabilísticas en el marcado en lugar del descarte de paquetes consecutivos favorece a los flujos TCP, al reducir la probabilidad en el descarte de paquetes dentro de una ventana, también favorece a flujos de voz UDP con ancho de banda baja. Estos flujos pueden beneficiar a los métodos de corrección de error (FEC), para reconstruir la pérdida de paquetes, si el número de pérdidas de paquetes se mantiene bajo. El efecto del marcador en los paquetes de un flujo de tráfico es asegurar que:

- Si la tasa promedio estimada es menor que o igual al CTR, los paquetes del flujo son verdes.
- Si la tasa promedio estimada es mayor que CTR, pero menor que o igual a PTR, los paquetes son marcados amarillos con probabilidad P_0 , y marcado verde con probabilidad $(1-P_0)$, donde P_0 es la fracción de los paquetes que contribuyen a la tasa media mayor que CTR.
- Si la tasa promedio estimada es mayor que PTR, los paquetes son marcados de color rojo con probabilidad P_1 , marcado de color amarillo con probabilidad P_2 , y marcado de color verde con probabilidad $(1 - (P_1 + P_2))$. P_1 es la fracción de los paquetes que contribuyen a la tasa media mayor que PTR. P_2 es la fracción de los paquetes que contribuyen a la tasa media entre CTR y PTR.

CTR y PTR pueden ser especificados en bits por segundo. Si PTR tiene el mismo valor que CTR, entonces todos los paquetes serán coloreados verdes o rojos, de tal manera que no habrá paquetes amarillos.

Si PTR tiene el mismo valor que la velocidad del enlace de entrada o menor y el CTR tiene un valor abajo del PTR, entonces todos los paquetes serán coloreados verdes o amarillos, de tal modo que no habrá paquetes rojos.

II.3.3.5.4.2 ACONDICIONADOR LEAKY BUCKET (LB)

El acondicionador de cubeta agujerada (Leaky Bucket), produce datos a una tasa de bits constantes, independientemente de las condiciones de la red, por lo tanto acondiciona el tráfico de una tasa variable a una tasa constante.

En la figura 14 se muestra este mecanismo. Se aprecia mejor haciendo la analogía con una cubeta la cual tiene un orificio que ocasiona una fuga (leak) a una tasa constante ρ . La capacidad de la cubeta σ determina la cantidad de datos que pueden ser almacenados antes de que sean descartados. Usa un mecanismo de encolamiento FIFO; por tanto, cuando se llena la cubeta, los paquetes de arribo adicionales son marcados con la prioridad de descarte más alta o definitivamente son descartados.

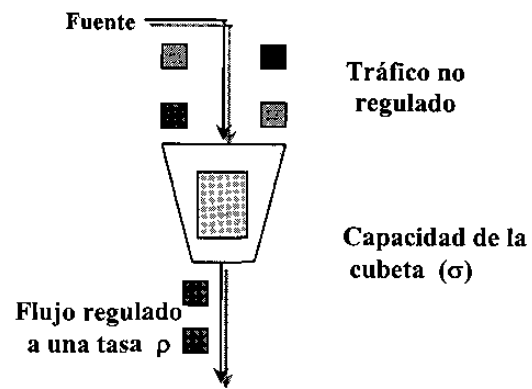


Figura 14. Acondicionador Leaky Bucket

CAPÍTULO III CONTROLADOR WFQ

III.1 INTRODUCCIÓN

La libertad de escoger una disciplina de servicio apropiada en los nodos de la red es clave para proporcionar un control de flujo. Un buen esquema debe permitir que la red trate a los usuarios de forma diferente, de acuerdo a su calidad de servicio deseado. Sin embargo, ésta flexibilidad no debe comprometer la imparcialidad de este esquema. Esto es, unas cuantas clases de usuario no deberían afectar el servicio a otras clases. La disciplina de servicio debe ser analizable, de tal modo que las garantías de servicio puedan establecerse de antemano.

III.2 MODELO DE FLUIDO

Con el *modelo de fluido* se consigue que el tráfico sea infinitamente divisible y un nodo pueda atender múltiples flujos simultáneamente. En una red real, los paquetes son procesados uno a la vez; así, el tamaño del paquete afectará el sistema de encolamiento. Sin embargo, la simplificación en el modelo de fluido nos permite visualizar la operación de las disciplinas de encolamiento más fácilmente.

Como un ejemplo supongamos que tenemos dos flujos compartiendo un enlace de salida y que todos los paquetes son del mismo tamaño como se muestra en la figura 15. En el modelo de fluido, los paquetes de los dos flujos (coloreados de verde y amarillo, respectivamente) son atendidos simultáneamente, empezando y terminando al mismo tiempo. Sin embargo, en el modelo de paquetes, el paquete verde ha sido atendido antes

que el paquete amarillo. Es importante saber que aunque el paquete verde termina mucho antes que el paquete amarillo, ambos paquetes terminan antes o al mismo tiempo que lo habrían hecho en el correspondiente sistema de fluido.

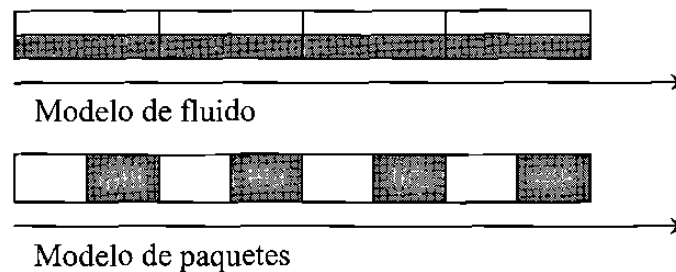


Figura 15. Modelo de Fluido y de paquetes

III.3 MECANISMO GENERAL PARA LA REPARTICIÓN DE UN PROCESADOR (GPS)

Se presenta una disciplina de multiplexión basado en el modelo de fluido llamado GPS (Generalized Processor Sharing), el cual es analizable, flexible en el servicio que se puede dar a los usuarios de diferentes aplicaciones y es eficiente en cada uno de los servidores. El servidor GPS es conservativo y opera a una tasa fija (r). A continuación se presenta el funcionamiento de GPS.

Cuando todos los flujos tienen por lo menos un paquete por transmitir todo el tiempo (debido a que generan paquetes a una tasa más alta que su ancho de banda asignado), el tiempo final del k -ésimo paquete del flujo i es igual al tiempo final del paquete anterior del mismo flujo mas el tiempo de transmisión del paquete actual.

$$F_i^k = F_i^{k-1} + \frac{L_i^k}{\phi_i \cdot r} \quad (1)$$

donde F_i^k representa el tiempo final del k -ésimo paquete del flujo i , L_i^k representa el tamaño del mismo paquete y $\phi_i \cdot r$ representa la fracción de ancho de banda asignado al flujo i .

Sin embargo, en el caso más general, el cálculo del tiempo final de servicio de los paquetes no es tan sencillo debido a que los flujos pueden pasar de activos a inactivos o viceversa, de tal manera que puede haber tiempos inactivos entre dos paquetes (por lo que el inicio de la transmisión puede no coincidir con el fin de la transmisión del paquete anterior), además de que la tasa de transmisión puede cambiar durante el envío de un mismo paquete debido a que los demás flujos pasan de activos a inactivos y viceversa (por lo que el tiempo de transmisión tendría una expresión mucho más compleja que la mostrada en la ecuación anterior).

El cálculo del tiempo final puede ser mejor explicado con la ayuda de lo que se conoce como curvas de entrada-salida de paquetes, como se explicará a continuación.

III.3.1 CURVAS DE LLEGADA - SALIDA

Para ejemplificar el cálculo del tiempo final de servicio de los paquetes en GPS, consideremos dos flujos que están compartiendo el mismo servidor, de tal manera que el flujo 1 y el flujo 2 son atendidos a una tasa de $[r/2]$, cada uno.

Las Figuras 16 y 17 muestran las curvas de llegada-salida correspondientes a este ejemplo. La línea punteada es el proceso de llegada (la cantidad de datos de llegada contra el tiempo), y cada flecha representa la llegada de un paquete. La altura de la flecha es el tamaño del paquete. Por ejemplo, en la figura 16, los paquetes que llegan en el tiempo $t = 1, 2, 3,$ y 11 con un tamaño de paquete $1, 1, 2, 2$ respectivamente. Después de la llegada de cada paquete, la curva de llegada (línea punteada) se incrementa por el tamaño del paquete. La línea sólida representa la cantidad de datos que han sido atendidos. Durante el intervalo de tiempo $(0, 1)$, el flujo 1 está ocioso y el flujo 2 envía un paquete en el tiempo $t = 0$. Por lo tanto, el primer paquete del flujo 2 es atendido con una tasa de 1 durante el intervalo de tiempo. En $t = 1$, llega el primer paquete del flujo 1, y ambos flujos tienen paquetes por transmitir y son atendidos con una tasa de $\frac{1}{2}$ hasta el tiempo $t = 9$. En este tiempo los tres paquetes del flujo 1 han dejado el sistema, y no llegan más paquetes. Durante el intervalo $(9,11)$, el flujo 2 es atendido otra vez con una tasa de 1. En el tiempo $t = 11$, llega un paquete del flujo 1 y por este tiempo todos los paquetes del flujo 2 son finalizados. El cuarto paquete del flujo 1 es entonces atendido con una tasa completa de 1 y terminan en el tiempo $t = 13$. Usando las curvas de llegada - salida, podemos calcular el tiempo final de un paquete en el sistema GPS. Veamos el flujo 1 de la figura 16. El tercer paquete del flujo 1 llega en el tiempo $t = 3$, y la cantidad total de datos de llegada es de 4 (línea punteada). La línea sólida indica que la cantidad de datos que han sido atendidos alcanzan 4 en el tiempo $t = 9$. Así el tercer paquete del flujo 1 llega en el tiempo $t = 3$ y sale en el tiempo $t = 9$; el retardo total que experimenta es 6.

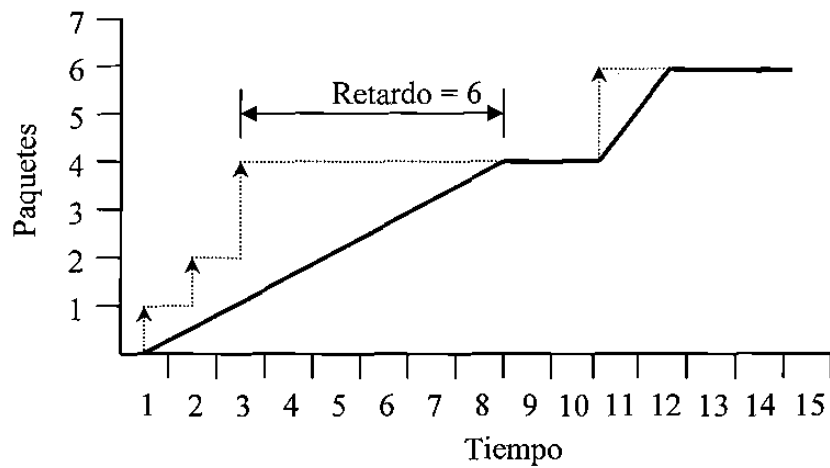


Figura 16. Curva de llegada – salida para el flujo 1

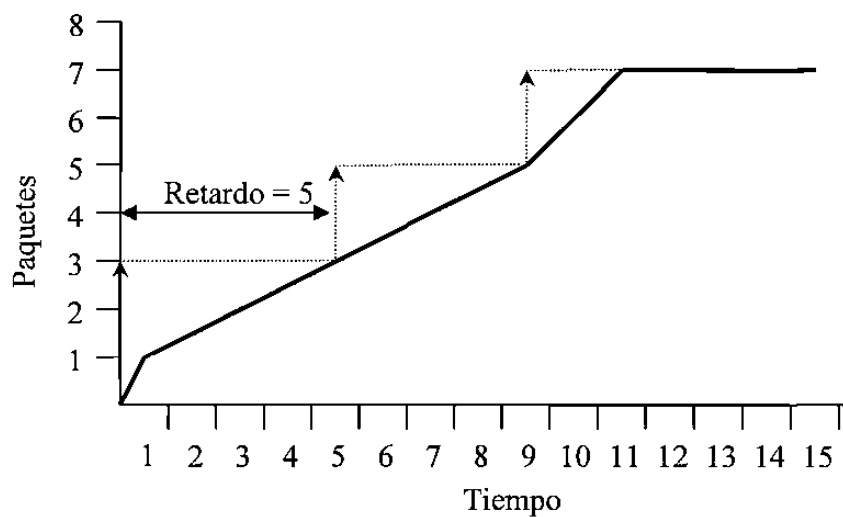


Figura 17. Curva de llegada – salida para el flujo 2

La distancia entre la línea punteada y la línea sólida a lo largo del eje X representa el retardo que experimenta el paquete correspondiente.

GPS es atractivo por varias razones:

- El retardo de un paquete de la clase i puede ser limitado como una función de la longitud de la cola de la clase i , independientemente de las colas y llegadas de las otras clases. El esquema FCFS no cuenta con ésta propiedad.
- Es caracterizado por números reales positivos: $\phi_1, \phi_2, \dots, \phi_N$
- Variando las ϕ_i 's, se tiene la flexibilidad de tratar a las clases de diferentes formas.

Por ejemplo:

- a) Cuando todas las ϕ 's son iguales, el sistema se reduce a un procesador uniforme compartido
 - b) A los flujos que requieren menor ancho de banda, se les puede asignar una tasa garantizada menor, permitiendo de esta forma un mejor tratamiento a los flujos que requieran mayor ancho de banda.
- Usar servidores GPS garantiza una cota superior para el retardo de encolamiento experimentado en la red completa cuando las fuentes son restringidas por un control de flujo de cubeta agujerada (ver sección II.3.3.5.4.2).
 - Es atractivo para las clases que envían tráfico en tiempo real, tal como voz y video.

Con GPS se tiene una cola FIFO y un servidor para cada clase compartiendo el mismo enlace, de manera que las colas son atendidas en forma simultánea. Para cualquier intervalo de tiempo, cuando existen exactamente N colas llenas, el servidor atiende a todas las colas que no están vacías en proporción al servicio asignado a sus correspondientes

clases, donde cualquier clase tiene garantizada una tasa mínima de servicio durante cualquier intervalo de tiempo en el que tenga paquetes por transmitir.

Desafortunadamente GPS es un sistema de fluido hipotético que no puede ser implementado en un mundo real, ya que supone que se puede atender simultáneamente a todas las clases con paquetes por transmitir o, equivalentemente, que el tráfico es infinitamente divisible. En un sistema de paquetes realista, solamente una clase puede recibir servicio en un tiempo dado y un paquete completo debe ser atendido antes de que otro paquete pueda iniciar su transmisión. La forma más aproximada de emular GPS, es utilizando el algoritmo (WFQ, Wiegthed Fair Queueing), como se verá en la siguiente sección.

III.4 ENCOLAMIENTO JUSTO PONDERADO (WFQ)

III.4.1 INTRODUCCIÓN

Es una clase de los algoritmos de calendarización que soportan asignación de ancho de banda y retardos. El WFQ original fue propuesto hace 10 años [Bennet J. C. R. y Zhang H. 1996]. Se han desarrollado algunas variantes con diferentes compromisos entre complejidad y funcionamiento. WFQ ha sido ampliamente implementado en enrutadores para soportar Calidad de Servicio (QoS).

Los algoritmos WFQ son frecuentemente explicados con el modelo de fluido, donde se supone que el tráfico es infinitamente divisible y un nodo puede atender múltiples flujos simultáneamente. En una red real, los paquetes son procesados uno a la vez, de esta manera el tamaño del paquete afectará el sistema de encolamiento. Sin embargo la simplificación en el modelo de fluido nos permite visualizar la operación de las disciplinas de colas más fácilmente, aunque en algunos casos es mejor derivar los resultados directamente para un sistema “paquetizado” que para un modelo de fluido.

III.4.2 MECANISMO GENERAL PARA LA REPARTICIÓN DE UN PROCESADOR DE PAQUETES (PGPS)

El mecanismo PGPS (Packet by Packet GPS) fue originalmente llamado WFQ, el cual es una aproximación de paquetes del calendarizador GPS aún cuando los paquetes son de longitud variable. El objetivo de WFQ es atender a los paquetes en la misma secuencia en el cual habrían sido atendidos en GPS. WFQ se auxilia en el cálculo de lo que se conoce como tiempo virtual, que representa el tiempo en el que los eventos (llegadas y salidas de paquetes) habrían sucedido en GPS. Se calcula, entonces, el tiempo virtual de salida de cada paquete y se selecciona aquel paquete que tenga el valor más pequeño para ser transmitido. En consecuencia, si el paquete A termina la transmisión antes que el paquete B bajo un sistema GPS, el paquete A será seleccionado para transmitirse antes que el paquete B en el correspondiente WFQ.

III.4.3 IMPLEMENTACIÓN DEL ALGORITMO WFQ

El algoritmo consiste en asignar un tiempo virtual a cada evento. Se usa este concepto de tiempo virtual para seguir el progreso de GPS que conducirá a una implementación práctica de PGPS. Este tiempo virtual está en función del número de clases activas en un intervalo pequeño de tiempo, así como de la suma de los pesos para todas las clases.

Sea t_j el tiempo en que transcurre el j – ésimo evento, el cual es ordenado arbitrariamente. El tiempo del primer arribo de un período ocupado se denota como $t_1=0$. Entonces para cada $j = 2, 3, \dots$, el conjunto de clases que están ocupados (produciendo tráfico) o activas durante el tiempo transcurrido entre dos eventos (t_{j-1}, t_j) es fijo y se denota este conjunto como B_j .

El tiempo virtual $V(t)$ se define como cero en todos los tiempos en los que el servidor se encuentra ocioso. Es decir, cuando no existen clases activas, $V(t) = 0$.

Si denotamos por t_0 el tiempo en el que inicia un período ocupado, y por $t_j, j=1,2,\dots$, los tiempos en los que ocurren los eventos subsecuentes., entonces $V(t)$ evoluciona de la siguiente manera:

$$V(t_0) = 0 \quad (3)$$

$$V(t_j) = V(t_{j-1}) + \frac{t_j - t_{j-1}}{\sum_{i \in B_j} \phi_i} \quad j = 1, 2, 3, \dots, \quad (4)$$

donde: $\sum_{i \in B_j} \phi_i$ Representa la suma de los pesos de las clases activas en el intervalo de tiempo (t_{j-1}, t_j) .

ϕ_i 's Son los pesos asignados de manera proporcional al ancho de banda requerido por los enlaces.

Una vez que se tenga el tiempo virtual de llegada del paquete, hay que obtener los tiempos virtuales en los cuales el paquete inicia y termina de ser procesado. Sea S_i^k y F_i^k los tiempos virtuales de inicio y final respectivamente. Si definimos $F_i^0 = 0$ para toda i :

$$S_i^k = \max\{F_i^{k-1}, V(a_i^k)\} \quad (5)$$

$$F_i^k = S_i^k + \frac{L_i^k}{\phi_i \cdot r} \quad (6)$$

donde: L_i^k representa el tamaño del paquete K de la clase i .

a_i^k tiempo en que llegó el k -ésimo paquete en la clase i

$V(a_i^k)$ es el tiempo virtual de llegada del paquete.

F_i^{k-1} tiempo en que terminó servicio el paquete anterior.

El orden en que los paquetes se van a transmitir está determinado por el tiempo final virtual (F_i^k). El paquete con el F_i^k más pequeño es el que se selecciona para ser transmitido.

Por tanto, las propiedades que hacen atractiva la interpretación del tiempo virtual desde el punto de vista de la implementación son:

- El tiempo final virtual esta basado en la clase a la que pertenece el paquete, en el tiempo de llegada del paquete, en el tiempo virtual de llegada del paquete ($V(a_i^k)$), en el tamaño del paquete, y en la fracción de ancho de banda asignada a la clase de tráfico correspondiente (ϕ), de acuerdo a las ecuaciones (2) y (3).
- Los paquetes son atendidos de acuerdo al tiempo final virtual.
- Es necesario actualizar el tiempo virtual cuando hay eventos en el sistema GPS.

III.4.3.1 DIAGRAMA DE FLUJO DEL ALGORITMO WFQ

En la figura 18 se muestra el diagrama de flujo de la implementación del algoritmo WFQ

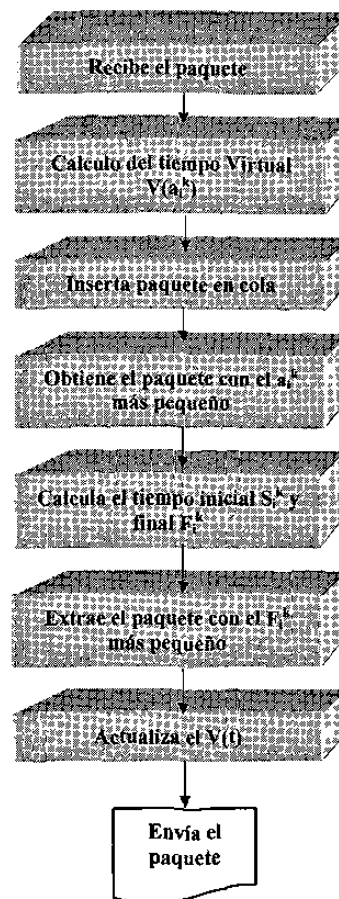


Figura 18. Diagrama de Flujo del algoritmo WFQ

El funcionamiento de WFQ se puede resumir de la siguiente manera. Cuando llega un paquete se calcula su tiempo virtual de llegada (que será usado a la hora de calcular el tiempo virtual de fin de servicio, como se mencionará más adelante) y se intenta almacenar el paquete en la cola que le corresponde de acuerdo con la clase a la que corresponda y prioridad de descarte (color). En caso de que no haya espacio en esta cola, se buscará espacio en las colas correspondientes a los paquetes con prioridad más alta de descarte. Si esta búsqueda también falla, se buscará un paquete con prioridad más alta de descarte que, en caso de encontrarse, será removido para hacer espacio para el recién llegado. Si el enlace de salida está libre cuando llega un paquete; se procederá a transmitirlo de manera inmediata, de lo contrario, el programa pasará a un estado de espera.

Cuando el enlace de salida se libera después de transmitir un paquete, el programa ejecuta la rutina en la que se decide cuál será el siguiente paquete transmitido. Para tal efecto, de las tres colas correspondientes a cada clase (usadas para almacenar paquetes de diferente color) se seleccionará el paquete que haya llegado antes (para usar una disciplina FIFO dentro de cada clase). Posteriormente, se calcula el tiempo virtual de fin de servicio de cada uno de estos paquetes seleccionados. El siguiente paquete transmitido será aquel con el tiempo virtual de fin de servicio más pequeño. Al ocurrir cada llegada y salida de algún paquete se actualiza el tiempo virtual del sistema.

III.4.4 POSIBLES MODALIDADES REFINADAS DEL ALGORITMO WFQ

Se mencionan tres esquemas modificados de WFQ que han sido propuestos para redistribuir de manera dinámica el exceso de ancho de banda que resulta después de dar a cada flujo de tráfico el ancho de banda que le ha sido garantizado.

Los modelos propuestos, en comparación con WFQ, reducen la probabilidad de pérdida de algunos paquetes y controlan las distribuciones del retardo en las colas usando diferentes criterios. Conservan la habilidad del calendarizador WFQ de proporcionar los límites de retardo en el peor caso y los esquemas trabajan como WFQ cuando no hay exceso de ancho de banda. Estas modalidades se presentan a continuación:

III.4.4.1 EL RETARDO MÁS LARGO PRIMERO (LDF, LONGEST DELAY FIRST)

Como se mencionó anteriormente, cuando en WFQ uno de los flujos de tráfico no está usando la parte que le corresponde de ancho de banda, ésta se redistribuye entre los que sí están activos de manera proporcional a las ϕ 's.

LDF propone una manera alternativa de redistribuir este ancho de banda excedente, usándolo para atender aquellos paquetes que hayan permanecido por más tiempo dentro de las colas de espera. Como lo que se redistribuye es únicamente el ancho de banda excedente, no se sacrifican las garantías de calidad de servicio.

El efecto de esta nueva política es el de reducir la varianza en la distribución de retardo. Esto ha sido comprobado en [Duffield N. G. *et al*, 1998] mediante simulaciones con tráfico de video y voz.

III.4.4.2 MENOR TIEMPO PARA SOBRE FLUJO (LTO, LEAST TIME TO OVERFLOW)

Al igual que LDF, esta es una propuesta para redistribuir el ancho de banda excedente, pero en esta ocasión se usa para atender a los paquetes de aquellos flujos que estén más cercanos a perder paquetes. La identificación del flujo más cercano a perder paquetes se basa en la ocupación de la cola correspondiente y en la tasa con la que llegan los paquetes.

El efecto de esta propuesta es la de reducir la pérdida de paquetes hasta donde sea posible dados los recursos con los que cuenta el sistema, como se muestra en [Gallardo J. R. y Makrakis D., 2001].

III.4.4.3 WFQ DINÁMICO BASADO EN PREDICCIÓN (DP, DYNAMIC PREDICTIVE)

Esta es una extensión del algoritmo LTO – WFQ en la que, se identifica el flujo que está más cercano a perder paquetes usando técnicas de predicción basadas en modelos realistas de tráfico. Este algoritmo y su desempeño se describen de manera detallada en [Gallardo J. R. y Makrakis D., 2001].

III.4.5 MODELO TÍPICO DE UN ENRUTADOR QUE INCLUYE UN CONTROLADOR WFQ

En la figura 19 se puede ver un escenario típico de un enrutador que soporta Servicios Diferenciados. El enrutador está recibiendo los paquetes de cuatro clases diferentes cada uno, con su respectiva fracción de ancho de banda asignada (peso). En la figura, las cuatro clases de tráfico están ejemplificados por tráfico de video, voz, datos y www, aunque aún no hay estándares para la definición de las clases www, aunque aún no hay estándares para la definición de las clases

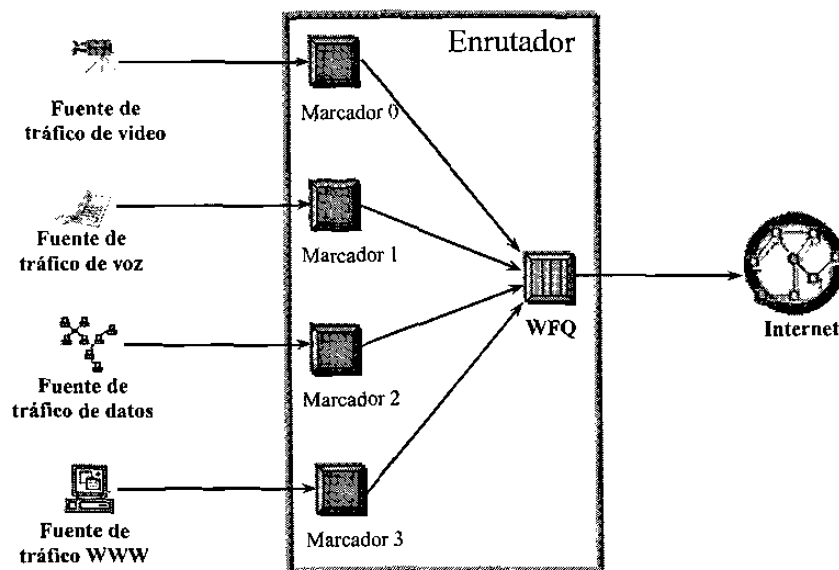


Figura 19. Estructura típica de un enrutador que soporta Servicios Diferenciados

Los algoritmos que se usen para asignar recursos a los diferentes flujos de tráfico deben satisfacer requerimientos mínimos de equidad y eficiencia. Se propone usar el algoritmo WFQ para garantizar a cada tipo de tráfico, como mínimo, una fracción (usualmente denotada por ϕ) ϕ_i del ancho de banda, sin importar lo que pase con los otros flujos que compiten por el recurso (aislamiento y equidad). Cuando uno de los flujos de

tráfico está inactivo, su fracción del ancho de banda se repartirá entre las clases que estén activas de manera proporcional a su propia porción (ϕ_i 's).

Además, para soportar las múltiples clases de servicios, los proveedores de servicio tendrán que usar técnicas eficientes para la repartición de los recursos, lo cual implica que los flujos de tráfico tendrán que ser conceptualmente agrupados y atendidos de acuerdo a la filiación administrativa, al protocolo, u otros criterios. Esto crea la necesidad de tener estrategias jerárquicas para administrar los recursos.

III.5 MODELO WFQ JERÁRQUICO

Los algoritmos de calendarización WFQ pueden ser usados de manera jerárquica para alcanzar una mejor distribución de recursos en los enlaces, debido a que existen escenarios en los que no es suficiente un solo nivel de diferenciación. Por ejemplo, es posible que un primer nivel de diferenciación se deba a que hay varias compañías, digamos **A**, **B**, **C** y **D**, que requieran una fracción del ancho de banda total de un equipo de comunicaciones. Denotemos a estas fracciones como ϕ_A , ϕ_B , ϕ_C , ϕ_D , respectivamente. Además, cada compañía puede requerir de todos los servicios que la red puede ofrecer. Supongamos que estos servicios son la transmisión de video, voz, datos de alta y datos de baja prioridad y que se le asigna a cada tipo de tráfico las proporciones ϕ_{Vi} , ϕ_{Vo} , ϕ_{Da} , ϕ_{Db} , respectivamente del total que le toca a la compañía correspondiente. De esta manera, el tráfico de video de la compañía **A** tiene garantizado el uso de la fracción ($\phi_A \cdot \phi_{Vi}$) del ancho de banda total del equipo de comunicaciones en cuestión. Ahora, si algún tipo de tráfico de

la compañía A no está usando el ancho de banda que tiene asignado, es justo que se redistribuya primero entre los demás usuarios de la compañía A, antes de usarlo para atender a los usuarios de otras compañías (si hubiera todavía ancho de banda disponible).

Este escenario le dá un carácter jerárquico a la manera en que hay que distribuir el ancho de banda entre los múltiples usuarios: primero hay que distribuir el ancho de banda entre los usuarios de una misma compañía y, posteriormente, si todavía hay excedentes, redistribuirlos entre los usuarios de otras compañías. Puede haber más de dos niveles en la estructura jerárquica.

La propuesta tradicional para implementar la distribución jerárquica de recursos, es usando controladores WFQ de un solo nivel y organizándolos en una estructura jerárquica, tal como se muestra en la figura 20.

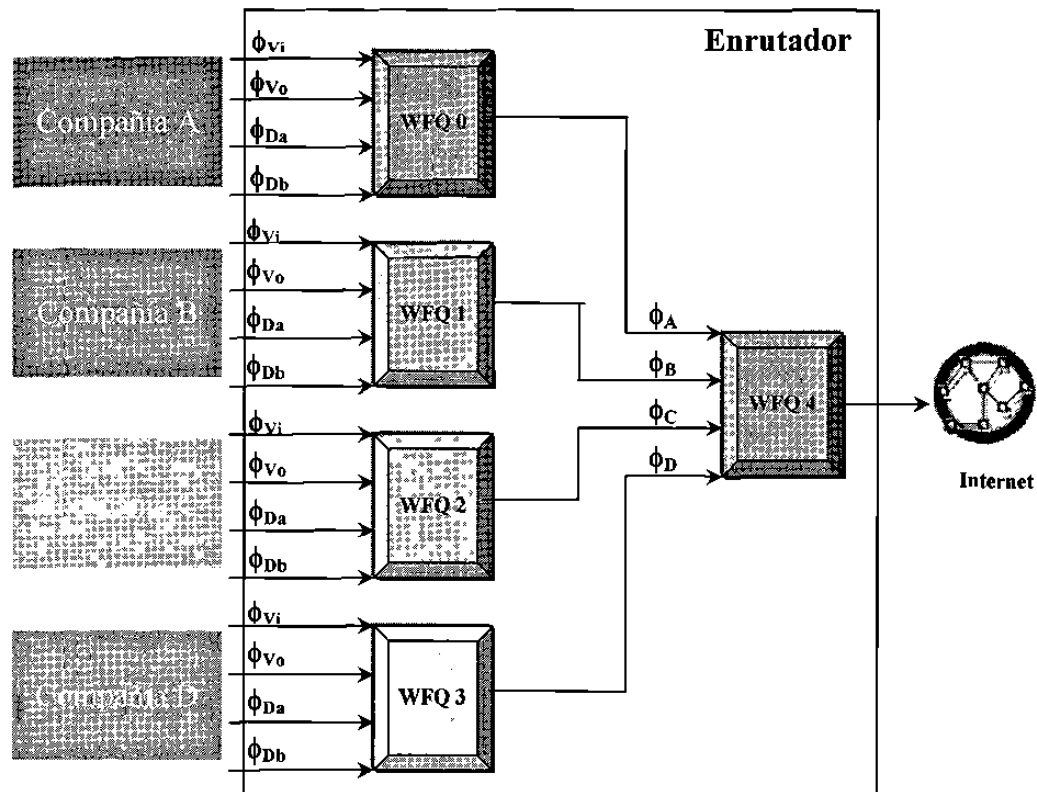


Figura 20. Propuesta tradicional para WFQ jerárquico

III.5.1 DEBILIDADES DE LOS SISTEMAS JERÁRQUICOS ACTUALES

La manera en que se implementan los sistemas jerárquicos actuales puede dar como resultado pérdidas y retardos innecesarios, incluso injusticia bajo algunas condiciones. El problema clave es cómo especificar la velocidad de los enlaces que unen a los WFQ de la primera etapa (los primeros 4 bloques WFQ's) con el WFQ de la segunda etapa. Cuando esta velocidad es muy baja, los paquetes permanecen innecesariamente durante mucho tiempo en el bloque de la primera etapa antes de pasar a la segunda etapa. Mientras que, por el contrario, cuando la velocidad es muy alta, la primera etapa tiende a enviar muy rápido los paquetes, ocasionando que se acumulen en los elementos de la última etapa. La

acumulación causa pérdida de paquetes debido al sobre flujo en las colas. Además, conforme la velocidad aumenta en los bloques de la primera etapa, éstos tienden a trabajar en forma FIFO, perdiendo su utilidad y efectividad.

El propósito de esta tesis es comprobar que estos problemas, previstos de manera intuitiva, son reales y proponer una alternativa de solución, que se describe en la sección III.6.

III.6 MODELO JERÁRQUICO DE DOS NIVELES (CH2WFQ)

De acuerdo a lo anterior, se propone un sistema jerárquico de dos niveles, el cual consta de un solo controlador que decidirá de manera centralizada cómo asignar los recursos en un sistema de telecomunicaciones, como se muestra en la figura 21. Llamamos a esta nueva propuesta CH2WFQ, que es la contribución principal de este trabajo de tesis. Este controlador operará de manera que no se coloque una serie de controladores simples en cascada, a fin de estar en contacto directo con los paquetes que vengan de todos los usuarios y con el enlace físico de transmisión para evitar tiempos muertos cuando haya paquetes por transmitir. Aunque en este trabajo nos limitamos a un WFQ jerárquico (HWFQ) de dos etapas, nuestras ideas pueden ser fácilmente extendidas para incluir más de dos etapas.

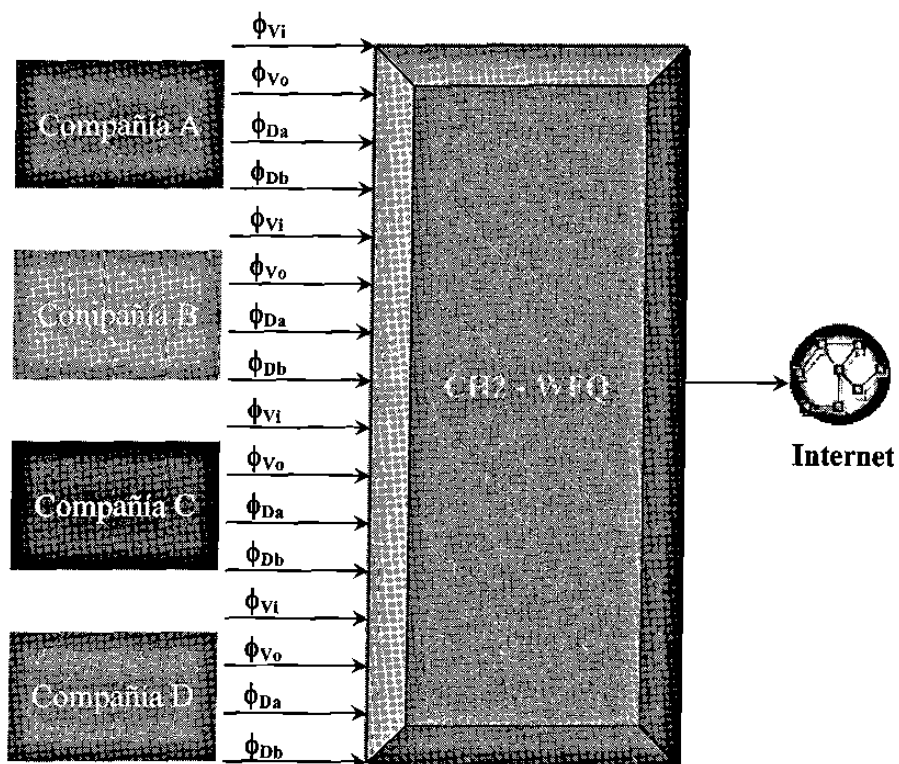


Figura 21. Nueva Propuesta CH2WFQ

CAPÍTULO IV. IMPLEMENTACIÓN DE LOS MODELOS JERÁRQUICOS WFQ Y CH2WFQ

IV.1 INTRODUCCIÓN

La implementación de los modelos jerárquicos se llevó a cabo a través de simulaciones, debido a que la implementación física es muy costosa y el tiempo con que se cuenta es poco. A través de un buen simulador es posible obtener resultados confiables sobre el desempeño del sistema. Para realizar la implementación de los modelos HWFQ y CH2WFQ se utilizó la Herramienta de Simulación de Ingeniería de Red Optimizada (OPNET, Optimized Network Engineering Tool), el cual es un sistema capaz de simular de forma detallada los diferentes elementos de una red y de predecir su desempeño. OPNET incluye herramientas para el diseño de modelos, simulación, recopilación y análisis de los datos. OPNET divide los modelos en tres ambientes: Red, Nodo y Procesos, llamado modelado de dominio, cubriendo esencialmente todos los niveles jerárquicos de un modelo. A continuación se describe brevemente cada dominio.

Dominio de Red.- Se refiere a las especificaciones de un sistema en términos de dispositivos de alto nivel llamados nodos, y enlaces de comunicación entre ellos. Es la descripción general de los objetos contenidos en el sistema, especificando la localización física, configuración e interconexión entre ellos. Para la construcción de las arquitecturas jerárquicas, se empleó un modelo de red que consta de un solo nodo, el cual define de manera global los sistemas que serán simulados.

Dominio de Nodos. Especifica las capacidades de los nodos en términos de la aplicación, del procesamiento, del encolamiento y de las interfaces de comunicación. Cada nodo, a su vez, puede contar con varios bloques que ejecuten tareas independientes. Físicamente, los nodos corresponden a varios tipos de equipo de comunicación, tales como: enrutadores, puentes, conmutadores, terminales, computadoras, servidores, etc.

En este trabajo se crearon dos modelos de nodos diferentes para la administración jerárquica de recursos: el modelo de nodo jerárquico tradicional WFQ, basado en cinco bloques WFQ y el modelo de nodo jerárquico propuesto, CH2WFQ, que como ya se mencionó, consta de un solo controlador. Ambos modelos de nodos serán descritos en la sección IV.2.

Dominio de Proceso.- Se refiere a las especificaciones de comportamiento para los procesos que operan dentro de los bloques que constituyen cada nodo del sistema. Un proceso se define a través de un diagrama de estados, de las acciones que se ejecuten en cada estado y de las condiciones para la transición entre estados.

El modelo de procesos en OPNET se puede usar para implementar una gran variedad de subsistemas (hardware y software), incluyendo protocolos de comunicación, algoritmos, disciplinas de servicio, generadores de tráfico, colectores de estadística, etc. Un proceso individual o un grupo de procesos se definen para implementar una tarea particular dentro de un módulo.

El modelo de proceso es implementado a través de máquinas de estado finito (FSM, Finite State Machine) conectadas por medio de líneas de transición, las cuales se dan a través de interrupciones. Dentro de los estados se puede introducir un código de programación para que realice una cierta tarea. El lenguaje de programación de OPNET es el PROTO C que es compatible con C++ con algunas adiciones que facilitan la compilación de los programas para su ejecución. Las FSM pueden ser representadas a través de estados forzados o no forzados. Dentro de OPNET los estados forzados son de color verde, mientras que los estados no forzados son de color rojo.

El código por ejecutar en cada estado se divide en dos partes: las instrucciones de entrada, que se ejecutan cuando la FSM llega al estado, y las instrucciones de salida, que se ejecutan cuando la FSM está a punto de transitar a otro estado. Si se trata de un estado no forzado después de ejecutar las instrucciones de entrada, la FSM detendrá temporalmente la ejecución de su código (estará latente) y sólo ejecutará las instrucciones de salida cuando se dé una interrupción para posteriormente pasar a otro estado. Cuando se trata de estados forzados, no hay pausa, entra la ejecución de las instrucciones de entrada /salida y la transición al siguiente estado.

Las interrupciones pueden provenir de otro módulo o bien puede tratarse de interrupciones calendarizadas por el mismo módulo.

Las transiciones entre los estados pueden ser condicionadas o incondicionales. Una transición condicionada implica que para pasar al siguiente estado forzosamente se debe

cumplir con alguna condición específica. Las transiciones incondicionales indican un flujo continuo entre estados.

La implementación de los modelos jerárquicos WFQ tradicional y CH2WFQ se realizó bajo los dominios de red, nodo y proceso.

IV.2 MODELO DE NODOS JERÁRQUICO TRADICIONAL

El modelo de nodos creado para la administración jerárquica de recursos mediante controladores simples en cascada consta de varios bloques (fuentes de tráfico, marcadores, controladores WFQ, y la red Internet) y conexiones (flujos de paquetes) que permiten que los paquetes fluyan entre los respectivos módulos. En la figura 22 se observa que un enrutador está recibiendo paquetes de 4 diferentes compañías y que cada compañía tiene usuarios que generan tráfico correspondiente a 4 diferentes clases. Cada clase cuenta con 3 colas separadas para almacenar los paquetes conforme a su prioridad de descarte (color) (ver sección III.4.5). Cada clase es atendida usando una disciplina FIFO.

Definimos al conjunto de los primeros cuatro bloques WFQ como la primera etapa y como segunda etapa al bloque WFQ que está conectado directamente con el enlace de salida del enrutador. La función que tienen las fuentes de tráfico en el modelo jerárquico es la de generar los paquetes en forma de ráfagas y especificar su tasa en términos de paquetes por segundo. El modelo de tráfico se describirá con más detalle en la sección V.1.1. Las

fuentes de tráfico de la 0 a la 3 corresponden a la compañía 0, las fuentes de tráfico de la 4 a la 7 (no mostrados en la figura 22) corresponden a la compañía 1, y así sucesivamente, de tal manera que las fuentes de tráfico de la 12 a la 15 corresponden a la compañía 3 (ver sección III.5).

El modelo de nodos jerárquico tradicional se puede describir a grandes rasgos de la siguiente manera: garantiza para cada fuente de tráfico, como mínimo, una fracción ϕ_i (peso) del ancho de banda. La compañía 0 tiene asignada una fracción ϕ_0 del ancho de banda total del enrutador, que es controlado por el bloque WFQ. El controlador WFQ0 tiene asignada una fracción ϕ_{00} del ancho de banda para la fuente de tráfico 0. Si tomamos en cuenta que la tasa de servicio para el enlace de salida del enrutador es de R , entonces la fuente de tráfico 0 de la compañía 0 tiene garantizado el uso de la fracción de $(\phi_{00} \cdot \phi_0 \cdot R)$ del ancho de banda total. De manera similar, a la compañía 3 le corresponde una porción ϕ_3 del ancho de banda total del enrutador. El controlador WFQ3 tiene reservada una fracción ϕ_{33} del ancho de banda para la fuente de tráfico 15. Por lo tanto, la fuente de tráfico 15 de la compañía 3 tiene garantizado el uso de la fracción de $(\phi_{33} \cdot \phi_3 \cdot R)$ del ancho de banda total del enrutador. Se hace lo mismo para las demás fuentes de tráfico. Ahora, si algún tipo de tráfico de la compañía 0 no está usando el ancho de banda que tiene asignado, entonces se va a redistribuir entre las fuentes de tráfico de la misma compañía, antes de redistribuirlo a las demás compañías (compañías 1, 2 y 3). De ahí el carácter jerárquico de la distribución de recursos.

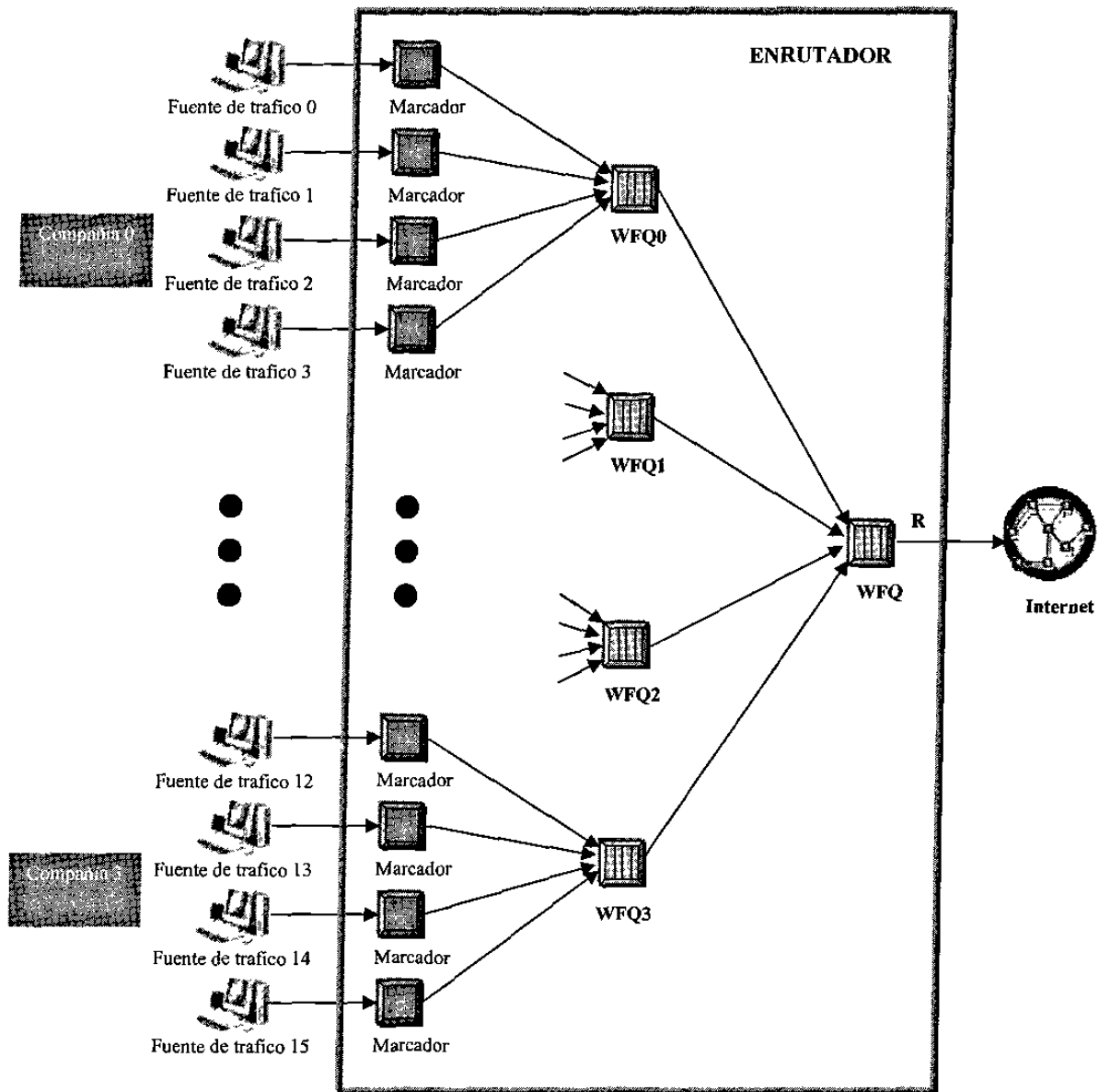


Figura 22. Modelo de nodos jerárquico WFQ

Los nodos marcadores mostrados en el modelo jerárquico tradicional son compatibles con el marcador de tres colores con ventana deslizante de tiempo (TSW-TCM) (ver sección II.3.3.4.4), el cual se encarga de asignar el color que le corresponde a cada paquete dependiendo de si viola o no el acuerdo de nivel de servicio (SLA) y, en caso de

que se viole, dependiendo de qué tanto. A cada color le corresponde una prioridad de descarte, así como un valor, tal y como se muestra en la tabla I.

TABLA I. ASIGNACIÓN DE COLOR

COLOR	PRIORIDAD DE DESCARTE	Valor asignado
Verde	Baja	0
Amarillo	Media	1
Rojo	Alta	2

Cada uno de los bloques WFQ, tiene una cola de entrada separada para cada clase de tráfico y para cada color, dando un total de 12 colas en cada uno de los bloques.

La única función que tiene el nodo de la Internet en la simulación, es destruir a los paquetes atendidos para liberar memoria.

IV.3 MODELO DE NODOS JERÁRQUICO CH2WFQ

Cuando usamos el modelo jerárquico tradicional, como se mostró en la figura 22, los cuatro bloques WFQ de la primera etapa están recibiendo independientemente los paquetes que llegan de cada una de las compañías correspondientes, y los almacenan en sus propias colas. El modelo de CH2WFQ se generó a partir de WFQ. La primera extensión que se tuvo que hacer fue la de agregarle más colas de tal manera que fuera capaz de recibir los paquetes de las cuatro compañías y almacenarlos apropiadamente en 48 colas, separándolos de acuerdo a la compañía, clase y color al que correspondan.

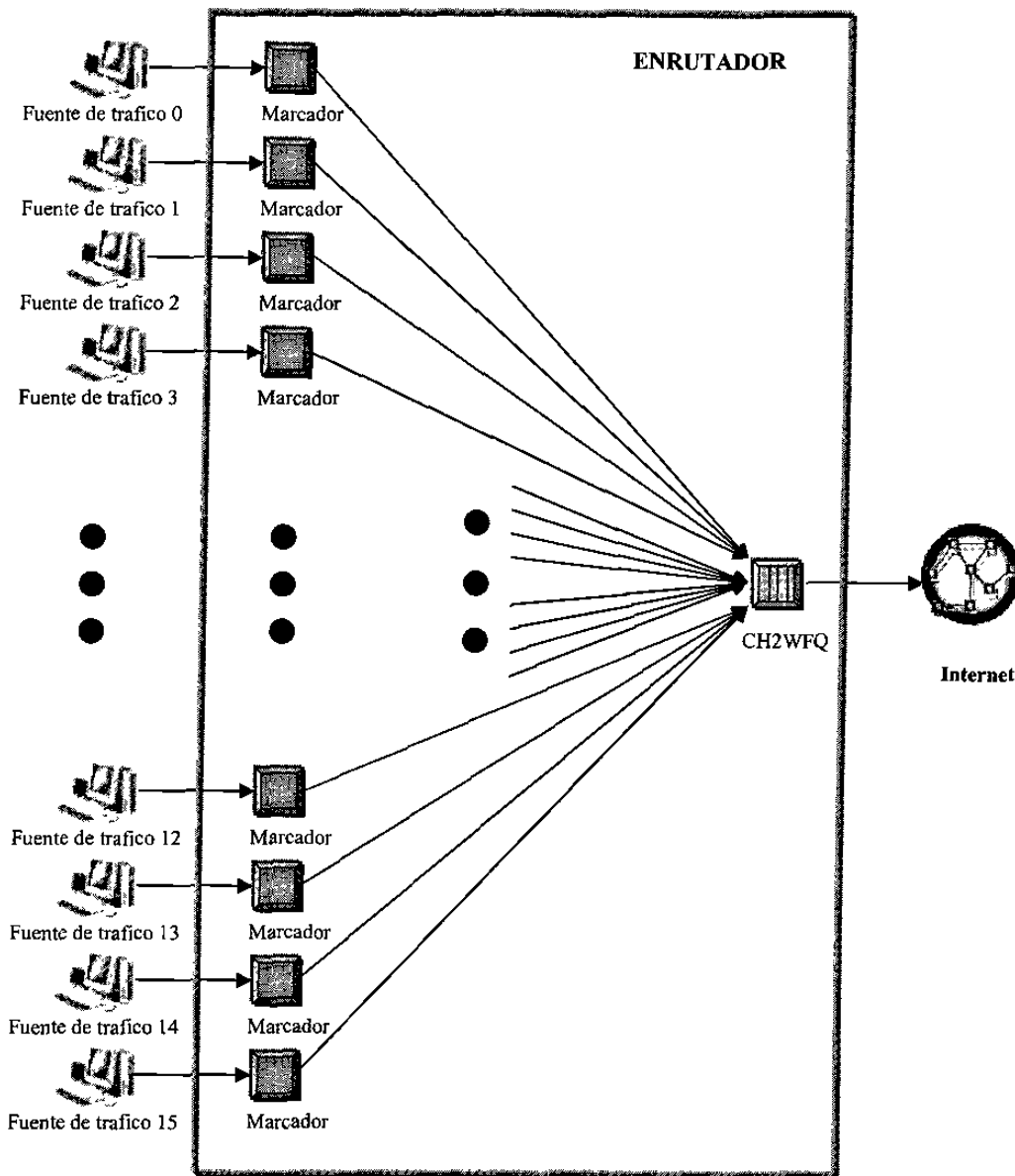


Figura 23. Modelo de nodos jerárquico de dos etapas CH2WFQ

De forma similar al modelo jerárquico tradicional, el modelo CH2WFQ tiene dos etapas de competencia, la única diferencia es que éstas etapas las maneja un solo controlador. Se puede pensar que hay 4 superclases y que cada superclase tiene 4 clases. Además, a cada clase le corresponden 3 colas separadas donde se almacenan los paquetes dependiendo de

su prioridad de descarte (color). Dentro de cada una de las superclases, las clases competirán para definir cuál paquete será el ganador. Una vez que haya un paquete ganador en cada una de las superclases, éstos deberán entrar en competencia en la segunda etapa, para decidir cuál será el paquete enviado a la salida del enrutador.

Los nodos marcadores, fuentes de tráfico y de la Internet usados para este modelo fueron los mismos que utilizamos para el modelo jerárquico WFQ tradicional.

Las fracciones de ancho de banda asignado a las fuentes de tráfico y a las compañías fueron las mismas que utilizamos para el modelo anterior, para después compararlos en las simulaciones.

IV.4 MODELO DE PROCESOS PARA WFQ

El modelo de procesos implementado para WFQ se muestra a continuación en la figura 24.

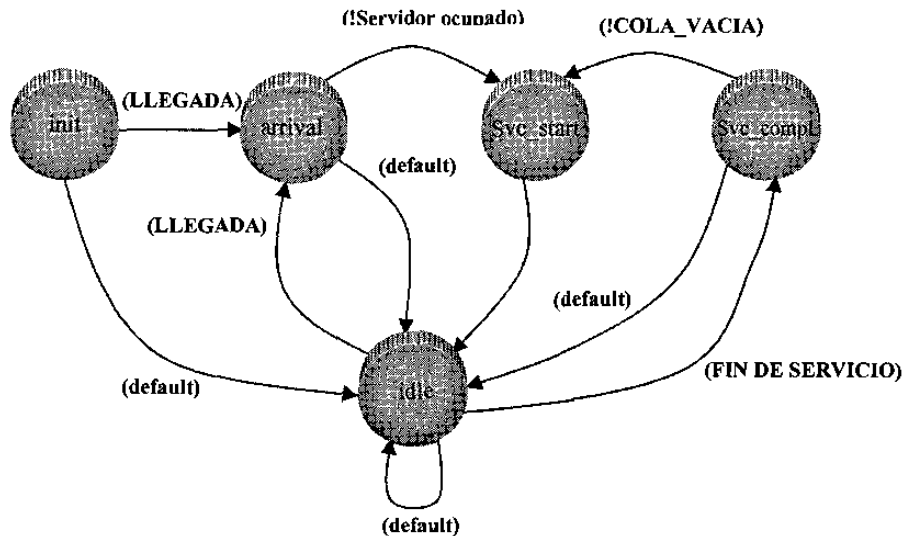


Figura 24. Modelo de Procesos WFQ

La manera de cómo funciona el sistema en cada uno de estos estados será descrita en forma general, con la ayuda de diagramas de flujo.

Estado inicio (init):

En este estado se leen los valores de los pesos (ϕ 's) correspondientes a cada una de las clases. El programa supone que los valores de las ϕ 's no están normalizados (no suman 1), así que se normalizan dividiendo cada valor entre la suma de todos; si todos los valores leídos son iguales a cero, se igualarán a 1 antes de hacer la normalización. En este mismo estado se inicializan en cero los contadores de los paquetes transmitidos, paquetes destruidos, retardo promedio y la variación del retardo (jitter). Si durante la inicialización

de variables llegó algún paquete, se realiza una transición al estado de arribo, de lo contrario se realiza una transición al estado inactivo. En la figura 25 se muestra lo anterior.

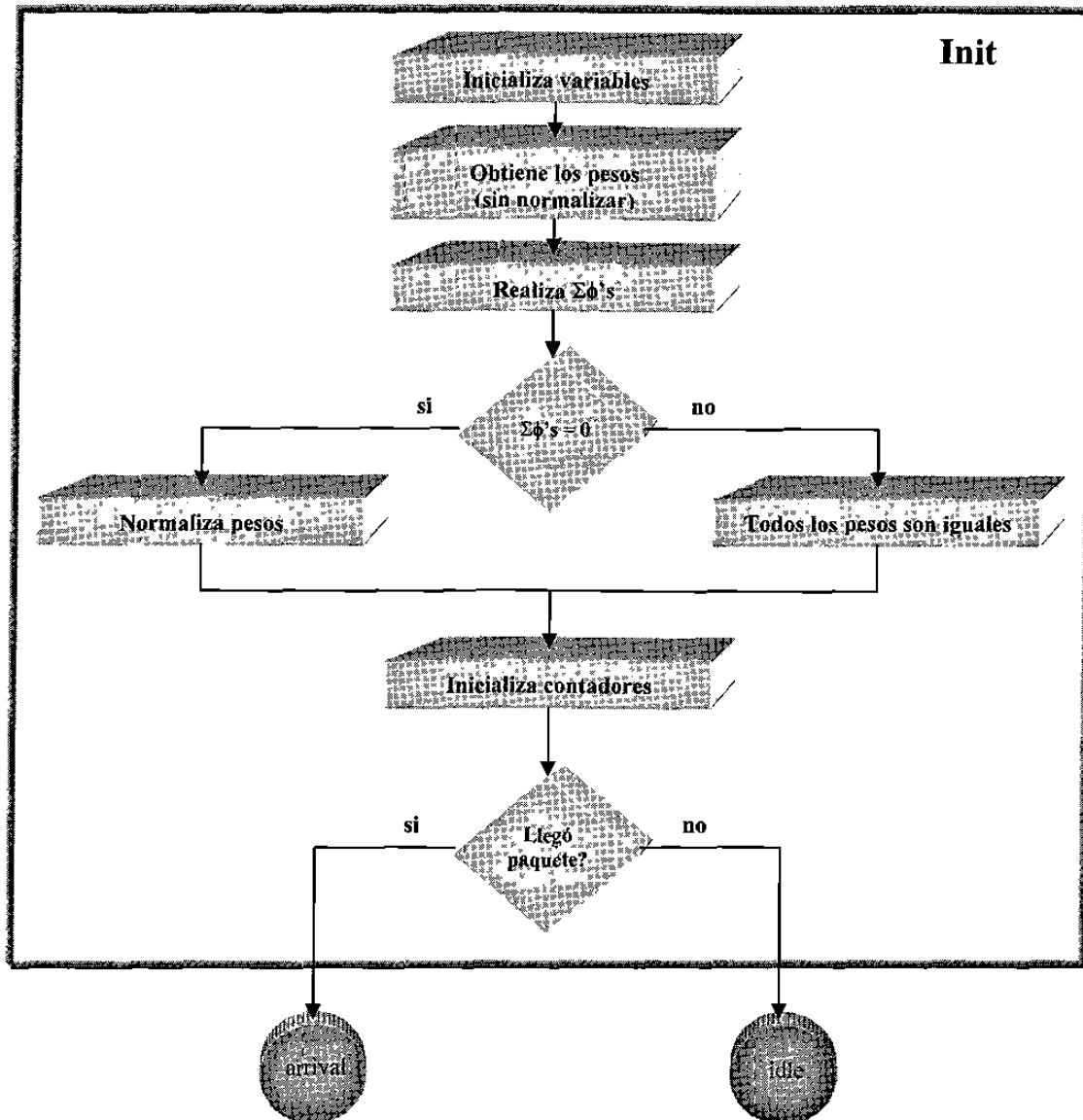


Figura 25. Diagrama de flujo del estado de inicio

Estado arribo (arrival):

El estado de arribo se encarga de recibir los paquetes que llegan de las diferentes clases, con su respectivo color. Se obtiene la información del tiempo en que llegó el paquete y se realiza el cálculo del tiempo virtual de llegada del paquete ($V(a_i^k)$) (ver sección III.4.3). Posteriormente, se identifica la cola donde le corresponde al paquete ser almacenado, determinado por la clase del paquete y su color. Las ecuaciones para calcular la cola destino para el modelo WFQ es la siguiente:

$$\text{identificador de cola} = \#\text{colores} * i + j \quad (7)$$

donde: i identifica la clase del paquete (0,3)

j identifica el color del paquete (0,2)

Si no hay espacio disponible en la cola que le corresponde, el sistema intentará guardarlo en las colas que correspondan a paquetes con prioridad más alta de descarte (a menos que se trate de un paquete rojo). En caso de que las otras colas también estén llenas, el sistema buscará algún paquete de más alta prioridad de descarte que esté guardado en alguna de las colas; si lo encuentra, será removido de la cola y descartado para hacer campo para el paquete recién llegado. En caso de que las tres opciones fallen, el paquete será descartado. Si la inserción fue exitosa y el enlace de transmisión está disponible, el sistema pasará al estado de inicio de servicio (Svc_start), de lo contrario, pasará al estado inactivo (idle) en espera de un evento (otra llegada de paquete o fin de la transmisión del que está siendo atendido).

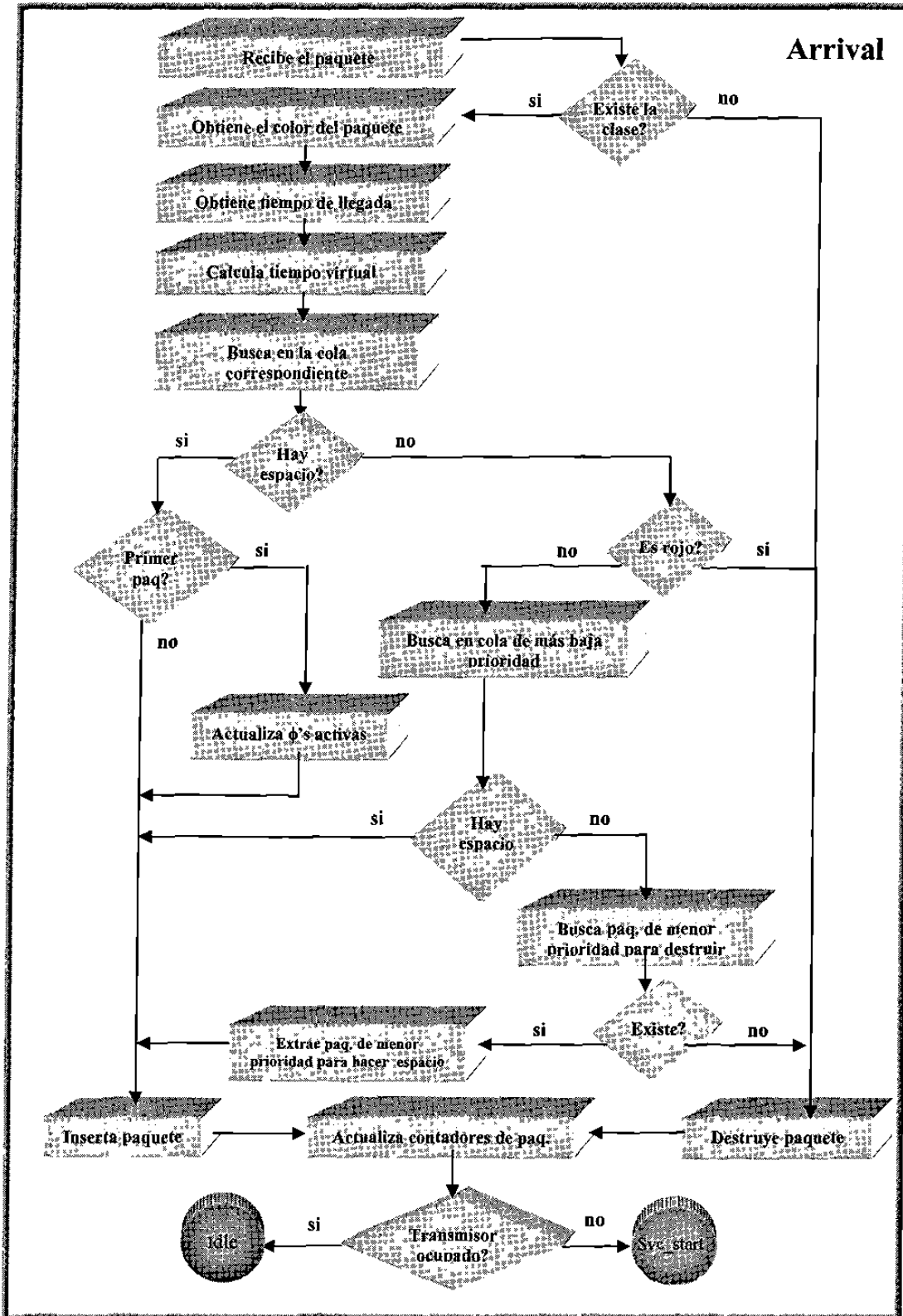


Figura 26. Diagrama de flujo del estado de arribo.

Estado inicio de servicio (Svc_Start)

Cuando el enlace de salida del enrutador está libre después de transmitir un paquete y hay más paquetes esperando en las colas para ser atendidos o cuando un nuevo paquete llega mientras el enlace está libre, el sistema entra al estado de inicio de servicio. En este estado, el sistema decide qué paquete será el siguiente en atenderse. Se hace un recorrido en las tres colas que corresponden a cada clase de tráfico que no estén vacías con el fin de obtener el paquete con el tiempo virtual de llegada más pequeño ($V(a_i^k)$) y se realiza para ellos el cálculo del tiempo virtual de terminación de servicio de ese paquete, denotado por F_i^k (ver sección III.4.3). El orden en que los paquetes se van a transmitir está determinado por este tiempo de terminación de servicio (F_i^k). De esta manera, se selecciona el paquete que tenga el F_i^k más pequeño para obtener un ganador.

Una vez que se obtiene un ganador, se guarda en memoria el identificador de la cola con el paquete que se enviará (cola en servicio). Posteriormente se determina el tamaño del paquete y se calcula el tiempo estimado para atender el paquete representado por $\frac{L_i^k}{r}$, donde r es la tasa de servicio del enrutador. Se programa en este momento una interrupción que sirve para simular el tiempo de transmisión del paquete. El servidor se marca como ocupado, tal como se muestra en la figura 27.

Es importante mencionar que el F_i^k de un paquete puede en principio ser calculado en el tiempo de llegada [Parekh A. y Gallager R., 1993], ya que toda la información

necesaria ($V(a_i^k)$ de los paquetes, el tamaño, la fracción de ancho de banda asignado ϕ , y el F_i^k del paquete que lo precede en la cola) ya está disponible. Sin embargo, el F_i^k del paquete dependerá de los F_i^k 's de todos los paquetes que lo preceden en la cola. Si uno de los paquetes (con excepción del último de la cola) es descartado, la secuencia se pierde y los valores del F_i^k de todos los paquetes que siguen son incorrectos. Así que, esta opción supone de manera implícita que una vez que un paquete entra a alguna de las colas, será atendido.

Considerando que en nuestro caso, los paquetes pueden ser removidos de una cola y destruidos, decidimos posponer el cálculo del F_i^k de un paquete hasta que llegue a la cabeza de la cola, y así estar enterados de cuántos paquetes han sido atendidos antes que él.

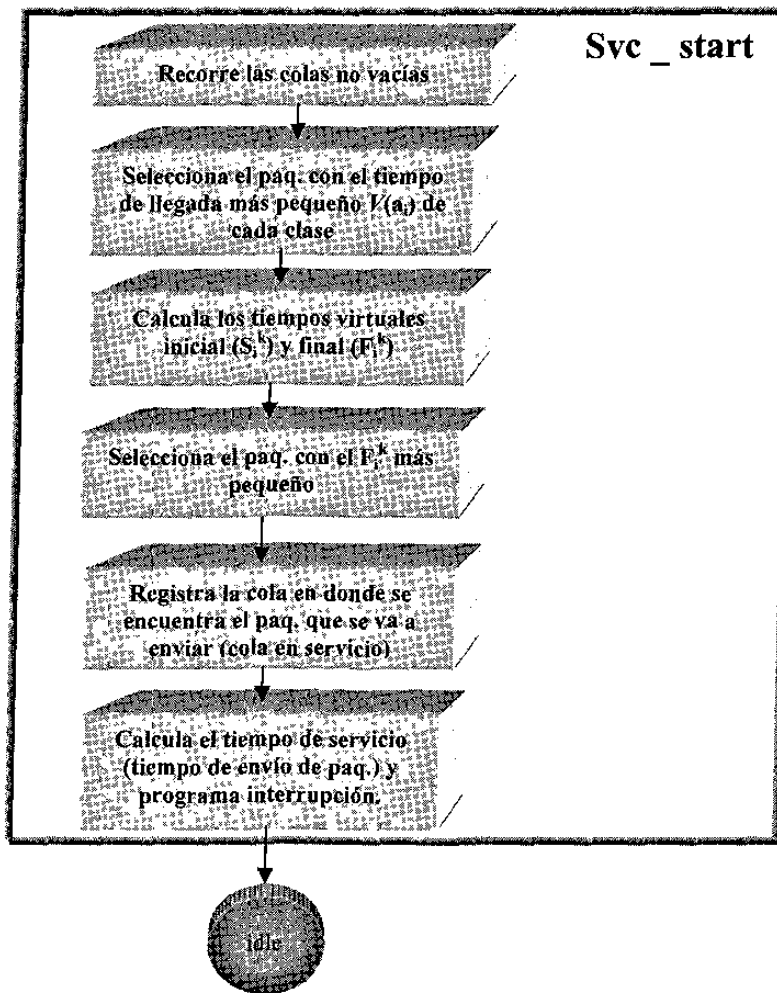


Figura 27. Diagrama de flujo del estado de inicio de servicio

Terminación de servicio (Svc _ completion)

El sistema entrará a este estado cuando ocurra la interrupción programada cuando se seleccionó el paquete que está siendo transmitido y que indica que el tiempo de transmisión ya ha transcurrido. En el diagrama de flujo de la figura 28, el sistema se encarga de extraer el paquete de la cola en servicio que fue seleccionada en el estado de inicio de servicio. Si el paquete es el último en la cola se calcula de nuevo la suma de los

ϕ 's de las clases que están ocupadas (que sirve para calcular los tiempos virtuales como se describe en la sección III.4.3). Registra el tiempo virtual de fin de servicio (F_i^k) de este paquete, que servirá para calcular el tiempo virtual de fin de servicio del próximo paquete por transmitir de esta misma clase. Se actualizan los contadores y se envía el paquete. Nuevamente el servidor se encuentra disponible. Si todavía hay paquetes en las colas esperando servicio, entonces se va al estado de inicio de servicio para solucionar el nuevo paquete por transmitir, de lo contrario pasará al estado inactivo a esperar la llegada de nuevos paquetes.

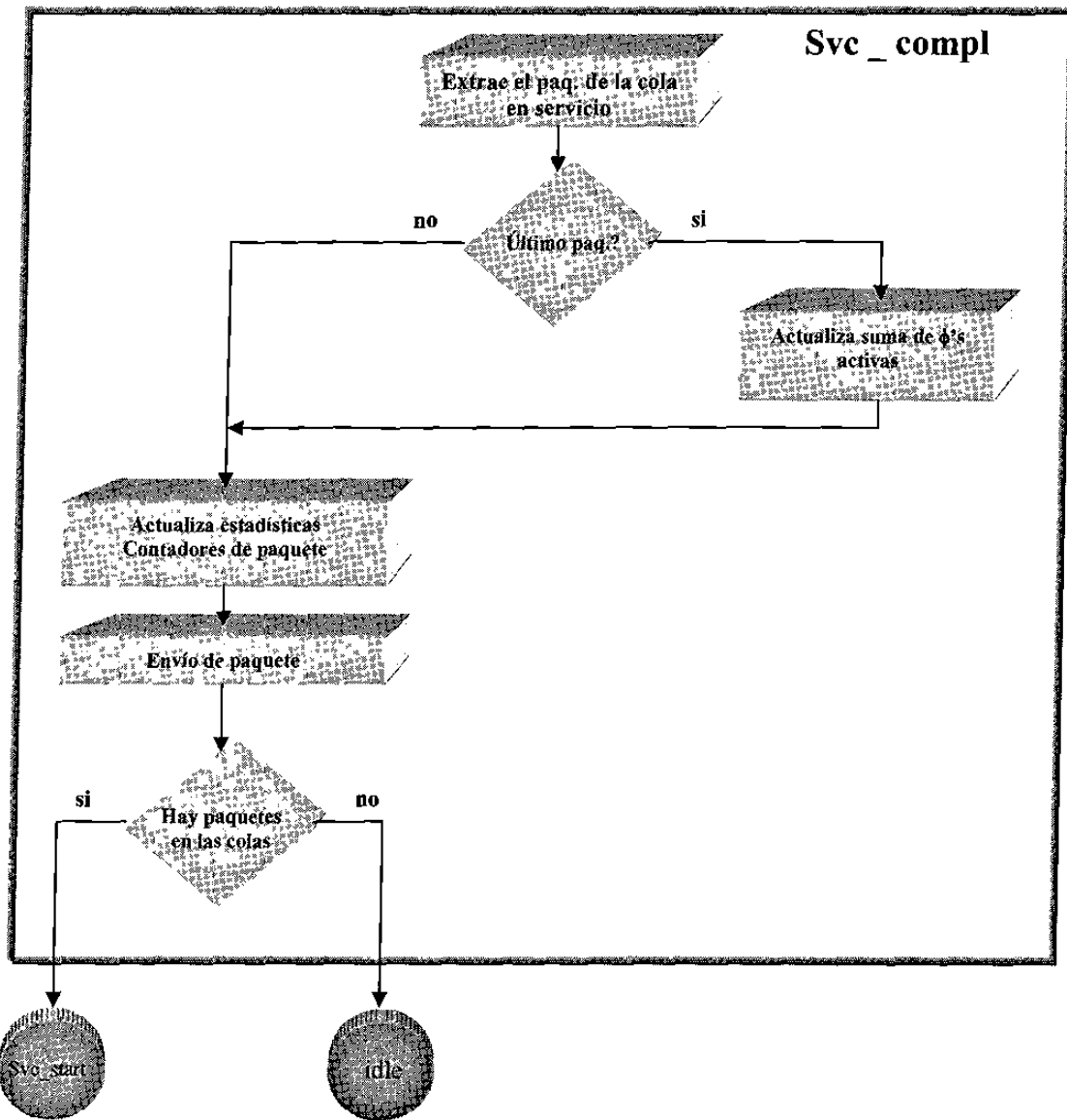


Figura 28: Diagrama de flujo del estado de terminación de servicio.

Estado Inactivo (idle)

En este estado el sistema se encuentra en espera de eventos para actuar de manera acorde. Los eventos se anuncian a través de interrupciones. Si se trata de la llegada de un nuevo paquete, el sistema pasará al estado de arriba. Si se trata de un aviso de que terminó

la transmisión de un paquete, pasará al estado de terminación de servicio (Svc_compl.). Finalmente, si se recibe una interrupción indicando el fin de la simulación, guardará en un archivo todas las estadísticas colectadas (paquetes transmitidos, pérdidas, retardo y variaciones del retardo). El diagrama de flujo se muestra en la figura 29.

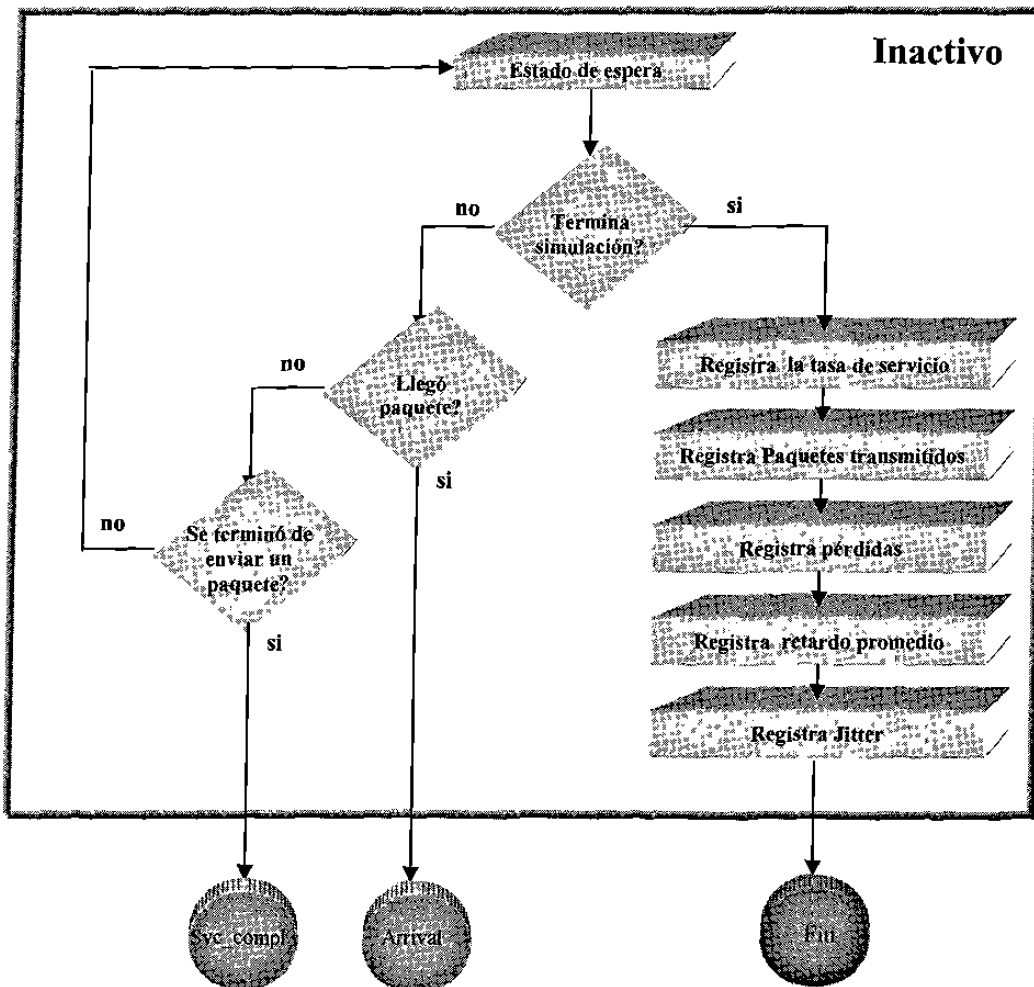


Figura 29: Diagrama de flujo del estado inactivo.

IV.5 MODELO DE PROCESOS CH2WFQ

El modelo de procesos de la nueva propuesta CH2WFQ tiene los mismos estados y condiciones de transición que el WFQ mostrado en la figura 24, pero las funciones realizadas en cada estado son diferentes, como se explica a continuación:

El modelo CH2WFQ maneja dos etapas de competencia, similar a lo que se hace de manera distribuida en el modelo WFQ jerárquico tradicional. Por tal motivo, CH2WFQ debe manejar tiempos virtuales independientes para cada bloque de competencia de la primera etapa (donde compiten los paquetes que se generan dentro de una misma compañía) y un tiempo virtual adicional para la competencia de los paquetes en la segunda etapa (donde compiten los paquetes provenientes de diferentes compañías). Un paquete podrá entrar a la competencia en la segunda etapa sólo cuando sea el ganador de la primera etapa.

En el estado de *inicio*, se obtiene el valor de las ϕ 's de las clases y de las superclases. Se realiza de nuevo la normalización de los pesos pero en esta ocasión se lleva a cabo dentro de cada uno de los bloques de competencia, de manera similar a lo que se hace en el modelo tradicional dentro de cada bloque WFQ individual.

En el estado de *arribo*, CH2WFQ calcula el tiempo virtual, de llegada de cada paquete usando el reloj virtual correspondiente al bloque de la competencia al que pertenece el paquete, el cual depende de la compañía de donde llega. Posteriormente, el

paquete es almacenado en la cola apropiada o destruido si no hay espacio disponible, como se describió para WFQ. El cálculo de la cola destino para CH2WFQ es el siguiente:

$$\text{identificador de cola} = \#clases * \#colores * k + (i * \#colores) + j \quad (8)$$

donde: k es el número de la compañía en que se generó (0,3)

i es el número de la clase a la que corresponde (0, 3)

j es el color del paquete (0, 2) (ver tabla I)

Note que el tiempo virtual de llegada del paquete correspondiente al reloj virtual de la segunda etapa no es calculado. Más adelante explicaremos la razón.

Las condiciones para entrar al estado *svc_start* en CH2WFQ son las mismas que las descritas para WFQ (Cuando el enlace de salida del enrutador está libre después de transmitir un paquete y hay más paquetes esperando para ser atendidos o cuando un nuevo paquete llega mientras el enlace está libre). La decisión para especificar cuál es el siguiente paquete que se va a transmitir se toma en dos etapas: en la primera etapa, se realiza la competencia entre los paquetes que llegan de la misma compañía, similar a lo que se hace para WFQ, pero esta vez tenemos que usar el reloj apropiado para calcular los tiempos virtuales pertinentes.

Una vez que tengamos un ganador en cada bloque de competencia en la primera etapa, calculamos su F_i^k correspondiente al reloj virtual de la segunda etapa y el paquete que se transmite será el que tenga el F_i^k más pequeño. Sin embargo, hay una diferencia en

la forma en que el F_i^k es calculado en la segunda etapa, comparado con la primera etapa, en el sentido de que el tiempo virtual de llegada no se toma en cuenta.

Es decir, en lugar de usar las ecuaciones (3) y (4) de la sección III.4.3. La expresión que se usa en la segunda etapa es la siguiente:

$$F_i^k = F_i^{k-1} + \frac{L_i^k}{\phi_i \cdot r} \quad (9)$$

Decidimos hacer esto porque definir el tiempo virtual de llegada en la segunda etapa no es sencilla. Se consideraron dos opciones para calcular el $V(a_i^k)$ de los paquetes en la segunda etapa de nuestro modelo:

- a) El tiempo virtual en el reloj de la segunda etapa correspondiente al tiempo en que llegó el paquete al sistema.
- b) El tiempo virtual en el reloj de la segunda etapa correspondiente al tiempo en que el paquete es un ganador en la primera etapa y está en posibilidad de entrar en la competencia en la segunda etapa.

Usar la opción (a) para el tiempo virtual de llegada de un paquete en CH2WFQ es equivalente a usar en el WFQ de la segunda etapa del modelo tradicional, el tiempo virtual correspondiente al tiempo en que el paquete entró al WFQ de la primera etapa, lo cual no tiene sentido, ya que el paquete no está aún en posibilidad de entrar a la competencia en la segunda etapa.

La opción (b) sería equivalente a usar en WFQ el tiempo virtual de llegada correspondiente al tiempo en el cual el paquete llega a la cabeza de la cola donde ha estado esperando a ser atendido desde su arribo, lo cual tampoco es congruente con la analogía que se trata de hacer con GPS.

Ambas opciones anteriores provocan que el algoritmo falle a la hora de repartir el ancho de banda a las diferentes clases. Decidimos, como una tercera opción, no tomar en cuenta el $V(a_i^k)$ en la segunda etapa y el problema se resolvió.

Con esta decisión, por cierto, no se violan los acuerdos de nivel de servicio (SLA), ya que el propósito de usar $V(a_i^k)$ es el de conocer en qué momento la clase de tráfico habría estado inactiva en un sistema GPS para distribuir su porción del ancho de banda entre los que sí estuvieran activos. No usar $V(a_i^k)$ es equivalente a decir que, cuando una clase de tráfico está activa, lo ha estado sin pausas desde la última vez que se envió un paquete de esa clase. Esto, por supuesto, no da a las otras fuentes de tráfico competir por el ancho de banda extra que habrían recibido en GPS, pero tampoco les quita lo que les toca.

Después de actualizar las estadísticas, se programa una interrupción al final del estado *svc_start* para simular el tiempo de transmisión del paquete. Cuando ocurra esta interrupción, el sistema entrará al estado *svc_compl*, de manera similar a lo que se hace en WFQ.

CAPÍTULO V. SIMULACIÓN Y ANÁLISIS DE RESULTADOS

V.1 SIMULACIÓN

Una simulación en OPNET opera bajo una secuencia de estados para un sistema determinado. El modelo evoluciona a través de estos estados en función del tiempo, basándose en el comportamiento de los componentes del modelo y de sus interacciones. La noción del tiempo en una simulación no está directamente relacionada con el tiempo que toma para correr tal simulación. Las simulaciones en OPNET están basados en eventos discretos, donde el progreso del modelo en términos del tiempo de simulación está dado por puntos individuales, en donde puede ocurrir un cambio. El término dado por OPNET para estos puntos es “eventos”. OPNET permite que ocurran múltiples eventos simultáneamente, lo que nos da una idea de que el tiempo de ejecución de una tarea puede ser cero.

Para llevar a cabo el proceso de simulación del sistema implementado a través de OPNET es necesario establecer los parámetros necesarios para su ejecución.

V.2 EVALUACIÓN

Para llevar a cabo la evaluación, se consideraron dos casos extremos para la configuración del modelo jerárquico tradicional. La diferencia entre estos casos es la velocidad de los enlaces que conectan los bloques WFQ de la primera etapa con el

bloque WFQ de la segunda etapa. Por comodidad llamaremos lo anterior como: *la velocidad de los enlaces internos*. En el primer caso, la velocidad de los enlaces internos es igual a la del enlace de salida del enrutador (caso R), mientras que en el segundo caso, la velocidad es igual al ancho de banda asignado a la compañía a la que corresponde cada enlace (caso $R \cdot \phi$).

Las simulaciones incluidas en este trabajo se diseñaron de tal manera que se hicieran evidentes los problemas que se describen enseguida y que se dan en el modelo jerárquico tradicional.

- Dar un tratamiento excesivamente bueno a los usuarios mal portados (caso R).
- Tener pérdidas innecesarias en la segunda etapa (caso R).
- Tener pérdidas innecesarias en la primera etapa y desperdicio de ancho de banda del enlace de salida del enrutador (caso $R \cdot \phi$).

Después de varias simulaciones de prueba, se seleccionan los parámetros que se describen enseguida.

V.2.1 PARÁMETROS DE SIMULACIÓN

La realidad del enlace de salida del enrutador, que se denota por R , se supone que es de 3 Mbps.

Con la intención de que una de las ϕ 's del WFQ de la segunda etapa en el modelo tradicional fuera suficientemente pequeña, se le asignó en los modelos jerárquicos de las figuras 22 y 23, a cada una de las compañías cuatro veces más ancho de banda que a la compañía que le antecede. Además, a cada fuente de tráfico dentro de cada compañía se le asigna el doble del ancho de banda que a la fuente de tráfico que le antecede. El criterio para esta asignación fue simplemente el de darles valores diferentes a las ϕ 's. La fracción de ancho de banda asignado a la compañía 0 (ϕ_0) se calculó resolviendo la siguiente ecuación:

$$\phi_0 + 4\cdot\phi_0 + 16\cdot\phi_0 + 64\phi_0 = 1 \quad (10)$$

Note que $\phi_0 = 1/85$

La fracción de ancho de banda asignado a la fuente de tráfico 0 (ϕ_{F0}) dentro de la compañía 0 se calculó resolviendo la siguiente ecuación:

$$\phi_{F0} + 2\cdot\phi_{F0} + 4\cdot\phi_{F0} + 8\phi_{F0} = 1 \quad (11)$$

De tal manera que la fracción de ancho de banda total asignado a la fuente de tráfico 0 está dada por $\phi_0 \cdot \phi_{F0}$. Las demás fracciones se calcularon de manera similar. Esta información se resume en la tabla II.

La manera en que se generó el tráfico fue usando fuentes compatibles con el modelo descrito en [Barford P., 1998] para fuentes de tráfico WWW, el cual por generar tráfico en ráfagas, acentúa los problemas relacionados con el modelo jerárquico tradicional WFQ (ver

sección V.2). Los marcadores usados son compatibles con la especificación TSWTCM (II.3.3.4.4.1). El tamaño de la ventana para todos los marcadores es de 100 milisegundos, compatible con los valores sugeridos en [Wenja F. *et al*, 1999]. Los parámetros restantes son incluidos en la tabla II. Note que la tasa pico es 20% mayor que la tasa garantizada en todos los casos.

TABLA II. PARÁMETROS UTILIZADOS EN LA SIMULACIÓN.

	Número de Fuentes De Tráfico	Fracción de Ancho de Banda Asignado (ϕ)		Tasa Garantizada TSW – TCM	Tasa Pico TSW – TCM
		Por fuente	Por Compañía		
Compañía 0	0	0.000784	0.011765	2.5 Kbps	3.0 Kbps
	1	0.001569		5.0 Kbps	6.0 Kbps
	2	0.003137		10.0 Kbps	12.0 Kbps
	3	0.006275		20.0 Kbps	24.0 Kbps
Compañía 1	4	0.003137	0.047059	10.0 Kbps	12.0 Kbps
	5	0.006275		20.0 Kbps	24.0 Kbps
	6	0.012548		40.0 Kbps	48.0 Kbps
	7	0.025098		80.0 Kbps	96.0 Kbps
Compañía 2	8	0.012549	0.188235	40.0 Kbps	48.0 Kbps
	9	0.025098		80.0 Kbps	96.0 Kbps
	10	0.050196		160.0 Kbps	192.0 Kbps
	11	0.100392		320.0 Kbps	384.0 Kbps
Compañía 3	12	0.050196	0.752941	160.0 Kbps	192.0 Kbps
	13	0.100392		320.0 Kbps	384.0 Kbps
	14	0.200784		640.0 Kbps	768.0 Kbps
	15	0.401569		1280.0 Kbps	1536.0 Kbps

V.2.2 PARÁMETROS DE DESEMPEÑO

Para determinar el desempeño de ambos modelos tomamos en cuenta los siguientes parámetros:

- Caudal Eficaz. Nos indica qué tan eficiente es la red, en función de la cantidad de información transmitida y recibida con éxito.
- Pérdidas. Se refiere a los paquetes que son destruidos por no haber capacidad en el búfer de la clase correspondiente.
- Retardo Promedio. Se define como el tiempo promedio transcurrido desde que el paquete entra al enrutador hasta que sale de él.
- Variación de retardo (Jitter). Es la variación del retardo de los paquetes dentro del enrutador. Se define como la desviación estándar del retardo experimentado. Es una característica inevitable de los enrutadores y otros equipos de comunicación, ya que el número de paquetes que llegan es aleatorio.

V.2.3 RESULTADOS DE LA SIMULACIÓN

Todas las simulaciones tienen una duración de 400 segundos, que es tiempo suficiente para que se generen más de un millón de paquetes, lo que se considera adecuado para que las estadísticas se estabilicen.

Cada simulación se corrió tres veces y los valores reportados son los promedios obtenidos de las tres simulaciones. Se manejaron dos conjuntos de experimentos, donde las condiciones en cada uno de éstos experimentos es diferente.

A continuación se muestran, a través de gráficas, los resultados obtenidos del conjunto de experimentos de la implementación y simulación de los modelos jerárquicos tradicional y CH2WFQ.

Con la intención de mejorar la resolución de las gráficas se incluyen solamente los resultados correspondientes a las 12 ranuras fuentes de tráfico (de la 0 a la 11). El eje X en todas las gráficas es el número que identifica a cada fuente de tráfico.

V.2.3.1 PRIMER CONJUNTO DE EXPERIMENTOS

En el primer conjunto de experimentos, casi todas las fuentes de tráfico están generando paquetes de acuerdo al perfil esperado (ver tabla I), excepto la fuente de tráfico 0, que está enviando tráfico 300 veces más rápido que lo esperado, y las fuentes de tráfico 13 y 14, que no están enviando prácticamente nada de tráfico. El ancho de banda que no es usado por las fuentes de tráfico 13 y 14 será redistribuido entre todos los demás usuarios.

En la figura 30 se muestra que el caso *R* del modelo jerárquico tradicional tiene el problema descrito en (i), en el sentido de que no protege a los usuarios con buen comportamiento en la primera compañía (fuentes de tráfico 1, 2 y 3) de los usuarios mal portados (fuente de tráfico 0). El número de bits transmitidos de las fuentes de tráfico 1, 2, y 3 en el caso *R* está muy por debajo de lo que se había prometido; de hecho es aproximadamente la mitad de lo que CH2WFQ transmite.

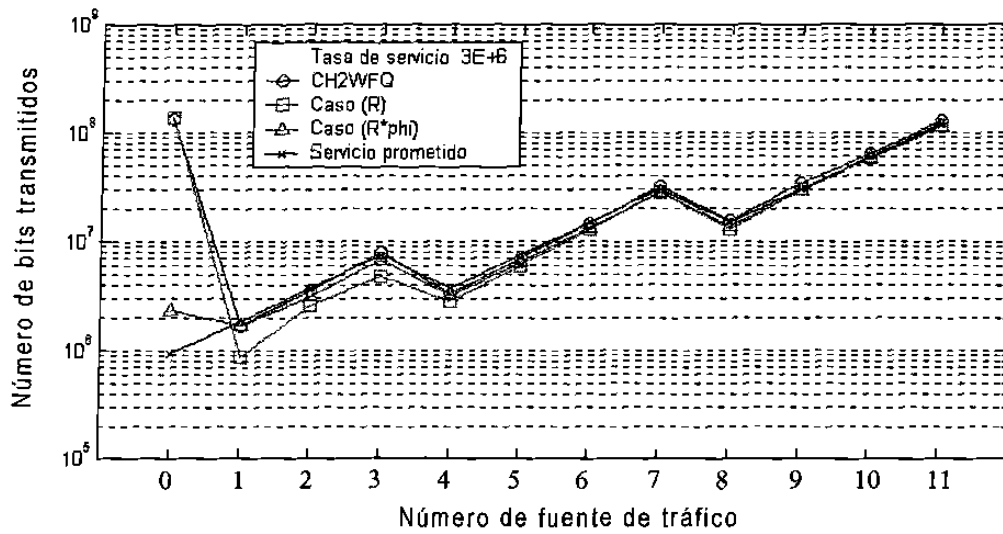


Figura 30. Bits transmitidos en el primer conjunto de simulaciones.

La razón de este problema en el enfoque tradicional se describe a continuación:

- Si la velocidad de salida de la primera etapa fuera infinita, las colas trabajarían en forma de FIFO ya que los paquetes serían enviados conforme fueran llegando, debido a que el tiempo de transmisión sería cero.
- Cuando la velocidad de salida de la primera etapa es finita pero muy grande, el efecto anterior empieza a ser notorio y la diferenciación empieza a perder su eficacia.

En la figura 31 se puede observar que los usuarios con buen comportamiento (fuentes de tráfico 1,2, y 3) tienen muchas más pérdidas (bits destruidos) en el caso *R* que en CH2WFQ.

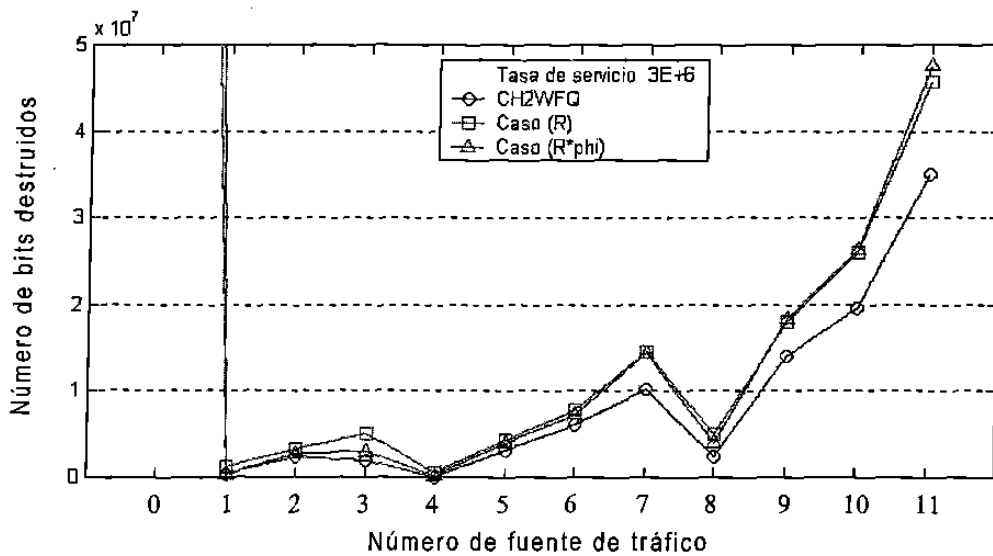


Figura 31. Bits destruidos en el primer conjunto de simulaciones.

Note que el número de pérdidas es mayor en el caso R que en CH2WFQ también para las otras fuentes de tráfico. Además, estas pérdidas innecesarias suceden en la segunda etapa (problema descrito en (ii)) debido a que gran parte del tiempo la primera etapa envía paquetes mucho más rápido que lo que la segunda etapa puede atender.

En las gráficas presentadas en las figuras 32 y 33 se puede observar que el modelo CH2WFQ experimenta un retardo y un jitter más grande que el que se presenta en el caso R . El pequeño retardo presentado en el caso R se debe a la gran velocidad con que trabajan los enlaces internos del modelo jerárquico tradicional, de tal manera que los paquetes que no se pierden llegan muy rápido a la segunda etapa de competencia.

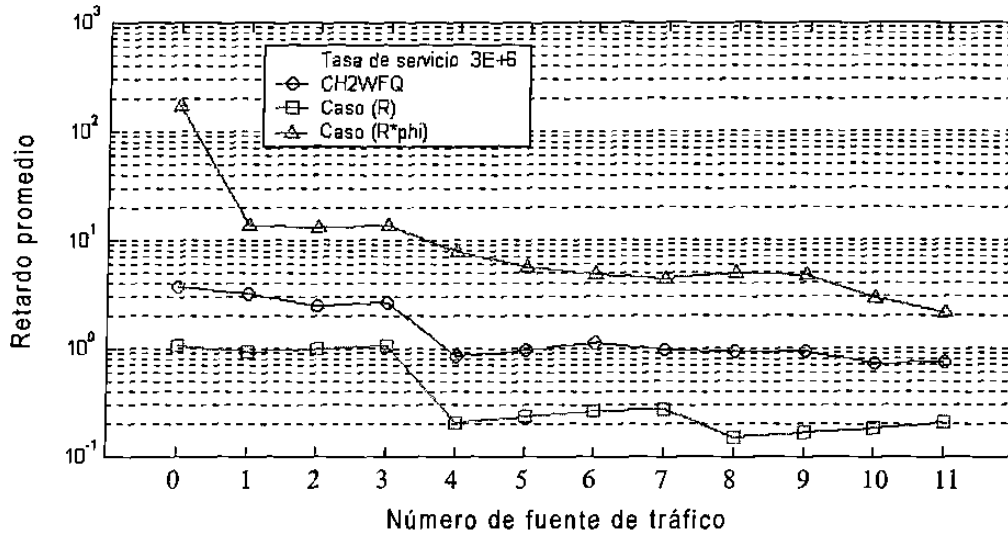


Figura 32. Retardo promedio del paquete en el primer conjunto de simulaciones.

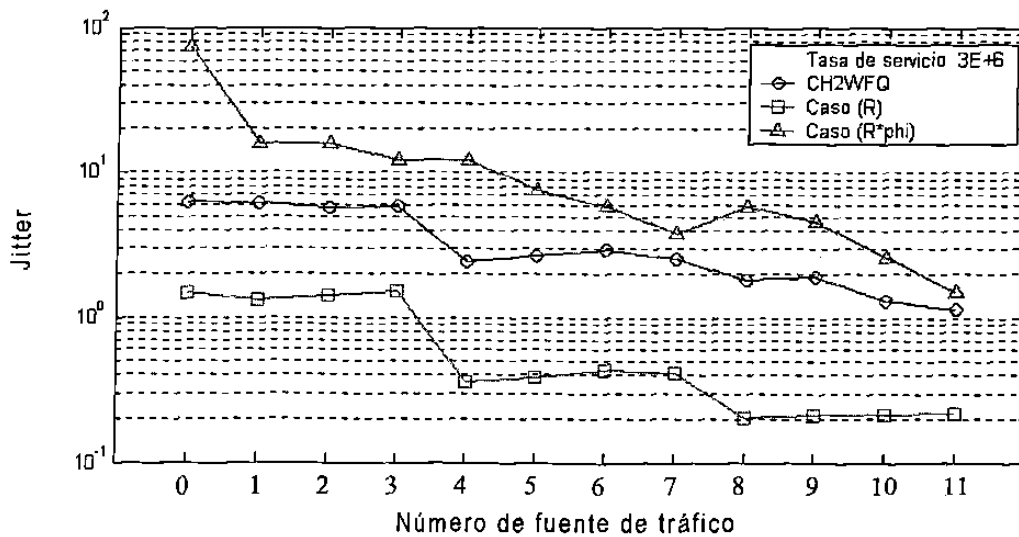


Figura 33. Jitter del paquete en el primer conjunto de simulaciones.

V.2.3.2 SEGUNDO CONJUNTO DE EXPERIMENTOS

En el segundo conjunto de experimentos, de nuevo casi todas las fuentes de tráfico están enviando paquetes de acuerdo al perfil esperado, excepto las fuentes de tráfico de la

12 a la 15, las cuales no están generando prácticamente nada de tráfico. Por lo anterior, se puede ver que ahora el ancho de banda disponible para repartir entre las fuentes de tráfico activas es mucho más grande en comparación con el primer conjunto de experimentos. Sin embargo, el caso $R\cdot\phi$ del modelo jerárquico tradicional no es capaz de aprovechar el ancho de banda extra para enviar más paquetes y, por lo tanto, permanece muy cerca del nivel del servicio prometido, tal como se muestra en la figura 34.

El motivo de que se genere esta deficiencia es que en el caso $R\cdot\phi$ los paquetes salen del bloque que le corresponde en la primera etapa a la velocidad que su ancho de banda asignada les permite y nunca sobrepasan ese límite, aún cuando el enlace de salida de la segunda etapa este siendo desperdiciado.

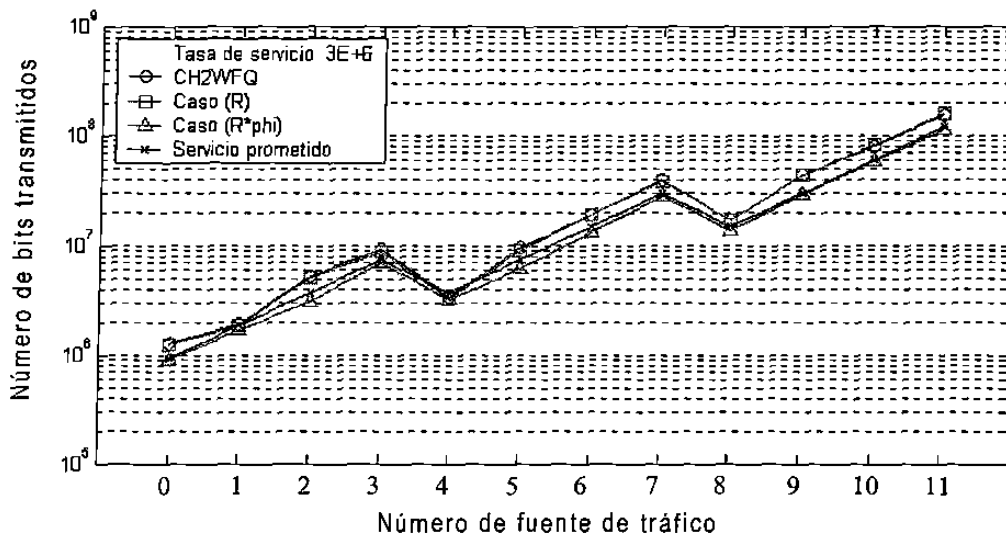


Figura 34. Número de bits transmitidos en el segundo conjunto de simulaciones.

En la figura 35 se puede observar el efecto que tiene el hecho de que el sistema no sea capaz de aprovechar el ancho de banda extra, en el caso $R\text{-}\phi$, sobre el número de pérdidas (bits destruidos). Como los paquetes toman mucho tiempo para dejar la primera etapa, muchos de los nuevos paquetes que llegan tienen que ser destruidos debido a la falta de espacio en las colas para poder almacenarlos. Este es el problema descrito en (iii).

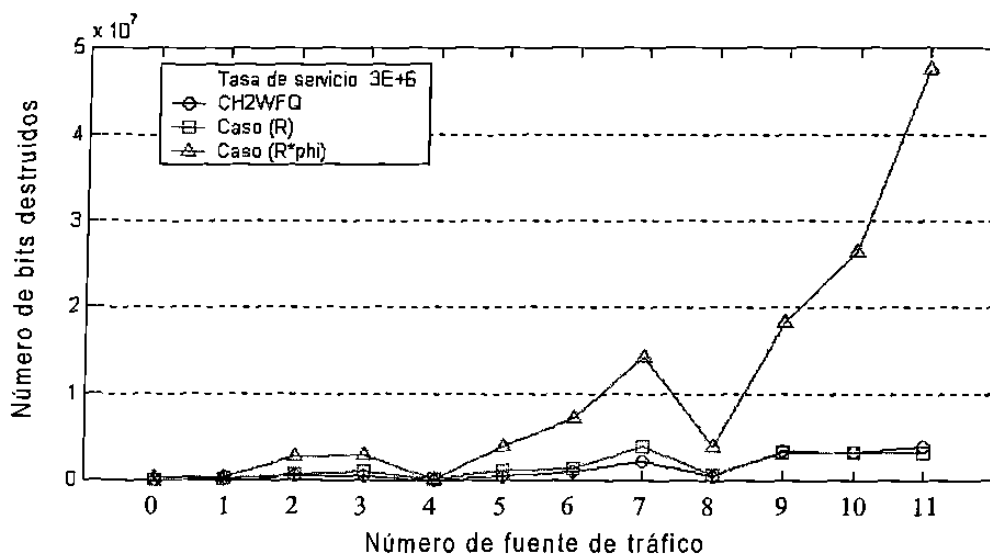


Figura 35. Número de bits destruidos en el segundo conjunto de simulaciones.

El tiempo que se necesita para desocupar las colas es bastante; por tanto, el retardo y el jitter son mucho más grandes en este caso que en el caso R y en el modelo CH2WFQ, tal como se muestra en las figuras 36 y 37.

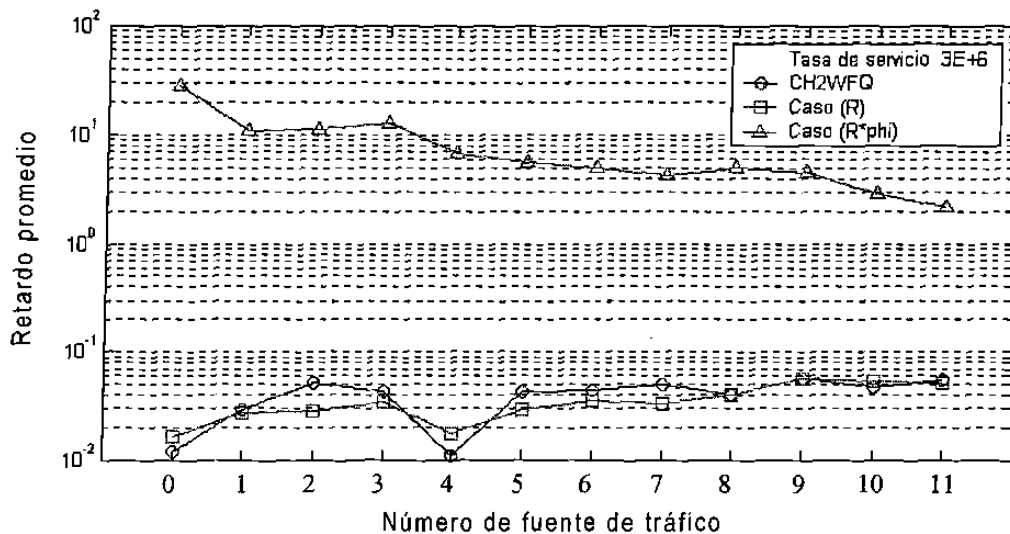


Figura 36. Retardo promedio del paquete en el segundo conjunto de simulaciones.

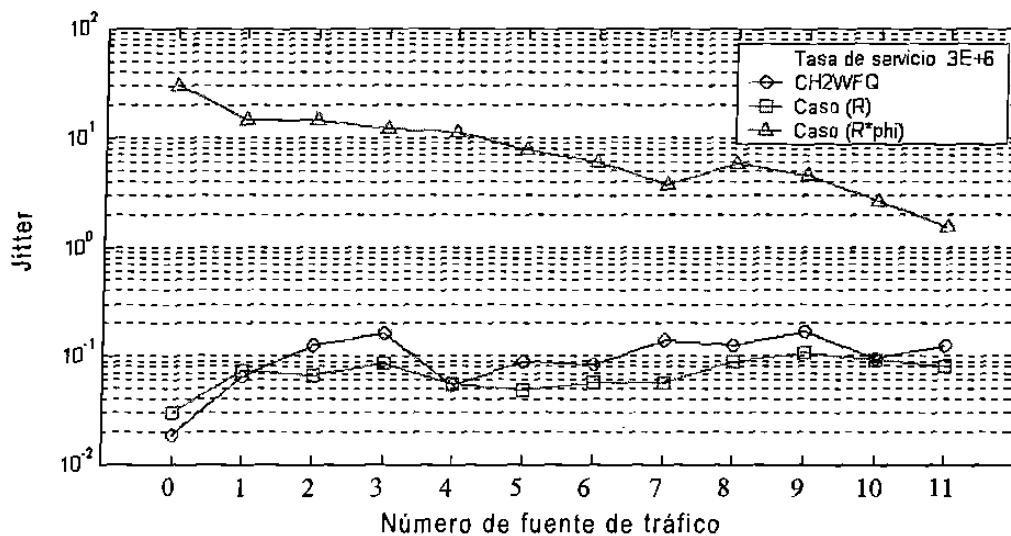


Figura 37. Jitter del paquete en el segundo conjunto de simulaciones.

Al comparar los parámetros del retardo y el jitter para el modelo CH2WFQ y el caso R en las figuras 36 y 37, podemos ver que la mayoría de estos valores son otra vez más grandes en CH2WFQ que en el caso R, debido a la gran velocidad con que trabajan los enlaces internos del modelo jerárquico tradicional.

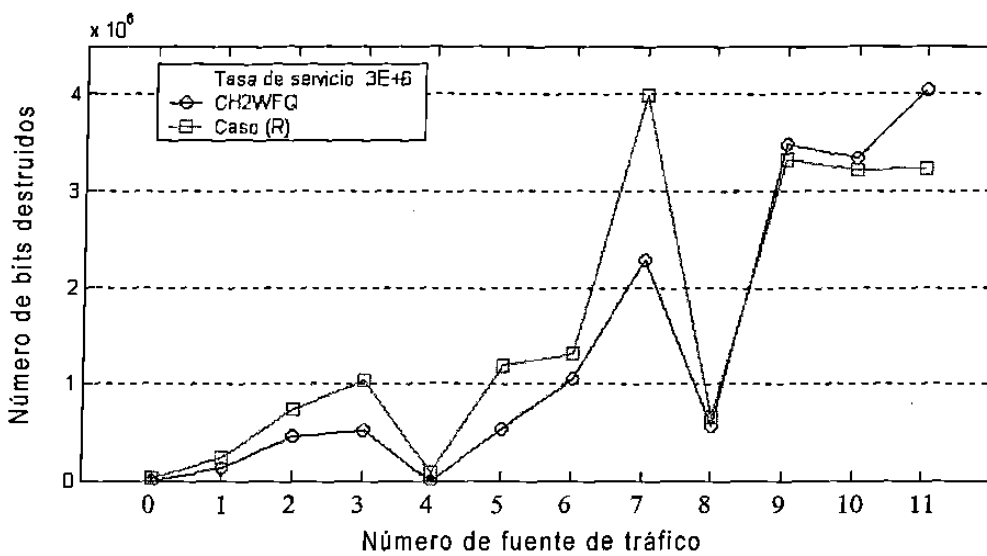


Figura 38. Número de bits destruidos en el segundo conjunto de simulaciones.

En la figura 38 se muestra con más detalle la diferencia entre CH2WFQ y el caso R con respecto al número de paquetes destruidos. Como puede verse, el número de pérdidas para la mayoría de las fuentes de tráfico es considerablemente más grande en el caso R , lo cual significa que otra vez existen pérdidas innecesarias en la segunda etapa, de tal forma que el problema (ii) también se presenta bajo éstas condiciones.

Después de analizar los resultados anteriores, lo que permite tener una visión más clara del funcionamiento de los tres enfoques simulados, vale la pena aclarar que se seleccionaron los casos R y $R\cdot\phi$ del modelo tradicional por las siguientes razones:

- Usar una velocidad menor que $R\cdot\phi$ en los enlaces internos, acentuaría el problema (iii).

- Usar una velocidad de los enlaces internos mayor que R , acentuaría los problemas *(i)* y *(ii)*.
- Manejar una velocidad entre los valores $R \cdot \phi$ y R para los enlaces internos, no garantiza que los problemas vayan a desaparecer, sino que por el contrario introduciría una combinación de todos los problemas juntos.

CAPÍTULO VI CONCLUSIONES Y TRABAJOS FUTUROS

VI.1 CONCLUSIONES

En el desarrollo de esta tesis se ha visto la importancia de ofrecer QoS en las redes, que ha sido posible con la incorporación de los diferentes mecanismos de control de tráfico y de la administración de recursos en los dispositivos enrutadores y conmutadores comerciales. Dicho sea de paso estas técnicas de administración de recursos también se usan en los principales sistemas operativos, tales como Windows NT/2000, Linux, FreeBSD, lo que aumenta la importancia de mejorar su eficiencia.

Al inicio de este trabajo de tesis se planteó que en la actualidad una de las propuestas para hacer que la Internet sea capaz de atender a los usuarios corriendo aplicaciones con diferentes requerimientos de calidad de servicio, son los Servicios Diferenciados (DiffServ) propuestos por la IETF, los cuales necesitan de algoritmos eficientes y justos para manejar los recursos de la red (colas y ancho de banda), siendo uno de sus principales pilares el calendarizador de Encolamiento Justo Ponderado (WFQ), al menos en un sentido genérico ya que existen algunas variantes.

La motivación de este trabajo es el hecho de que al implementar una estrategia para la distribución jerárquica de recursos usando controladores WFQ de una sola etapa y organizándolos en una estructura jerárquica, se introducen pérdidas y retardos innecesarios, y lo que es más, se produce injusticia bajo ciertas condiciones. Con base en este argumento,

se propuso e implementó un sistema jerárquico de dos niveles, en donde hay un solo controlador que decide de manera centralizada cómo asignar los recursos. Este controlador toma las decisiones basadas en la información completa relacionada con la ocupación de las diferentes colas y la disponibilidad del enlace de salida del enrutador.

Se desarrollaron los programas para llevar a la práctica el algoritmo propuesto y poderlo comparar con el modelo tradicional, que como se mencionó anteriormente consiste de cinco bloques conectados en cascada.

En este trabajo se describen los detalles de la implementación del modelo tradicional WFQ y del modelo CH2WFQ, así como los problemas presentados en la misma. Uno de los problemas clave del modelo tradicional, y que lo hace ineficiente con respecto al modelo propuesto, es cómo especificar la velocidad de los enlaces internos.

Con los resultados obtenidos mediante las simulaciones realizadas a través del programa OPNET, fue posible concluir con claridad que CH2WFQ tiene las siguientes ventajas en comparación con el modelo jerárquico tradicional:

- Se eliminan las pérdidas de paquetes y el desperdicio de ancho de banda debidos a la falta de sincronización entre la primera y la segunda etapa del modelo tradicional. Cuando los enlaces internos son lentos, las pérdidas en el modelo tradicional ocurren en la primera etapa, mientras que cuando los enlaces son muy rápidos, las pérdidas ocurren en la segunda etapa.

- Se eliminan los retardos excesivos que resultarían si se seleccionara una velocidad muy pequeña para los enlaces internos.
- Se eliminan las injusticias que se introducirían si se seleccionara una velocidad muy grande para los enlaces internos.

Lo anterior se refleja en un desempeño superior de CH2WFQ en términos de paquetes transmitidos y destruidos. El precio que hay que pagar es un incremento en el retardo promedio y jitter de los paquetes.

Es importante mencionar que, a pesar de que este trabajo se limitó a un modelo jerárquico de dos etapas, las conclusiones obtenidas son también válidas para un sistema más general con n niveles (más de 2) en la jerarquía, (el cual se llamaría CH n WFQ) y para aquellos sistemas que usen alguna variante de WFQ. Se desarrollaron los programas para manejar HWFQ de dos etapas, sin embargo la implementación puede ser fácilmente extendida para incluir más etapas.

V1.2 TRABAJOS FUTUROS

En este trabajo se mostraron los beneficios del algoritmo propuesto (CH2WFQ), en comparación con el enfoque tradicional, de una manera genérica. Es decir, no se usaron valores reales (que se usan en la práctica) para las velocidades de los enlaces de transmisión, para los tamaños de las colas ni para las tasas de generación de tráfico en las fuentes. Además, se usó el mismo comportamiento estadístico para modelar el tráfico

correspondiente a todas las clases de servicio. Por último, no se incluyó el manejo de protocolos, tales como TCP, que también contribuyen a modificar la manera en que se envía el tráfico a las redes.

Una posible extensión de este trabajo sería precisamente el medir los beneficios del modelo propuesto en un ambiente más apegado al que se da en las redes reales de telecomunicaciones.

Una extensión más sería incluir algoritmos no ideales de descarte de paquetes, tales como RED (Random Early Detection) [Floyd Sally y Jacobson Van, 1993], DRED (Dinamic -Random Early Detection) [Aweya J. *et al*, 2002], BLUE [Feng Wu – Chang *et al*, 2001] y tantos otros que existen en la literatura.

Y, por supuesto, una extensión interesante también sería el desarrollar los programas para manejar un sistema jerárquico de n etapas, donde n sea un número entero positivo arbitrario. Creemos que los lineamientos más importantes para evitar complicaciones a la hora de desarrollar tales programas se encuentran plasmados en esta tesis.

