

**VERIFICACIÓN DEL MODELO MATEMÁTICO DE UN ROUTER DE SERVICIOS
INTEGRADOS**

**CARLOS ANTONIO ACEVEDO SOLANO
FABIÁN ARMANDO ARANGO FONTECHA**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
SECCIONAL BUCARAMANGA
ESCUELA DE INGENIERÍAS Y ADMINISTRACIÓN
INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2008**

**VERIFICACIÓN DEL MODELO MATEMÁTICO DE UN ROUTER DE SERVICIOS
INTEGRADOS**

**CARLOS ANTONIO ACEVEDO SOLANO
FABIÁN ARMANDO ARANGO FONTECHA**

**TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARA OPTAR
AL TÍTULO DE INGENIERO ELECTRÓNICO**

**Director
Ph.D. JHON JAIRO PADILLA AGUILAR**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
SECCIONAL BUCARAMANGA
ESCUELA DE INGENIERÍAS Y ADMINISTRACIÓN
INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2008**

Nota de aceptación:

Presidente del jurado

Jurado

Jurado

Ciudad y fecha:

A Dios y a nuestros padres, por habernos dado la oportunidad de estudiar una carrera universitaria, así como por su apoyo y comprensión durante todo este proceso de cumplir uno de los objetivos de nuestros proyectos de vida.

AGRADECIMIENTOS

A **Ph.D. Jhon Jairo Padilla Aguilar**, como docente y como amigo, por su colaboración y consejos que fueron esenciales para el desarrollo de esta investigación, de igual forma agradecemos a todo el personal administrativo, docentes, y empleados de la Universidad Pontificia Bolivariana por brindarnos su cálido apoyo y haber ayudado en nuestra formación académica y personal.

También a todos los que sin ningún compromiso colaboraron dedicando parte de su tiempo y conocimientos.

OBJETIVOS

Objetivo general

Comprobar la exactitud del modelo matemático de un router de servicios integrados comparando dos tecnologías para el soporte de Calidad de Servicio en redes IPv6: *IntServ* e *IntServ6*.

Objetivos específicos

- Desarrollar y probar los modelos de simulación del proceso de tráfico IP para dos modelos: *POISSON* y *D-MMPP* [2].
- Reutilizar y probar el modelo de simulación del proceso de control de admisión para dos métodos diferentes: *New Call Bounding Scheme* y *Cutoff Priority Scheme* desarrollados por Yuguang Fang en [10].
- Desarrollar y probar el modelo de simulación del proceso de clasificación para dos tipos de tecnología: *IntServ* e *IntServ6*.
- Desarrollar y probar el modelo de simulación del proceso de planificación para dos tipos de planificadores: *PGPS* y *Priority Queuing*.
- Integrar los modelos obtenidos para obtener el modelo de simulación completo del router de servicios integrados.
- Comprobar diferentes combinaciones de los módulos para simular el desempeño de diferentes tipos de routers.

- Verificar la exactitud del modelo matemático de un router de servicios integrados simulado en [8], comparando las dos tecnologías de QoS especificadas.

TABLA DE CONTENIDO

	Pág.
PARTE 1 MARCO TEÓRICO	22
1. INTRODUCCIÓN A LA CALIDAD DE SERVICIO	23
1.1 EVOLUCIÓN DE LA CALIDAD DE SERVICIO EN INTERNET	23
1.2 INTRODUCCIÓN A LOS SERVICIOS INTEGRADOS	25
1.3 INTRODUCCIÓN A LOS SERVICIOS DIFERENCIADOS	28
2. SERVICIOS INTEGRADOS	31
2.1 APLICACIONES DE TIEMPO REAL	32
2.2 USO DEL BUFFER SUAVIZADO	35
2.3 TRÁFICO MULTICAST MBONE	36
2.3.1 IP Multicast	37
2.4 ARQUITECTURA DE SERVICIOS INTEGRADOS	39
2.4.1 Aproximación Básica	39
2.4.2 Modelo de referencia	40
2.4.3 Funcionamiento del plano de control y del plano de Datos	41
2.5 MODELOS DE SERVICIO	44
2.5.1 Servicio garantizado	44
2.5.2 Servicio de carga controlada	45
2.5.3 Protocolo de reservación de recursos (resource reservations setup – RSVP)	47
2.5.4 Control de reserva	48
2.6 CLASIFICACIÓN DE PAQUETES CON INTSERV	50
2.6.1 El proceso de Clasificación de flujos (Flow Identification)	50
2.7 CLASIFICACIÓN DE PAQUETES CON INTSERV6	52
2.8 MODELO DEL CLASIFICADOR	52
2.9 PLANIFICACIÓN DE PAQUETES	55
2.9.1 Priority Queuing (PQ)	56

2.9.2 GPS, PGPS (WFQ) Y PFQ	57
2.10 MODELO DEL ROUTER DE ACCESO INALÁBRINCO	60
2.10.1 Modelos de control de acceso	61
3. TEORÍA DE COLAS	64
3.1 MODELOS DE TRÁFICO	64
3.1.1 Modelo Poisson	64
3.1.2 MMPP (Harkov Modulated Poisson Process)	65
3.2 AUTOSIMILITUD	67
3.2.1 Dependencia a corto plazo y largo plazo	67
3.2.2 Estimación del parámetro de Hurst	68
PARTE 2 DESARROLLO DE LA PROPUESTA	71
4. DESCRIPCIÓN DE LA METODOLOGÍA	72
4.1 INTRODUCCIÓN	72
5. DESCRIPCIÓN DE LOS PROGRAMAS	74
5.1 INTERFAZ DE USUARIO	74
5.1.1 Módulo de generación de paquetes (tráfico de entrada)	75
5.1.2 Módulo de control de Admisión de Llamadas (CAC)	75
5.1.3 Módulo de Clasificación de Paquetes	75
5.1.4 Módulo de Planificación de Paquetes	76
5.2 DESCRIPCIÓN GENERAL DEL SISTEMA	76
5.3 PROCESO DE TRÁFICO DE PAQUETES	78
5.3.1 Modelo POISSON	78
5.3.2 Modelo D-MMPP	82
5.3.3 Estimación del parámetro de Hurst	90
5.4 ESQUEMA DE CONTROL DE ADMISIÓN DE LLAMADAS CAC (CALL ADMISIÓN CONTROL)	93
5.4.1 Etapa de control de admisión de llamadas utilizando el modelo Ancho de Banda Fijo	93
5.4.2 Etapa de control de admisión de llamadas utilizando el modelo de ancho de banda aleatorio	96

5.4.3 Etapa de control de admisión de llamadas utilizando el esquema de New Call Bounding	99
5.4.4 Etapa de control de admisión de llamadas utilizando el esquema Cutoff Priority	104
5.4.5 Generación de tabla de colisiones	109
5.5 ETAPA DE CLASIFICACIÓN DE PAQUETES	113
5.5.1 Clasificadores IntServ e IntServ6	114
5.6 ETAPA DE PLANIFICACIÓN DE PAQUETES	118
5.6.1 Planificación de paquetes con el esquema PQ (Priority Queuing)	119
5.6.2 Planificación de paquetes con el esquema PGPS (Packet-by-packet Generalized Processor Sharing)	124
5.7 ETAPA DE INTEGRACIÓN DE SUBPROGRAMAS	129
5.7.1 Programa para simulación con número de reservas fijo	129
5.7.2 Programa para simulación con número de reservas incremental	134
5.7.3 Programa para simulación con router inalámbrico	137
PARTE 3 RESULTADOS Y CONCLUSIONES	143
6. SIMULACIONES Y ANÁLISIS DE RESULTADOS	144
6.1 RESULTADOS OBTENIDOS PARA TIEMPO DE RETARDO EN CLASIFICACIÓN	144
6.2 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE CLASIFICACIÓN PARA UNA SIMULACIÓN DE RESERV INCREMENTAL	146
6.3 RESULTADOS OBTENIDOS PARA TIEMPO DE RETARDO EN PLANIFICACIÓN	148
6.4 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE PLANIFICACIÓN PARA UNA SIMULACIÓN DE RESERV INCREMENTAL	153
6.5 RESULTADOS OBTENIDOS DE ANCHO DE BANDA USADO EN PLANIFICACIÓN	155
6.6 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE PLANIFICACIÓN PARA UNA SIMULACIÓN DE RESERV FIJA	157

6.7 RESULTADOS OBTENIDOS PARA TIEMPO DE RETARDO EN CLASIFICACIÓN	158
6.8 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE CLASIFICACIÓN PARA UNA SIMULACIÓN DE RESERV FIJA	160
6.9 COMPROBACIÓN DE FUNCIONAMIENTO DEL MODELO DE TRÁFICO D-MMPP	161
6.10 VERIFICACIÓN DEL MODELO MATEMÁTICO	170
6.11 COMPROBACIÓN DE FUNCIONAMIENTO DEL MODELO DE TRÁFICO D-MMPP	171
6.12 VERIFICACIÓN DEL MODELO MATEMÁTICO	171
7. CONCLUSIONES Y RECOMENDACIONES	173
GLOSARIO	177
BIBLIOGRAFÍA	181
ANEXOS	185

LISTA DE FIGURAS

- Figura 1.** Distribución del retardo y variación del retardo
- Figura 2.** Uso del Buffer suavizador
- Figura 3.** Realización de una reserva de recursos
- Figura 4.** Modelo de referencia de *IntServ*
- Figura 5.** Procedimiento de establecimiento de la Reserva en *IntServ6*
- Figura 6.** Proceso de Identificación de flujo basado en Hashing en *IntServ*
- Figura 7.** Diagrama de transición de estados del proceso de clasificación de paquetes de un Router *IntServ* e *IntServ6*.
- Figura 8.** Esquema gráfico (*PQ*)
- Figura 9.** Esquema gráfico (*PGPS*)
- Figura 10.** Modelo de un router de acceso inalámbrico
- Figura 11.** Distribución Poisson
- Figura 12.** Modelo *MMPP* dos estados
- Figura 13.** Proceso de diseño de un modelo
- Figura 14.** Grafica varianza-tiempo para la estimación del parámetro *H*.
- Figura 15.** Módulo de la Interfaz Grafica del Router
- Figura 16.** Diagrama de flujo general del Simulador
- Figura 17.** Diagrama de flujo *POISSON*
- Figura 18.** Diagrama de estado *MMPP*
- Figura 19.** Diagrama de flujo del Módulo de *D-MMPP*
- Figura 20.** Programa estimación del Parámetro *H*
- Figura 21.** Diagrama de flujo del Módulo Ancho de Banda Fijo
- Figura 22.** Diagrama de flujo del Módulo Ancho de Banda Aleatorio
- Figura 23.** Diagrama de flujo del Módulo de *New Call Bounding Scheme*
- Figura 24.** Diagrama de flujo del Módulo de *Cutoff Priority Scheme*
- Figura 25.** Diagrama de flujo del Módulo general de tabla de colisiones

- Figura 26.** Diagrama de flujo del Módulo Clasificación
- Figura 27.** Diagrama de flujo del Módulo de *PQ*
- Figura 28.** Diagrama de flujo del Módulo *PGPS*
- Figura 29.** Diagrama de flujo del Simulador de Reservas Fijo
- Figura 30.** Diagrama de flujo del Simulador de Reservas Incrementales
- Figura 31.** Diagrama de flujo del Simulador router inalámbrico
- Figura 32.** Tiempo promedio de retardo de *Clasificación1*
- Figura 33.** Tiempo promedio de retardo de *Clasificación2*
- Figura 34.** Tiempo promedio de retardo de *Clasificación3*
- Figura 35.** Tiempo promedio de retardo de *Clasificación4*
- Figura 36.** Tiempo promedio de retardo de *PQ1*
- Figura 37.** Acercamiento a la gráfica Tiempo promedio de retardo de *PQ1*
- Figura 38.** Tiempo promedio de retardo de *PQ2*
- Figura 39.** Tiempo promedio de retardo de *PGPS1*
- Figura 40.** Tiempo promedio de retardo de *PGPS2*
- Figura 41.** Tiempo promedio de retardo de *PQ3*
- Figura 42.** Tiempo promedio de retardo de *PQ4*
- Figura 43.** Tiempo promedio de retardo de *PGPS3*
- Figura 44.** Tiempo promedio de retardo de *PGPS4*
- Figura 45.** Anchos de banda de planificación *PGPS1*
- Figura 46.** Anchos de banda de planificación *PGPS2*
- Figura 47.** Anchos de banda de planificación *PQ1*
- Figura 48.** Anchos de banda de planificación *PQ2*
- Figura 49.** Retardos de paquetes en clasificación para router fijo *IntServ* haciendo uso de *Poisson*
- Figura 50.** Retardos de paquetes en clasificación para router fijo *IntServ6* haciendo uso de *Poisson*
- Figura 51.** Retardos de paquetes en clasificación para router fijo *IntServ* haciendo uso de *D-MMPP*

- Figura 52.** Retardos de paquetes en clasificación para router fijo *IntServ6* haciendo uso de *D-MMPP*
- Figura 53.** Retardos de paquetes en clasificación para router inalámbrico *IntServ New Call Bounding Scheme* y variación de nuevas llamadas usando *Poisson*
- Figura 54.** Retardos de paquetes en clasificación para router inalámbrico *IntServ6 New Call Bounding Scheme* y variación de nuevas llamadas usando *Poisson*
- Figura 55.** Retardos de paquetes en clasificación para router inalámbrico *IntServ New Call Bounding Scheme* y variación de nuevas llamadas usando *D-MMPP*
- Figura 56.** Retardos de paquetes en clasificación para router inalámbrico *IntServ6 New Call Bounding Scheme* y variación de nuevas llamadas usando *D-MMPP*
- Figura 57.** Retardos de paquetes en clasificación para router inalámbrico *IntServ New Call Bounding Scheme* y variación de traspaso usando *Poisson*
- Figura 58.** Retardos de paquetes en clasificación para router inalámbrico *IntServ6 New Call Bounding Scheme* y variación de traspaso usando *Poisson*
- Figura 59.** Retardos de paquetes en clasificación para router inalámbrico *IntServ New Call Bounding Scheme* y variación de traspaso usando *D-MMPP*
- Figura 60.** Retardos de paquetes en clasificación para router inalámbrico *IntServ6 New Call Bounding Scheme* y variación de traspaso usando *D-MMPP*
- Figura 61.** Retardos de paquetes en clasificación para router inalámbrico *IntServ Cutoff Priority Scheme* y variación de nueva llamada usando *Poisson*
- Figura 62.** Retardos de paquetes en clasificación para router inalámbrico *IntServ6 Cutoff Priority Scheme* y variación de nueva llamada usando *Poisson*
- Figura 63.** Retardos de paquetes en clasificación para router inalámbrico *IntServ Cutoff Priority Scheme* y variación de nueva llamada usando *D-MMPP*
- Figura 64.** Retardos de paquetes en clasificación para router inalámbrico *IntServ6 Cutoff Priority Scheme* y variación de nueva llamada usando *D-MMPP*

- Figura 65.** Retardos de paquetes en clasificación para router inalámbrico IntServ *Cutoff Priority Scheme* y variación de traspaso usando *Poisson*
- Figura 66.** Retardos de paquetes en clasificación para router inalámbrico IntServ6 *Cutoff Priority Scheme* y variación de traspaso usando *Poisson*
- Figura 67.** Retardos de paquetes en clasificación para router inalámbrico IntServ *Cutoff Priority Scheme* y variación de traspaso usando *D-MMPP*
- Figura 68.** Retardos de paquetes en clasificación para router inalámbrico IntServ6 *Cutoff Priority Scheme* y variación de traspaso usando *D-MMPP*
- Figura 69.** Comprobación del modelo de tráfico D-MMPP
- Figura 70.** Retardo medio de clasificación IntServ e IntServ6 del simulador y el modelo matemático

LISTA DE TABLAS

- Tabla1.** Parámetros de Simulación del procesador de red *IXP 1200*
- Tabla2.** Variable entrada-salida de Poisson
- Tabla3.** Variable de salida-entrada *D-MMPP*
- Tabla4.** Variables entrada-salida del parámetro *H*
- Tabla5.** Variable entrada-salida Bw Fijo
- Tabla6.** Variable entrada-salida Bw Aleatorio
- Tabla7.** Matriz de salida admit para los esquemas *New Call Bounding* y *Cutoff Priority*
- Tabla8.** Variable de entrada-salida de los esquemas *New Call Bounding* y *Cutoff Priority*
- Tabla9.** Variable entrada-salida de la tabla de reservas de colisiones
- Tabla10.** Variables de entrada-salida del Clasificador *IntServ* e *IntServ6*
- Tabla11.** Variables de entra-salida del esquema *PQ*
- Tabla12.** Esquema de prioridades *PQ*
- Tabla13.** Variables de entra-salida del esquema Simulador
- Tabla14.** Variables de entrada-salida del esquema *Simulador del router inalámbrico*

LISTA DE ANEXOS

- Anexo A.** Artículos tesis
- Anexo B.** Manual simulador
- Anexo C.** CD Simulador MATLAB y documentación en formato digital de la tesis

RESUMEN

En este documento se plasma una contribución a la solución para soportar calidad de servicio en Internet (QoS), llamada *verificación del modelo matemático de un router de servicios integrados*, se realizó con el fin de crear un simulador de un router de tráfico en Internet, Que incluya diferentes mecanismos para el soporte de QoS, tales como controles de admisión, procesos de la clasificación y planificación de paquetes.

Este proyecto hace parte de la línea de investigación del proyecto "*Evaluación de IntServ6, una propuesta de Calidad del Servicio en Redes IPv6*" apoyada por la Dirección General de Investigaciones de la UPB Bucaramanga y contribuye al proyecto *IRHMA* (Interconexión de Redes Heterogéneas Malladas Autoconfigurables), que es apoyado por la *CICYT* (Comisión Interministerial de Ciencia y Tecnología de España) a través de la consolidación de la propuesta *IntServ6*, desarrollada en la tesis de doctorado del profesor Jhon Jairo Padilla Aguilar "*Contribución al soporte de calidad de servicio en redes móviles*" en la Universidad Politécnica de Cataluña – España

El documento esta dividido en tres partes, la primera comprende el marco teórico donde se explica los temas relacionados de la investigación. La segunda parte hace referencia a la metodología del proyecto donde se explica en que consiste y que proceso se lleva a cabo en la investigación mostrando la descripción general de cada programa, y una tercera parte que son los resultados y conclusiones.

La tesis consta de 7 capítulos y 3 anexos que este último comprende el artículo de la tesis (Anexo A), Manual Simulador (Anexo B), CD Simulador MATLAB y documento en formato digital de la tesis (Anexo C).

ABSTRACT

This document reflects a contribution to the solution to support quality of service on the Internet, called “Verificación del modelo matemático de un router de servicios integrados”, was conducted in order to create a simulator of a router using the simulation software MATLAB, which includes several mechanisms to support quality of service, such as admission controls, processes of classification and packages scheduling.

This project is part of the research project “Evaluación de IntServ6, una propuesta de Calidad del Servicio en Redes IPv6” supported by the General Direction for Research at the University Pontificia Bolivariana sectional Bucaramanga and contributes to the project "Interconexión de Redes Heterogéneas Malladas Autoconfigurables", which is supported by the "Comisión Interministerial de Ciencia y Tecnología de España" through the consolidation of the IntServ6 proposal, which was developed in the doctoral thesis of Professor Jhon Jairo Padilla Aguilar “Contribución al soporte de calidad de servicio en redes móviles” at the Universidad Politécnica de Cataluña -Spain.

The document is divided into three sections: the first covers the theoretical framework that explains the issues related to the investigation. The second part refers to the methodology of the project, which explains it and describe which process is carried out in the investigation by showing the outline of each program, and a third part contains the results and conclusions.

The thesis consists of 7 chapters and 3 annexes which include the thesis article (Annex A), Manual Simulator (Annex B), CD with the simulator and enforceable document in a digital format of the thesis (Annex C).

INTRODUCCIÓN

En este proyecto grado se simulan las tecnologías IntServ6 desarrollada en [24] y la tecnología IntServ (*Servicios Integrados*) [5], haciendo uso del software de simulación MATLAB.

Esta simulación esta compuesta por varios módulos que al ser acoplados logran simular el comportamiento de un router de servicios integrados. En total son cuatro grandes módulos equivalentes a los tres procesos que se presentan en las arquitecturas de calidad de servicio *IntServ* e *IntServ6* y un *módulo de generación de trafico*. Los módulos correspondientes a las arquitecturas son: *módulo de Control de Admisión*, *el módulo de Clasificación* y *módulo de Planificación*. Dichos módulos tendrán como mínimo dos protocolos de funcionamiento cada uno, suministrando al usuario la posibilidad de seleccionar diferentes tipos configuraciones del router, creando así una base para futuras investigaciones en el campo de las comunicaciones.

De estas simulaciones se toma como resultado principal los tiempos de retardo promedio en el clasificador para cada tipo de arquitectura y se contrastan con los tiempos de retardo obtenidos en [8] del modelo matemático desarrollado en [24], determinando así la similitud entre el comportamiento obtenido a partir del modelo matemático [8] y el comportamiento que se observa en estas simulaciones; además se entregan resultados correspondientes a la etapa de planificación, que permiten tener una pauta de cómo afecta este proceso a la prestación de la calidad de servicio.

Con el fin de mejorar la finalidad didáctica de este proyecto, se desarrolló una interfaz gráfica de usuario que integra los procesos antes mencionados, permitiendo una selección de características del router de fácil acceso y

entregando los resultados en gráficas y tablas para el análisis del comportamiento del la calidad de servicio del router seleccionado.

PARTE 1 MARCO TEÓRICO

- 1. INTRODUCCIÓN A LA CALIDAD DE SERVICIO**
- 2. SERVICIOS INTEGRADOS**
- 3. TEORÍA DE COLAS**

1. INTRODUCCIÓN A LA CALIDAD DE SERVICIO

1.1 EVOLUCIÓN DE LA CALIDAD DE SERVICIO EN INTERNET

Los años 60's, cuando en los Estados Unidos, estaba buscando una forma de mantener las comunicaciones vitales del país en el posible caso de una Guerra Nuclear. Este hecho marcó profundamente su evolución, ya que aún ahora los rasgos fundamentales del proyecto se hallan presentes en lo que hoy conocemos como Internet [28].

En primer lugar, el proyecto contemplaba la eliminación de cualquier "autoridad central", ya que sería el primer blanco en caso de un ataque; en este sentido, se pensó en una red descentralizada y diseñada para operar en situaciones difíciles. Cada máquina conectada debería tener el mismo status y la misma capacidad para mandar y recibir información [28].

El envío de los datos debería descansar en un mecanismo que pudiera manejar la destrucción parcial de la Red. Se decidió entonces que los mensajes deberían de dividirse en pequeñas porciones de información o paquetes, los cuales contendrían la dirección de destino pero sin especificar una ruta determinada para su arribo; por el contrario, cada paquete buscaría la manera de llegar al destinatario por las rutas disponibles y el destinatario reensamblaría los paquetes individuales para reconstruir el mensaje original. La ruta que siguieran los paquetes no era importante; lo importante era que llegaran a su destino [28].

Curiosamente fue en Inglaterra donde se experimentó primero con estos conceptos; y así en 1968, el Laboratorio Nacional de Física de la Gran Bretaña estableció la primera red experimental. Al año siguiente, el Pentágono de los EU. decidió financiar

su propio proyecto, y en 1969 se establece la primera red en la Universidad de California (*UCLA*) y poco después aparecen tres redes adicionales. Nació así *ARPANET (Advanced Research Projects Agency NETWORK)*, antecedente de la actual Internet [28].

El Protocolo utilizado en ese entonces por las máquinas conectadas a *ARPANET* se llamaba *NCP (Network Control Protocol ó Protocolo de Control de Red)*, pero con el tiempo dio paso a un protocolo más sofisticado: TCP/IP, que de hecho está formado no por uno, sino por varios protocolos, siendo los más importantes: el protocolo TCP (*Transmission Control Protocol*) y el Protocolo IP (*Internet Protocol*).

En el transcurso de los años fue creciendo la red llamada *WWW (world wide web)*, y fueron naciendo nuevos retos para los científicos. Hoy en día, por ejemplo, para los tráficos con requerimientos de tiempo real (voz o video), es deseable que no ocurra pérdida de información, que exista un gran ancho de banda disponible, y que los retrasos en los envíos de estos paquetes de datos sean mínimos. Es por ello que surge la necesidad de aplicar *Calidad de Servicio (QoS)* en el nivel del transporte de datos y métodos de diferenciación de tráficos particulares con el fin de otorgar preferencia a estos datos sensibles [7].

Se entiende por “Calidad de Servicio”, a la capacidad de una red para sostener un comportamiento adecuado del tráfico que transita por ella, cumpliendo a su vez con los requerimientos de ciertos parámetros relevantes para el usuario final. Esto puede entenderse también como el cumplimiento de un conjunto de requisitos estipulados en un contrato (*SLA: Service Level Agreement*) entre un *ISP (Internet Service Provider, proveedor de servicios de Internet)* y sus clientes [7].

El protocolo de comunicación *IPv4 (Internet Protocol Version 4)* contiene especificaciones que permiten ejercer manipulaciones sobre estos paquetes, las cuales deben ser manejadas por los enrutadores al momento de implementar

QoS. Sin embargo, en los últimos años, se han estado perfeccionando detalles acerca de un nuevo estándar para el protocolo de Internet (*IP*), éste es llamado *IPv6 (Internet Protocol Version 6)*, el cual contiene nuevas y reestructuradas especificaciones para ejercer QoS [7].

Al contar con QoS, es posible asegurar una correcta entrega de la información necesaria o crítica, para ámbitos empresariales o institucionales, dando preferencia a aplicaciones de desempeño crítico, donde se comparten simultáneamente los recursos de red con otras aplicaciones no críticas. QoS hace la diferencia, al prometer un uso eficiente de los recursos ante la situación de congestión, seleccionando un tráfico específico de la red, priorizándolo según su importancia relativa, y utilizando métodos de control y evasión de congestión para darles un tratamiento preferencial. Implementando QoS en una red, hace al rendimiento de la red más predecible, y a la utilización de ancho de banda más eficiente [7].

1.2 INTRODUCCIÓN A LOS SERVICIOS INTEGRADOS [5]

Aunque los problemas con los modelos de *best-effort*, han sido reorganizados a lo largo del tiempo, la propuesta para la arquitectura de servicio garantizado surgió a principios de 1990 después de algunos experimentos de teleconferencia a gran escala en Internet. Las aplicaciones en tiempo real como video conferencias son sensibles a las líneas de tiempo de información y además no tienen un óptimo funcionamiento en el Internet, donde el retardo es típicamente impredecible. El riguroso retardo y la inquietud requerida en estas aplicaciones requieren un nuevo tipo de servicio el cual pueda proveer un nivel de seguridad de recursos a estas aplicaciones.

A principios de 1990 la *IETF (Internet Engineering Task Force)* inicio el grupo de trabajo de servicios integrados para estandarizar una nueva arquitectura de

asignación de recursos y nuevos modelos de servicio. En este tiempo el *World Wide Web*, como lo conocemos, aun no existía, y la conferencia multimedia era vista por mucha gente como una aplicación potencialmente destructora para el Internet. Así los requerimientos de las aplicaciones de tiempo real tuvieron mayor impacto en la arquitectura de servicios integrados.

La arquitectura de servicios integrados esta basada en *reservación de recursos por flujo*. Para recibir un nivel de servicio garantizado, una aplicación debe hacer la reservación antes de que pueda transmitir tráfico sobre la red. La reservación de recursos implica varios casos. Primero, la aplicación debe caracterizar las fuentes de tráfico y el aseguramiento del recurso. Esta red utiliza un protocolo de enrutamiento para encontrar una trayectoria basada en solicitud de recursos. Un protocolo de reservas se utiliza después para instalar el estado de la reservación a lo largo de esa trayectoria. En cada salto de admisión de control revisa si hay suficiente recursos disponibles para aceptar una nueva reserva. Después de que la primera reservación es establecida, la aplicación puede iniciar el envío de paquetes sobre la trayectoria para la cual esta tiene un uso exclusivo del recurso. La reservación de recursos se ha hecho cumplir por clasificación de paquetes y mecanismos de programación en los elementos en la red, como los *routers*.

La propuesta de trabajo del grupo de servicios integrados propuso dos modelos de servicio que el usuario puede seleccionar el *servicio garantizado* y el *modelo de servicio de carga controlada*. Los modelos de *servicio garantizado*, de característica determinista, en el peor caso retrasan el límite a través de un estricto control de admisión y un programa de encolamiento justo (*fair queuing scheduling*). Este servicio fue diseñado para aplicaciones que requieren absolutas garantías en los retardos. El otro modelo de servicio, es el servicio de carga controlada, que proporciona menos garantía y es un servicio que se aproxima a una red *best-effort* cargada ligeramente. El protocolo *RSVP (The resource Reservation Setup Protocol)* también está diseñado para señalar los

requerimientos de las aplicaciones para la red y para creación de recursos a lo largo de la trayectoria.

El modelo de servicios integrados fue la primera tentativa de adicionar a Internet las capacidades de QoS. La investigación y esfuerzos desarrollados proporcionaron valiosas y complejas aplicaciones en el soporte QoS en Internet.

El despliegue de la arquitectura de servicios integrados proporcionados en los *Backbones* de los proveedores de servicio, ha estado algo lento por numerosas razones. Primero, la arquitectura de servicios integrados esta enfocada principalmente a aplicaciones de larga duración y retardo sensitivo. La palabra *World Wide Web*, sin embargo, representa perceptiblemente un paisaje cambiante de Internet. La Web basado en aplicaciones que ahora dominan el Internet y gran parte de su tráfico es una transacción de corta vida. Si bien la reservación por flujo tiene sentido para las reservaciones duraderas, por ejemplo las video conferencias, estas no son apropiadas para el tráfico de la Web. Los esfuerzos para establecer una reserva por cada sesión son simplemente demasiado altos. Las preocupaciones también se presentaron alrededor de la escalabilidad de los mecanismos para soportar servicios integrados. Para soporte de reservaciones por flujo cada nodo en una red tiene que implementar clasificación y programación por flujo. Estos mecanismos posiblemente no puedan hacer frente a un largo número de flujos a altas velocidades.

La reservación de recursos requiere un soporte de contabilidad y el arreglo entre los diferentes prestadores de servicios. Aquellos que requieren la reservación tienen que pagar por los servicios, cualquier reservación tiene que ser autorizada,

autenticada y registrada. Semejante infraestructura simple de soporte no existe en la Internet. Cuando múltiples proveedores de servicio están involucrados en una reservación, ellos tienen que agregar la carga por acarreamiento de tráfico de los otros proveedores de servicio y establecer estas cargas entre ellos.

La mayoría de proveedores de servicio de redes están conectados actualmente a través acuerdos bilateralmente iguales. Para extender estos acuerdos bilaterales a un acuerdo establecido para la red de Internet es difícil debido al gran número de usuarios.

La arquitectura de servicios integrados podría devolverse en un marco de trabajo viable para las asignaciones de recursos en redes corporativas. Las redes corporativas son típicamente limitadas en tamaño y operadas por un solo dominio administrativo por lo tanto muchos de los temas de escalamiento y establecimiento discutidos anteriormente se desvanecen. Los servicios integrados pueden soportar un ancho de banda garantizado para telefonía IP, video conferencias sobre redes internas corporativas. *RSVP* también puede ser usado para asignación de recursos y control de admisión para tráfico saliendo a redes de áreas abiertas.

Las ideas, conceptos, y mecanismos desarrollados en los servicios integrados también encuentran sus caminos en posteriores trabajos en QoS. Por ejemplo, el servicio de carga controlada ha influenciado el desarrollo de servicios diferenciados, y una similar capacidad de reservación de recursos ha sido incorporada al *MPLS* para garantizar un ancho de banda sobre tráfico estancado en los *backbones*.

1.3 INTRODUCCIÓN A LOS SERVICIOS DIFERENCIADOS [5]

La arquitectura de servicios integrados fue convertida también como alternativa de esquema de admisión para redes proveedoras de servicio. Para 1997 los proveedores sintieron que servicios integrados no estaban listos para un proveedor a mayor escala, y al mismo tiempo la necesidad de realzar el modelo de servicios había llegado a ser mas urgente. La comunidad del Internet comenzó a buscar un acercamiento más simple y más escalable ofreciendo un mejor servicio que el *best-effort* (*servicio del mejor esfuerzo*).

Después de la gran discusión, la *IETF* (Internet Engineering Task Force) formó un nuevo grupo de trabajo para convertirse en un marco y estándar para asignación de diferentes niveles de servicios de Internet. El nuevo acercamiento, llamado servicios diferenciados, es una diferencia significativa para los servicios integrados. En vez de marcar reservaciones por flujo, la arquitectura de servicios diferenciados es aplicada una combinación de limpiar la frontera, aprovisionamiento, y priorizar el tráfico, para alcanzar los servicios diferenciados.

En la arquitectura de servicios diferenciados, el tráfico de los usuarios es dividido en pequeños números de clase de envíos, la cantidad de tráfico que los usuarios pueden inyectar en la red está limitada en la frontera de la misma. Cambiando la cantidad total de tráfico permitido en la red, los proveedores de servicio pueden ajustar el nivel de aprovisionamiento de recursos y por lo tanto controla los recursos de los usuarios.

La frontera de una red de servicios diferenciados es responsable por trazar paquetes en sus apropiadas clases de envío. Esta clasificación de paquetes es típica de un hecho basado en el *acuerdo de nivel de servicio (Service Level Agreement SLA)* entre los usuarios y los proveedores de servicio. Los nodos en la frontera de la red también realizan limpiezas del tráfico para la protección de la red. El tráfico inconforme debe ser desechado, retrasado, o marcado con una clase avanzada diferente.

La clase de envío es directamente codificada dentro del encabezado del paquete. Después de que los paquetes son marcados con sus clases respectivas en los extremos de la red, los nodos internos pueden usar esta información para dar el trato respectivo a los paquetes. Las clases de envío pueden indicar la prioridad de pérdida o la prioridad de recursos. Por ejemplo: cuando un enlace está congestionado, la red desechará primero los paquetes que tengan la más alta prioridad de pérdidas.

Los Servicios Diferenciados no requieren hacer un establecimiento de reserva de recursos; la asignación de las clases de envío es generalmente especificada como parte del *SLA* entre el proveedor y el usuario, y estas clases aplican más para tipos de flujo que para flujos individuales. Esta forma de funcionamiento es buena para aplicaciones web orientadas a transacción. La arquitectura de *Servicios Diferenciados* también mejora la escalabilidad que se da en los *Servicios Integrados*. Las funciones que tienen que realizar los nodos interiores son relativamente simples, los procesos complejos se llevan a cabo por los routers de frontera, donde las tasas de tráfico son generalmente más bajas.

La propuesta de los Servicios Integrados se fundamenta en el aprovisionamiento para proveer aseguramiento de recursos. La calidad del aseguramiento depende de cómo sea llevado a cabo y como sean administrados los recursos en la red. Debido a la naturaleza dinámica de los flujos de tráfico, los aprovisionamientos precisos son difíciles de lograr, entonces generalmente es más difícil y también más costoso proporcionar garantías determinísticas a través del aprovisionamiento que a través de reservaciones.

2. SERVICIOS INTEGRADOS

El desarrollo de los *Servicios Integrados* fue el primer gran intento de realzar el Internet con las capacidades de *QoS*. Los investigadores fueron motivados en parte por los experimentos sobre la *MBONE*, una red *multicast* construida en el Internet para audio y video-conferencia. La experiencia de la *MBONE* demostró que se necesitaban modificaciones significativas a la arquitectura de Internet para poder soportar aplicaciones de tiempo-real. [5]

A principios de los años 90, la comunidad investigativa de trabajo en redes hizo también un progreso substancial en algoritmos y mecanismos para soportar asignación de recursos avanzados sobre la red. En detalle, el trabajo en el planificado de los paquetes por un número de investigaciones demostró que es absolutamente factible proporcionar un retardo y un ancho de banda garantizado con una clase de algoritmos para la planificación, generalmente conocidos como *FQ* o de cola justa, en inglés *Fair Queuing*. Estos esquemas se pueden utilizar para hacer cumplir la asignación de recursos y al mismo tiempo mantener una alta utilización de los mismos. [5]

El desarrollo de los *Servicios Integrados* fue un desafío técnico importante para la comunidad del Internet. El nuevo acercamiento representaba una salida significativa del modelo de datagrama en el cual el Internet fue construido. Muchos nuevos conceptos, mecanismos, y protocolos tuvieron que ser desarrollados, incluyendo los nuevos modelos de servicio, especificaciones de flujo, protocolos de reserva de recursos, mecanismos de encaminamiento, control de admisión, y algoritmos de planificación. El *IETF* estableció varios grupos de trabajo para estandarizar los modelos de servicio y los protocolos para los

Servicios Integrados. Los principales grupos de trabajo y sus funciones fueron los siguientes:

- El grupo de trabajo de (*INTSERV*), responsable de definir la arquitectura, los modelos del servicio, las especificaciones del flujo, y el marco total para el resto de los componentes relacionados en los Servicios Integrados, tales como el control de admisión, la identificación de flujos, y la planificación de los mismos.
- El grupo de trabajo de Servicios Integrados específico sobre las capas de enlace *ISSLL (Integrated Services over Specific Link Layers)*. Donde se definen las especificaciones y las técnicas necesarias para implementar las capacidades de los Servicios Integrados dentro de tecnologías específicas de capa de enlace tales como *Ethernet* y *ATM*.
- El grupo de trabajo del protocolo de reserva de recursos (*RSVP*) en inglés *Resource Reservation Setup Protocol*. Estandariza un protocolo del establecimiento de la reserva de recursos, que instala el estado de reserva dentro de la red.

2.1 APLICACIONES DE TIEMPO REAL [4]

La necesidad de transmisión de información multimedia (audio y vídeo) tiene un gran impacto en los sistemas de comunicación. Esta transmisión requerirá no sólo el soporte de las redes, sino también de los protocolos de comunicación de niveles superiores. En este sentido se trabaja tanto en redes locales (*LAN*) como en *internetworks*.

Podemos hablar de transmisión de información en tiempo real cuando se puede asegurar que la información llegue a su destino con unos parámetros determinados (retraso, rendimiento, fiabilidad, etc.). En este sentido se puede

asumir que la transmisión multimedia tiene unos requerimientos temporales que necesitan del uso de esta transmisión en tiempo real.

En general las aplicaciones multimedia requieren una calidad de servicio (QoS) por parte de los servicios de red. De las nuevas redes se exige poder especificar esta calidad de servicio y asegurar su cumplimiento. Otro problema es quién proporciona esta calidad de servicio: la red (los *routers*, nodos, conmutadores, etc.) o el sistema operativo.

El retraso puede afectar las prestaciones de las aplicaciones en dos maneras. Primero, el tiempo del retraso determina la latencia de la aplicación. Segundo, el retraso individual de los paquetes puede hacer que la fidelidad decaiga si se excede el tiempo de retraso determinado; en este caso la aplicación puede retrasar la ejecución para reproducir estos paquetes retrasados (lo que introduce distorsión) o bien simplemente descartarlos (lo que crea una señal incompleta).

En este sentido se pueden distinguir dos tipos de aplicaciones:

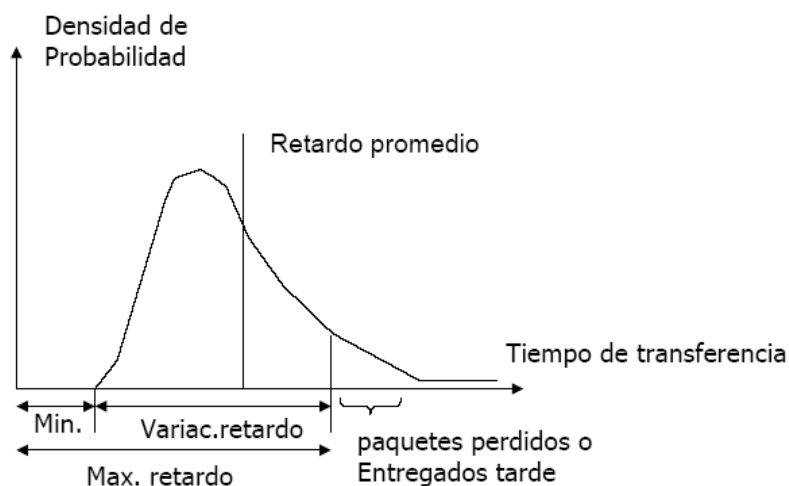
- Aplicaciones intolerantes: Estas aplicaciones no se pueden adaptar a que un paquete se retrase más que el límite predeterminado. Necesitan por lo tanto un límite superior del retraso fiable. Estas aplicaciones requieren un modelo de servicio denominado *servicio garantizado*.
- Aplicaciones adaptativas: Estas aplicaciones pueden tolerar que lleguen paquetes con un mayor retraso. Estas aplicaciones requieren un modelo de servicio denominado *servicio predictivo*, que proporciona un servicio bastante fiable pero no seguro. Este tipo de aplicaciones pueden aceptar una pérdida en la calidad presumiblemente por el menor valor de este modelo, ya que se incrementa el uso de los recursos de red.

Para proporcionar un límite en el retraso el tráfico se tiene que caracterizar y debe haber algún algoritmo de control de admisión que asegure que una petición de flujo puede ser aceptada.

La variación en el retardo es conocida como *jitter* (diferencia entre el mayor y menor retardo). Debido a la distorsión de la sincronización causada por el jitter, la calidad de la señal de la voz no sería buena si el receptor simplemente envía los datos al dispositivo de audio al mismo tiempo que los va recibiendo. En la *figura 1* se observa como el retardo promedio se encuentra entre la variación de retardo, no necesariamente en la mitad, y también se puede ver cómo los paquetes perdidos y los que llegan demasiado tarde no son tenidos en cuenta.

En la *figura 1* se observa como el retardo promedio se encuentra entre la variación de retardo, no necesariamente en la mitad, y también se puede ver cómo los paquetes perdidos y los que llegan demasiado tarde no son tenidos en cuenta.

Figura 1. Distribución del retardo y variación del retardo



Fuente: Jhon j. Padilla, *calidad de servicio en Internet (Octubre 2007)*
<http://jpadilla.docentes.upbbga.edu.co/QoS/contenido.htm>

2.2 USO DEL BUFFER SUAVIZADO [5]

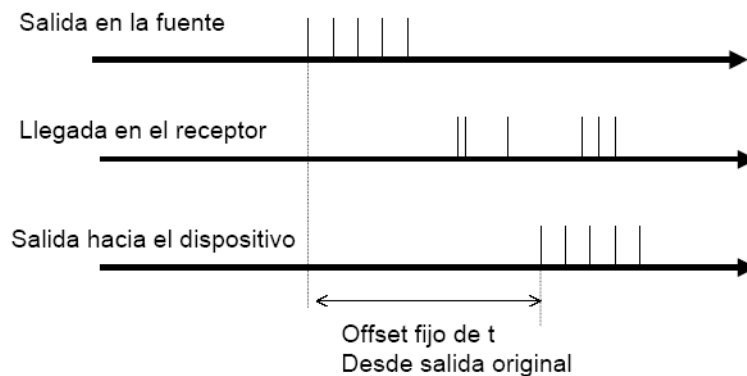
El *jitter* es la variación del tiempo entre la llegada de distintos paquetes. Estas variaciones son debidas a la saturación de la red, la falta de sincronismo o los cambios dinámicos en las rutas. En redes con grandes cambios de velocidad se puede usar un *jitter buffer* para mejorar la calidad de la conversación. Un buffer es un espacio intermedio donde se almacenan los paquetes hasta su procesamiento. La idea básica del *jitter buffer* es retrasar deliberadamente en el caso de *VoIP* la reproducción del sonido para garantizar que los paquetes más lentos hayan llegado. Los paquetes se almacenan en el buffer, se reordenan si es necesario y se reproducen a una velocidad constante. La calidad de voz mejora al precio de incrementar la latencia total.

Una técnica común para remover el *jitter* es usar un almacenador intermedio (*play-out buffer*). Con esto, los paquetes entrantes son guardados en un almacenador intermedio, y el retardo debido al *jitter* es aislado al agregar atrasos extras a los paquetes que experimenten menos atrasos en la red. (*Figura 2*). Los datos son mandados al dispositivo de salida a un retardo fijo desde el tiempo de partida original; de esta manera la sincronización original de la señal se reconstruye fielmente. El término *playback-point* (punto de *Playback*) se refiere al punto en el tiempo que es el offset (nivel de referencia) del tiempo original de salida; las aplicaciones que usan *playback-buffers* o *buffers* intermedios para reconstruir la señal original son llamadas aplicaciones de *playback*.

El *playback-point* juega un papel crucial en las aplicaciones de *playback*, los datos que llegan antes del *playback point* se almacenan en el *playback buffer*, esperando por el *play-out point* o punto de comienzo de ejecución. Los datos que lleguen después del *playback point* llegarán muy tarde para la reconstrucción de la señal original para fijar un punto apropiado de *playback*, una aplicación debe saber el máximo retardo que un paquete puede experimentar; este retardo o retraso

límite podría ser proporcionado por la red como un servicio o puede ser estimado basado en retardos experimentados por llegadas de paquetes previos.

Figura 2. Uso del Buffer suavizador



Fuente: Jhon j. Padilla, *calidad de servicio en Internet* (Octubre 2007)
<http://jpadilla.docentes.upbbga.edu.co/QoS/contenido.htm>

2.3 TRÁFICO MULTICAST MBONE

MBONE existe desde 1992 como una red virtual para la experimentación del uso del IP *Multicast* en Internet. Esta red se ha empleado principalmente para el estudio de herramientas de audio/vídeo conferencias multipunto, aunque en principio puede ser empleada para el intercambio de cualquier tipo de información multimedia. Su principal ventaja, o mejor su principal característica, es la de proporcionar el intercambio de información de uno a muchos, pero sin los inconvenientes de tener que duplicar dicha información para cada uno de los receptores y en función del número de ellos. [6]

Ha permanecido por muchos años principalmente por ser una *red virtual*; una red creada como la unión de múltiples *islas* interconectadas entre sí. Esto es así debido bien a que la mayor parte de equipos de comunicaciones que interconectan la infinidad de redes que forman Internet aun no hablan el lenguaje

del *MBone*, es decir, el *IP Multicast*, o lo hacen de forma incompatible con otros. Pese a que el panorama está cambiando muy rápidamente y que un gran número de ingenieros, tanto de empresas privadas como pertenecientes al ámbito académico y/o investigador, se están esforzando conjuntamente para convertir este experimento global en un servicio operativo, aún existen ciertas complicaciones que apuntan a que *MBone* seguirá siendo por el momento una red experimental en la Internet. Esto, evidentemente, no quiere decir que los conceptos de *MBone* no puedan ser empleados con éxito y de manera totalmente funcional dentro de redes corporativas o incluso entre distintas intranets, sino que el objetivo final, el hacer de las comunicaciones *multicast* algo inherente a Internet, es algo que aún está lejano.

A continuación se presentan ciertos conceptos que forman la base tecnológica de *MBone* [6].

2.3.1 IP Multicast [5]

Internet es una red en la que el intercambio de información entre equipos locales o remotos se hace a través de datagramas IP (*Internet Protocol*). Un datagrama IP (o paquete IP) podríamos decir que es la unidad mínima de información en el lenguaje que hablan todos los equipos que forman parte de Internet. Estos datagramas IP están formados principalmente por una dirección origen y una dirección destino, y cada equipo de comunicaciones situado en la ruta entre ambos se encarga de enviar dicho datagrama por el camino adecuado. Esto implica que cada estación conectada a Internet debe tener una dirección que la identifique, lo que se llama dirección IP, y constituye un sello de identidad global y única para cada equipo en Internet.

Pueden clasificarse en tres tipos en función de la dirección de destino:

- **IP unicast:** La dirección corresponde a un solo receptor y será éste el único que procese los datagramas IP con ese destino (conexión uno-a-uno).
- **IP broadcast:** La dirección corresponde a todos los equipos conectados en un mismo tramo de red local y el datagrama IP es procesado por todos ellos (conexión uno-a-todos dentro de la misma subred).
- **IP multicast:** La dirección corresponde a un grupo de equipos, y sólo estos procesarán los datagramas IP con ese destino (conexión uno-a-muchos, o uno-a-varios).

Cuando un equipo envía un datagrama IP a una determinada dirección IP *multicast*, sólo es recibida por aquellos equipos que están a la escucha de esa dirección y, que por tanto, son capaces de entender las direcciones *multicast*.

La experiencia del *MBONE* también destacó la necesidad de una mejor política de tráfico. En particular *multicast*, el cual, si se usa mal, puede potencialmente causar una tremenda interrupción a una larga porción del Internet. Aplicaciones como video digital son capaces de generar muy altas tasas de tráfico; en los años 90 fueron lo suficientemente altas como para saturar muchos *backbones*. Un usuario malicioso o descuidado puede extender la congestión a muchos lugares con solo enviar un flujo de video a un grupo *multicast*. De hecho ocurrieron muchos incidentes de este tipo en el *MBONE*, causados por malas configuraciones. Por ejemplo, una característica de una aplicación de video se encendió accidentalmente por un usuario y toda su pantalla fue transmitida por el *MBONE*. La congestión que esto causó fue tan grande que muchos usuarios no pudieron conectarse a Internet y tuvieron que usar el viejo sistema telefónico.

La carencia de un control de tráfico explícito dentro de la red puede causar que aplicaciones basadas en *UDP* comiencen a comerse las aplicaciones basadas en *TCP* en su competencia por el ancho de banda. Cuando una aplicación *TCP* detecta una congestión en la red, el mecanismo de control de congestión en *TCP* retrocede y reduce la tasa de transmisión. La mayoría de aplicaciones de tiempo

real basadas en *UDP* no reaccionan a signos de congestión (o reaccionan menos agresivamente). Por lo tanto las aplicaciones *TCP* tienden a sufrir más las pérdidas por congestión.

2.4 ARQUITECTURA DE SERVICIOS INTEGRADOS [3, 7]

El modelo de Servicios Integrados (*IntServ: Integrated Services*) provee a las aplicaciones de un nivel garantizado de servicio, negociando parámetros de red, de extremo a extremo. La aplicación solicita el nivel de servicio necesario para ella con el fin de operar apropiadamente, y se basa en la *QoS* para que se reserven los recursos de red necesarios antes de que la aplicación comience a operar. Estas reservaciones se mantienen en pie hasta que la aplicación termina o hasta que el ancho de banda requerido por ésta sobrepase el límite reservado para dicha aplicación. El modelo *IntServ* se basa en el Protocolo de Reservación de Recursos (*RSVP*) para señalar y reservar la *QoS* deseada para cada flujo en la red. Debido a que la información de estados para cada reserva necesita ser mantenida por cada enrutador a lo largo de la ruta, la escalabilidad para cientos de miles de flujos a través de una red central, típicos de una red óptica, se convierte en un problema.

Esta arquitectura debería cumplir con los requisitos básicos de las aplicaciones de tiempo real. La idea básica es hacer reservaciones de recursos por flujos.

El objetivo es preservar el modelo de *datagrama* de las redes basadas en *IP* y al mismo tiempo soportar reservas de recursos para aplicaciones de tiempo real.

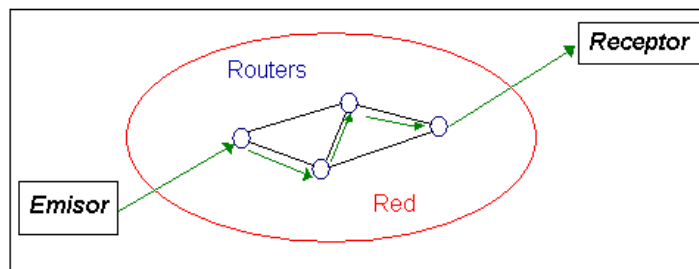
2.4.1 Aproximación Básica [3]

En esta aproximación hace reserva de recursos por flujos. Un flujo es una cadena de paquetes que fluyen por la red desde una aplicación en un ordenador origen

hasta una aplicación en un ordenador destino. La reserva de recursos debe establecerse previamente en cada uno de los *routers* que hacen parte del camino entre los dos terminales. Para ello cuando una aplicación desea iniciar una comunicación debe seguir los pasos siguientes que se ilustrados en la *figura 3*.

- a. La fuente inicia el establecimiento de una reserva describiendo primero a la red las características del flujo y los requerimientos de los recursos.
- b. La red puede aceptar este nuevo flujo de aplicación sólo si hay suficientes recursos para comprometerse con los recursos solicitados.
- c. Una vez la reserva es establecida, la aplicación puede enviar sus paquetes a lo largo del camino reservado y la red cumplirá su compromiso.

Figura 3. Realización de una reserva de recursos



Fuente: Wang Z., *Internet QoS. Architectures and Mechanisms for Quality of Service*, Ed. Morgan Kaufman. U.S.A. 2001.

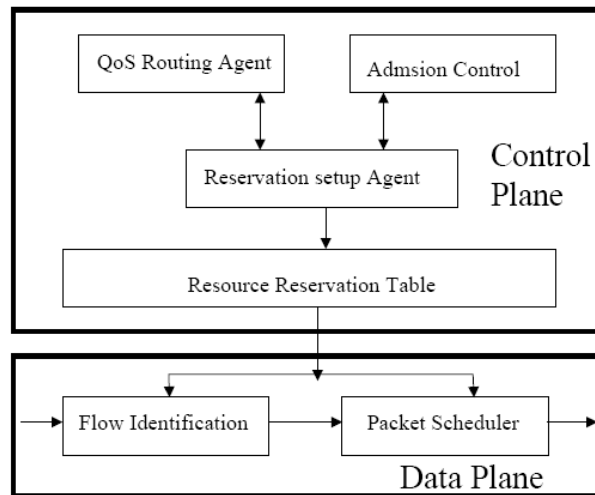
2.4.2 Modelo de referencia

El modelo de referencia de *IntServ* (Figura 4) puede ser dividido en dos partes:

- El plano de Control, que establece la reserva de recursos

- El plano de datos, que re-envía los paquetes de datos basado en el estado de la reserva.

Figura 4. Modelo de referencia de *IntServ*



Fuente: Jhon j. Padilla, *calidad de servicio en Internet* (Octubre 2007)
<http://jpadilla.docentes.upbbga.edu.co/QoS/contenido.htm>

2.4.3 Funcionamiento del plano de control y del plano de Datos

2.4.3.1 El plano de Control (Control Plane) [3]

Selección de la ruta

En cada nodo la red debe determinar cual trayectoria usará para establecer la reserva de recursos. La trayectoria debe ser seleccionada de tal modo que tenga suficientes recursos.

Actualmente los protocolos de enrutamiento, típicamente usan una métrica simple como un retardo, un conteo de saltos, o un manejo de los pesos para calcular la trayectoria mas corta a todos los destinos de red.

Ya que un enrutamiento que soporte reservación de recursos sigue siendo un problema difícil, la arquitectura de Servicios Integrados deliberadamente desacopla el proceso de enrutado de los procesos de reserva. La selección de ruta puede ser inicialmente hecha basada en el enrutamiento actual de IP o usando un servidor centralizado hasta que un sistema mejor sea desarrollado.

Establecimiento de reserva

Se necesitó un protocolo de establecimiento de reserva que fuera de salto en salto a través de la trayectoria e instalara el estado de reserva en los enrutadores. El protocolo contendría la información acerca de la caracterización del tráfico y los recursos requeridos de tal forma que cada nodo a través de la trayectoria pueda determinar si la nueva solicitud de reserva puede o no ser aceptada. El protocolo de establecimiento de la reserva debe lidiar con los cambios en la topología de la red.

El usuario que solicite una reservación debe ser autenticado y la reservación será registrada para la tarificación correspondiente.

En los Servicios Integrados, el protocolo *RSVP* ha sido desarrollado como el protocolo de establecimiento de reserva para el Internet. El protocolo *RSVP* permite diferentes estilos de reservación y usa un modelo de estado suave (*soft state*) para tratar con los cambios de ruta.

Control de admisión

Para ofrecer recursos garantizados a los flujos con reserva, una red debe monitorear el uso de los recursos, y debe negar las solicitudes de reserva cuando no hay suficientes recursos disponibles.

El control de admisión tiene dos funciones básicas. La primera función es la de determinar si una nueva reserva puede ser instalada basándose en las políticas de control de admisión. La segunda función es monitorear y medir los recursos disponibles.

2.4.3.2 El plano de datos (data plano) [3]

Identificación de flujos

En el procesamiento de paquetes, un enrutador debe examinar cada paquete que entra y decidir si el paquete pertenece a uno de los flujos reservados *RSVP*. Un flujo *IP* es identificado por los cinco campos del encabezado del paquete: Dirección de la fuente de *IP*, destino de la fuente *IP*, identificación del protocolo, puerto de la fuente y puerto de destino. Los cinco campos usualmente están referidos como la quintupla. Para determinar si un paquete pertenece a un flujo *RSVP*, el motor de identificación del flujo debe comparar las quintuplas del paquete entrante con las quintuplas de todos los flujos en la tabla de reservas. Si hay una igual, el estado de reservación correspondiente es recuperado de la tabla de reservas y el paquete es enviado al planificador de paquetes con el estado de reservación asociado con el flujo.

Planificación de paquetes

El planificador de paquetes es el responsable de hacer cumplir la asignación de los recursos. Directamente afecta el retardo que los paquetes y pueden llegar a descartasen cuando los *buffers* están llenos.

La tarea central de un planificador de paquetes es seleccionar un paquete para transmitirlo cuando el enlace (*link*) esté listo.

2.5 MODELOS DE SERVICIO

Los *Modelos de Servicio* describen las interfaces entre la red y sus usuarios en una arquitectura de asignación de recursos; esto es, cuáles son los servicios que los usuarios pueden solicitar a la red y que tipo de compromiso puede ofrecer la red frente a los recursos reservados. Los Servicios Integrados estandarizaron dos modelos de servicio: el *servicio garantizado* y el *servicio de carga controlada*. [5]

2.5.1 Servicio garantizado [5]

El servicio garantizado provee garantías de ancho de banda y límites estrictos de retardo de encolamiento extremo a extremo para sus flujos. El servicio está orientado a aplicaciones que requieran el más alto aseguramiento de ancho de banda y retardo; por ejemplo las aplicaciones de playback intolerantes. El servicio garantizado también puede ser utilizado para aplicaciones que tienen requerimientos de tiempo-real fuertes, como los sistemas de control de una misión.

2.5.1.1 Comportamiento de extremo a extremo

Intuitivamente, el comportamiento de extremo a extremo de un trayecto que soporta el servicio garantizado, puede ser visto como un circuito virtual con un

ancho de banda garantizado. El circuito virtual es más flexible que un circuito real; la partición del ancho de banda entre circuitos virtuales es lógica, y por tanto también los límites de las particiones permitiendo que se puedan expandir. Por ejemplo, el flujo de tráfico *best-effort* podría oportunamente usar el ancho de banda no consumido por los flujos reservados.

2.5.2 Servicio de carga controlada [5]

El aseguramiento estricto del ancho de banda y el retardo límite proporcionado por el servicio garantizado tiene un precio: los recursos tienen que ser reservados para el peor caso. Para fuentes de tráfico fluctuante, esto conllevará a una baja utilización de la red y a incrementar el costo por reserva de recursos. Además, es usualmente difícil conocer exactamente el ancho de banda y los requerimientos de retardo para una aplicación.

Para algunas aplicaciones, un modelo de servicio con menos garantías estrictas y bajo costo podría servir mejor las necesidades. El grupo de trabajo de los Servicios Integrados consideró varias ofertas y estandarizó el servicio de carga controlada. El servicio de carga controlada no provee ninguna garantía cuantitativa sobre los límites de retardos ni del ancho de banda. En lugar de eso trata de emular una red de carga ligera para las aplicaciones que requieran el servicio. Este modelo de servicio permite multiplexación estadística y por tanto puede ser implementado de una forma más eficiente que garantice el servicio. Sus características se ajustan bien con aplicaciones adaptables que requieren algún grado de aseguramiento de desempeño mas no límites absolutos.

2.5.2.1 Comportamiento de extremo a extremo

El comportamiento de extremo a extremo del servicio de carga controlada es de alguna manera imprecisa comparado con el servicio de carga garantizada; de

cierta forma, esto es intencional. El servicio de carga controlada, intenta crear un modelo de servicio intermedio entre el servicio de mejor esfuerzo (*best-effort*) y el servicio garantizado a través de un control de admisión apropiado y mecanismos de aislamiento de tráfico. Por esta razón, también es referido como el servicio *better-than-best-effort*. El comportamiento visible a las aplicaciones puede ser mejor descrito como al de una red ligeramente cargada tipo *best-effort*. Más específicamente:

- Un porcentaje muy alto de los paquetes transmitidos serán exitosamente entregados por la red a los receptores.
- El retardo de encolamiento de tránsito experimentado por un porcentaje muy alto de paquetes entregados no excede grandemente al mínimo retardo.

El servicio de carga controlada no hace uso de valores específicos para el control de parámetros como el retardo y las pérdidas. La aceptación de una solicitud para el servicio de carga controlada implica que la red tiene suficientes recursos para acomodar el tráfico sin causar congestión.

Otra forma de definir el servicio de carga controlada es describiendo los eventos que se espera que no ocurran con ninguna frecuencia.

- Poco o nada de retardo promedio de encolamiento sobre todas las escalas de tiempo significativamente más largas que el tiempo de "*burst*". (tiempo requerido para transmitir la máxima fluctuación o ráfaga de mayor tamaño a la tasa requerida).
- Poco o nada de pérdidas por congestión sobre todas las escalas significativamente mas largas que el tiempo de "*burst*".

En esencia, el servicio de carga controlada permite un valor ocasional de retardos o pérdidas. Sin embargo, la probabilidad de tales eventos debe ser suficientemente baja para que el retardo promedio de encolamiento y la tasa de pérdidas promedio sobre un periodo razonable se sigan acercando a cero.

2.5.3 Protocolo de reservación de recursos (*resource reservations setup - RSVP*) [3, 5, 9]

El Protocolo de reservación de recursos es, un protocolo de control de la red que permite que los programas que se irán a trabajar en Internet puedan obtener la calidad de servicio que sus flujos de datos puedan requerir.

Este protocolo no es, en contra de lo que pudiera parecer, un protocolo de enrutamiento. Es un protocolo que se inscribe dentro de la capa de Transporte del modelo de conectividad *OSI* y se apoya en las tablas de rutas dinámicas que manejan los protocolos de enrutado clásicos para establecer una conexión a modo de circuito virtual entre emisor y receptor o receptores implicados. [9]

Para *RSVP* el flujo de datos es simplemente una secuencia de paquetes que tienen un mismo origen, uno o varios destinos, según sea la difusión, *unicast* o *multicast*, y una calidad de servicio, todo ello caracterizado mediante sesiones. Una sesión *RSVP* es cada torrente de datos que el protocolo maneja de forma independiente. [9]

Las especificaciones de operación de este protocolo se materializan en un programa, *RSVP* estructurado en módulos, cada uno de ellos con unas funciones específicas. Por una parte están el módulo de Control de Admisión y el módulo de Control de Política. El primero se encarga de determinar si el nodo tiene los recursos solicitados disponibles para soportar la calidad de Servicio pedida. El

Control de Política determina si el solicitante tiene los permisos necesarios para poder disponer de los recursos que solicita. En otro lado se encuentra el motor de la reserva, el módulo de Clasificación, encargado de determinar su ruta y QoS necesaria de los paquetes, y el módulo Esquemático al que se le encomienda la transmisión de los paquetes. [9]

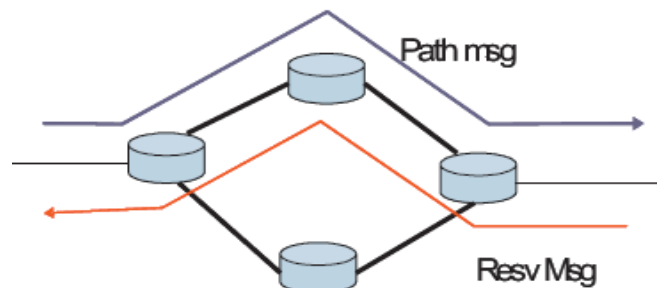
Característica *RSVP*

- *RSVP* fue diseñado para ser un protocolo agregado de la arquitectura en los protocolos *TCP/IP*.
- Su diseño fue influenciado por los requerimientos de videoconferencia *multicast*.
- Principio de diseño: Reservación simple, orientado a receptor, independencia del enrutamiento, independencia de las políticas, adaptabilidad a cambios en la red (*Soft-State*), varios estilos de reserva.

2.5.4 Control de la reserva

La labor de gestión de recursos la realiza este protocolo mediante el intercambio de mensajes que se incorporan en los paquetes *RSVP*, que le permiten gestionar ese circuito virtual que se llega a establecer entre emisor y receptores. En *RSVP* hay dos tipos de mensajes: los *mensajes PATH* y los *mensajes RESV*. Ver figura 5

Figura 5. Procedimiento de establecimiento de la Reserva en *IntServ6*



Fuente. Jhon Jairo Padilla Aguilar, Josep ParadellAspas, Mónica Karel Huertas, Xavier Hesselbach, *SOPORTE DE QOS SOBRE IPV6 CON INTSERV6*, *Revista Gerencia Tecnológica Informática*, GTI, Número 8, Volumen 4, 6-05-2005.

El primer tipo de mensajes que ha de manejarse es el de solicitud de reservación *PATH*, mensaje que tiene que generar el receptor. Es decir, el host de destino ha de enviar un mensaje que recorra la ruta de llegada a la inversa para que el *host* emisor pueda determinar qué control de flujo es necesario para que pueda iniciarse el tráfico desde el primer salto. El protocolo no mantiene informado al emisor sobre el éxito del establecimiento de la conexión con el receptor. [3]

Para que esos mensajes de solicitud de reservación se produzcan, el emisor tiene que lanzar un mensaje de trayectoria *RESV* que trace la ruta según la información que le proporcionan los protocolos de enrutamiento y que cada nodo del camino ha de conocer para mantenerla.

El protocolo de reservación de recursos almacena el estado de los nodos enrutadores y de los equipos terminales regenerando periódicamente la información mediante los mensajes de trayectoria y solicitud de reservación. El estado se elimina cuando no llegan los mensajes oportunos de regeneración dentro del tiempo de validez que se marca, o cuando llegan mensajes explícitos de revocación. Cuando hay modificaciones respecto a una ruta, el mensaje de

trayectoria que se produce a continuación inicializa un nuevo camino y los mensajes de solicitud de reservación establecen la reserva, expirando la senda que no se utiliza. Con *RSVP*, los cambios de las rutas tienen un mínimo retardo, porque si los mensajes que informan sobre el estado difieren sobre el almacenado, se actualizan inmediatamente y se propagan las actualizaciones al resto de las etapas. [9]

2.6 CLASIFICACIÓN DE PAQUETES CON *INTSERV*

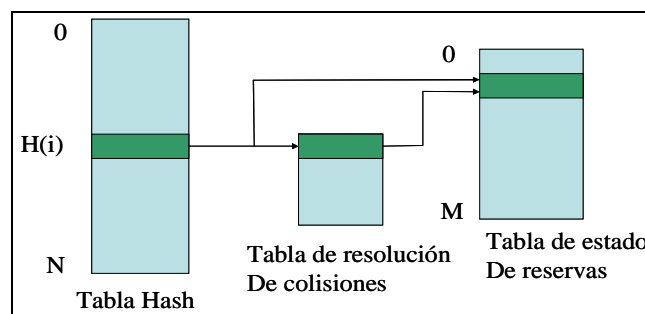
2.6.1 *El proceso de Clasificación de flujos (Flow Identification) [3]:*

El módulo de Identificación de flujos debe examinar cada paquete entrante y decidir si el paquete pertenece a uno de los flujos *RSVP* reservados. La identificación de un flujo IP se hace mediante 5 campos de la cabecera del paquete (también conocidos como *Quíntupla*): Dirección *IP* fuente, Dirección *IP* destino, Identificación del protocolo, Puerto fuente, Puerto destino. Para determinar si un paquete coincide con un flujo *RSVP*, el motor de identificación de flujos debe comparar la *Quíntupla* del paquete entrante con la *Quíntupla* de todos los flujos en la tabla de reservas, si hay una coincidencia, se obtiene el estado de la reserva de la tabla de reservas y el paquete es enviado al planificador de paquetes junto con el estado de la reserva asociado con el flujo. Este proceso debe hacerse en tiempos muy cortos pues la cantidad de paquetes de un enlace entrante de un *router* puede ser muy grande, al igual que el número de flujos.

Para solucionar el problema de la eficiencia en la clasificación de los paquetes, *IntServ* utiliza una solución denominada *Hashing*. Esta consiste en que, el establecimiento de Reserva, el *router* calcula un valor a partir de una *función Hash* que utiliza la *Quíntupla* como semilla. El valor *Hash* de salida es usado para identificar el estado de la reserva del flujo en la *Tabla de Reservas* del *router* (ver *figura. 6*) y se almacena en una *Tabla Hash* que contiene el valor *Hash* y un

apuntador a la *Tabla de Reservas*. Si el valor *Hash* calculado ya existe en la tabla se dice que hay una *colisión* y se coloca para esta entrada de la tabla el *bit* de *colisión* en 1, indicando que hay una colisión. En este caso se almacena en la *tabla de resolución de colisiones* el valor *Hash* y la *Quíntupla* correspondiente, junto con un apuntador a la reserva respectiva en la tabla de reservas.

Figura 6. Proceso de identificación de flujos basado en *Hashing* en *IntServ*



Fuente: Jhon Jairo Padilla, "Contribución al soporte de calidad del servicio en redes móviles" Tesis Doctoral, Universidad Politécnica de Cataluña, Barcelona, España, Febrero 2008.

Durante el proceso de retransmisión de paquetes, el *router* obtiene el valor *Hash* a partir de la *Quíntupla* de cada paquete entrante y luego verifica en la tabla *Hash* si este valor existe. Si no existe, el paquete es tratado con un servicio tipo *Best-Effort*, mientras que si el número *Hash* existe en la tabla, se debe verificar si hay una colisión. En caso de no haber colisión, el número *Hash* apunta directamente a la *Tabla de reservas*, mientras que, si hay colisión, se debe resolver esta haciendo una búsqueda del número *Hash* y de la *Quíntupla* en la *tabla de resolución de colisiones*.

2.7 CLASIFICACIÓN DE PAQUETES CON INTSERV6 [24]

La propuesta *IntServ6* se basa en la arquitectura *IntServ* [5] para el soporte de QoS y mejora del desempeño de los *routers*. Para lograrlo, *IntServ6* usa la etiqueta de flujo de *IPv6* para acelerar el proceso de clasificación en los *routers*.

Este número se usa para buscar un apuntador en la *Tabla Hash* hacia la *Tabla de Reserva de Recursos*, donde la *Tabla de Reserva de Recursos* a su vez es usada para almacenar información de planificación y enrutamiento.

En *IntServ6*, el proceso de clasificación de paquetes en los *routers* es mejorado haciendo el cálculo del *número Hash* sólo en el *host* en lugar de hacerlo en todos los *routers* como se hace en *IntServ*. Esto es posible hacerlo y no constituye un problema porque el *número Hash* resultante es siempre el mismo independientemente del sitio donde éste sea calculado, ya que, tanto los parámetros usados (*Quíntupla*), como el algoritmo que hace los cálculos con éstos es siempre el mismo.

IntServ6 usa el *número Hash* para identificar los flujos de paquetes llevando este número en el campo de *etiqueta de flujo* disponible en el formato de paquetes de *IPv6*.

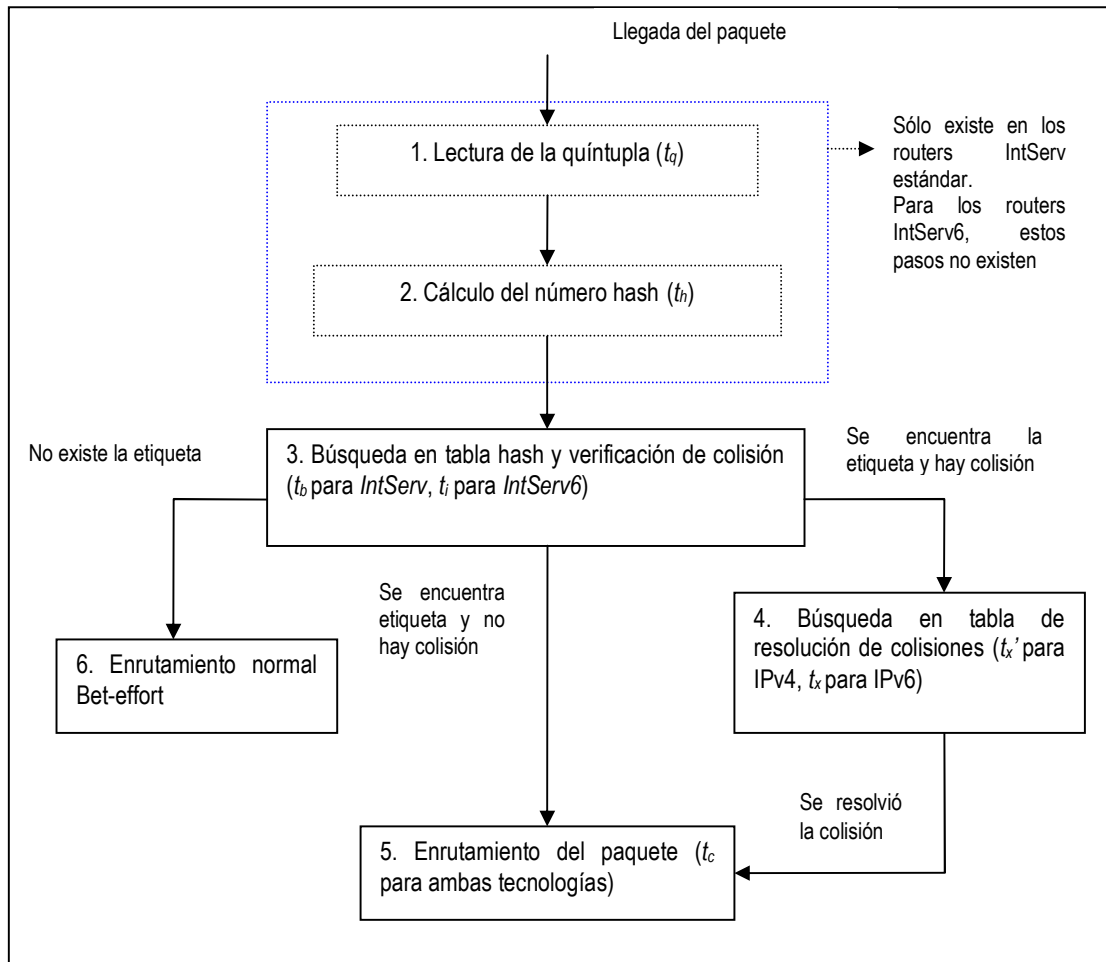
2.8 MODELO DEL CLASIFICADOR [8]

Para el clasificador se obtiene un modelo estocástico del retardo medio de los paquetes descrito en [20], el cual se basa en el Diagrama de Transición de Estados (*STD*) del proceso de la *figura 10*.

El método para obtener una expresión para el retardo de los paquetes en el clasificador es asignar un retardo a cada estado (tiempos entre paréntesis en la figura del *STD* o figura 7). Luego, para obtener los tiempos de retardo totales en el clasificador se lleva un conteo de cuantas iteraciones necesita el simulador para llevar a cabo cada estado del *STD* y se multiplican por las respectivas constantes de tiempo que se encuentran en la *tabla 1*. La mayoría de los procesos tienen un retardo constante calculado de la secuencia seguida para cada proceso específico. Sin embargo hay dos procesos que tienen un retardo aleatorio. Primero, el proceso de búsqueda de la *tabla Hash* para *IntServ* tradicional, porque usa un esquema *matching* (búsqueda uno por uno hasta encontrar una coincidencia); luego la búsqueda tiene un comportamiento aleatorio con distribución geométrica ya que, cada intento representa una lectura de la tabla, donde el resultado puede ser de éxito o de falla, y la prueba se realiza hasta que halla una coincidencia. Por otro lado en *Intserv6*, este proceso es determinista porque usa un esquema *tagging* o de etiquetado, y sólo toma una sola lectura de la *tabla Hash* para esta tecnología.

El otro proceso aleatorio es el de la búsqueda en la *tabla de resolución de colisiones*, este es también del tipo *matching* (y posee distribución geométrica), y similar para ambas tecnologías *IntServ* e *IntServ6*.

Figura 7. Diagrama de transición de estados del proceso de clasificación de paquetes de un Router *IntServ* e *IntServ6*.



Fuente: Padilla, J., J. Paradells, and A. Rodriguez, Supporting QoS over IPv6 wireless networks with *IntServ6*. (Accepted) PIMRC'06 The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Helsinki -Finlandia, 2006.

Los tiempos correspondientes a los procesos indicados en el diagrama de estados se encuentran en la *tabla 1*. Se aclara que son calculados con base en un procesador de red *IXP 1200* de Intel [20], el cual fue diseñado para soportar *routers* con enlaces OC-3 (100-500Mbps). Este usa la memoria *DRAM* para hacer

de *Buffer* de los paquetes, y la *DRAM* para almacenar la información de control de frecuentemente acceso.

Tabla 1. Parámetros de Simulación del procesador de red *IXP 1200*

IXP 1200 Parámetros del Procesador de Red		
Parámetro		Valor
L_{sdram}	(tiempo de lectura sdram)	160 ns
L_{sram}	(tiempo de lectura sram)	80 ns
Tamaño de palabra del procesador		32 bits
Tiempos de Procesamiento		
$t_i = 1L_{sram}$;	(tiempo de verificación de la tabla hash - IntServ, IntServ6)	80ns
$t_e = 2L_{sram} + 32L_{sdram}$;	(tiempo de enrutamiento de paquete, es el mismo para IntServ e IntServ6)	5280ns
$t_{comp} = 10 * L_{sram}$	(tiempo de comparación de la quintupla para el proceso de resolución de colisiones)	800ns
$t_q = 10 * L_{sram}$	(tiempo de lectura de la quintupla para búsqueda de la tabla hash)	800ns

Fuente: Padilla, J., J. Paradells, and A. Rodriguez, Supporting QoS over IPv6 wireless networks with IntServ6. (Accepted) PIMRC'06 The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Helsinki -Finlandia, 2006.

2.9 PLANIFICACIÓN DE PAQUETES [7]

Una forma de llevar a cabo una planificación, es diseñar una estrategia de manejo de los paquetes en caso de congestión, o evitar que la red alcance este estado descartando paquetes a medida que estos ingresan.

El *manejo de congestión* es un término general usado para nombrar los distintos tipos de estrategia de encolamiento que se usan para manejar situaciones donde la demanda de ancho de banda solicitada por las aplicaciones excede el ancho de banda total de la red, controlando la inyección de tráfico a la red, para que ciertos flujos tengan prioridad sobre otros.

Los modelos de planificación utilizados en este proyecto de grado son el modelo *PQ* y el modelo *PGPS*.

2.9.1 Priority Queuing (PQ)

El Encolamiento de Prioridad (*PQ: Priority Queueing*) consiste en un conjunto de colas, clasificadas desde alta a baja prioridad. Las cuales son servidas en estricto orden de prioridad. Las colas de mayor prioridad son siempre atendidas primero, luego la siguiente de menor prioridad y así. Si una cola de menor prioridad está siendo atendida, y un paquete ingresa a una cola de mayor prioridad, ésta es atendida inmediatamente (*ver figura 8*). Este mecanismo se ajusta a condiciones donde existe un tráfico importante, pero puede causar la total falta de atención de colas de menor prioridad (*starvation*) [7].

Características

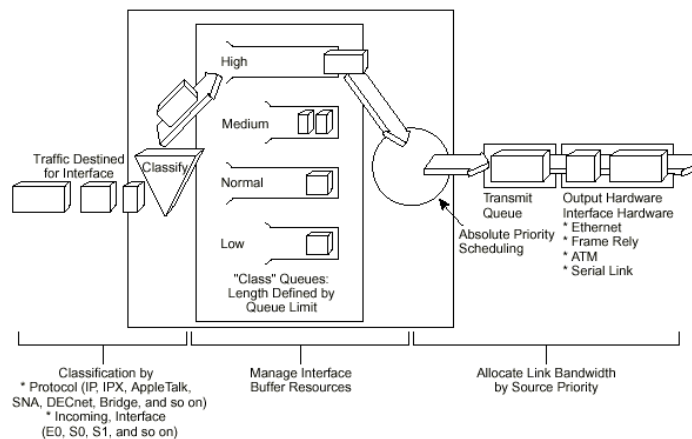
- Dar prioridad estricta al tráfico importante

Asegura que el tráfico importante reciba un servicio rápido en cada punto de la red.

- Existen 4 clases de prioridad de tráfico
 - En el mecanismo *PQ*, cada uno de los paquetes debe ser colocado en una de las cuatro posibles colas (alta, media, normal, baja prioridad), servidas en riguroso orden de prioridad.
 - Ofrece garantías totales.
 - Las prioridades se definen por filtros en los *routers*.

- Clasificación: Protocolo de red, interfaz de acceso del *router*, dirección origen y destino, tamaño del paquete.
 - La prioridad de los paquetes puede diferenciarse por diversos medios, como: el protocolo de red, el interfaz del *router* por el que llegue el paquete, el tamaño del paquete y la dirección de origen o destino.
 - Los paquetes que no se puedan clasificar serán asignados a la cola de prioridad normal

Figura 8. Esquema gráfico (PQ)



Fuente: *Santiago Felici*, "Evolución de mecanismos de calidad de servicio en los routers para servicios multimedia" Doctorado: Sistemas y servicios telemáticos, Universidad de Valencia, España 1999.

2.9.2 GPS, PGPS (WFQ) Y PFQ

La disciplina *GPS* (*Generalized Processor Sharing*) tiene dos propiedades importantes: primero, puede garantizar un servicio de retardo limitado extremo-a-extremo a una sesión cuyo tráfico está controlado por *leaky bucket*; y segundo, puede asegurar asignación justa de ancho de banda a todas las sesiones que tienen trabajos en espera, estén o no sometidas al control de tráfico. Mientras la

primera propiedad es importante para soportar tráfico *best-effort* y servicios jerárquicos de enlace compartido, la segunda propiedad es la base para aportar *GoS(Grade of Service)* al tráfico [30].

Debido a que *GPS* usa un modelo fluido ideal que no puede ser realizado en el mundo real, se han propuesto varios algoritmos de paquetes como aproximación a *GPS*. En la literatura se denominan *PGPS (Packet Generalized Processor Sharing)* a estas aproximaciones, aunque en otros casos también se denominan como *PFQ (Packet Fair Queueing)*. Entre los algoritmos que implementan *GPS(Generalized Processor Sharing)* en redes de paquetes destaca *WFQ (Weighted Fair Queueing)* que es actualmente el más popular método para ofrecer garantía de funcionamiento y es también conocido como *PGPS (Packet Generalized Processor Sharing)*. [30]

En este caso se aplicara la aproximación *PGPS* como parte del proyecto actual, es un método automatizado que provee una justa asignación de ancho de banda para todo el tráfico de la red, utilizado habitualmente para enlaces de velocidades menores a 2048 [Mbps]. *PGPS* ordena el tráfico en flujos, utilizando una combinación de parámetros. Por ejemplo, para una conversación *TCP/IP*, se utiliza como filtro el protocolo *IP*, dirección *IP* fuente, dirección *IP* destino, puerto de origen, etc. Una vez distinguidos estos flujos, el enrutador determina cuáles son de uso intensivo o sensibles al retardo, priorizándolos y asegurando que los flujos de alto volumen sean empujados al final de la cola, y los volúmenes bajos, sensibles al retardo, sean empujados al principio de la cola. *PGPS* es apropiado en situaciones donde se desea proveer un tiempo de respuesta consistente ante usuarios que generen altas y bajas cargas en la red, ya que *PGPS* se adapta a las condiciones cambiantes del tráfico en ésta. Sin embargo, la carga que significa para el procesador en los equipos de enrutamiento, hace de esta metodología poco escalable, al requerir recursos adicionales en la clasificación y manipulación dinámica de las colas [7]. Ver figura 9.

PGPS es adaptativo a los cambios en la red

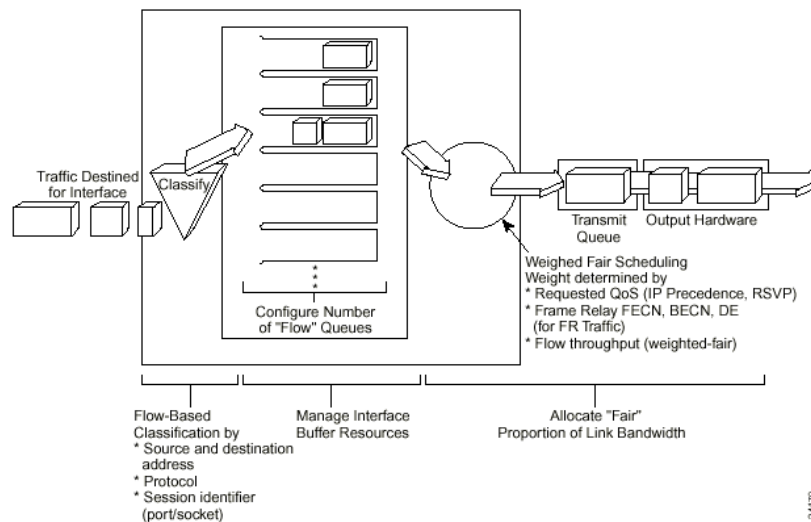
- Los mecanismos vistos anteriormente son estáticos, y por lo tanto no se adaptan a los cambios producidos en la red. Por ello ha sido necesario un mecanismo como *PGPS*, que es adaptativo.
- No proporciona garantías totales como *PFQ*.

Proporciona un buen tiempo de respuesta

- *PGPS* es adecuado para situaciones donde se necesite un buen tiempo de respuesta, para usuarios que hagan tanto un uso elevado de la red, tanto como para los que hagan un uso más leve, sin añadir ancho de banda adicional.
- Cisco lo utiliza por defecto en enlaces inferiores a T1 (1,5 Mbps)
PGPS es un algoritmo basado en flujos
- *PFQ* es un algoritmo de cola basado en flujos (o sesiones), que realiza dos tareas simultáneamente y de forma automática:
 - Organiza el tráfico (de tiempo real), poniéndolo al principio de la cola, reduciendo así el tiempo de respuesta.
 - Comparte equitativamente el resto del ancho de banda, entre el resto de tráfico de alta prioridad
- *PGPS* asegura que las diferentes colas no se queden privadas de un mínimo ancho de banda, de modo que el servicio proporcionado al tráfico es más predecible.

- Considera flujos de poco caudal con flujos sensibles al retardo, por ej. *VoIP*.
- No es escalable dentro de una gran red.

Figura 9. Esquema gráfico (*PGPS*)



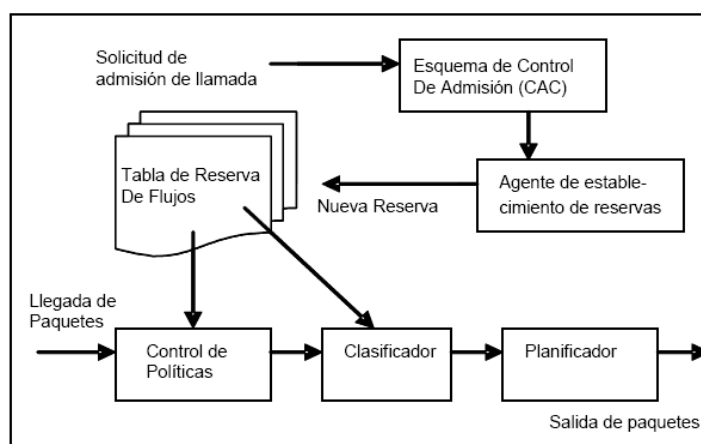
Fuente: *Santiago Felici*, "Evolución de mecanismos de calidad de servicio en los routers para servicios multimedia" Doctorado: Sistemas y servicios telemáticos, Universidad de Valencia, España, 1999.

2.10 MODELO DEL ROUTER DE ACCESO INALÁMBRICO

En la actualidad, la necesidad de movilizarse de un lugar a otro y estar continuamente actualizado es muy indispensable, por eso, las futuras redes celulares estarán basadas en la conmutación de paquetes y en el soporte de un alto manejo de información como lo es el manejo de voz, datos, video y/o aplicaciones en tiempo real como las video conferencias o juegos en línea, debido a estos estándares, las investigaciones apuntan al uso de la arquitectura de *QoS* (calidad de servicio) como *IntServ* e *IntServ6*, con el fin de optimizar estas arquitecturas.

Para lograr avances en esta investigación, un concepto clave a tener en cuenta es que, un *router* de *Servicios Integrados* esta compuesto principalmente por dos partes, la primera es el *gestor de las reservas*, el cual esta compuesto por el esquema de control de admisión (CAC) y el *agente de Establecimiento de las Reserva*. La segunda parte del *router* realiza el proceso de retransmisión de paquetes y esta compuesto por el proceso de *Clasificación* y el proceso de *Planificación*. Como se observa en la *Figura 10*, Estas dos partes principales están relacionadas entre si a través de la *Tabla de Reserva de Flujos*, la cual depende de control de admisión ya sea de nuevas llamadas o de traspaso que se lleve a cabo en el módulo CAC y determina la prestación del servicio en las demás etapas del *router*.

Figura 10. Modelo de un router de acceso inalámbrico



Fuente: Jhon Jairo Padilla, “*Contribución al soporte de calidad del servicio en redes móviles*” Doctorado Ingeniería Telemática, Universidad Politécnica de Cataluña 2008.

2.10.1 Modelos de control de acceso [8, 10, 24]

Este modelo asume que los procesos de Llegadas de *Nuevas Llamadas* y *Llamadas de Traspaso* son de tipo *Poisson*. Los tiempos de retención del canal

para *Nuevas Llamadas*, y para la *Llamadas de Traspaso*, se asumen independientes y exponencialmente distribuidos pero con diferentes valores medios. Sean λ_n y λ_h , $1/\mu_n$ y $1/\mu_h$ las tasas de llegada de *Nuevas Llamadas* y de *Traspaso* y valores medios de *tiempos de retención del canal* para *llamadas nuevas* y de *traspaso respectivamente*. Sean $\rho_n = \lambda_n / \mu_n$ y $\rho_h = \lambda_h / \mu_h$ las intensidades de tráfico de nuevas y llamadas de traspaso. Sea C el número total de canales en una celda. El modelo desarrollado en [10] obtiene para diferentes esquemas de *Control de Admisión de Llamadas (CAC)* los parámetros de desempeño a nivel de llamadas tales como: bloque de nueva llamadas e Interrupción de Llamadas de Traspaso.

En un *router* de la arquitectura de Servicios Integrados, el número de flujos afecta el límite de retardo medio de paquetes ya que el número de colisiones cambia. Como resultado, el número de llamadas promedio en un *router* de acceso inalámbrico para *IntServ* tiene un efecto importante sobre el límite de retardo medio de paquetes. Por tanto, se puede obtener una relación entre el límite superior de retardo medio de paquetes, los esquemas de control de admisión de llamadas o CAC y el modelo de movilidad.

2.10.1.1 Esquema de Limitación de Nuevas Llamadas (New Call Bounding Scheme)

Este esquema limita la admisión de nuevas llamadas en las redes inalámbricas y funciona de la siguiente forma:

Si el número de llamadas nuevas en una celda excede un umbral cuando una nueva llamada llega, la nueva llamada será bloqueada; de otra forma ésta será admitida. Una llamada de traspaso se rechaza sólo cuando todos los canales en la celda están siendo usados [8].

2.10.1.2 Esquema de Prioridades de Limitación (*Cutoff Priority Scheme*)

En este esquema en vez de colocar limitaciones al número de nuevas llamadas, se basa en el número total de llamadas establecidas en la celda para tomar la decisión, si una llamada entrante es aceptada o no [8].

El esquema *Cutoff Priority* funciona así:

Sea m quien denota el umbral sobre una nueva llamada, si el número total de canales ocupados es menor que m , la nueva llamada es aceptada, de lo contrario es bloqueada. Las llamadas de traspaso son siempre aceptadas a no ser que no haya canal disponible para las llamadas [8].

3. TEORÍA DE COLAS

Se ha demostrado recientemente que el tráfico tiene un comportamiento autosimilar o fractal en la mayoría de los servicios [23]. Históricamente, la modelización de tráfico tiene sus orígenes en la telefonía, y se ha basado casi exclusivamente en suponer casi independencia en los tiempos entre llegadas (fundamentalmente los modelos poissonianos), y duraciones exponenciales en los tiempos de los recursos. Los modelos telegráficos no son los más adecuados para simular tráfico *IP*, sin embargo se utiliza un modelo *Poisson*, que no tiene en cuenta la correlación entre llegadas, sin embargo es aplicado a la presente investigación como condición de observar la fiabilidad los resultados del *router* de servicios integrados del proyecto de grado actual con el modelo matemático simulado en [8].

Para simular las características básicas del tráfico *IP* actual, se desarrolló un modelo simulado de cola de comportamiento autosimilar y con dependencia a largo plazo [2].

A continuación se muestra un concepto de modelos de tráfico, como el modelo *Poisson* y un modelo *MMPP* de tráfico autosimilar, y luego se mostrara un modelo que captura las estructuras complejas de correlación en los tráficos simulados con el fin de obtener el parámetro de *Hurst* comprobando así la autosimilitud del simulador *MMPP*.

3.1 MODELOS DE TRÁFICO

3.1.1 Modelo *Poisson* [25]

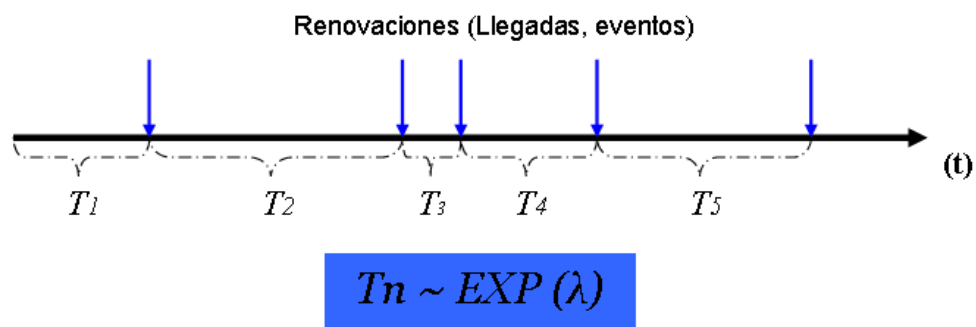
En teoría de probabilidad y estadística, la distribución de Poisson es una distribución de probabilidad discreta. Expresa la probabilidad de un número de

eventos ocurriendo en un tiempo fijo si estos eventos ocurren con una tasa media conocida, y son independientes del tiempo desde el último evento.

En otras palabras la propiedad del proceso de *Poisson* es la secuencia de tiempos entre llegadas T_n es independiente e idénticamente distribuida (iid) con distribución exponencial de parámetro λ , media $1/\lambda$. (Ver figura 11).

En la literatura de sistemas de espera, estos procesos de llegada se denotan por *GI* ("General independent").

Figura 11. Distribución Poisson



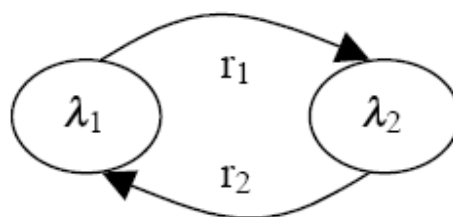
Fuente: Néstor Misael Peña Traslaviña, *Modelos de Trafico en Tiempo Real*, GEST – Grupo de Electrónica y Sistemas de Telecomunicaciones, Universidad de los Andes, 2006.

3.1.2 MMPP (*Markov Modulated Poisson Process*) [13]

El modelo de tasa modulada por *Markov* sintetiza la tasa de generación según una función de distribución de probabilidad que depende del estado en que se encuentre la cadena de *Markov*. Como caso particular, se emplea una distribución exponencial de generación de tasa dando lugar a *MMPP* (*Markov Modulated Poisson Process*). Dicho modelo combina la simplicidad de la conmutación markoviana con la de la generación poissoniana.

El modelo *MMPP* más común y que es base para el modelo aplicado en este proyecto de grado y es ilustrado en la *Figura 12*. Este modelo básico consta de dos estados es decir, un proceso de Poisson y los modelos ON/OFF, también llamado procesos Poisson interrumpidos (*interrupted Processes Poisson, IPP*). Los *IPP* son procesos de dos estados donde la tasa de generación poissoniana en un estado es nulo y el otro es constante en los cuales la generación sigue procesos de Poisson de tasas medias λ_1 y λ_2 , respectivamente. Por su parte, la permanencia en cada estado se rige por distribuciones exponenciales de valores r^{-1} y r_2^{-1} .

Figura 12. Modelo *MMPP* dos estados

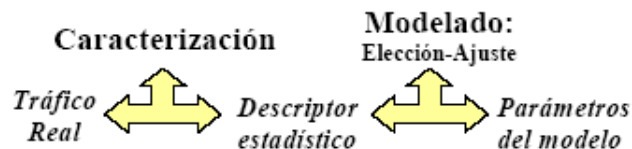


Fuente. Jurado, E. Casilari, A. Reyes, A. Díaz-Estrella y F. Sandoval, *Modelado Markoviano de Tráfico Agregado ATM, Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga, 1999*

Sin embargo, y tal como se ilustra en la *Figura 13*, el diseño de un modelo exige, previamente a la elección (en este caso el modelo *MMPP*) y ajuste de cierto proceso, que es llamado caracterización. La caracterización supone analizar la realidad a imitar (el tráfico) con el objeto de describirla mediante un conjunto reducido de variables (descriptor estadístico) que resumen y abstraen sus aspectos más importantes. Tras la definición de este descriptor, que, en ocasiones puede actuar directamente como un modelo en sí, la tarea de modelado se centra en elegir un proceso matemático que aproxime dicho descriptor, ya con independencia de la realidad física detallada que resume. El objetivo de esta fase,

tras decidir qué modelo se emplea, es una parametrización adecuada del mismo que asegure un ajuste óptimo del descriptor [13].

Figura 13. Proceso de diseño de un modelo



Fuente. Jurado, E. Casilari, A. Reyes, A. Díaz-Estrella y F. Sandoval, *Modelado Markoviano de Tráfico Agregado ATM*, Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga, 1999

3.2 AUTOSIMILITUD

La autosimilitud y la fractalidad describen el fenómeno en el que cierta propiedad de un objeto se preserva con respecto a la escalización en el tiempo o en el espacio. Este fenómeno sucede en imágenes naturales, en subdominio de convergencia de ciertos sistemas dinámicos y en muchas series de tiempo. En un objeto autosimilar o fractal, sus magnificadas se asemejan a la forma del objeto completo. Donde la semejanza se mide en algún sentido adecuado. [1]

Hoy día la autosimilitud se considera un concepto fundamental para comprender la naturaleza dinámica del tráfico, el desempeño de las redes y los procedimientos del control de tráfico que buscan proporcionar una calidad del servicio. [1]

3.2.1 Dependencia a corto plazo y largo plazo [15]

El fenómeno de *LRD* era conocido mucho antes que se desarrollara un modelo apropiado. En los más diversos campos se observaba empíricamente que muchas series de tiempo o espacio tienen funciones de autocorrelación que decaen a cero a una tasa mucho menor que la esperada en series de datos independientes o que siguen modelos clásicos como *AIRMA* o *Markov*.

La *dependencia a largo plazo* (*Long Range Dependence - LRD*) es una característica que está íntimamente relacionada con los procesos autosimilares y también indica que por muy pequeños que sean los valores de la función de autocorrelación, al aumentar la escala su acumulación no es menoscabable y pueden provocar efectos importantes. Dicho de otro modo, el tráfico medido en cualquier instante de tiempo depende del tráfico anterior. Así pues, los procesos *LRD* tienen memoria (a diferencia de los procesos Poissonianos) y sus componentes de baja frecuencia son muy importantes.

En cambio una función que presente *dependencia a corto plazo* (*Short Range Dependence - SRD*) no tiene memoria y por tanto sus componentes de baja frecuencia no son significativos.

3.2.2 Estimación del parámetro de Hurst [16]

Al simular tráfico como movimiento browniano fraccional, debemos ajustar el valor medio, la varianza y el parámetro de H para que correspondan con los respectivos estimadores estadísticos tomados de mediciones de tráfico real. Más allá de la lenta convergencia de los estimadores tradicionales para la media y la varianza, no existen mayor dificultad en obtener valores adecuados para estos dos parámetros (claro, si la ergodicidad es una suposición válida). Sin embargo, la estimación del parámetro H es mucho más fácil. Una de las técnicas más utilizadas es el diagrama varianza-tiempo. En efecto, tomando la varianza de ambos lados de la *ecuación 1*, obtenemos:

$$\text{Var}[X^{(m)}] = m^\beta \text{Var}[X] \quad (\text{ecuación 1})$$

$\beta = 2(1 - H)$ Tomando el logaritmo a cada lado de la *ecuación 2* obtenemos

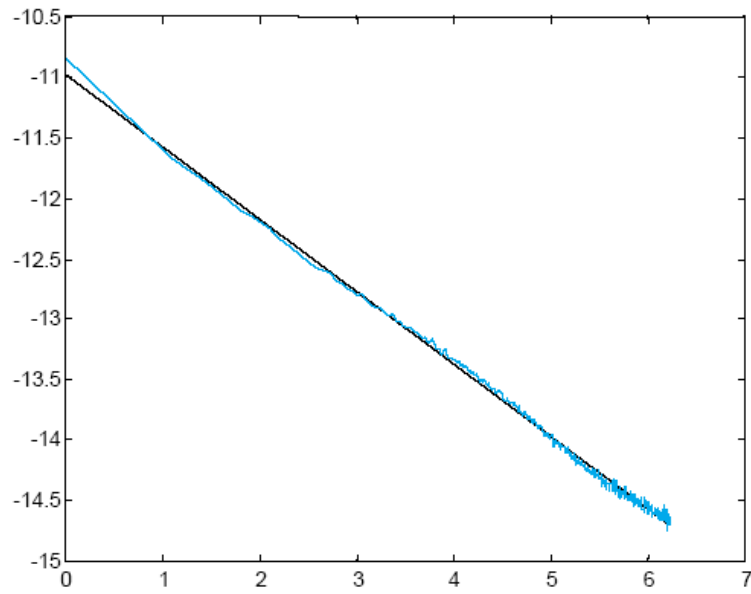
$$\log(\text{Var}[X^{(m)}]) = \log(\text{Var}[X]) - \beta \log(m)$$

(ecuación 2)

de manera que al graficar $\log(\text{Var}[X^{(m)}])$ contra $\log(m)$ obtenemos una curva cuya pendiente es $2(H - 1)$.

Cuando se tiene una serie de tiempo suficientemente larga como para estimar con suficientemente precisión de las series agregadas para un amplio rango de valores m , se podrá construir una grafica. Si detectamos algún tipo de alineación con una pendiente entre -1 y 0, podemos interpretarla como un fenómeno *LRD* y la pendiente (obtenida Regresión con Mínimos Cuadrados) resulta de la ecuación $2(H - 1)$. La *figura 14* muestra un ejemplo que se obtuvo de los 100000 primeros paquetes (tomado de los laboratorios *Bellcore*). La curva de color azul es el diagrama de varianza-tiempo y la curva negra es el ajuste de mínimos cuadrados, que produce un pendiente de 0.6, para una estimación de H igual a 0.7 (Nótese que la técnica de estimación no es solo validad para movimiento browniano fraccional sino para todo proceso *LRD*, para lo cual toca encontrar m_0 mínimo a partir del cual se detecte alineación antes de efectuar la regresión de mínimos cuadrados).

Figura 14. Grafica varianza-tiempo para la estimación del parámetro H.



Marco Aurelio Alzate Monroy, *Introducción al tráfico autosimilar en redes de comunicaciones*, Universidad Distrital Francisco José de Caldas, revista de ingeniería Vol 6 No. 2 año 2001.

PARTE 2 DESARROLLO DE LA PROPUESTA

4. INTRODUCCIÓN

5. DESCRIPCIÓN DE LOS PROGRAMAS

4. DESCRIPCIÓN DE LA METODOLOGÍA

4.1 INTRODUCCIÓN

Para desarrollar la presente investigación fue necesaria la ejecución de diferentes pasos que serán explicados a continuación.

Inicialmente se realizó una revisión del estado del arte en diferentes temas como: programación en *Matlab*, conceptos de Calidad de Servicio(QoS), la Arquitectura de Servicios Integrados, distribuciones de entrada del tipo *Poisson* y *MMPP*, métodos de control de acceso al medio del tipo *New Call Bounding Scheme* y *Cutoff Priority Scheme*, planificación de paquetes con mecanismos de cola del tipo *PGPS* y *PQ*, clasificación de paquetes en *IntServ* e *IntServ6*, procedimiento de establecimiento de la reserva en *IntServ6*, procesamiento de paquetes en un Router *IntServ6*.

En segunda instancia, el proyecto se realizó bajo unos supuestos o condiciones iguales a los utilizados en [8] con el fin de comprobar el modelo teórico simulado en [8] y desarrollado en [24].

Como software de simulación se utilizó *Matlab*. Esta es una herramienta que permite crear modelos de simulación fácilmente en el campo de las comunicaciones y posee grandes ventajas como: amplias librerías en comunicaciones y estadística, opciones de graficación, fácil programación, excelente desempeño, fácil verificación, modificación y adaptabilidad de código, sin mencionar la aceptación que tiene *Matlab*, como software para modelar y simular sistemas, lo que a su vez da sustento a la validez de la propuesta. La simulación del modelo propuesto en *Matlab*, se considera esencial ya que es la base para cualquier otra simulación derivada del proceso.

Los algoritmos se desarrollaron por etapas, están divididos en cuatro partes y a su vez incluyen subprogramas que ofrecen al usuario una amplia variedad de opciones con el fin de que la simulación se adapte a las necesidades ajuste a las necesidades. Estas partes son:

- Generación de tráfico IP
- Módulo de Control de Admisión (*CAC*)
- Módulo Clasificador
- Módulo Planificador

Se aclara que la primera parte es el Generador de tráfico IP y no hace parte de la estructura interna del router de *Servicios Integrados*.

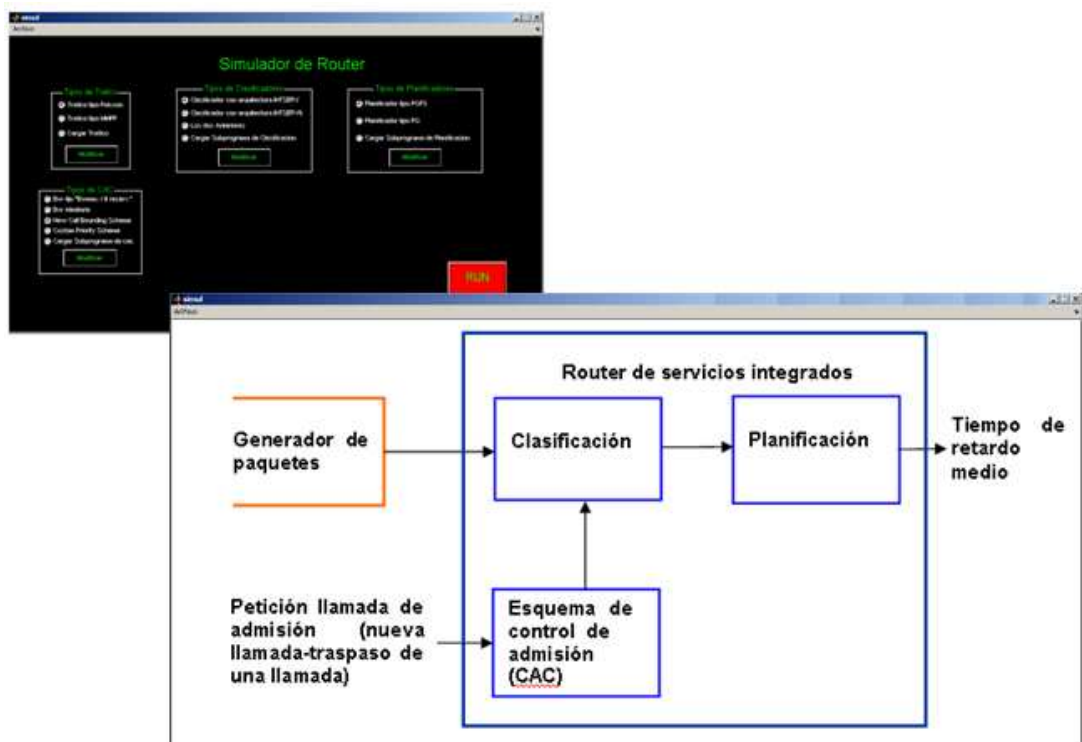
Por último, se elaboró la documentación de la Tesis, donde se describe toda la investigación realizada, análisis de las simulaciones, conclusiones, el manual de usuario y el artículo con el formato *IEEE* donde especifican los métodos que se utilizaron en la investigación y el análisis con las respectivas conclusiones.

5. DESCRIPCIÓN DE LOS PROGRAMAS

5.1 INTERFAZ DE USUARIO

La *figura 15* muestra el diagrama de bloques que representa la interfaz gráfica del simulador. Este simulador permite modificar las opciones de simulación y variables iniciales de cada uno de los módulos que componen este router; a continuación se especifican las opciones a modificar en cada módulo.

Figura 15. Módulos de la interfaz grafica del Router



5.1.1 Módulo de generación de paquetes (tráfico de entrada)

En este módulo, el usuario puede seleccionar el tipo de generación de tráfico que desea usar (*POISSON* ó *D-MMPP*). También permite modificar el número de usuarios con reserva, el número de paquetes a procesar y en el caso de ser seleccionado el generador de tráfico *D-MMPP*, permite la selección del número de ciclos de simulación; además puede seleccionar la opción de cargar tráfico, que le permite cargar una traza cualquiera guardada en un documento de texto como se especifica en el *Anexo B*.

5.1.2 Módulo de control de Admisión de Llamadas (CAC)

En este módulo, el usuario puede seleccionar el tipo de control de admisión de llamadas que desea usar (*Bw fijo*, *Bw aleatorio*, *New Call Bounding Scheme*, *Cutoff Priority Scheme*), dependiendo del tipo de router seleccionado, ya sea Inalámbrico ó Fijo. También permite seleccionar el tipo de simulación teniendo en cuenta el número de usuarios (Número de usuarios fijos ó incremento de usuarios); así como, el número de reservas inicial para llevar a cabo el incremento. Como condición inicial a modificar esta el *Bw* máximo en *Mbps*.

5.1.3 Módulo de Clasificación de Paquetes

En este módulo, se podrá seleccionar la *arquitectura de servicios integrados* que tipo de arquitectura se desea simular, ya sea *IntServ*, *IntServ6* o si desea simular los de dos tipos de arquitectura y como resultado se visualizan la graficas correspondientes a cada arquitectura

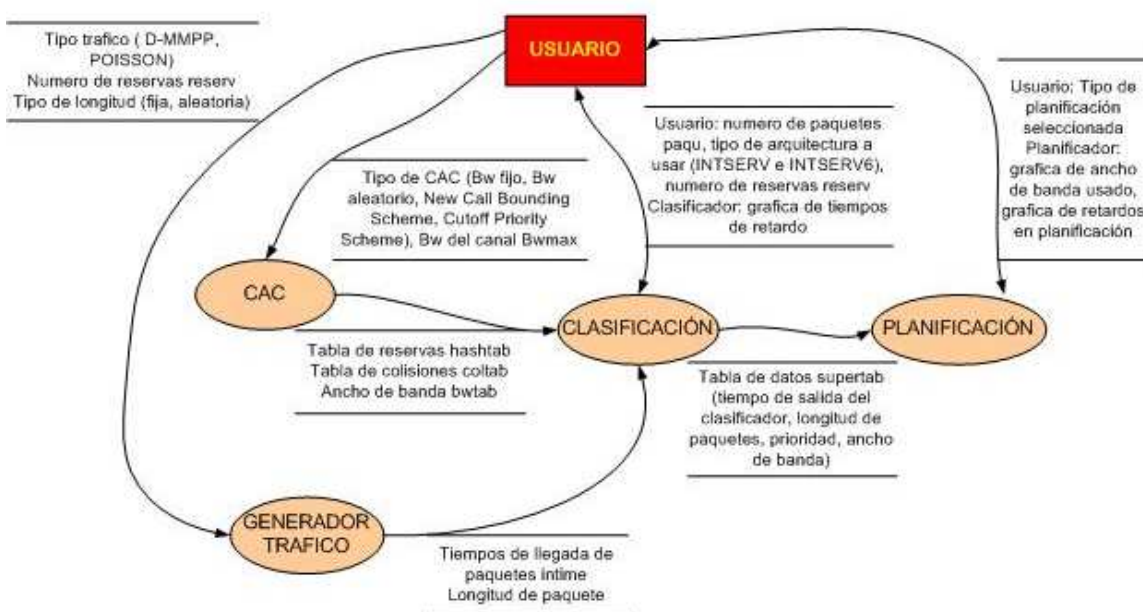
5.1.4 Módulo de Planificación de paquetes

En este módulo, se podrá seleccionar el tipo de planificador, ya sea *PGPS* (*Packet-by-Packet Generalized Processor Sharing*) o *PQ* (*Priority Queuing*).

5.2 DESCRIPCIÓN GENERAL DEL SISTEMA

La *figura 16* muestra cada una de las etapas que componen el simulador del router y la comunicación entre ellas.

Figura 16. Diagrama de flujo general del simulador



En la *figura 16* se puede observar la interacción que se da entre el usuario y las diferentes etapas de router. Inicialmente el módulo de control de admisión recibe datos del usuario, tales como (tipo control de admisión seleccionado y ancho de banda del canal), datos que permiten llevar a cabo el proceso de admisión, y por

tanto la creación de la tabla de reservas *Hash* y la *tabla de resolución de colisiones*.

Una vez se ha realizado el establecimiento de las reservas, se procede a iniciar el proceso de generación de paquetes, para lo cual, haciendo uso de la interfaz gráfica, el usuario ingresa datos tales como (tipo de generador de tráfico deseado, número de usuarios con reserva, número de paquetes a simular), entre otros, con el fin de generar una simulación de llegada de paquetes, que se aproxime al tráfico real de Internet, donde los datos simulados son guardados en la variable *intime*.

A continuación del proceso de generación de tráfico, se procede a iniciar la etapa de clasificación de paquetes, la cual implementa las variables obtenidas de la etapa de *CAC (tabla Hash y tabla de resolución de colisiones)*, y la tabla de tráfico *intime*, para llevar a cabo el proceso de clasificación, tanto para una arquitectura *IntServ*, como para *IntServ6*, este proceso entrega como resultado al usuario gráficas que indican el retardo de los paquetes en la etapa de clasificación, además de almacenar los tiempos de salida del clasificador en la tabla *supertab*, que va a ser usada por el módulo de Planificación.

Finalmente se lleva a cabo el proceso de Planificación de paquetes, ya sea para un esquema *PGPS (Packet-by-packet Generalized Processor Sharing)*, o un esquema *PQ (Priority Queuing)*, según indique el usuario en la interfaz gráfica; para lo cual, hace uso de la tabla *supertab* que contiene los tiempos de salida del clasificador, con sus respectivos anchos de banda y prioridades, entregando como resultado de este proceso al usuario, gráficas de ancho de banda usado por paquete, y tiempos de retardo de Planificación.

5.3 PROCESO DE TRÁFICO DE PAQUETES

Se utilizan dos tipos de generadores de tráfico, el modelo *Poisson* y el *D-MMPP*. El modelo *Poisson* es común por ser útil para sacar modelos matemáticos y se adapta a modelos de tráfico de datos por ráfagas con llegadas de ráfagas exponenciales. Adicionalmente, debido a que en la actualidad existe un gran manejo de información en Internet como lo es el video, sonido, y aplicaciones en tiempo real como video conferencias y video juegos en línea, se pretende simular estas demandas de información con el modelo *D-MMPP* que combina la simplicidad de la *conmutación Markoviana* con la de la generación *Poissoniana*. El modelo *D-MMPP* posee características de tráfico autosimilar que modelan de forma bastante cercana el tráfico actual de Internet [8, 33].

5.3.1 Modelo POISSON

5.3.1.1 Descripción

El modelo utilizado para este subprograma es el modelo de *Poisson*, que consiste en generar paquetes de información, donde los tiempos entre llegadas son de tipo exponencial y dependen de una tasa media λ .

Los datos que debe ingresar el usuario a través de la interfaz gráfica para el correcto funcionamiento de este módulo son:

- Número de usuarios con reserva
- Número de paquetes a simular
- Longitud de paquetes fijos y aleatorios

5.3.1.2 Variables Entrada-Salida

Las variables de entrada y salida son almacenadas en archivos, que posteriormente son cargadas en el módulo de clasificación. Estos archivos y variables de entrada se muestran en la *tabla 2* y son explicadas a continuación.

Datos de entrada:

La variable de entrada *paqord*, es una variable simple que guarda el número de paquetes y la variable simple *reserv* guarda el número usuarios con reservas.

Como parámetros predeterminados en el simulador se encuentra:

- Número de paquetes: 50 paquetes
- Ancho de banda: 64Kbps con longitud de paquete de 64bytes
- El número de reservas a simular es de 800 flujos.

Datos de salida:

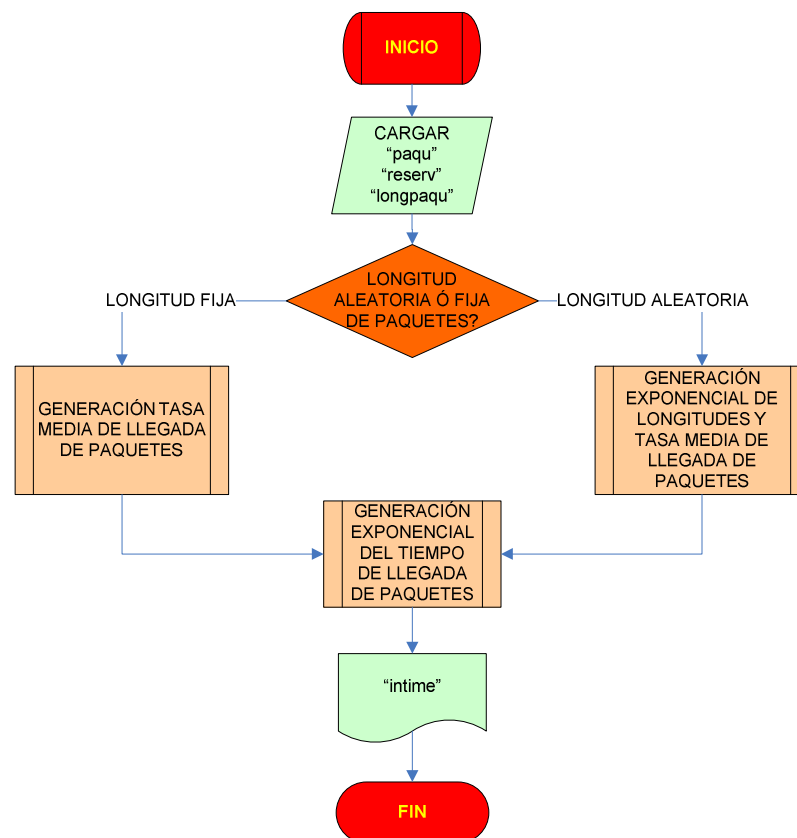
La variable de salida *intime* es una matriz de tamaño [paqux1], donde se guardan los tiempos de llegada de los paquetes, donde *paqu* es el número de paquetes seleccionado por el usuario en la interfaz grafica del simulador..

Tabla 2. Variable entrada-salida de Poisson

	Variable	Archivo
Datos Entrada	paqord	ordpaqu
	reserv	resnum
Datos Salida	intime	trafic

5.3.1.3 Algoritmo del módulo de Poisson

Figura 17. Diagrama del flujo del módulo de Poisson



A continuación se hace una explicación general del algoritmo que es mostrado en la *figura 17*.

Pasos Generales:

1. Hallar el tiempo medio de entre-llegada de paquetes
2. Generación de tiempos exponenciales
3. Crear variable que simula el tráfico de entrada al router

A continuación se explican los diferentes pasos generales:

Tiempo medio de entre-llegada de paquetes

De acuerdo a las variables de longitud de paquetes (*longpaqu*) y número de reservas (*reserv*), se calcula el tiempo medio de entre-llegada de paquetes λ_5 (*la5*) de la siguiente manera:

```
lambda1=(64*1024)/(longpaqu); % Donde lambda1 es la tasa media de
                                % llegadas de paquetes y longpaqu es la
                                % longitud del paquete.
lambda_i=lambda1*reserv;      % lambda_i es la tasa media de llegada de
                                % paquetes que dependerá del la cantidad de
                                % flujos 'reserv'.
la5=1/lambda_i;               % la5 es el tiempo medio entre-llegada de
                                % paquetes dado en segundos.
Xcin1=la5*1000;               % Escala de 1000 para obtener valores
                                % manejables.
```

El anterior segmento de código muestra la ecuación *lambda1* para calcular el tiempo medio entre-llegada de los paquetes, el cual depende de la longitud del paquete *longpaqu*. Posteriormente se crea *lambda_i*, que es el tiempo medio con el número de usuarios con reservas *reserv*. Este código aplica para dos casos:

1. La opción de longitud fija: esta variable debe ser ingresarla por el usuario a través de la interfaz gráfica del simulador y es guarda con el nombre de *longpaqu* y posteriormente se utiliza para determinar el tiempo medio de llegadas de los paquetes.
2. El otro caso es de longitud aleatoria, donde la longitud que se ingresa a través de la interfaz grafica del simulador, es la longitud promedio de los paquetes, haciendo uso de este valor ingresado el subprograma utiliza una función que genera longitudes aleatorias con distribución *Pareto* [34] con un valor mínimo de 64 bytes.

Generación de tiempos exponenciales

Una vez creado el tiempo medio entre-llegadas de paquetes X_{cin1} , se procede a generar los paquetes, como se muestra en el siguiente código:

```
Pin=0; % Inicialización del contador.
while Pin<paqord(1,1) % Realiza el conteo de uno en uno
    Pin=Pin+1; % hasta el número máx. de paquetes 'paqord'
    tone(Pin,1)=exprnd(Xcin1); % Conteo de posición del vector
    % La función R = EXPRND (MU) genera
    % exponencialmente tiempos aleatorios
    % con tiempo medio 'Xcin1' y son
    % guardados en 'tone'
end
ttwo=cumsum(tone); % La función CUMSUM(X) toma la matriz
% 'tone' sumando la fila siguiente con
% la anterior para obtener los tiempos
% de llegada de cada paquete y son
% guardados en la nueva matriz 'ttwo'.
```

Creación de la variable que simula el tráfico de entrada al router [8]

Una vez asignado un tiempo medio entre-llegadas de paquetes X_{cin1} exponencial a cada uno de los paquetes, se organizan uno tras otro, y posteriormente se crea la variable de salida *intime*, que contiene los tiempos de llegada paquetes generados y será utilizada en el módulo de clasificación.

5.3.2 Modelo D-MMPP

5.3.2.1 Introducción

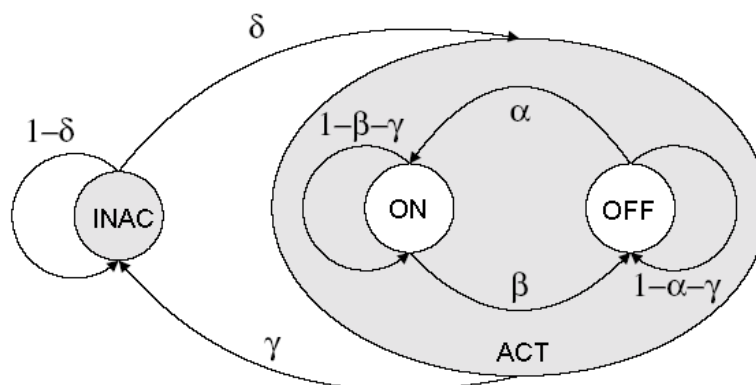
Debido a la necesidad de crear un modelo de tráfico que simule el comportamiento del tráfico de Internet, se utilizó un modelo de tráfico de tipo D-MMPP (*Proceso de Poisson Modelado por Markov Discreto*) para generar una traza discreta [1]. Este modelo cuenta con la característica de poseer un comportamiento *autosimilar*.

Para comprobar esto es necesario calcular un parámetro denominado *Hurst* mediante el método *diagrama varianza-tiempo* [1, 2, 31].

5.3.2.2 Descripción del Modelo [2]

El modelo *D-MMPP* representa aquellas aplicaciones de datos con comportamiento a ráfagas, cuya tasa de generación de información es variable en el tiempo y tiene unos requerimientos de retardo ajustados. Algunos ejemplos típicos de estas aplicaciones que se pretenden simular son aplicaciones de datos iterativos, como por ejemplo conversaciones en línea (chats) o juegos en red. En [2] se ha modelado este tipo de usuarios haciendo uso de una estructura *D-MMPP* mediante un esquema que presenta el comportamiento descrito en la *figura 18*.

Figura 18. Diagrama de estados *MMPP*



Fuente: Alonso Zárate, Luís Gonzaga "Técnicas de acceso y gestión de recursos para garantizar calidad de servicio en sistemas de comunicaciones móviles basados en CDMA" Tesis Doctoral-UPC, 2001

En el diagrama de estados de la *figura 18* los usuarios están inicialmente en el estado de reposo o silencio (*INAC*) en el que no generan bits de información. Con una tasa media (λ) se puede pasar desde el estado de reposo a un estado activo (*ACT*). En este estado se distinguen a su vez dos nuevos estados, uno de silencio (*OFF*) en el que tampoco se genera información, y otro de actividad (*ON*) en el

que se genera información a tasa aleatoria con distribución exponencial y con un valor medio λ . Visto de otro modo, el proceso de generación consiste en un modelo *ON-OFF* en el que la generación de paquetes de datos en los periodos *ON* no es constante si no que sigue un proceso de *Poisson* con llegada de paquetes exponenciales, es decir un modelo *ON-OFF* dentro de otro *ON*. Los estados *ON* y *OFF* representan las ráfagas pequeñas de información o de silencio que se generan dentro del estado activo. Podemos considerar que dentro de los periodos del estado *ACT* la tasa de generación es variable y que se caracteriza por una cierta tasa media que sería la equivalente al factor de actividad del modelo *ON-OFF* contenido en este estado. [2]

La interfaz gráfica del simulador solicita algunos datos para ejecutar la simulación del proceso *D-MMPP*. Estos datos son:

- Número de usuarios con reserva
- Número de Ciclos a simular
- Longitud de paquetes fijos y aleatorios.

Y como valores predefinidos se tienen:

- El número de reservas es de 800 flujos
- El número de Ciclos es: un ciclo
- La selección de longitud del paquete se encuentra inicialmente en “Longitud fija”.

Se debe aclarar que un *ciclo* comprende la suma de un solo tiempo de inactividad y un tiempo de actividad; por tanto, si el usuario desea aumentar el número de paquetes debe aumentar el número de ciclos.

5.3.2.3 Variables Entrada-Salida

Las variables de entrada y salida, son almacenadas en archivos como se muestra en la *tabla 3*, para posteriormente utilizarlas en los programas.

Datos de entrada:

La variable simple de entrada *reserv*, corresponde al número usuarios con reservas y la variable *tsim*, corresponde al número de *ciclos*. Estas variables son almacenadas en archivos para posteriormente ser cargadas en el subprograma.

Datos de salida:

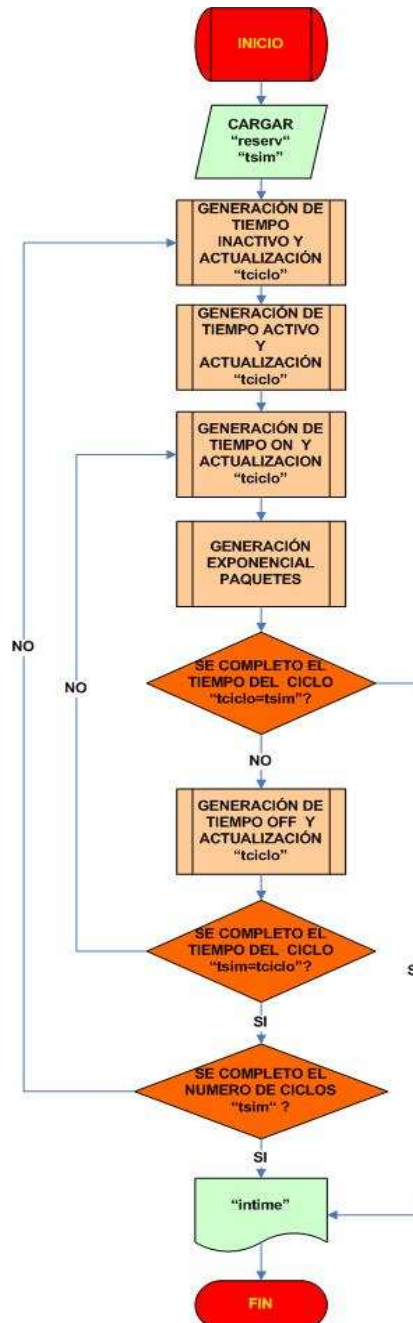
La variable de salida correspondiente a los tiempos de llegada de los paquetes se guarda en variable *intime*, y es una matriz de una columna y *reserv* filas, donde *reserv* es el número de usuarios con reservas. (dimensiones: [*reserv* x 1]).

Tabla 3. Variable entrada-salida de D-MMPP

	Variable	Archivo
Datos Entrada	<i>reserv</i>	<i>resnum</i>
	<i>tsim</i>	<i>ciclos</i>
Datos Salida	<i>intime</i>	<i>trafic</i>

5.3.2.4 Algoritmo del módulo de D-MMPP

Figura 19. Diagrama de flujo del módulo de D-MMPP



A continuación se hace una explicación general del algoritmo que es mostrado en la *figura 19*.

Pasos Generales:

1. Estimación del tiempo media de estado
2. Creación de los tiempos de permanencia de estado
3. Generación Tiempo medio de llegada de paquetes
4. Creación de la variable *intime*

A continuación se explican los diferentes pasos generales:

Estimación del tiempo media de estado

Las tasas λ_2 (*la2*) y λ_3 (*la3*) tomadas de [26], λ_5 (*la5*) tomada de [8], λ_1 (*la1*) y λ_2 (*la2*) determinadas de esta forma para generar una mayor cantidad de paquetes con el fin de obtener las características autosimilares; son utilizadas en el algoritmo de *D-MMPP*, e indican el tiempo medio de permanencia en los diferentes estados. Estas tasas de tiempo son:

VARIABLES *la*, *la1*, *la2*, *la3*, *la4* indica la media del tiempo de permanencia de estados:

```
la = 0.1;           % media del tiempo de inactividad.
la2 = 0.36;        % media de tiempo de ton.
la3 = 0.54;        % media de tiempo toff.
la4 = 0.9;         % media de tiempo de actividad.

                    % y lambda5 (la5):
la5 = 1/lambda_i   % Tiempo medio entre-llegada de paquetes en
                    % segundo, donde lambda_i es el tiempo medio
                    % entre-llegada de paquetes.
```

Los valores de las tasas se asignaron como se explica a continuación:

La interfaz gráfica del simulador solicita datos para ejecutar la simulación del proceso *D-MMPP*, donde uno de ellos es Número de Ciclos que comprende la suma de un solo tiempo medio de permanencia en estado inactivo mas uno activo. Si se desea simular varias veces el programa, solo se le incrementa la cantidad *Ciclos*, obteniendo más cantidad de paquetes. Cada ciclo de simulación tiene un tiempo máximo de duración de 1s; en este segundo se repartirán los tiempos para cada subestado teniendo en cuenta los tiempos de permanencia de cada uno, estos tiempos son multiplicadas por una escala que depende del tiempo máximo que se le ha otorgado a cada subestado, calculando así el tiempo promedio de permanencia en cada uno. Los porcentajes de tiempo elegidos fueron tomados del modelo de voz. Esta distribución se basa en los tiempos medios del modelo de voz en paquetes para los estados *ON* y *OFF* descrito en [26]. La duración del intervalo de silencio (*OFF*) es una *función de densidad de probabilidad (fdp)* exponencial con duración media $1/\lambda = 600ms$ [26] y a la duración del intervalo de habla (*ON*) es una *fdp* exponencial con media $1/\mu = 400ms$ [26]. A partir de esto y asumiendo un 100% equivalente a 1seg; al estado *ON* se le otorga una duración en media del 40% y al estado *OFF* se le otorga una duración en media del 60% del tiempo de actividad ACT, el cual es equivalente en media al 90% del tiempo de un Ciclo, el 10% restante determina el tiempo medio de inactividad INACT.

Por ultimo, para generar los tiempos entre llegada de paquetes en el estado *ON*, se utilizó una tasa λ_5 (*la5*) que es la misma usada en el modelo de *Poisson* [8].
Ver sección 5.3.1.

Creación de los tiempos de permanencia de estado

En el siguiente segmento de código, se muestra la forma en que se generaron los tiempos para el caso del tiempo inactividad *INACT*. Este mismo principio se aplica los otros estados (*ACT*, *ON*, *OFF*), aplicando una distribución exponencial haciendo uso del tiempo medio de permanencia λ correspondiente a cada estado estado.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TIEMPO INACTIVO: los tiempos inactivos dependen
% de una tasa "la" que genera los tiempos
% aleatorios de tipo exponencial.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while tinact>=tmax           % Genera un tiempo de inactividad
                             % menor que tmax(tiempo medio)
    expinact=exprnd(la);      % Matriz de números aleatoria de
                             % distribución exponencial donde,
                             % R = EXPRND (MU) convierte un vector
                             % de los números aleatorios de
                             % distribución exponencial con el
                             % parámetro MU.
    tinact=round(expinact... % multiplica el tiempo exponencial
                 *scal);    % generado en 'expinact' por la escala
                             % 'scal' y Redondea hacia el número
                             % entero más cercano.
    if tinact<(.06*scal)     % compara si el tiempo exponencial es un
                             % valor menor a (.06*scal), si es menor
                             % continúa en while hasta que sea mayor.
                             % scal es la escala de tiempo.
        tinact=tmax+1;      % permite continuar en el while
                             % convirtiendo a tinact mayor que tmax
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Generación Tiempo medio entre-llegada de paquetes

De acuerdo a las variables de longitud de paquetes (*longpaqu*) y número de reservas (*reserv*), se calcula el tiempo medio de entre-llegada de paquetes λ_5 (*la5*), de la misma forma que en modelo *Poisson* descrito [8]. Ver sección 5.3.1.3.

Creación de la variable *intime*

Después de generar los tiempos de llegada de los paquetes, se guardan en la variable de salida *intime* para utilizarse como datos de entrada en el clasificador del router. Esta variable es una matriz de una columna por *reserv* filas [*reservx1*], siendo *reserv* número de usuarios con reserva.

5.3.3 Estimación del parámetro de Hurst

5.3.3.1 Descripción

Este estimador es utilizado para determinar la autosimilitud de las muestras en el tiempo; los procesos autosimilares producen un incremento del índice de dispersión de la forma $m^{-1}t^{2(H-1)}$, graficando $\log(\text{var}(t))$ contra $\log(t)$ [27], donde *var* es la varianza y *t* es el tiempo. El *diagrama varianza-tiempo* [16], determina el parámetro *Hursts* (*H*), que dependerá de la alineación de la pendiente *m*, esto es explicado mas adelante con detalle.

5.3.3.2 Variable entrada-salida

Las variables de entrada y salida, son almacenadas en archivos como se muestra en la *tabla 4*, para posteriormente utilizarlas en los programas.

Datos de entrada:

La variable de entrada llamada *intime*, que guarda los tiempos de entre-llegada de paquetes, es una matriz de una columna por *paqord* (número de paquetes) que va hacer el numero de filas y se representa de la forma [*paqordx1*].

Datos de salida:

La variable de salida H , es el parámetro de Hurts. Si H se encuentra en el rango $0.5 < H < 1$, se dice que el tráfico de salida es autosimilar [16, 33].

Tabla 4. Variable entrada-salida del parámetro H

	Variable	Archivo
Datos Entrada	<i>Intime</i>	<i>trafic</i>
Datos Salida	<i>H</i>	<i>auto</i>

5.3.3.3 Algoritmo del Diagrama Varianza-tiempo

A continuación se hace una explicación general del código de simulación que es mostrado en la *figura 20*.

Pasos Generales:

1. Grafica *Log-Log*
2. Parámetro de H

A continuación se explica con los diferentes pasos generales:

Grafica log-log

Se debe tener una serie de tiempo suficientemente larga como para estimar con suficientemente precisión las series agregadas para un amplio rango de valores m . Se podrá construir una grafica (*ver figura 14*) aplicando la siguiente función tomada de [16]:

```
plot(log(1:N),log(v),'r-',[0 log(N)],[p(2) p(2)+log(N)*p(1)],'b-')
```

Donde v es la varianza calculada, p es el resultado de calcular mínimos cuadrados y N es el número de niveles de agregación que se ingresa por teclado.

Parámetro H

Si detectamos en la grafica *log-log* (ver figura 18) algún tipo de alineación con una pendiente entre -1 y 0, podemos interpretarla como un fenómeno de dependencia de largo plazo (*LRD*). Despejando la ecuación de la pendiente resultante $2(H - 1)$, obtenemos el parámetro H que determina el grado de *autosimilitud*; se debe estar en un rango entre $0.5 < H < 1$, donde un valor de $H=0.5$ indica ausencia de *autosimilitud*. Cuanto más cerca de 1 esté H , mayor será el grado de persistencia y de la dependencia de largo plazo. Este algoritmo fue tomado de [16] y se observa en la figura 20.

Figura 20. programa estimación del parámetro H

```
function H=VarianzaTiempo(NP,N)
% function H=VarianzaTiempo(NP,Nm)
% Este función calcula el diagrama varianza-tiempo de los tiempos entre llegadas
% de los primeros NP paquetes de los datos de Bellcore usando Nm niveles de agregación.
% Retorna el estimado de H, calculado mediante regresión de mínimos cuadrados
%
% Marco A. Alzate, Universidad Distrital, 2002

load Bc_paug0
x=Bc_paug0(2:NP+1,1)-Bc_paug0(1:NP,1); % Los datos originales traen el tiempo de llegada
v(1)=var(x); % Calcula los tiempos entre llegadas
for m=2:N % Varianza de la secuencia total
    Ns=floor(NP/m); % Número de niveles de agregación
    y=zeros(Ns,1); % Número de muestras en esta escala de agregación
    for i=1:Ns % Señal agregada
        y(i)=sum(x(1+m*(i-1):m*i))/m; % Calcula las sumas acumuladas
    end
    v(m)=var(y); % y calcula la varianza a este nivel de agregación
end
p=polyfit(log(1:N),log(v),1); % Regresión de mínimos cuadrados
plot(log(1:N),log(v),'r-',[0 log(N)],[p(2)+log(N)*p(1)],'b-') % Grafica los resultados
H=p(1)/2 + 1; % Estimación de H
```

Fuente: Marco Aurelio Alzate Monroy, *Introducción al tráfico autosimilar en redes de comunicaciones*, Universidad Distrital Francisco José de Caldas. Revista de ingeniería Vol. 6 No. 2 año 2002.

5.4 ESQUEMA DE CONTROL DE ADMISIÓN DE LLAMADAS CAC (CALL ADMISSION CONTROL)

5.4.1 Etapa de control de admisión de llamadas utilizando el modelo Ancho de Banda Fijo

5.4.1.1 Descripción

Este subprograma es el más sencillo, parte de un ancho de banda máximo que se divide en el número de reservas totales.

La interfaz gráfica del simulador solicita un dato para ejecutar la simulación del proceso de ancho de banda fijo; este dato es el ancho de banda máximo dado en *Mbps* y toma un valor por defecto de *10Mbps*.

5.4.1.2 Datos de entrada-salida:

Las variables de entrada y salida, son almacenadas en archivos como se muestra en la *tabla 5*, para posteriormente utilizarlas en los programas.

Datos de entrada:

Una de las variables de entrada que se usan en este subprograma es el número de usuarios con reservas, que se guarda en una variable simple llamada *reserv*. Para modificar esta variable, el usuario debe dirigirse a la interfaz gráfica del simulador en las opciones de tráfico y modificar las condiciones iniciales, ya que el número de reservas usado en la etapa de tráfico es el mismo que se utiliza para este subprograma. Otra variable de entrada es el ancho de banda máximo, que se encuentra guardado en la variable simple *bwmax*, esta variable simple corresponde al ancho de banda del canal, el cual se distribuye equitativamente en todos los usuarios.

Datos de salida:

La variable de salida es *bwtab* y es una matriz de una columna por *reserv* filas de la forma [*reserv*x1], donde *reserv* es el número usuarios con reserva.

Tabla 5. Variable entrada-salida BW Fijo

	Variable	Archivo
Datos Entrada	<i>reserv</i>	<i>resnum</i>
	<i>bwmax</i>	<i>maxbw</i>
Datos Salida	<i>bwtab</i>	<i>ancho</i>

5.4.1.3 Algoritmo del módulo de Ancho de Banda Fijo

Figura 21. Diagrama de flujo del módulo de Ancho de Banda Fijo



A continuación se hace una explicación general del algoritmo que es mostrado en la *figura 21*.

Pasos Generales:

1. Calcular el ancho de banda fijo para cada reserva
2. Creación de la variable *bwtab*

A continuación se explican los diferentes pasos generales:

Calcular el ancho de banda fijo para cada reserva

Básicamente se divide el ancho de banda del canal *bwmax* entre el número de reservas para así obtener un valor de ancho de banda igual *bw_fiji* para todos los usuarios. Se calcula con la *ecuación 3*.

$$bw_fiji = bwmax/reserv \quad (ecuación 3)$$

donde *bwmax* es el ancho de banda del canal dado Mbps y la variable *reserv* es el número de usuarios con reserva que dividirá el ancho de banda del canal.

Creación de la variable bwtab

Se crea una matriz de una columna con tamaño de *reserv* filas de la forma:

```
for n=1:reserv
    bwtab(n,1)=bw_fiji;
end
```

donde se obtendrán los anchos de banda para cada reserva que serán guardados en la matriz *bwtab*.

5.4.2 Etapa de control de admisión de llamadas utilizando el modelo de ancho de banda aleatorio

5.4.2.1 Descripción

Los anchos de banda para este subprograma, se generan aleatoriamente hasta completar un 100% de Bw del canal, ingresado por el usuario en la interfaz grafica del simulador; esto se logra asignando un Bw medio de 0.01% para cada reserva, obteniendo así el número total de reservas admitidas.

En la interfaz gráfica del simulador solicita un dato para ejecutar la simulación del proceso de ancho de banda aleatorio, este dato es el ancho de banda máximo dado en *Mbps* y como parámetro predeterminado se encuentra *10Mbps*.

5.4.2.2 Datos de entrada-salida:

Las variables son almacenadas en archivos para posteriormente utilizarlas en los programas, estas variables se muestran en la *tabla 6*.

Datos de entrada:

La variable simple de entrada *bwmax*, guarda el ancho de banda máximo del canal que es ingresado por usuario en la interfaz grafica del simulador.

Datos de salida:

La variable *bwtab*, es la matriz de datos resultante del programa, los Anchos de Banda para cada reserva se distribuyen en la primera columna por *reserv* filas [*reservx1*], donde *reserv* es el número total de reservas.

Tabla 6. Variable entrada-salida BW Aleatorio

	Variable	Archivo
Datos Entrada	<i>bwmax</i>	<i>maxbw</i>
Datos Salida	<i>bwtab</i>	<i>ancho</i>

5.4.2.3 Algoritmo del módulo de Ancho de Banda Aleatorio

Figura 22. Diagrama de flujo del módulo de Ancho de Banda Aleatorio



A continuación se hace una explicación general del algoritmo que es mostrado en la figura 22.

Pasos Generales:

1. Generación de ancho de banda aleatorio
2. Control de admisión para anchos de banda aleatorios.

A continuación se explican los diferentes pasos generales:

Generación de ancho de banda aleatorio

Haciendo uso de la función *normrnd* de *Matlab*, se hace un arreglo de datos aleatorios de distribución normal y se aplica de siguiente forma:

$$r = \text{normrnd}(\mu, \sigma) \quad (\text{ecuación 4})$$

Donde μ es la posición de la *campana de Gauss*, es decir, la media y σ será el parámetro de dispersión o desviación estándar, cuanto menor sea este valor, mayor será la cantidad de masa de probabilidad que habrá concentrada alrededor de la media, esto con el fin de generar una matriz de datos que almacena los anchos de banda otorgados a cada reserva.

En este programa utiliza una media de $\mu = 0.01$ y centralización de $\sigma = 0.0009$, donde μ es un porcentaje mínimo que corresponde a una centésima parte del ancho de banda total del canal y σ se le ha determina el valor de 0.0009 con el fin de no generar pocos usuarios con reservas. La *ecuación 4* es una función de *Matlab*, que genera números aleatorios alrededor de μ y con un valor máximo de $\mu + \sigma$ y un valor mínimo de $\mu - \sigma$; para evitar que se generen anchos de banda muy pequeños, se restringe el valor mínimo a $B_{wmin} = 0.01 * B_{wmax}$, siendo B_{wmin} el ancho de banda mínimo y B_{wmax} el ancho de banda máximo.

Control de admisión para anchos de banda aleatorios

En esta etapa se admite un número limitado de reservas, otorgando un Ancho de Banda a cada uno. El número de reservas depende de un umbral b_{wmax} , que no permite exceder el ancho de banda asignado a cada reserva y que es ingresado por el usuario en la interfaz gráfica del simulador. Cada vez que se desee admitir

una reserva, se debe preguntar si la sumatoria de los anchos de banda asignados superó el ancho de banda total; si es superado, finalizará la asignación de Bw a las reservas y se procederá a crear el registro final en la variable *bwtab de los Bw*; si no, se continúan admitiendo las reservas.

5.4.3 Etapa de control de admisión de llamadas utilizando el esquema de New Call Bounding

5.4.3.1 Descripción

El esquema consiste en controlar las llamadas que serán admitidas ya sea llamadas nuevas o llamadas de traspaso para posteriormente ser procesadas por el clasificador y el planificador de paquetes. El proceso de admisión para *New call bounding scheme* desarrollado por *Yuguang Fang* en [10] se realiza de la siguiente forma:

Si el número de Llamadas Nuevas excede un valor de umbral cuando llega una llamada, la nueva llamada será bloqueada; en otro caso, la llamada será admitida. Por otro lado, una llamada de traspaso es rechazada solo cuando todos los canales en el router están siendo utilizados.

5.4.3.2 Datos de entrada-salida:

Datos de entrada:

En este caso el usuario no ingresa valores, solo selecciona en la interfaz gráfica del simulador alguna opciones para ejecutar la simulación del módulo de *New Call Bounding Scheme*. Estas opciones son:

- variar el parámetro ρ_n (*ro_n*) (tasa media de llegadas de n uevas llamadas).
- variar el parámetro de ρ_h (*ro_h*) (tasa media de llamadas de traspaso).

Datos de salida:

La variable de salida *admit*, que guarda los datos de las reservas admitidas, donde es una matriz, que se distribuye en tres columnas por C (canales) filas de la forma $[Cx3]$. Esta matriz se muestra como un ejemplo en la *tabla 7*.

Tabla 7. Matriz de salida *admit* para los esquemas *New Call Bounding* y *Cutoff Priority*

	1	2	3
C	Tiempos	Estado	T. Permanecía
1	t_1	x	tp_1
2	t_2	x	tp_2
3	t_3	x	tp_3
4	t_4	x	tp_4
5	t_5	x	tp_5
"	"	"	"
"	"	"	"
30	t_n	x	tp_n

La primera columna son los tiempos de petición t_n , ya sea de llamadas nuevas o llamadas de traspaso, donde los tiempos son organizados de forma ascendente; la segunda columna identifica el estado x , ya sea de nueva llamada que se identifica con el número uno o si es llamada de traspaso con el número cero) y una tercera columna guarda los tiempos de permanencia tp_n para cada estado. La tabla 7. Se organizo de esta forma para tener una lógica de tal manera que los demás subprogramas puedan usar la información que en esta se almacena.

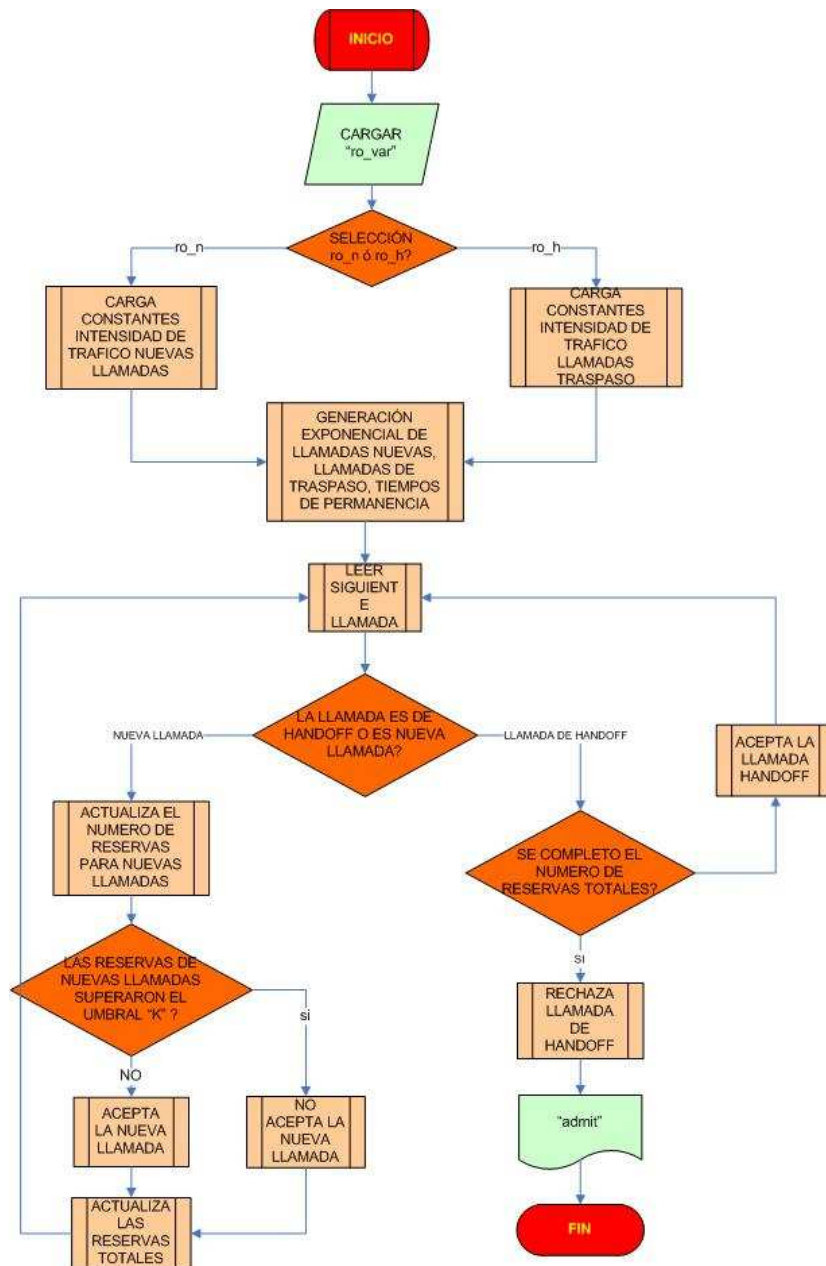
La variable de salida es almacenada en un archivo para posteriormente utilizarla en los programas, estas variables se muestran en la *tabla 8*.

Tabla 8. Variable de salida de los esquemas New Call Bounding y Cutoff Priority

	Variable	Archivo
Datos Salida	admit	CAC

5.4.3.3 Algoritmo del módulo de New Call Bounding Scheme

Figura 23. Diagrama de flujo del módulo de *New Call Bounding Scheme*



A continuación se hace una explicación general del algoritmo que es mostrado en la *figura 23*.

Pasos Generales:

1. Generación de llamadas
2. Control de admisión de llamadas

A continuación se explican los diferentes pasos generales:

Generación de llamadas

Según las opciones (variación de la intensidad tráfico de nueva llamada ro_n y variación de intensidad de tráfico de llamada de traspaso ro_h), que son seleccionadas por el usuario en la interfaz grafica del simulador, según la opción, se aplican las variables internas según el caso.

Estas variables se encuentran en el código fuente como predeterminada, cumpliendo con el modelo de *Yuguang Fang* [10] y son listadas a continuación:

```
ro_variable=1;
if ro_variable==variable

% Variando el parámetro 'ro' (200 a 600) con un ro_h=15
% 'ro' intensidad de tráfico de nuevas llamadas.

    lan=30;           % Tasa medio de nueva llamada
    t_lan=1/lan;     % Tiempo media de nueva llamada

    lah=30;          % Tasa media de llamada de traspaso (Handoff)
    t_lah=1/lah;    % Tiempo medio de llamada de traspaso
    muh=450;        % Tasa media de permanencia de traspaso
    t_muh=1/muh;   % Tiempo media de permanencia de traspaso

    c=30;           % Número de canales en el router
    k=15;           % Umbral de nueva llamada
    fin_ro=20;     % Finaliza la intensidad de tráfico
```

```

else
% Variando el parámetro de 'ro_h' entre (100 a 1200), con un ro_n=15;
% 'ro_h' intensidad de tráfico de llamadas de traspaso.

    lan=20;           % Tasa media nueva llamada
    t_lan=1/lan;     % Tiempo media de nueva llamada
    mun=300;         % Tasa media de permanencia de traspaso
    t_mun=1/mun;    % Tiempo media de permanencia de traspaso
    lah=30;          % Tasa medio de llamada traspaso(Handoff)
    t_lah=1/lah;    % Tiempo media de llamada traspaso

    c=30;            % número de canales
    k=20;            % umbral de nueva llamada
    fin_ro=40;       % finaliza la intensidad de tráfico
end

```

Para generar peticiones de llamadas se utiliza la distribución de tipo exponencial con una tasa de nueva llamada λ_n (la_n) y una tasa de llamadas de traspaso λ_h (la_h) y cada tipo de llamada tiene un tiempo de retención del canal $1/\mu_n$ $1/\mu_h$ respectivamente. También posee una identificación que es el número *Hash*, que se genera aleatoriamente que es asignado a cada llamada. Esto se muestra en el segmento de código siguiente:

```

for i=1:25
    llegada_tn(i,1)=exprnd(la_n); % Con la función 'exprnd' se crea
    % los tiempos de petición de nuevas
    % llamadas exponencialmente y
    % aleatorias con una tasa media
    % 'la_n' que se guardan en la matriz
    % llegada_tn'.
    llegada_th(i,1)=exprnd(la_h); % Crea los tiempos de llamadas de
    % handoff exponencialmente y
    % aleatorias con una tasa media
    % 'la_h' y se guarda en la matriz
    % llegad_th.
end
sum_lan=cumsum(llegada_tn); % Con la función 'cumsum' suma los
% tiempos de llamadas nuevas de los
% usuarios 'llegada_tn' y son
% guardados en un nuevo vector
% 'sum_lan'.
sum_lah=cumsum(llegada_th); % Con la función 'cumsum' suma los
% tiempos de llamadas de traspaso
% 'llegada_tn' y son guardados
% en un nuevo vector 'sum_lah'
for j=1:25
    stan_tn(j,1)=exprnd(mu_n); % Se crea los tiempos de permanencia

```

```

% de nuevas llamadas con una tasa
% media de mu_n, que es guardado en
% la matriz stan_tn.
stan_th(j,1)=exprnd(mu_h);
% Crea los tiempos de permanencia
% de llamadas de traspaso o handoff
% con una tasa media de mu_n, que es
% guardo en la matriz stan_th.
end
end

```

Como se desea calcular el número de llamadas (*New Call* ó *Handoff*) admitidas con respecto a la intensidad de tráfico, esto se determina variando $\rho_n = \lambda_n / \mu_n$ y $\rho_h = \lambda_h / \mu_h$, la intensidad de tráfico para los caso de nueva llamada y de traspaso respectivamente.

Control de admisión de llamadas

Este esquema limita la admisión de nuevas llamadas en las redes inalámbricas y funciona así:

Si el número de llamadas nuevas en una celda excede un umbral K cuando una nueva llamada llega, la nueva llamada será bloqueada; de otra forma ésta será admitida. Una llamada de traspaso se rechaza sólo cuando todos los canales en la celda están siendo usados.

5.4.4 Etapa de control de admisión de llamadas utilizando el esquema Cutoff Priority

5.4.4.1 Descripción

Este modelo asume procesos de Llegadas de Nuevas Llamadas y Llamadas de Traspaso de tipo Poisson. Los tiempos de retención del Canal para Nuevas Llamadas, y para la Llamadas de Traspaso, se asumen Independientes y exponencialmente distribuidos pero con diferentes valores medios.

5.4.4.2 Datos de entrada-salida

Datos de entrada:

En este caso el usuario no ingresa valores, solo selecciona en la interfaz gráfica del simulador alguna opciones para ejecutar la simulación del módulo de *Cutoff Priority Scheme* desarrollado por *Yuguang Fang* en [10]. Estas opciones son:

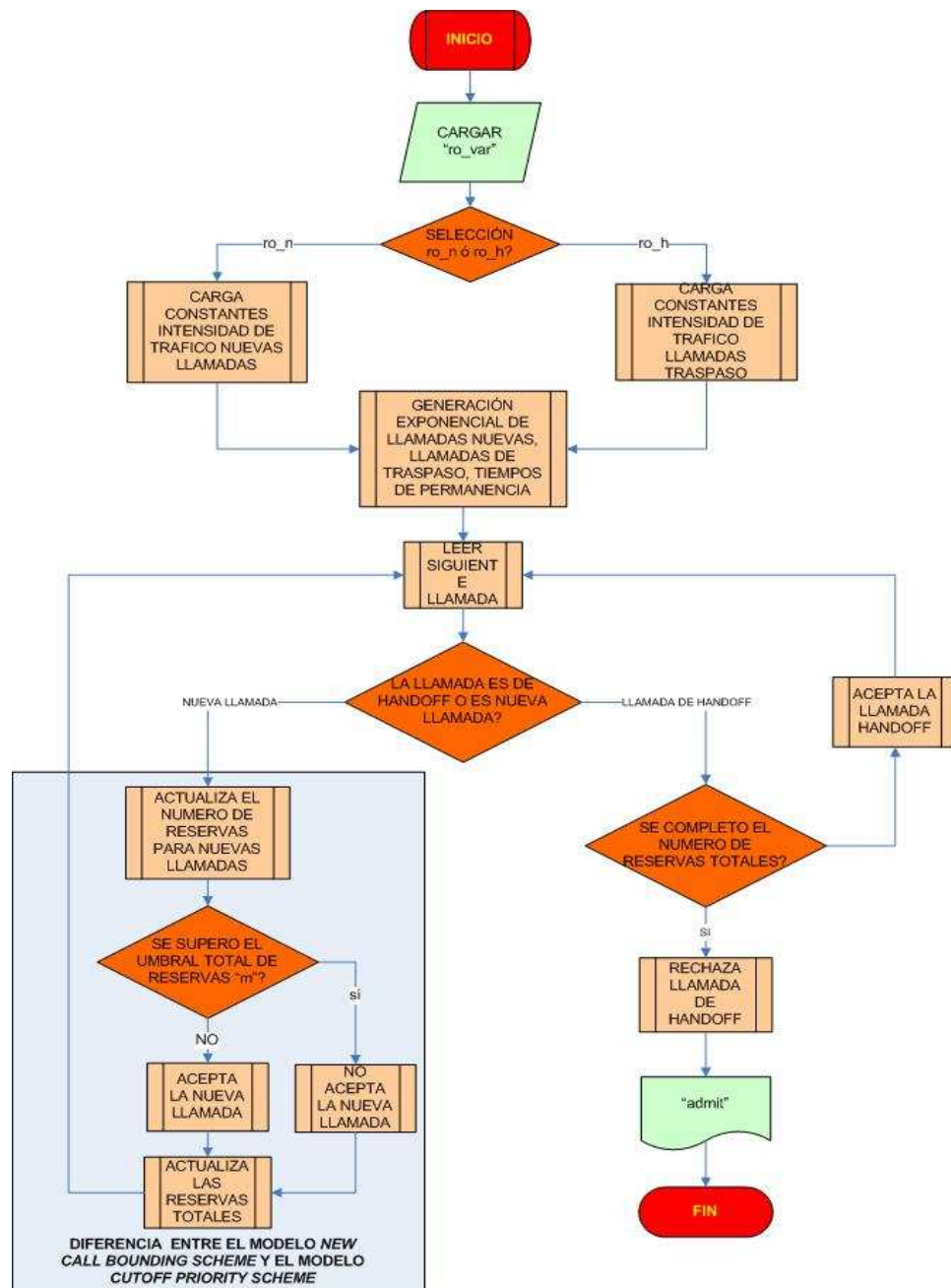
- variar el parámetro ρ_n o ro_n (tasa media de llegadas de nuevas llamadas).
- variar el parámetro de ρ_h o ro_h (tasa media de llamadas de traspaso).

Datos de salida:

La variable de salida *admit* es una matriz, distribuida en cuatro columnas por *C* canales [4xC]. Esta Variable es la misma del programa anterior y es explicada para el programa *New Call Bounding Scheme*, para una explicación detallada referirse a la *sección 5.4.3.2*.

5.4.4.3 Algoritmo del módulo de Cutoff Priority Scheme

Figura 24. Diagrama de flujo del módulo de Cutoff Priority Scheme



A continuación se hace una explicación general del algoritmo como se muestra en la figura 24.

Pasos Generales:

1. Generación de llamadas
2. control de admisión de llamadas

A continuación se explican los diferentes pasos generales:

Generación de llamadas

Según las opciones (variación de la intensidad tráfico de nueva llamada *ro_n* y variación de intensidad de tráfico de llamada de traspaso *ro_h*), que son seleccionadas por el usuario en la interfaz grafica del simulador, según la opción, se aplican las variables internas según el caso.

Estas variables se encuentran en el código fuente como predeterminadas, cumpliendo con el modelo de *Yuguang Fang* [10] y son listadas a continuación:

```
ro_variable=1;
if ro_variable==variable

% Variando el parámetro 'ro' (100 a 1200) con un ro_h=10
% 'ro' intensidad de tráfico de nuevas llamadas.

    lan=20;           % Tasa medio de nueva llamada
    t_lan=1/lan;      % Tiempo media de nueva llamada

    lah=30;           % Tasa media de llamada de traspaso (Handoff)
    t_lah=1/lah;      % Tiempo medio de llamada de traspaso
    muh=300;          % Tasa media de permanencia de traspaso
    t_muh=1/muh;      % Tiempo media de permanencia de traspaso

    c=30;             % Número de canales en el router
    m=25;             % Umbral de nueva llamada
    fin_ro=60;        % Finaliza la intensidad de tráfico
else

% Variando el parámetro de 'ro_h' entre (100 a 1100), con un ro_n=15;
% 'ro_h' intensidad de tráfico de llamadas de traspaso

    lan=20;           % Tasa media nueva llamada
    t_lan=1/lan;      % Tiempo media de nueva llamada
    mun=300;          % Tasa media de permanencia de traspaso
    t_mun=1/mun;      % Tiempo media de permanencia de traspaso
```

```

lah=40;           % Tasa medio de llamada traspaso(Handoff)
t_lah=1/lah;     % Tiempo media de llamada traspaso

c=30;           % número de canales
m=25;           % umbral de nueva llamada
fin_ro=28;      % finaliza la intensidad de tráfico
end

```

Para generar peticiones de llamadas se utiliza la distribución de tipo exponencial con una tasa de nueva llamada $\lambda_n(la_n)$ y una tasa de llamadas de traspaso $\lambda_h(la_h)$ y cada tipo de llamada tiene un tiempo de retención del canal $1/\mu_n$ $1/\mu_h$ respectivamente. También posee una identificación que es el número *Hash*, que se genera aleatoriamente que es asignado a cada llamada. Esto se muestra en el segmento de código siguiente:

```

for i=1:25
    llegada_tn(i,1)=exprnd(la_n); % Con la función 'exprnd' se crea
                                   % los tiempos de petición de nuevas
                                   % llamadas exponencialmente y
                                   % aleatorias con una tasa media
                                   % 'la_n' que se guardan en la matriz
                                   % 'llegada_tn'.
    llegada_th(i,1)=exprnd(la_h); % Crea los tiempos de llamadas de
                                   % handoff exponencialmente y
                                   % aleatorias con una tasa media
                                   % 'la_h'y se guarda en la matriz
                                   % 'llegad_th'.
end
sum_lan=cumsum(llegada_tn); % Con la función 'cumsum' suma los
                             % tiempos de llamadas nuevas de los
                             % usuarios 'llegada_tn' y son
                             % guardados en un nuevo vector
                             % 'sum_lan'.
sum_lah=cumsum(llegada_th); % Con la función 'cumsum' suma los
                             % tiempos de llamadas de traspaso
                             % 'llegada_tn' y son guardados
                             % en un nuevo vector 'sum_lah'

for j=1:25
    stan_tn(j,1)=exprnd(mu_n); % Se crea los tiempos de permanencia
                                % de nuevas llamadas con una tasa
                                % media de mu_n, que es guardado en
                                % la matriz stan_tn.
    stan_th(j,1)=exprnd(mu_h); % Crea los tiempos de permanencia
                                % de llamadas de traspaso o handoff
                                % con una tasa media de mu_n, que es
                                % guardado en la matriz stan_th.
end
end

```

Como se desea calcular el número de llamadas (*New Call* ó *Handoff*) admitidas con respecto a la intensidad de tráfico, esto se determina variando $\rho_n = \lambda_n / \mu_n$ y $\rho_h = \lambda_h / \mu_h$, la intensidad de tráfico para los caso de nueva llamada y de traspaso respectivamente.

Control de admisión de llamadas

En este esquema de control de admisión de llamadas, en lugar de poner limitación al número de nuevas llamadas, se utiliza el número total de llamadas en curso en la celda para tomar una decisión acerca de aceptar o rechazar una nueva llamada entrante, este cambio se observa en el recuadro de la *figura 25*.

Sea m , que representa el Umbral cuando ocurre la llegada de una Nueva Llamada. Si el Número Total de Canales Ocupados es menor que m , se aceptara la Nueva Llamada; de otra forma, la Nueva Llamada será rechazada. Las Llamadas de traspaso son aceptadas siempre a menos que no haya un canal disponible cuando ocurre la llegada. Se asumen que para un router inalámbrico su capacidad máxima de canales es 30.

5.4.5 Módulo de generación de tabla de reserva Hash y tabla de resolución de colisiones

5.4.5.1 Descripción

Para generar estos números que identifica el flujo *flowid* del usuario, se generan números aleatorios entre 1 y 10000. Para esto se usa la función de Matlab *RAND* la cual genera números pseudo-aleatorios uniformemente distribuidos. Esta función es aplicada de la forma: $r = a + (b - a) \cdot \text{rand}(x,1)$; de forma que se generan x números aleatorios en el intervalo $[a,b]$. Esta tabla de reservas inicial se recorre para determinar si hay dos o mas números iguales; además se le agrega un

número de identificación correspondiente a las veces que se ha repetido el *flowid* para simular la *quíntupla*, estos números repetidos, es decir, con el mismo *flowid* y diferente *quíntupla*, pasan a crear la *tabla de resolución de colisiones*.

5.4.5.2 Datos de entrada-salida:

Datos de entrada

Este subprograma tiene como única variable de entrada, la variable simple *reserv*, que es el número de usuarios con reserva que es definida por el usuario en la interfaz grafica del simulador.

Datos de salida

La variable *Hashtab* es la matriz de reservas de datos obtenida de este subprograma, la cual consta de dos columnas, la primera contiene las identificaciones de flujo y la segunda contiene 1 ó 0, donde estos identifican si este flujo tiene o no colisión. *hashfa* es la matriz de reservas original que contiene todas las reservas ya sea que tengan o no colisión. *Hashfa* consta de 2 columnas, la primera contiene las identificaciones de flujo, la segunda contiene el número que identifica la *quíntupla* (toda identificación de flujo *flowid* con *quíntupla* $\neq 0$ tiene colisión). Además está la tabla de resolución de colisiones *coltab*, que contiene las identificaciones de flujo *flowid* repetidos con su respectiva *quíntupla*, también esta representada como una matriz de 2 columnas, una con *flowid* y la otra con el número que simula la *quíntupla*.

Las variables son almacenadas en archivos para posteriormente utilizarlas en los programas, estas variables se muestran en la tabla 9.

Tabla 9. Variables de entrada-salida de tabla de reserva de colisiones

	Variable	Archivo
Datos entrada	<i>reserv</i>	<i>resnum</i>
Datos de salida	<i>hashtab</i>	<i>tablah</i>
	<i>hashfa</i>	<i>tabla2</i>
	<i>coltab</i>	<i>tablac</i>

5.4.5.3 Algoritmo del módulo de Generación de tabla de colisiones

Figura 25. Diagrama de flujo del módulo de Generación de tabla de colisiones



A continuación se hace una explicación general del algoritmo como se muestra en la figura 25.

Pasos Generales:

1. Generación Tabla de Reserva
2. Generación Tabla de Colisiones

A continuación se explican los diferentes pasos generales:

Generador tabla de reservas

Para generar la *tabla de reservas* se hace uso de la función *rand* de *MATLAB* y se utiliza la *ecuación 5*.

$$x = (a * (b - a) .* rand(m,n)) \quad (\text{ecuación 5})$$

Donde [a,b] es el rango dentro del cual se van a encontrar los número generados, *m* es el número de filas y *n* el número de columnas de la matriz de número generada, la función se aplica como en la *ecuación 6*.

$$hashf = abs(1*(10000 - 1) .* rand(reserv,1)) \quad (\text{ecuación 6})$$

Donde *reserv* es el número de reservas a procesar. Es decir, se genera una matriz con *reserv* filas, en la cual se almacenan los números que representan las identificaciones de flujo de los usuarios con reserva.

Generador tabla de colisiones

En esta subrutina se hace una identificación de cuales reservas tienen *colisiones*, asignándoles un número entero. Por ejemplo al encontrar varias reservas con una misma *identificación de flujo* se dice que existe colisión, por

tanto a la primera colisión se le asigna un número que va hacer su *quíntupla* y esta dado inicialmente por el número uno, a la segunda colisión se le asigna la *quíntupla* con el número dos y así sucesivamente hasta el número de colisiones que haya. Esto se hace con el fin de facilitar la búsqueda en la *tabla de colisiones*. La comparación de la *quíntupla* se hace en la etapa del clasificador, donde son aceptadas las reservas.

5.5 ETAPA DE CLASIFICACIÓN DE PAQUETES

Dentro de los procesos de un router encontramos el proceso de clasificación, que se encarga de identificar los paquetes entrantes y verificar que estos posean una reserva previa para poder ser tratados. Este proceso se lleva a cabo en el simulador de dos formas:

La primera es mediante el uso de la arquitectura *INTSERV*, la cual calcula la identificación de flujo del paquete entrante haciendo uso de la *quíntupla*, este número es calculado y buscado en la tabla de reservas y de ser necesario en la tabla de resolución de colisiones, en caso tal que no se encontrara en ninguna de las dos tablas este paquete será desechado [5].

También se puede llevar a cabo haciendo uso de una arquitectura *INTSERV6*, esta arquitectura lleva a cabo un proceso más simple ya que el cálculo de la identificación de flujo se lleva a cabo en el origen, ahorrando este proceso al router. Así mismo, el clasificador no necesita realizar la búsqueda en la *tabla de reservas* ya que la posición en la misma es apuntada directamente utilizando como índice la identificación del flujo, determinando así mediante el mismo criterio, si se lleva o no a cabo la prestación del servicio [24].

5.5.1 Clasificadores IntServ e IntServ6

5.5.1.1 Descripción

Este módulo hace uso de las tablas anteriormente generadas (*hash* y de *resolución de colisiones*) para llevar a cabo el proceso de clasificación. Es decir, se genera un número aleatorio entre 1 y 10000 que simula la identificación de flujo *flowid* del paquete entrante y se busca este *flowid* en la *tabla de reservas*; en caso de encontrar la identificación del flujo del proceso, se registra el número de instrucciones que le tomó al programa encontrarlo. Así mismo, verifica si este número presenta colisiones, ya que de ser así, también habría que registrar las instrucciones de búsqueda en la *tabla de resolución de colisiones*. Esto se hace con el fin de tomar estos valores y multiplicarlos por constantes de tiempo (tomadas de la tabla 2) equivalentes a los ciclos de máquina que se toma el router en llevar a cabo cada una de estas lecturas. Ver figura 36.

5.5.1.2 Datos de entrada-salida

Datos de entrada:

La variable simple *paqord*, que es el número de paquetes a simular. Otra variable es la *hashtab* que es una *matriz de reservas* depurada, es decir, sin colisiones. Esta consta de dos columnas, en las cuales se encuentra almacenada la *identificación de flujo* correspondiente a cada reserva, además de una bandera que identifica la presencia o ausencia de colisiones respectivamente. Y finalmente, una última variable de entrada es *coltab* que es la matriz de *resolución de colisiones*, la cual consta de dos columnas, en las cuales se encuentra almacenada la *identificación de flujo* correspondiente a cada reserva, así como la *quíntupla* que identifica a la misma respectivamente.

Datos de salida:

La variable *datatab* es la matriz de datos obtenida de este subprograma, la cual consta de cuatro columnas que contienen datos de *identificación de flujo*, número de instrucciones de lectura en *tabla Hash*, número de instrucciones de lectura en la *tabla de resolución de colisiones*, e instrucciones de lectura de *quíntupla*. Esta última columna se refiere a cuantos *flowid* iguales tuvo que leer en la tabla de resolución de colisiones, antes de encontrar el indicado, organizados respectivamente para cada paquete procesado.

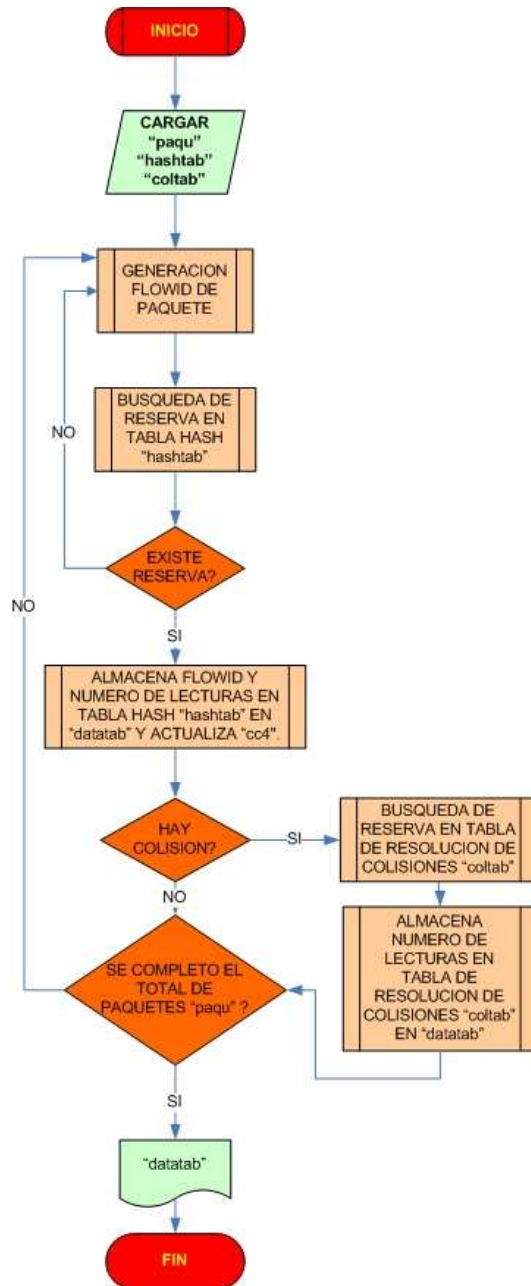
Las variables son almacenadas en archivos para posteriormente utilizarlas en los programas, estas variables se muestra en la *tabla 10*.

Tabla 10. Variables de entrada-salida del Clasificador IntServ e IntServ6

	Variable	Archivo
Datos de entrada	<i>paqord</i>	<i>ordpaqu</i>
	<i>hashtab</i>	<i>tablah</i>
	<i>coltab</i>	<i>tablac</i>
Datos salida	<i>datatab</i>	<i>tablaf</i>

5.5.1.3 Algoritmo del módulo de Clasificación

Figura 26. Diagrama de flujo del módulo de Clasificación



A continuación se hace una explicación general del algoritmo como se muestra en la figura 26.

Pasos Generales:

1. Generación de identificación de flujos de paquetes
2. Búsqueda en tabla de reservas
3. Búsqueda en tabla de resolución de colisiones

A continuación se explican los diferentes pasos generales:

Generación de identificación de flujos de paquetes

Para generar la identificación de flujos de paquetes *hashn*, se hace uso de la función *rand* la cual genera número aleatorios con una distribución exponencial uniforme y se aplica la función:

$$x = (a * (b - a) .* rand(m, n)) \quad (\text{ecuación 7})$$

Donde *a* y *b* es el rango dentro del cual se van a encontrar los número generados, *m* es el número de filas y *n* el número de columnas de la matriz de número generada, para la simulación esta función queda de la forma:

$$hashn = abs(1 * (10000 - 1) .* rand(1, 1)) \quad (\text{ecuación 8})$$

La cual genera un número aleatorio entre 1 y 10000 que representa la identificación de flujos del paquete que se va a procesar esto se lleva a cabo hasta completar el total de paquetes que el usuario desea simular.

Búsqueda en tabla de reservas

Como se muestra en la *figura 36*, para realizar la búsqueda en la *tabla de reservas*, se usa una subrutina *while* que compara este número generado con las identificaciones de flujos presentes en la *tabla de reservas*, si la *identificación de flujo* de paquetes se encuentra presente, se procede a verificar si presenta colisión y de ser así, se realiza una búsqueda en la *tabla de resolución de colisiones* llevando un registro de las lecturas necesarias para encontrar esta reserva; en caso tal de que no se encuentre reserva, se ignora esta identificación de flujo y se procede a generar una nueva. El proceso continuará así hasta completar el número de paquetes que el usuario desea simular. Esto se determina comparando la variable de entrada *paqord* con *cc4*, que es una variable interna del programa que almacena el número de paquetes que han sido procesados.

Búsqueda en tabla de resolución de colisiones

Para llevar a cabo esta búsqueda se hace uso de una subrutina *while*, en la cual se busca la primera colisión que corresponda a la *identificación de flujo* que se está procesando. En esta ubicación de la tabla se toma el dato de cuántas colisiones hay para esta reserva, esto con el fin de generar un número aleatorio entre uno y este valor, el cual representa la *quíntupla*. Calculado este número se continúa la búsqueda en la tabla hasta encontrar la reserva respectiva para estos valores (*Flowid-Quíntupla*).

5.6 ETAPA DE PLANIFICACIÓN DE PAQUETES

El proceso de planificación, es el que se encarga de organizar la salida de los paquetes del router hacia su destino teniendo en cuenta criterios intrínsecos de cada método; en este simulador se implementaron dos esquemas, el de *PQ* (*Priority Queuing*), el cual se basa en un criterio de prioridades, es decir,

dependiendo de los requerimientos de la reserva se le otorga a esta un nivel de prioridad. en este caso se otorgan cuatro tipos de prioridades (alta, media, normal y baja) de esta manera, mientras hallan paquetes en cola de prioridad alta, los paquetes de las demás prioridades serán ignorados y así sucesivamente, teniendo en cuenta al final los paquetes de prioridad baja; también está el esquema de *PGPS (Packet-by-packet Generalized Processor Sharing)*, el cual tiene un criterio de selección basado en anchos de banda, otorgándole servicio a todos los paquetes sin importar la aplicación "sin prioridad", pero limitando la transferencia de bits según lo indique el ancho de banda; es decir, si los paquetes de un usuario llegan a superar el máximo de bits permitidos para este, estos son retenidos en el router por un tiempo para permitir así que pasen paquetes de los demás usuarios y evitar que uno solo acapare todos los recursos [5, 24, 30].

5.6.1 Planificación de paquetes con el esquema PQ (Priority Queuing)

5.6.1.1 Descripción

De acuerdo a la figura 37, en el subprograma se toman los tiempos de proceso de la clasificación para actualizarlos; para esto, se tiene en cuenta la prioridad que tiene cada paquete, es decir, el primer paquete que se transmite es el primero en llegar al planificador, a este se le calcula el tiempo de servicio (longitud de paquete/ancho de banda) y se le suma al tiempo respectivo de clasificación; después se verifica si quedaron paquetes en cola y, si es así, se toman los paquetes según su prioridad y se siguen transmitiendo. De no ser así, se transmite el siguiente en llegar. Para este planificador se tienen cuatro prioridades (alta, media, normal, baja) y los paquetes se transmiten según este orden, es decir, mientras hallan paquetes en cola de prioridad alta, no se van a transmitir los de las demás prioridades y así sucesivamente teniendo en cuenta por último la prioridad baja.

5.6.1.2 Datos de entrada-salida

Datos de entrada:

Una de las variables de entrada es *supertab*, la cual es una matriz que contiene los resultados de la clasificación (*tiempo de búsqueda en tabla Hash, tiempo de búsqueda en tabla de resolución de colisiones, tiempo de salida de clasificación*), después de haber sido convertidos a una misma escala (segundos). Además contiene datos necesarios en la planificación tales como *anchos de banda* en el caso de *PGPS (Packet-by-packet Generalized Processor Sharing)*, tiempos de llegada al router, prioridad "*en el caso de PQ*" y la longitud de paquete. Otra de las variables de entrada es la variable *paqord*, que es una matriz de una columna y tantas filas como incrementos de reserva, es decir al llevar a cabo una simulación con número de reservas fijo, esta matriz tendrá una sola fila; y al llevar a cabo una simulación con incremento de reservas, tendrá tantas filas como incrementos hallan sido necesarios para alcanzar el número de reservas máximo definido por el usuario.

Datos de salida:

Como variable de salida está la matriz *tvact*, la cual consta de cuatro columnas, en las cuales se almacenan, los tiempos de salida del router para *INTSERV*, prioridad del paquete, tiempos de salida del router para *INTSERV6* y prioridad del paquete, esto para cada columna y para cada paquete procesado respectivamente. Otra variable de salida es *promes*, la cual es una matriz que almacena los tiempos medios de retardo del planificador para cada número de flujos con reserva; esta consta de dos columnas en las cuales se almacena los tiempos medios de retardo para *INTSERV* e *INTSERV6* respectivamente.

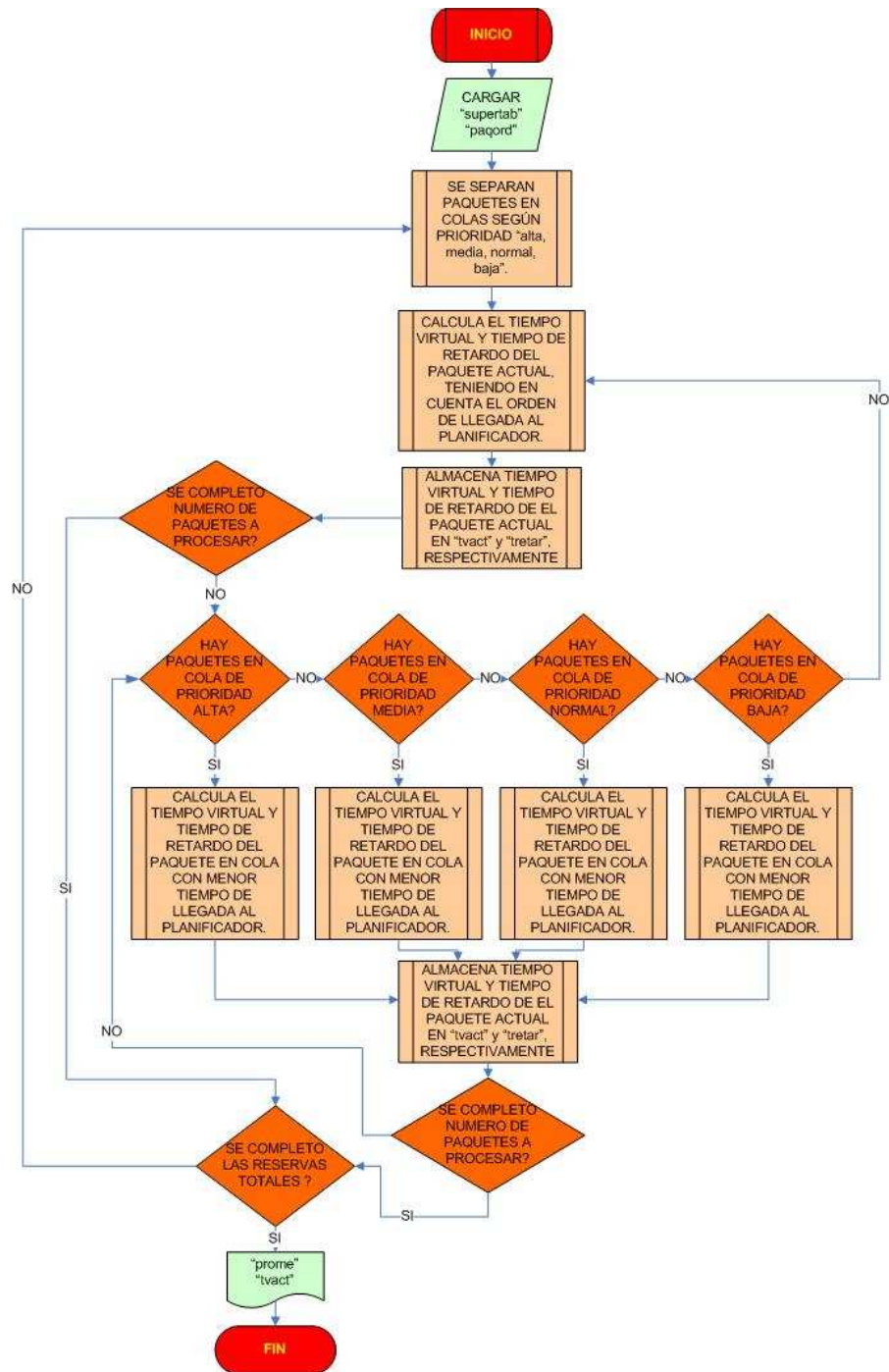
Las variables son almacenadas en archivos para posteriormente utilizarlas en los programas, estas variables se muestran en la *tabla 11*.

Tabla 11. Variables de entra-salida del esquema *PQ*

	Variable	Archivo
Datos Entrada	<i>paqord</i>	<i>ordpaqu</i>
	<i>supertab</i>	<i>tabplan</i>
Datos Salida	<i>prome</i>	<i>pqprom</i>
	<i>tvact</i>	<i>tabpq</i>

5.6.1.3 Algoritmo del módulo del esquema PQ

Figura 27. Diagrama de flujo del Esquema PQ



A continuación se hace una explicación general del algoritmo como se muestra en la *figura 27*.

Pasos Generales:

1. Organización de paquetes en colas según prioridad
2. Revisión de colas y procesamiento de paquetes según prioridad

A continuación se explican los diferentes pasos generales:

Organización de paquetes en colas según prioridad

Haciendo uso de los datos de resultados obtenidos en el clasificador, se organizan los mismos en cuatro colas de planificación dependiendo de la prioridad otorgada previamente a las reservas, esta distribución se identifica dentro del programa mediante etiquetas que identifican este grado de prioridad, las cuales se encuentran organizadas como se indica en la *tabla 12*:

Tabla 12. Esquema de prioridades PQ

<i>Prioridad</i>	<i>Alta</i>	<i>Media</i>	<i>Normal</i>	<i>Baja</i>
<i>Etiqueta</i>	1	2	3	4

Las colas de planificación quedan representadas en forma de matrices que contienen la información anterior.

Revisión de colas y procesamiento de paquetes según prioridad

En esta etapa del planificador PQ, se revisan constantemente las colas de prioridad en orden descendente buscando la presencia de paquetes en ellas; de esta forma, mientras hayan paquetes en las colas de mayor prioridad, los

paquetes presentes en las demás colas tendrán que esperar a que estos terminen de ser procesados; para llevar acabo este proceso se hace uso de la *ecuación 9*.

$$t_s = t_i + (long)/(bw) \quad (\text{ecuación 9})$$

Donde t_s es el tiempo de salida del planificador, t_i es el tiempo en que el paquete empieza a ser procesado en el planificador ya sea el tiempo de llegada al planificador o bien el tiempo de salida del paquete anterior (en caso de que halla esperado en cola), $long$ es la longitud del paquete en *bits* y bw es el ancho de banda otorgado en *bits/seg*. También se calcula el tiempo que dura el paquete en el planificador haciendo uso de la *ecuación 10*.

$$t_r = t_s - t_{in} \quad (\text{ecuación 10})$$

Donde t_r es el tiempo de permanencia en el planificador, t_s es el tiempo de salida y t_{in} es el tiempo de llegada al planificador, estos tiempos de permanencia son calculados tanto para *INTSERV* como para *INTSERV6*, y se guardan en la matriz llamada *tretar*, compuesta por dos columnas y *paqord* filas, donde *paqord* es el número de paquetes.

5.6.2 Planificación de paquetes con el esquema PGPS (Packet-by-packet Generalized Processor Sharing)

5.6.2.1 Descripción

Este subprograma toma los tiempos de proceso de la clasificación para actualizarlos; para esto, se tiene un conteo de los bits que cada usuario ha transmitido por unidad de tiempo, este número de bits no puede superar el ancho de banda del usuario *bits/segundo*, usando como referencia una ventana de tiempo de 100ms. Si esto llegase a suceder, el paquete sería demorado para permitir que los demás usuarios transmitan en este tiempo, logrando así una

distribución justa de recursos para todos los usuarios; los tiempos se actualizan sumándole a los tiempos de clasificación el tiempo que el router se demora en sacar cada paquete (longitud de paquete/ancho de banda), y el tiempo que se demora en cola de planificación; obteniendo así el tiempo de salida de los paquetes del planificador.

5.6.2.2 Datos de entrada-salida:

Los datos de entrada-salida se muestran en la *tabla 13* y se explican a continuación.

Datos de entrada:

Una de las variables de entrada es *supertab*, la cual es una matriz que contiene los resultados de la clasificación (*tiempo de búsqueda en tabla Hash, tiempo de búsqueda en tabla de resolución de colisiones, tiempo de salida de clasificación*), después de haber sido convertidos a una misma escala (segundos). Además se contiene datos necesarios en la planificación tales como: *anchos de banda* en el caso de *PGPS (Packet-by-packet Generalized Processor Sharing)*, tiempos de llegada al router, prioridad "*en el caso de PQ*" y la longitud de paquete. Otra de las variables de entrada es la variable simple *paqord* que contiene el número de paquetes que van a ser procesados.

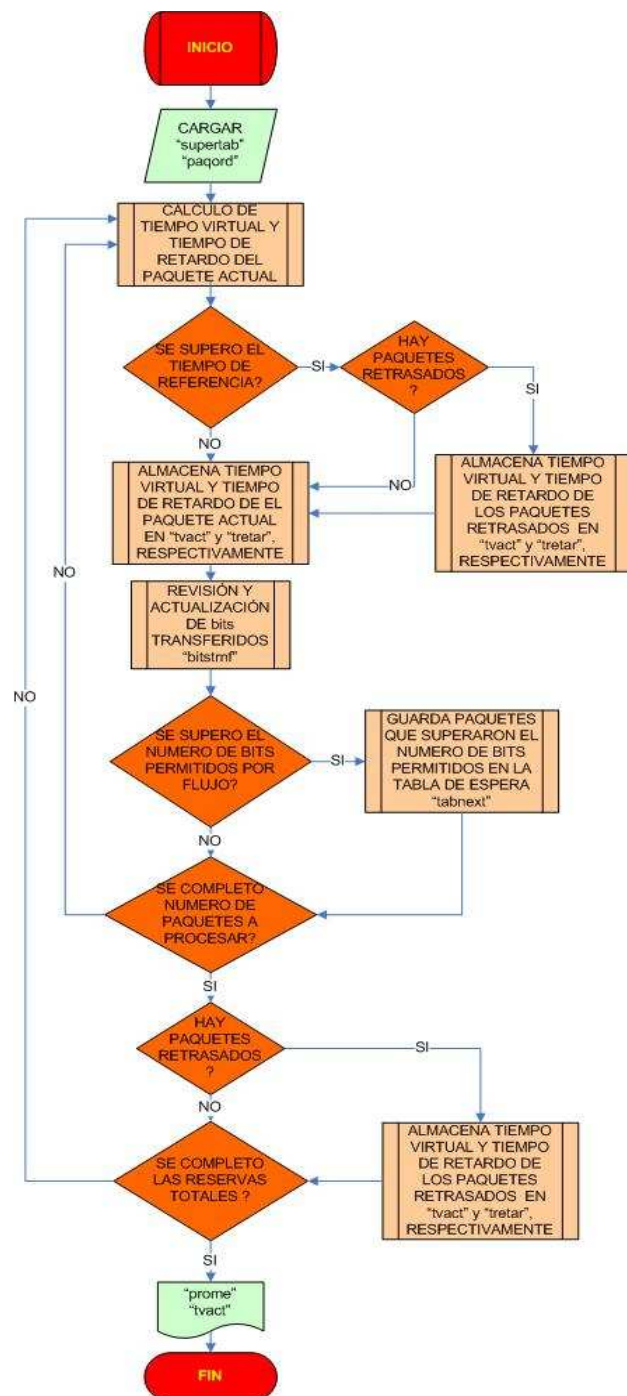
Datos de salida:

Como variable de salida esta la matriz *tvact*, la cual consta de cuatro columnas, en las cuales se almacenan, los tiempos de salida del router para *INTSERV* y tiempos de salida del router para *INTSERV6*, esto para cada columna y para cada paquete procesado respectivamente. Otra variable de salida es *prom*, la cual es una matriz que almacena los tiempos medios de retardo del planificador para cada número de flujos con reserva, consta de dos columnas en las cuales se almacena

los tiempos medios de retardo para *INTSERV* e *INTSERV6* respectivamente. El tiempo que dura el paquete en el planificador se calcula como en el programa de modelo *PQ* explicado en la *sección 5.6.1*.

5.6.2.3 Algoritmo del módulo de PGPS

Figura 28. Diagrama de flujo del módulo de PGPS



A continuación se hace una explicación general del algoritmo como se muestra en la figura 28.

Pasos Generales:

1. Cálculo de tiempo virtual de planificación
2. Revisión de cola de paquetes
3. Revisión de *bits* transferidos por usuario

A continuación se explican los diferentes pasos generales:

Cálculo de tiempo virtual de planificación

En esta parte se lleva a cabo el cálculo del tiempo de salida del planificador haciendo uso de la *ecuación 11*.

$$t_s = t_i + (long)/(bw) \quad (\text{ecuación 11})$$

Donde t_s es el tiempo de salida del planificador, t_i es el tiempo en que el paquete empieza a ser procesado en el planificador, ya sea el tiempo de llegada al planificador o bien el tiempo de salida del paquete anterior (en caso de que halla esperado en cola), *long* es la longitud del paquete en *bits* y *Bw* es el ancho de banda otorgado en *bits/seg*.

Revisión de cola de paquetes

En esta etapa del proceso se verifica si la ventana de tiempo ha sido superada, en caso tal, se procesan los paquetes presentes en la cola que han sido retrasados

por sobrepaso de *bits* transferidos y se les calcula el tiempo de salida del planificador.

Revisión de bits transferidos por usuario

Al llevar a cabo el procesamiento de un paquete, se le toma registro de la identificación de flujo y de la longitud, llevando así un conteo de los *bits* transferidos por cada usuario, que permitan determinar si los paquetes se deben retrasar por sobrepaso de *bits*, o si deben ser procesados normalmente.

5.7 ETAPA DE INTEGRACIÓN DE SUBPROGRAMAS

En esta etapa, se integran los subprogramas que componen el simulador del router, teniendo en cuenta, la configuración y condiciones iniciales definidas por el usuario en la interfaz grafica; para esto se desarrollaron tres programas cuyo funcionamiento se explica a continuación.

5.7.1 Programa para simulación con número de reservas fijo

5.7.1.1 Descripción

Este programa es usado cuando esta seleccionada la opción de simulación con número de *reservas* fijo en la interfaz grafica del simulador; en el, se pregunta que opción seleccionó el usuario para cada etapa y llama los subprogramas respectivos, además toma los datos entregados por la etapa de *clasificación* (*datatab*), de *CAC* (*bwtab*) y por la etapa de tráfico (*intime*), para acoplarlos en una sola variable llamada *supertab*, asegurándose que los datos de tiempo, ancho de banda y longitud de paquetes tengan las mismas unidades (*segundos*, *bps*, *bits*) respectivamente, adicionalmente, se encarga de procesar los tiempos de búsqueda en tabla de *reservas hashing* y *tabla de resolución de colisiones*, para

calcular los tiempos de salida de los paquetes y tiempos de retardo de *clasificación*. Ver figura 29

5.7.1.2 Datos de entrada-salida:

Datos de entrada:

Este subprograma tiene como variables de entrada las opciones y condiciones iniciales especificadas por el usuario en la interfaz grafica, como *número de reservas*, *numero de paquetes a procesar* y *opciones de simulación*. Otra variable de entrada es *datatab*, representada como una matriz de cuatro columnas y *paqu* filas en las que se almacenan datos de *identificación de flujo*, número de lecturas en tabla *hash*, número de lecturas en *tabla de resolución de colisiones* y *quíntupla*, para cada paquete respectivamente; también esta como variable de entrada *intime*, que esta representada como una matriz de una columna por *paqu* filas, y que contiene los tiempos de llegada de los paquetes al *router* según el tráfico seleccionado. Ver *Tabla 13*.

Datos de salida:

Una de las variables de salida de este subprograma es *supertab*, que es una matriz de once columnas y *paqu* filas, en la que se almacenan datos de *identificación de flujo*, *quíntupla*, ancho de banda, tiempo de búsqueda en tabla *hash*, tiempo de búsqueda en *tabla de resolución de colisiones*, tiempo de llegada de paquetes, tiempo de salida de los paquetes para una arquitectura *INTSERV*, tiempo de salida de los paquetes para una arquitectura *INTSERV6*, número de paquetes procesados por usuario, longitud de paquete y prioridad, para cada paquete respectivamente, también esta como variable de salida *retar*, que es una matriz de dos columnas en la que se almacenan los tiempos de retardo de *clasificación* para *INTSERV* e *INTSERV6* respectivamente. Ver *tabla 13*.

Tabla 13. Variables de entra-salida del esquema Simulador

	Variable	Archivo
Datos Entrada	<i>datatab</i>	<i>tabplan</i>
	<i>paqu</i>	<i>paqnum</i>
	<i>intime</i>	<i>trafic</i>
Datos Salida	<i>supertab</i>	<i>tabplan</i>
	<i>retar</i>	<i>tretrar</i>

5.7.1.3 Algoritmo del módulo de Reservas fijo

Figura 29. Diagrama de flujo del simulador de Reservas Fijo



A continuación se hace una explicación general del algoritmo que es mostrado en la *figura 29*.

Pasos Generales:

1. Llamar subprogramamas
2. Creación de la variable *supertab*

3. Calcular tiempos de retardo y salida de *clasificación*

A continuación se explican los diferentes pasos generales:

Llamar subprogramas

Según las opciones de las etapas de *tráfico*, *control de admisión*, *clasificación* y *planificación*, seleccionadas por el usuario en la interfaz grafica, se llaman los subprogramas respectivos que componen el tipo de simulación deseada.

Creación de la variable *supertab*

Se toman los datos entregados por la etapa de *clasificación* (*datatab*), de CAC (*bwtab*) y por la etapa de tráfico (*intime*), y se integran en la variable *supertab*, para cada paquete procesado respectivamente, teniendo en cuenta que los datos de tiempo, ancho de banda y longitud de paquetes tengan las mismas unidades (*segundos*, *bps*, *bits*) respectivamente.

Calcular tiempos de retardo y salida de *clasificación*

Para calcular estos tiempos se hace uso de la *ecuación 12*.

$$t_s = t_e + t_{ch} + t_{in} + t_{bh} + t_{bc} \quad (\text{ecuación 12})$$

Donde t_s es el tiempo de salida del proceso de *clasificación* para cada paquete, t_e es el tiempo de enrutamiento que se toma de la *Tabla 2*, t_{ch} es el tiempo que se demora en calcular el número *hash* y es tomado de la *Tabla 2*, t_{in} es el tiempo en que el paquete empieza a ser tratado en la etapa de *clasificación*, t_{bh} es el tiempo que demoro la búsqueda en tabla *hash* y t_{bc} es el tiempo que demoro la búsqueda en la *tabla de resolución de colisiones*, estos últimos, se calculan multiplicando el número de iteraciones de búsqueda realizadas, por constantes de tiempo equivalentes a

lectura de número *Hash* y lectura de *quíntupla* respectivamente, tomadas de *Tabla 1*; para el caso de *INTSERV6*, no se tiene en cuenta el calculo de número *hash*, ya que esta *identificación de flujo* es calculada en el origen; además t_{bh} es el equivalente a una lectura de número *hash* según condiciones de este tipo de arquitectura, este valor de tiempo que demora el router en leer una *identificación de flujo* es tomado de *Tabla 2*, este calculo se hace aplicando la *ecuación 13*.

$$t_s = t_e + t_{ch} + t_{in} + t_{bh} + t_{bc} \quad (\text{ecuación 13})$$

Para calcular los tiempos de retardo del clasificador, se hace uso de la función: $t_r = t_s - t_{ll}$. Donde t_r es el tiempo de retardo del paquete en el clasificador, t_s es el tiempo de salida y t_{ll} es el tiempo de llegada del paquete al clasificador.

5.7.2 Programa para simulación con número de reservas incremental

5.7.2.1 Descripción

Este programa es usado cuando esta seleccionada la opción de simulación con número de *reservas* incremental en la interfaz grafica del simulador; en el, se pregunta que opción seleccionó el usuario para cada etapa y llama los subprogramas respectivos, además toma los datos entregados por la etapa de *clasificación (datatab)*, de *CAC (bwtab)* y por la etapa de tráfico (*intime*), para acoplarlos en una sola variable llamada *supertab*, asegurándose que los datos de tiempo, ancho de banda y longitud de paquetes tengan las mismas unidades (*segundos, bps, bits*) respectivamente, adicionalment.e, se encarga de procesar los tiempos de búsqueda en tabla de *reservas hashing* y *tabla de resolución de colisiones*, para calcular los tiempos de salida de los paquetes y tiempos de retardo de *clasificación*, este proceso se lleva a cabo tantas veces como incrementos de *reservas* existan y el incremento de *reservas* se realiza de diez en diez. Ver *figura 30*

5.7.2.2 Datos de entrada-salida:

Los datos de entrada-salida se muestran en la *tabla 13*, estos datos son explicados a continuación.

Datos de entrada:

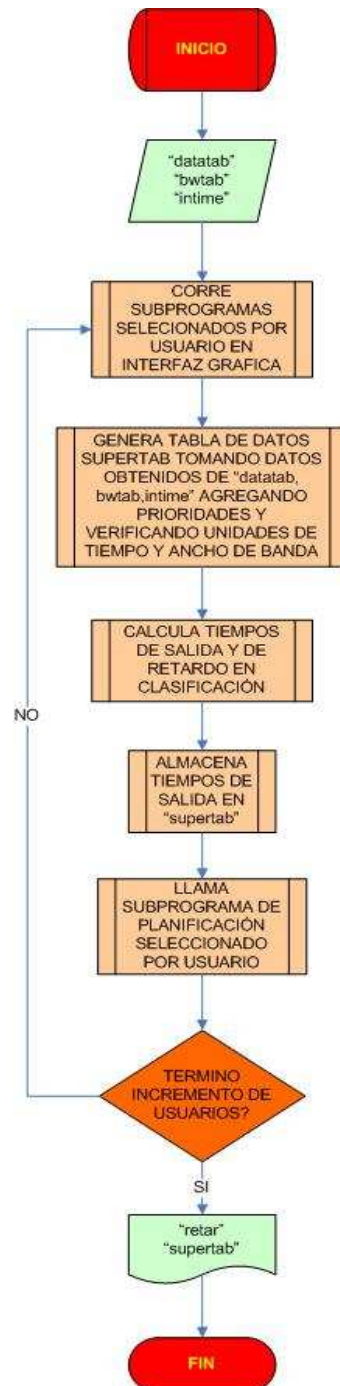
Este subprograma tiene como variables de entrada las opciones y condiciones iniciales especificadas por el usuario en la interfaz grafica, como *número de reservas inicial*, *número de reservas final*, *numero de paquetes a procesar* y *opciones de simulación*. Otra variable de entrada es *datatab*, representada como una matriz de cuatro columnas y *paqord* filas (número de paquetes) en las que se almacenan datos de *identificación de flujo*, número de lecturas en tabla *hash*, número de lecturas en *tabla de resolución de colisiones* y *quíntupla*, para cada paquete respectivamente; también esta como variable de entrada *intime*, que esta representada como una matriz de una columna por *paqord* filas, y que contiene los tiempos de llegada de los paquetes al *router* según el tráfico seleccionado.

Datos de salida:

Una de las variables de salida de este subprograma es *supertab*, que es una matriz de once columnas y *paqord* filas, en la que se almacenan datos de *identificación de flujo*, *quíntupla*, ancho de banda, tiempo de búsqueda en tabla *hash*, tiempo de búsqueda en *tabla de resolución de colisiones*, tiempo de llegada de paquetes, tiempo de salida de los paquetes para una arquitectura *INTSERV*, tiempo de salida de los paquetes para una arquitectura *INTSERV6*, número de paquetes procesados por usuario, longitud de paquete y prioridad, para cada paquete respectivamente, también esta como variable de salida *retar*, que es una matriz de dos columnas en la que se almacenan los tiempos de retardo de *clasificación* para *INTSERV* e *INTSERV6* respectivamente.

5.7.2.3 Algoritmo del simulador de reservas incrementales

Figura 30. Diagrama de flujo del *Simulador de reservas incremental*



A continuación se hace una explicación general del algoritmo que es mostrado en la *figura 30*.

Pasos Generales:

1. Llamar subprogramamas
2. Creación de la variable *supertab*
3. Calcular tiempos de retardo y salida de *clasificación*

Los pasos generales que tiene este subprograma, son iguales a los descritos en el subprograma de simulador de *reserv fijo sección 5.7.1*

5.7.3 Programa para simulación con router inalámbrico

5.7.3.1 Descripción

Este programa es usado cuando esta seleccionada la opción de simulación con *router* inalámbrico fijo en la interfaz grafica del simulador; en el, se pregunta que opción seleccionó el usuario para cada etapa y llama los subprogramas respectivos, además toma los datos entregados por la etapa de *clasificación (datatab)*, de *CAC (bwtab)* y por la etapa de tráfico (*intime*), para acoplarlos en una sola variable llamada *supertab*, asegurándose que los datos de tiempo, ancho de banda y longitud de paquetes tengan las mismas unidades (*segundos, bps, bits*) respectivamente, adicionalmente, se encarga de procesar los tiempos de búsqueda en tabla de *reservas hashing* y *tabla de resolución de colisiones*, para calcular los tiempos de salida de los paquetes y tiempos de retardo de *clasificación*, finalmente, hace uso de la variable *admit* obtenida del control de admisión para revisar constantemente los tiempos de permanencia de las *reservas* admitidas y por tanto llevar a cabo la actualización de la tabla *hash*. Ver *figura 41*

5.7.3.2 Datos de entrada-salida:

Los datos de entrada-salida se muestran en la *tabla 14*. estos datos son explicados a continuación.

Datos de entrada:

Este subprograma tiene como variables de entrada las opciones y condiciones iniciales especificadas por el usuario en la interfaz grafica, como *número de reservas*, *numero de paquetes a procesar* y *opciones de simulación*. Otra variable de entrada es *datatab*, representada como una matriz de cuatro columnas y *paqord* filas en las que se almacenan datos de *identificación de flujo*, número de lecturas en tabla *hash*, número de lecturas en *tabla de resolución de colisiones* y *quíntupla*, para cada paquete respectivamente; también esta como variable de entrada *intime*, que esta representada como una matriz de una columna por *paqord* filas, y que contiene los tiempos de llegada de los paquetes al *router* según el tráfico seleccionado, finalmente, están las variables *admit* y *nuevos*, que están representadas como matrices de tres columnas, en las que se almacenan datos de tiempo de establecimiento de *reserva*, tipo de llamada y tiempo de permanencia de *reserva*, siendo *admit* la tabla de reservas admitidas y *nuevos* la tabla de reservas que piden admisión.

Datos de salida:

Una de las variables de salida de este subprograma es *supertab*, que es una matriz de once columnas y *paqord* filas, en la que se almacenan datos de *identificación de flujo*, *quíntupla*, ancho de banda, tiempo de búsqueda en tabla *hash*, tiempo de búsqueda en *tabla de resolución de colisiones*, tiempo de llegada de paquetes, tiempo de salida de los paquetes para una arquitectura *INTSERV*, tiempo de salida de los paquetes para una arquitectura *INTSERV6*, número de

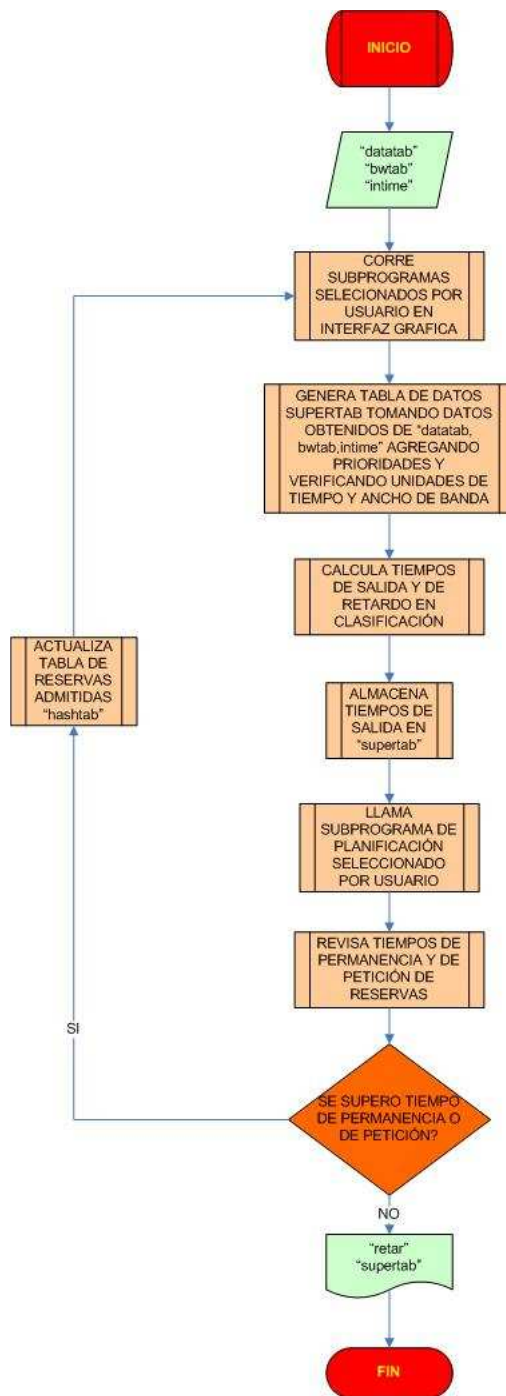
paquetes procesados por usuario, longitud de paquete y prioridad, para cada paquete respectivamente, también esta como variable de salida *retar*, que es una matriz de dos columnas en la que se almacenan los tiempos de retardo de *clasificación* para *INTSERV* e *INTSERV6* respectivamente, finalmente, este subprograma reutiliza y actualiza la variable *hashtab* según condiciones del proceso de admisión; esta variable, está representada como una matriz de una columna por *reserv filas (numero de usuarios con reserva)*, que almacenan datos de *identificación de flujo* y presencia de *colisión* respectivamente *sección 5.4.5*.

Tabla 14. Variables de entra-salida del esquema Simulador del router inalámbrico

	Variable	Archivo
Datos Entrada	<i>datatab</i>	<i>tabplan</i>
	<i>paqord</i>	<i>ordpaqu</i>
	<i>admit</i>	<i>regllam</i>
	<i>intime</i>	<i>trafic</i>
Datos Salida	<i>supertab</i>	<i>tabplan</i>
	<i>retar</i>	<i>tretar</i>

5.7.3.3 Algoritmo del simulador para router inalámbrico

Figura 31. Diagrama de flujo del simulador para router inalámbrico



A continuación se hace una explicación general del algoritmo que es mostrado en la *figura 31*.

Pasos Generales:

1. Llamar subprogramamas
2. Creación de la variable *supertab*
3. Calcular tiempos de retardo y salida de *clasificación*
4. Actualización de tabla de *reservas admitidas*

Actualización de tabla de reservas admitidas

Para realizar este proceso, se revisa constantemente el tiempo de permanencia de las reservas admitidas, haciendo uso de la *ecuación 14*.

$$t_p = t_s - t_{est} \quad (\text{ecuación 14})$$

Donde t_p es el tiempo de permanencia, t_s es el tiempo de salida del paquete del router y t_{est} es el tiempo de establecimiento de la reserva; este tiempo de permanencia es comparado con el calculado en el proceso de control de admisión, con el fin de sacar de la tabla de reservas, aquellas que lo hallan superado; también se deben tener en cuenta los nuevos tiempos de petición de reserva que están almacenados en la variable *nuevos*, con el fin de determinar su admisión e ingreso a la tabla de reservas.

Los pasos generales que no se explican en este subprograma, son iguales a los descritos en el subprograma de simulador de *reserv fijo* sección 5.7.1

PARTE 3 RESULTADOS Y CONCLUSIONES

6. SIMULACIONES Y ANÁLISIS DE RESULTADOS

7. CONCLUSIONES Y LÍNEAS FUTURAS

6. SIMULACIONES Y ANÁLISIS DE RESULTADOS

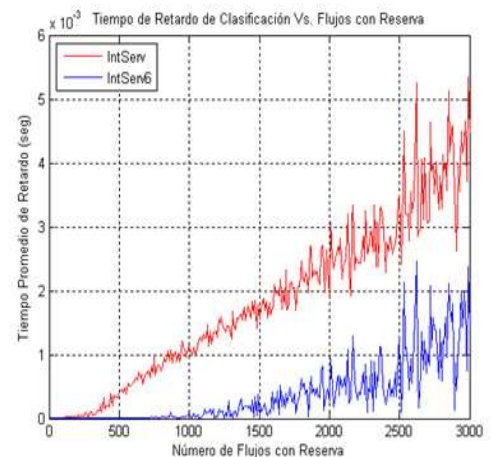
6.1 RESULTADOS OBTENIDOS PARA TIEMPO DE RETARDO EN CLASIFICACIÓN

La configuración de simulación de las gráficas que se muestran a continuación, se hizo en base a las simulaciones hechas en [8], que muestran el retardo promedio de los paquetes en la etapa de clasificación para los dos tipos de arquitectura *IntServ* e *IntServ6*, teniendo en cuenta el procesamiento de 50 paquetes, solo se varia el tipo de tráfico simulado y la configuración de longitud de paquetes.

La tabla de la *figura 32*, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 32. Tiempo promedio de retardo de *Clasificación1*

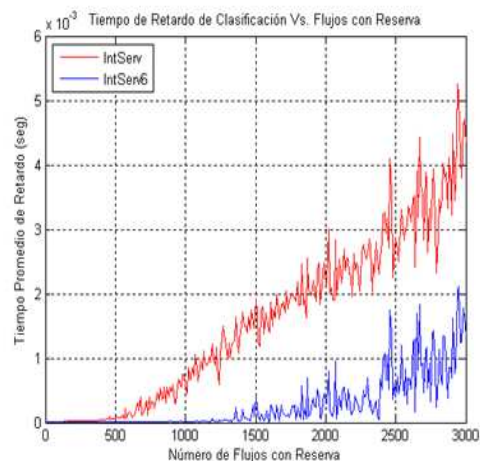
Proceso	Stección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[3000]
		Número de Paquetes =[50]
	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[10Mbps]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de NewCall
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	- - - - -
	PQ	- - - - -



La tabla de la *figura 33*, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 33. Tiempo promedio de retardo de *Clasificación2*

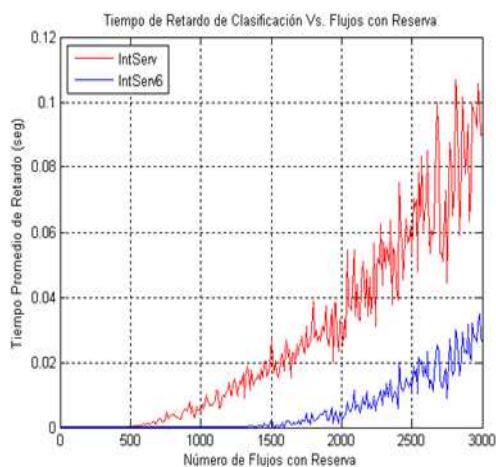
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[512 bits] en media
		Número Max. de reservas =[3000]
	D-MMPP	Número de Paquetes =[50]
		Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	New Call Bounding Scheme	Número Max. de reservas =[]
		Ciclos de Simulación de Proceso =[]
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 34*, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

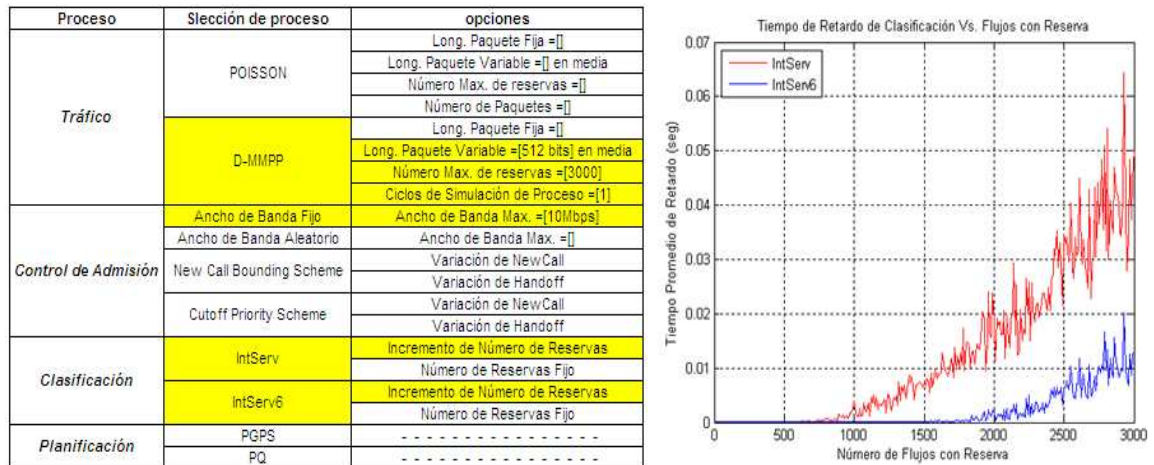
Figura 34. Tiempo promedio de retardo de *Clasificación3*

Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
	D-MMPP	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[512 bits] en media
		Número Max. de reservas =[3000]
Control de Admisión	New Call Bounding Scheme	Ciclos de Simulación de Proceso =[1]
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 35*, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 35. Tiempo promedio de retardo de *Clasificación4*



6.2 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE CLASIFICACIÓN PARA UNA SIMULACIÓN DE RESERV INCREMENTAL.

En todas las gráficas anteriores, la línea roja corresponde a los tiempos de retardo promedio que se presentan en el clasificador con una arquitectura del tipo *INTSERV*, teniendo como referencia el número de usuarios con reserva presentes en la tabla *Hash*. El comportamiento que se observa se debe a que a medida que aumenta el número de usuarios con reservas presentes en la tabla *Hash*, aumenta el tiempo de búsqueda de reservas, logrando que el proceso de clasificación sea mas lento, es decir, aumenta el tiempo de retardo de búsqueda de reservas y por tanto aumenta el tiempo de espera de los paquetes en cola de clasificación, incrementando así el tiempo de retardo total de los paquetes en el clasificador. Este tiempo también se ve afectado por la búsqueda en la *tabla de resolución de colisiones*, el cálculo del número *Hash*, las lecturas de *quíntupla* y el tiempo de enrutamiento que es el mismo para los dos tipos de arquitectura.

La línea de color azul, corresponde a los tiempos de retardo promedio que se presentan en el clasificador con una arquitectura del tipo *INTSERV6*, teniendo como referencia, el número de usuarios con reserva presentes en la tabla *Hash*. El comportamiento observado se debe a que los tiempos de retardo dependen en su mayor parte de la búsqueda en la *tabla de resolución de colisiones* y las lecturas de *quíntupla* que se llevan a cabo en ella; otra variable que incrementa este tiempo de retardo, es el equivalente a una instrucción de lectura en tabla *Hash*, teniendo en cuenta para *INTSERV6*, la búsqueda de la reserva necesita llevar a cabo una sola instrucción de lectura en tabla *Hash* para ubicar su posición, es decir, el cálculo del número *Hash* se lleva a cabo en por el usuario y no por el *router*, además las reservas en la tabla *Hash* se encuentran almacenadas en la posición equivalente al número *Hash*, reduciendo el número de búsquedas a una sola. A los tiempos antes mencionados hay que sumarle el tiempo de enrutamiento, que es el mismo para los dos tipos de arquitectura.

En las *figuras (32 y 33)*, se observa que la escala de el tiempo promedio de retardo es mucho menor que en las *figuras (34 y 35)*. Esto se debe a que las dos primeras hicieron uso de un generador de tráfico *POISSON*, que permite mantener un número de paquetes simulados fijo y definido por el usuario, logrando tener un control de la cola de paquetes a la entrada de *clasificación*. Caso contrario, para las *figuras (34 y 35)*, se usó un generador de tráfico *D-MMPP*, para el cual, el número de paquetes se incrementa proporcionalmente al número de usuarios en la simulación, además este tipo de distribución simula tiempos de llegada más cercanos entre si en algunos puntos del tráfico, generándose una mayor cola de paquetes a la entrada del clasificador, y así mismo, un mayor tiempo de retardo promedio de clasificación.

La longitud de paquete, no afecta el proceso de clasificación; sin embargo, se presenta una leve diferencia entre las gráficas obtenidas haciendo uso de una longitud variable (*figuras 33 y 34*), y las obtenidas haciendo uso de una longitud fija (*figuras 32 y 35*). Esta diferencia se da a la hora de calcular los tiempos de llegada de

paquetes en la etapa de tráfico ya que el tiempo medio de llegada de paquetes es directamente proporcional a la longitud del paquete. Es decir, los tiempos entre llegada de paquetes son calculados teniendo en cuenta la longitud de los paquetes, generándose así, tiempos de llegada de paquetes mas cercanos entre si; esto aumenta la probabilidad de cola de paquetes en la entrada del clasificador.

6.3 RESULTADOS OBTENIDOS PARA TIEMPO DE RETARDO EN PLANIFICACIÓN

La configuración de simulación de las gráficas que se muestran a continuación, es la misma que la usada para obtener las gráficas de los tiempos de retardo medio en el clasificador, solo que en estas gráficas se analiza el retardo promedio de los paquetes en el planificador, para esto se simulo el procesamiento de 50 paquetes, variando el tipo de tráfico simulado, la configuración de longitud de paquetes y el tipo de planificador usado.

La tabla de la *figura 36*, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 36. Tiempo promedio de retardo de PQ1

Proceso	Selección de proceso	Opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[3000]
	D-MMPP	Número de Paquetes =[50]
		Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[10Mbps]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----

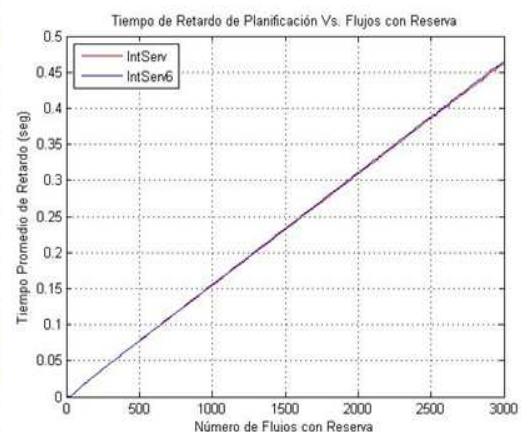
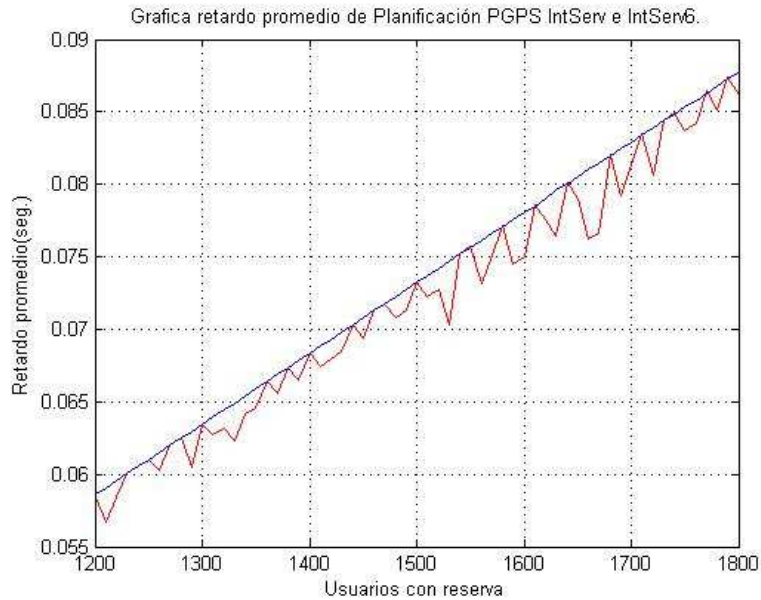


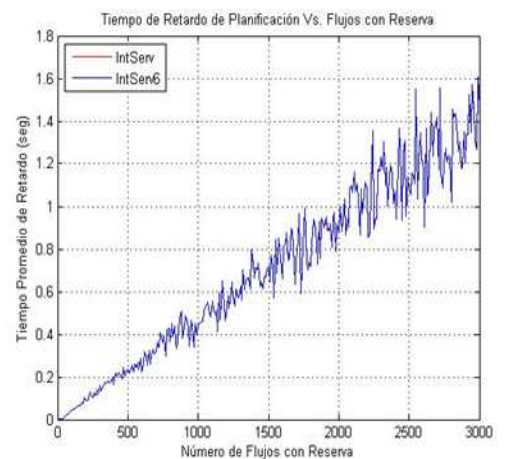
Figura 37. Acercamiento a la gráfica Tiempo promedio de retardo de PQ1



La tabla de la figura 38, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 38. Tiempo promedio de retardo de PQ2

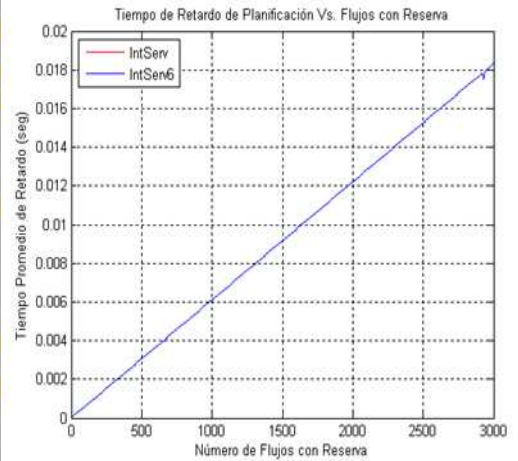
Proceso	Sección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[512 bits] en media
		Número Max. de reservas =[3000]
		Número de Paquetes =[50]
	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[10Mbps]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de New Call
	Cutoff Priority Scheme	Variación de Handoff
		Variación de New Call
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 39*, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 39. Tiempo promedio de retardo de *PGPS1*

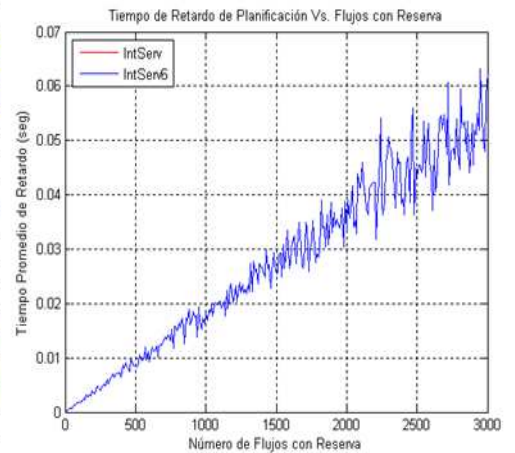
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[3000]
	D-MMPP	Número de Paquetes =[50]
		Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[10Mbps]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	Número de Reservas Fijo
	PQ	- - - - -



La tabla de la *figura 40*, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 40. Tiempo promedio de retardo de *PGPS2*

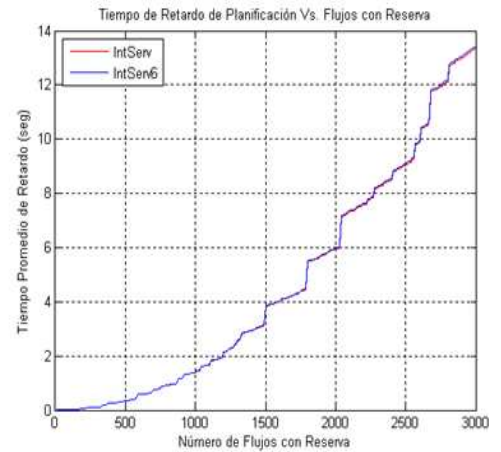
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[512 bits] en media
		Número Max. de reservas =[3000]
	D-MMPP	Número de Paquetes =[50]
		Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[10Mbps]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	Número de Reservas Fijo
	PQ	- - - - -



La tabla de la *figura 41*, corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 41. Tiempo promedio de retardo de *PQ3*

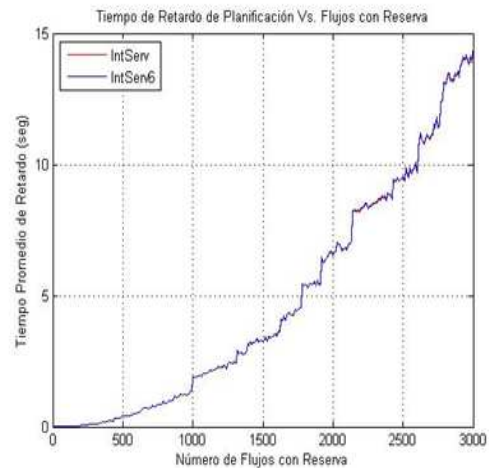
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
	D-MMPP	Número de Paquetes =[]
		Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[512 bits] en media
Número Max. de reservas =[3000]		
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[10Mbps]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de NewCall
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 42* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 42. Tiempo promedio de retardo de *PQ4*

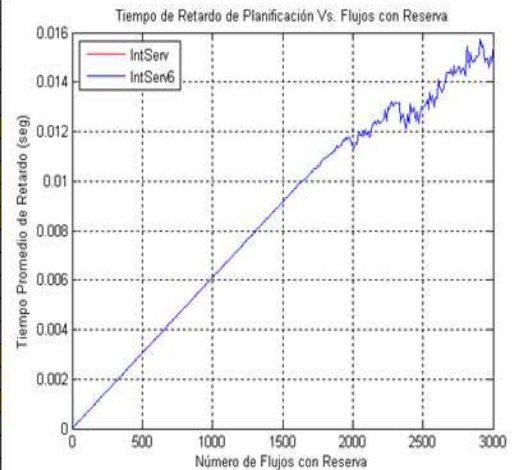
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
	D-MMPP	Número de Paquetes =[]
		Long. Paquete Fija =[]
		Long. Paquete Variable =[512 bits] en media
Número Max. de reservas =[3000]		
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[10Mbps]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de NewCall
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 43* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 43. Tiempo promedio de retardo de *PGPS3*

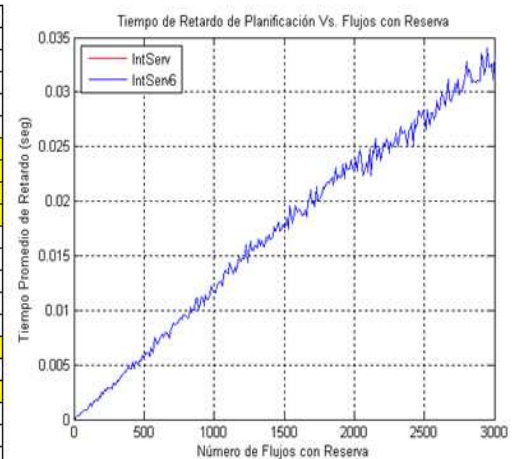
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
	D-MMPP	Número de Paquetes =[]
		Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[512 bits] en media
Control de Admisión	Ancho de Banda Fijo	Número Max. de reservas =[3000]
	Ancho de Banda Aleatorio	Ciclos de Simulación de Proceso =[1]
	New Call Bounding Scheme	Ancho de Banda Max. =[10Mbps]
	Cutoff Priority Scheme	Ancho de Banda Max. =[]
Clasificación	IntServ	Variación de NewCall
	IntServ6	Variación de Handoff
		Variación de Handoff
Planificación	PGPS	Variación de NewCall
	PQ	Variación de Handoff



La tabla de la *figura 44* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 44. Tiempo promedio de retardo de *PGPS4*

Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
	D-MMPP	Número de Paquetes =[]
		Long. Paquete Fija =[]
		Long. Paquete Variable =[512 bits] en media
Control de Admisión	Ancho de Banda Fijo	Número Max. de reservas =[3000]
	Ancho de Banda Aleatorio	Ciclos de Simulación de Proceso =[1]
	New Call Bounding Scheme	Ancho de Banda Max. =[10Mbps]
	Cutoff Priority Scheme	Ancho de Banda Max. =[]
Clasificación	IntServ	Variación de NewCall
	IntServ6	Variación de Handoff
		Variación de Handoff
Planificación	PGPS	Variación de NewCall
	PQ	Variación de Handoff



6.4 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE PLANIFICACIÓN PARA UNA SIMULACIÓN DE RESERV INCREMENTAL.

Se obtuvieron gráficas de retardo en planificación tanto para *INTSERV*, como para *INTSERV6*. Como se puede observar, a simple vista no se nota diferencia alguna, debido a que la discrepancia entre los tiempos de retardo de Planificación de *IntServ* y los de *IntServ6* es pequeña, esto teniendo en cuenta, que los tiempos de retardo en planificación dependen de el tiempo en cola de los paquetes mas el tiempo de servicio de planificación (*longitud / ancho de banda*), es decir, a menos que un paquete tenga que esperar en cola de planificación, el tiempo de retardo de planificación va a ser el igual al tiempo de servicio, de todas maneras se presenta una pequeña diferencia como se observa en la *figura 37*; esta figura es simulada bajos las mismas condiciones que la *figura 36*, pero se analiza en un intervalo de usuarios con reserva entre 1200 y 1800; este acercamiento permite observar que para el caso de la arquitectura *IntServ*, identificada con la línea roja, el tiempo de retardo de Planificación es menor que el de *IntServ6* identificada con la línea azul, esto debido a que al tiempo de retardo de clasificación de *IntServ6* es menor, es decir, los paquetes salen del clasificador mas rápido acrecentando la cola de planificación y por tanto aumentando el tiempo de retardo medio de planificación; este comportamiento se da tanto para *Pgps* como para *PQ*.

Para las simulaciones con longitud de paquetes fija, y tráfico de entrada *Poisson*, se observa un incremento lineal en la gráfica *figuras (36 y 39)*. Este comportamiento se presenta debido a que el número de paquetes para tráfico *Poisson* se mantiene constante al incrementar las reservas, por tanto, el ancho de banda fijo otorgado a cada usuario, va a ser inversamente proporcional al incremento de las reservas, es decir, al aumentar el número de reservas, disminuye el ancho de banda por usuario, aumentando a su vez el tiempo de servicio del planificador, otra razón para este comportamiento, es que el número de paquetes usado en la simulación es pequeño, disminuyendo así la probabilidad

de que se presente una cola de paquetes a la entrada del planificador, tanto para *Pgps* como para *PQ*.

Al llevar a cabo las simulaciones con longitud de paquetes variable (*figuras 38 y 40*), se observa un comportamiento más aleatorio, debido a que el tiempo de servicio del planificador va a ser diferente para cada paquete procesado, esto teniendo en cuenta que al haber paquetes con longitudes mayores, estos van a tener un mayor tiempo de servicio, logrando así, aumentar el tiempo de espera de los paquetes presentes en la cola de entrada al Planificador y por tanto el tiempo de retardo promedio de planificación.

Al implementar el modelo de tráfico *D-MMPP*, se debe tener en cuenta que el número de paquetes aumenta a medida que aumenta el número de reservas. En el caso de *PGPS*, y longitud de paquetes fija (*figura 43*) a medida que el número de reservas sea mayor, aumenta la probabilidad de que se presente una cola y por tanto los tiempos de retardo de planificación presentan un comportamiento más aleatorio cuando el número de reservas es mayor a 2000. En el caso de *PQ* (*figura 41*), al aumentar la llegada de paquetes al planificador, el tamaño de las colas de prioridad aumenta, motivo por el cual los paquetes en las colas de prioridad mas bajas, demoran aun más en ser procesados, incrementando así considerablemente el tiempo promedio total de retardo de planificación. Al aplicar una longitud variable a la simulación, el comportamiento va a ser el mismo antes descrito, tanto para *PGPS* como para *PQ*, solo que presentará un comportamiento aleatorio desde el principio de la simulación, debido a la variación de la longitud de los paquetes (*figuras 42 y 44*), teniendo en cuenta que el tiempo de servicio de planificación es (*longitud / ancho de banda*).

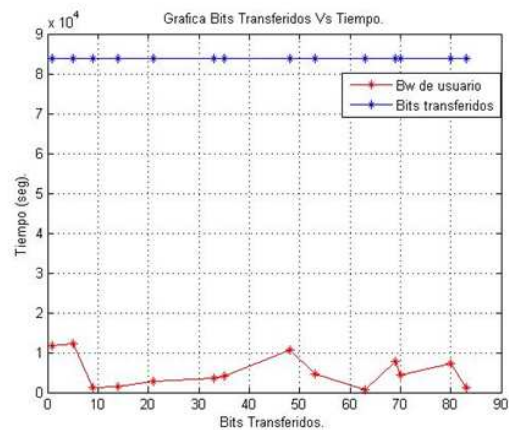
6.5 RESULTADOS OBTENIDOS DE ANCHO DE BANDA USADO EN PLANIFICACIÓN

La configuración de simulación de las gráficas que se muestran a continuación, es diferente a la usada para obtener las gráficas de los tiempos de retardo medio en el clasificador y los tiempos de retardo medio en el planificador, esto debido a que se quiere analizar el comportamiento del ancho de banda usado por un usuario en el planificador para un número de reservas fijo; para observar estos resultados se debe llevar a cabo una simulación para un número de paquetes grande, con el fin de aumentar el número de paquetes procesados por el planificador, además, se simula una longitud de paquetes mas grande aumentando el número de bits transferidos por usuario.

La tabla de la *figura 45* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 45. Anchos de banda de planificación *PGPS1*

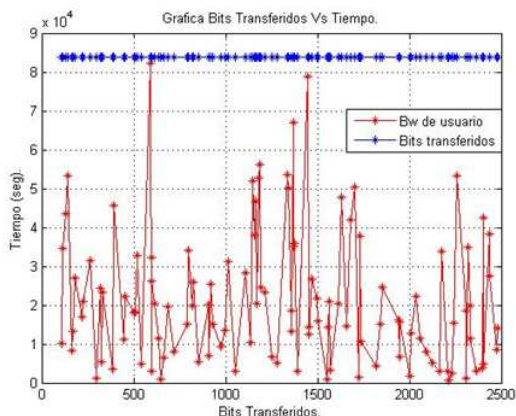
Proceso	Selección de Proceso	Opciones
Trafico	POISSON	Long. Paquete Fija
		Long. Paquete Variable=[20kbits]
		Número Max. de Reservas =[100]
		Número de Paquetes =[10000]
	D-MMPP	Long. Paquete Fija
		Long. Paquete Aleatorio=[20kbits]
Control de Admisión	Ancho de Banda fijo	Ancho de Banda max. =[1Kbps]
	Ancho de Banda Aleatorio	Ancho de Banda max. =[1Mbps]
	New Call Bounding Scheme	Variación de New Call
		Variación de Handoff
	Cutoff Priority Scheme	Variación de New Call
		Variación de Handoff
Clasificación	INTSERV	Incremento Número de Reservas
		Número de Reservas fijo
	INTSERV6	Incremento Número de Reservas
		Número de Reservas fijo
Planificación	PGPS	- - - - -
	PQ	- - - - -



La tabla de la *figura 46* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 46. Anchos de banda de planificación PGPS2

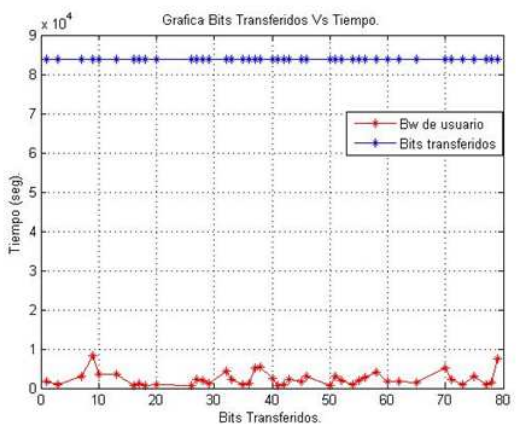
Proceso	Selección de Proceso	opciones
Tráfico	POISSON	Long. Paquete Fija
		Long. Paquete Variable=[20kbits]
		Número Max. de Reservas =[100]
		Número de Paquetes =[10000]
	D-MMPP	Long. Paquete Fija
		Long. Paquete Aleatorio=[20kbits]
		Número Max. de Reservas =[100]
		Ciclos de Simulación de Proceso=[3]
Control de Admisión	Ancho de Banda fijo	Ancho de Banda max. =[1Kbps]
	Ancho de Banda Aleatorio	Ancho de Banda max. =[1Mbps]
	New Call Bounding Scheme	Variación de New Call
	Cutoff Priority Scheme	Variación de Handoff
Clasificación	INTSERV	Incremento Número de Reservas
	INTSERV6	Incremento Número de Reservas
Planificación	PGPS	Número de Reservas fijo
	PQ	- - - - -



La tabla de la figura 47 corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 47. Anchos de banda de planificación PQ1

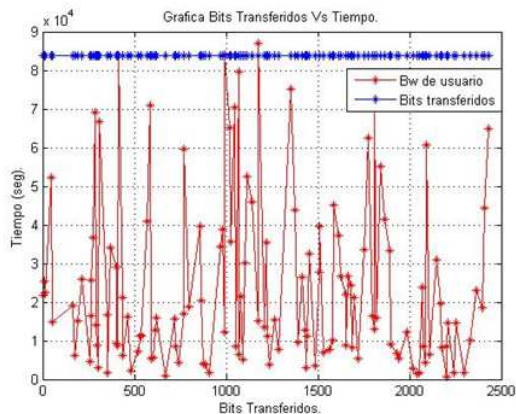
Proceso	Selección de Proceso	opciones
Tráfico	POISSON	Long. Paquete Fija
		Long. Paquete Variable=[20kbits]
		Número Max. de Reservas =[100]
		Número de Paquetes =[4450]
	D-MMPP	Long. Paquete Fija
		Long. Paquete Aleatorio=[20kbits]
		Número Max. de Reservas =[100]
		Ciclos de Simulación de Proceso=[3]
Control de Admisión	Ancho de Banda fijo	Ancho de Banda max. =[1Kbps]
	Ancho de Banda Aleatorio	Ancho de Banda max. =[1Mbps]
	New Call Bounding Scheme	Variación de New Call
	Cutoff Priority Scheme	Variación de Handoff
Clasificación	INTSERV	Incremento Número de Reservas
	INTSERV6	Incremento Número de Reservas
Planificación	PGPS	Número de Reservas fijo
	PQ	- - - - -



La tabla de la figura 48 corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 48. Anchos de banda de planificación PQ2

Proceso	Selección de Proceso	Opciones
Tráfico	POISSON	Long. Paquete Fija
		Long. Paquete Variable=[20kbits]
		Número Max. de Reservas =[100]
		Número de Paquetes =[10000]
	D-MMPP	Long. Paquete Fija
		Long. Paquete Aleatorio=[20kbits]
		Número Max. de Reservas =[100]
		Ciclos de Simulación de Proceso=[3]
Control de Admisión	Ancho de Banda fijo	Ancho de Banda max.=[1Kbps]
	Ancho de Banda Aleatorio	Ancho de Banda max.=[1Mbps]
	New Call Bounding Scheme	Variación de New Call
		Variación de Handoff
	Cutoff Priority Scheme	Variación de New Call
		Variación de Handoff
Clasificación	INTSERV	Incremento Número de Reservas
		Número de Reservas fijo
	INTSERV6	Incremento Número de Reservas
		Número de Reservas fijo
Planificación	PGPS	- - - - -
	PQ	- - - - -



6.6 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE PLANIFICACIÓN PARA UNA SIMULACIÓN DE RESERVA FIJA.

Al implementar el generador de tráfico *D-MMPP* y planificador *PGPS* ó *PQ* (figuras 45 y 47) el número de paquetes que se simulan son pocos, es decir, al llevar a cabo la sumatoria de bits transferidos para el usuario que mas paquetes proceso (*ancho de banda usado*), este valor en ningún momento supera el valor de ancho de banda asignado al usuario, motivo por el cual se observa una gran diferencia entre el ancho de banda asignado y el ancho de banda usado. Para poder obtener una gráfica en la que se observe una mayor transferencia de bits, se deben procesar aproximadamente 10 mil paquetes, aumentando así la probabilidad de que lleguen mas paquetes de un mismo usuario y por tanto la probabilidad que se supere el ancho de banda otorgado; para lograr esto haciendo uso del modelo *D-MMPP* tendrían que simularse muchos ciclos, haciendo que la simulación sea muy demorada para llevar a cabo en el computador que se usó para realizar estas simulaciones; se recomienda que en estudios posteriores se lleven a cabo estas simulaciones haciendo uso de un computador mas adecuado para las mismas.

Al hacer uso del generador de tráfico *POISSON*, tanto para *PQ* como para *PGPS*, se llevaron a cabo simulaciones para mas de 10000 paquetes, permitiendo así observar el comportamiento de los planificadores ver *figuras (46 y 48)*; para el caso de *PGPS*, se puede observar que el ancho de banda usado nunca supera el máximo otorgado al usuario; mientras que para el caso de *PQ* se observa que el ancho de banda usado, supera el ancho de banda otorgado al usuario, esto se debe a que el planificador tipo *PGPS* verifica que el ancho de banda otorgado a un usuario no sea superado por el mismo, mientras que en *PQ* el ancho de banda no importa, el planificador verifica es que los usuarios de prioridad alta sean procesados primero, y así sucesivamente en orden de prioridad.

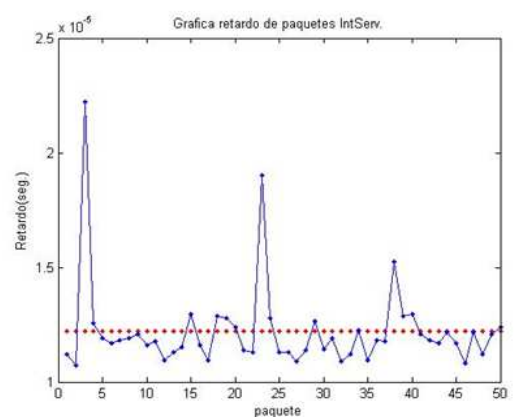
6.7 RESULTADOS OBTENIDOS PARA TIEMPO DE RETARDO EN CLASIFICACIÓN DE UN ROUTER FIJO

La configuración de simulación de las graficas que se muestran a continuación, tiene como nuero de reservas 30 y hace el procesamiento de 50 paquetes, esto con el fin de obtener los tiempos de retardo en clasificación, y compararlos con los obtenidos con la configuración de router inalámbrico.

La tabla de la *figura 49* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 49. Retardos de paquetes en clasificación para router fijo *IntServ* haciendo uso de *Poisson*

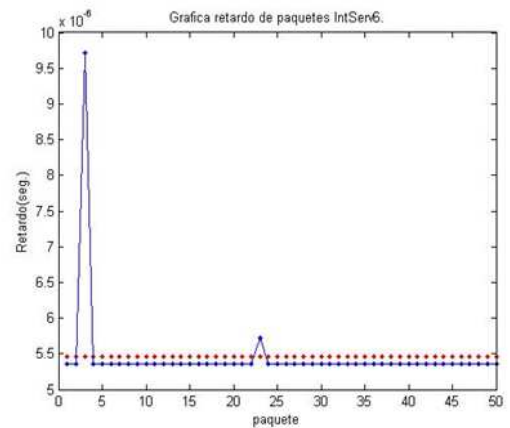
Proceso	Selección de proceso	opciones	
Tráfico	POISSON	Long. Paquete Fija =[512bits]	
		Long. Paquete Variable =[] en media	
		Número Max. de reservas =[30]	
		Número de Paquetes =[50]	
		Long. Paquete Fija =[]	
D-MMPP	D-MMPP	Long. Paquete Variable =[] en media	
		Número Max. de reservas =[]	
		Ciclos de Simulación de Proceso =[]	
		Ancho de Banda Fijo	Ancho de Banda Max. =[]
		Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
Control de Admisión	New Call Bounding Scheme	Variación de NewCall	
	Cutoff Priority Scheme	Variación de Handoff	
		Variación de NewCall	
		Variación de Handoff	
Clasificación	IntServ	Incremento de Número de Reservas	
	IntServ6	Número de Reservas Fijo	
		Incremento de Número de Reservas	
Planificación	PGPS	Número de Reservas Fijo	
	PQ	-----	



La tabla de la *figura 50* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 50. Retardos de paquetes en clasificación para router fijo *IntServ6* haciendo uso de *Poisson*

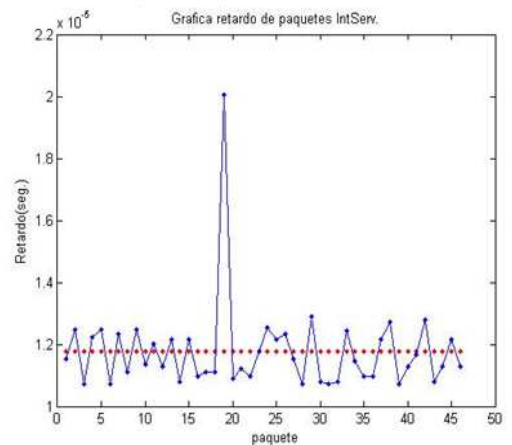
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Número de Paquetes =[50]
	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de NewCall
	Clasificación	IntServ
Número de Reservas Fijo		
IntServ6		Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 51* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 51. Retardos de paquetes en clasificación para router fijo *IntServ* haciendo uso de *D-MMPP*

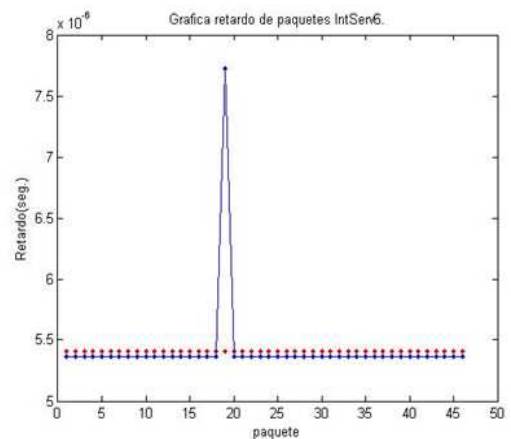
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Número de Paquetes =[]
	D-MMPP	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de NewCall
	Clasificación	IntServ
Número de Reservas Fijo		
IntServ6		Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 52* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 52. Retardos de paquetes en clasificación para router fijo *IntServ6* haciendo uso de *D-MMPP*

Proceso	Selección de proceso	Opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
	D-MMPP	Número de Paquetes =[]
		Long. Paquete Fija =[512 bits]
Long. Paquete Variable =[] en media		
Número Max. de reservas =[30]		
		Ciclos de Simulación de Proceso =[5]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
Variación de Handoff		
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



6.8 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE CLASIFICACIÓN PARA UNA SIMULACIÓN DE RESERVA FIJA.

En las gráficas anteriores se observa el comportamiento del tiempo de retardo de clasificación para *IntServ* figuras (49, 51), e *IntServ6* figuras (50, 52); en ellas se puede observar que el tiempo de retardo de *IntServ6* sigue siendo menor respecto al tiempo de retardo de *IntServ*; esto se debe a que para *IntServ6* la mayor parte de paquetes procesados no presentaron colisión, es decir, las variaciones que se observan en las gráficas de *IntServ6* son debidas a la presencia de colisiones, para las simulaciones con *D-MMPP* se lleva a cabo una simulación para 5 ciclos, esto con el fin de aproximar el número de paquetes generados por *D-MMPP* al número de paquetes definidos en las simulaciones con *Poisson*.

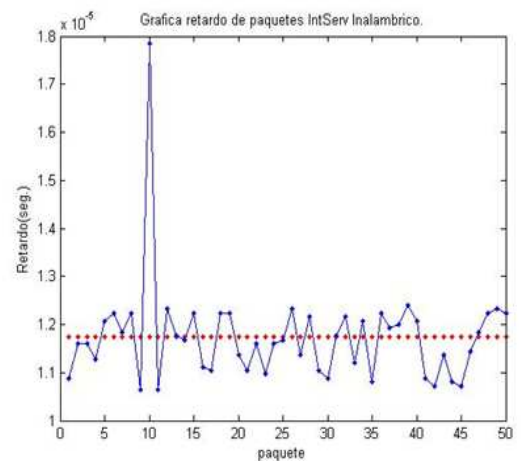
6.9 RESULTADOS OBTENIDOS PARA TIEMPO DE RETARDO EN CLASIFICACIÓN DE UN ROUTER INALAMBRICO

La configuración de simulación de las graficas que se muestran a continuación, tiene como número de reservas máximo 30, esto se da por características del router inalámbrico, además al llevar a cabo las simulaciones con *D-MMPP* se usó un número de ciclos igual a 5 y un número de paquetes igual a 50, con el fin de tener características de simulación iguales a las usadas para obtener los tiempos de retardo de clasificación en un router fijo.

La tabla de la *figura 53* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 53. Retardos de paquetes en clasificación para router inalámbrico *IntServ New Call Bounding Scheme* y variación de nuevas llamadas usando *Poisson*

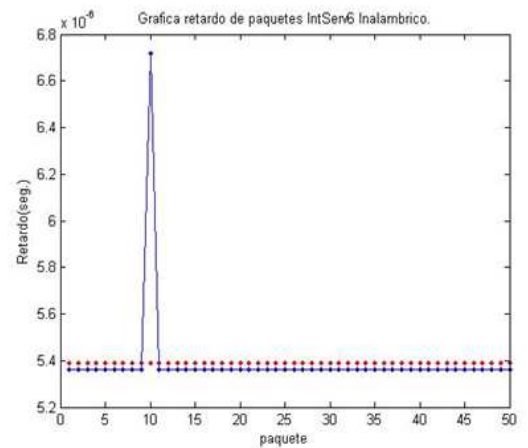
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
	D-MMPP	Número de Paquetes =[50]
		Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de New Call
	Cutoff Priority Scheme	Variación de Handoff
		Variación de New Call
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 54* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 54. Retardos de paquetes en clasificación para router inalámbrico IntServ6 *New Call Bounding Scheme* y variación de nuevas llamadas usando *Poisson*

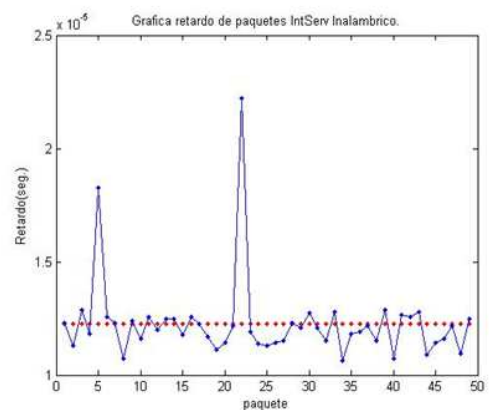
Proceso	Selección de proceso	Opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Número de Paquetes =[50]
	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la figura 55 corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 55. Retardos de paquetes en clasificación para router inalámbrico IntServ *New Call Bounding Scheme* y variación de nuevas llamadas usando *D-MMPP*

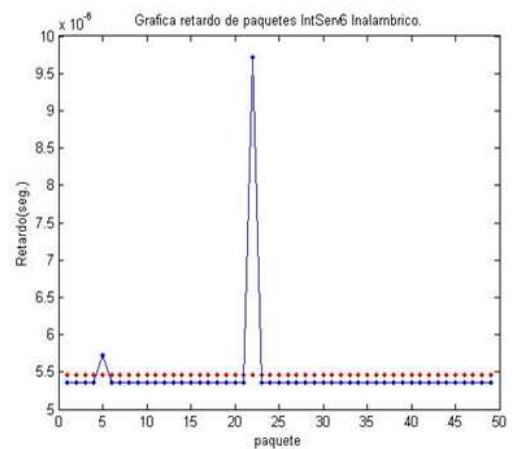
Proceso	Selección de proceso	Opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Número de Paquetes =[]
	D-MMPP	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 56* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 56. Retardos de paquetes en clasificación para router inalámbrico IntServ6 *New Call Bounding Scheme* y variación de nuevas llamadas usando *D-MMPP*

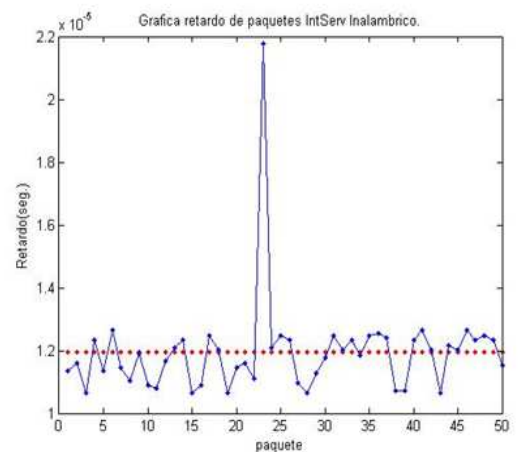
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Número de Paquetes =[]
	D-MMPP	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Ciclos de Simulación de Proceso =[5]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 57* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 57. Retardos de paquetes en clasificación para router inalámbrico IntServ *New Call Bounding Scheme* y variación de traspaso usando *Poisson*

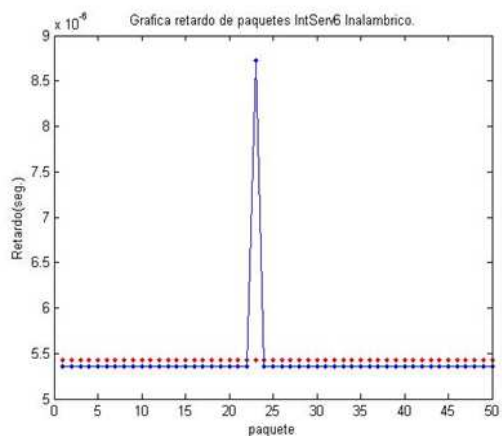
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Número de Paquetes =[50]
	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Ciclos de Simulación de Proceso =[]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 58* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 58. Retardos de paquetes en clasificación para router inalámbrico IntServ6 *New Call Bounding Scheme* y variación de traspaso usando *Poisson*

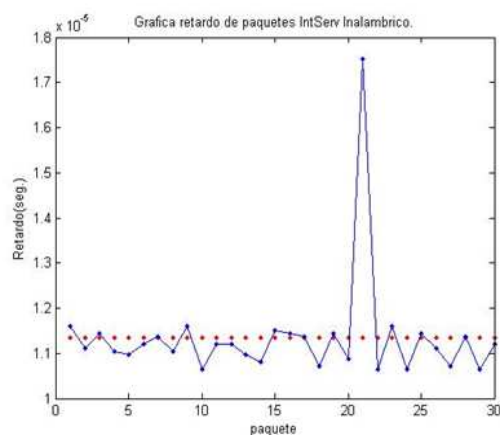
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Número de Paquetes =[50]
	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Ciclos de Simulación de Proceso =[]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 59* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 59. Retardos de paquetes en clasificación para router inalámbrico IntServ New Call Bounding Scheme y variación de traspaso usando D-MMPP

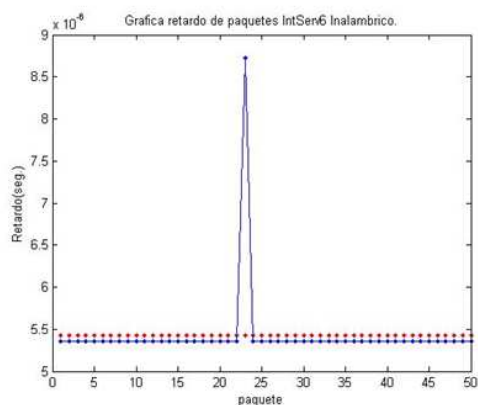
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
	D-MMPP	Número de Paquetes =[]
		Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
Número Max. de reservas =[30]		
		Ciclos de Simulación de Proceso =[5]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la figura 60 corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 60. Retardos de paquetes en clasificación para router inalámbrico IntServ6 New Call Bounding Scheme y variación de traspaso usando D-MMPP

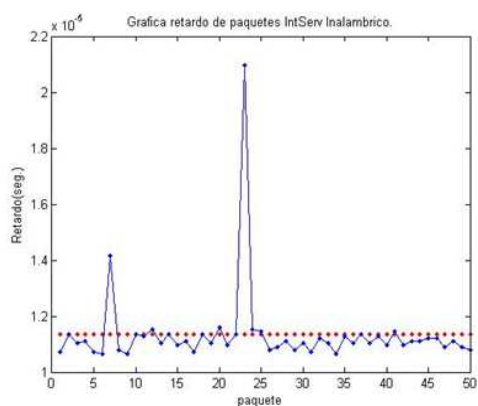
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
	D-MMPP	Número de Paquetes =[50]
		Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
Número Max. de reservas =[30]		
		Ciclos de Simulación de Proceso =[5]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la figura 61 corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 61. Retardos de paquetes en clasificación para router inalámbrico IntServ *Cutoff Priority Scheme* y variación de nueva llamada usando *Poisson*

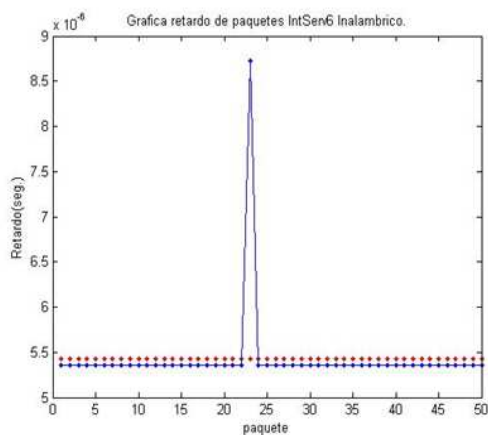
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Número de Paquetes =[50]
Tráfico	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Ciclos de Simulación de Proceso =[]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 62* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 62. Retardos de paquetes en clasificación para router inalámbrico IntServ6 *Cutoff Priority Scheme* y variación de nueva llamada usando *Poisson*

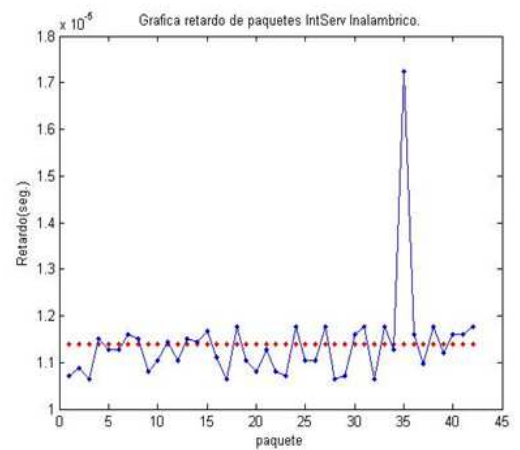
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Número de Paquetes =[50]
Tráfico	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Ciclos de Simulación de Proceso =[]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 63* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 63. Retardos de paquetes en clasificación para router inalámbrico IntServ *Cutoff Priority Scheme* y variación de nueva llamada usando *D-MMPP*

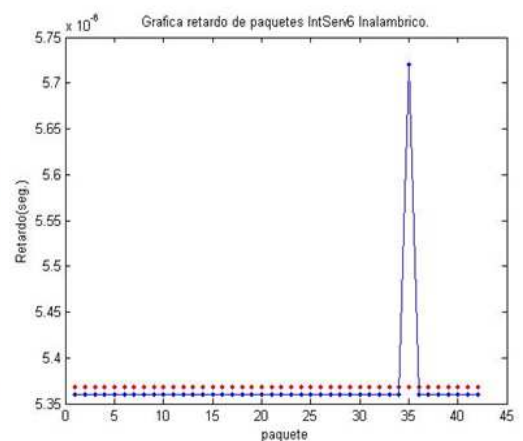
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
	Número de Paquetes =[50]	
	D-MMPP	Long. Paquete Fija =[512 bits]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de NewCall
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 64* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 64. Retardos de paquetes en clasificación para router inalámbrico IntServ6 *Cutoff Priority Scheme* y variación de nueva llamada usando *D-MMPP*

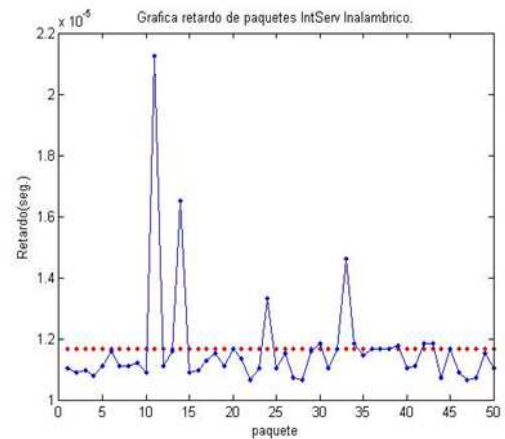
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
	Número de Paquetes =[50]	
	D-MMPP	Long. Paquete Fija =[512 bits]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de NewCall
Clasificación	IntServ	Incremento de Número de Reservas
	IntServ6	Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 65* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 65. Retardos de paquetes en clasificación para router inalámbrico IntServ *Cutoff Priority Scheme* y variación de traspaso usando *Poisson*

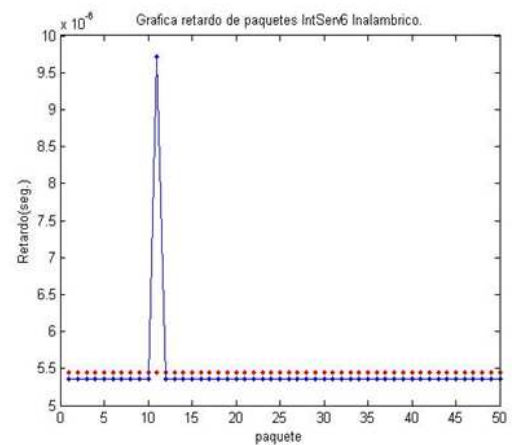
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Número de Paquetes =[50]
	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de Handoff
	Clasificación	IntServ
IntServ6		Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 66* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 66. Retardos de paquetes en clasificación para router inalámbrico IntServ6 *Cutoff Priority Scheme* y variación de traspaso usando *Poisson*

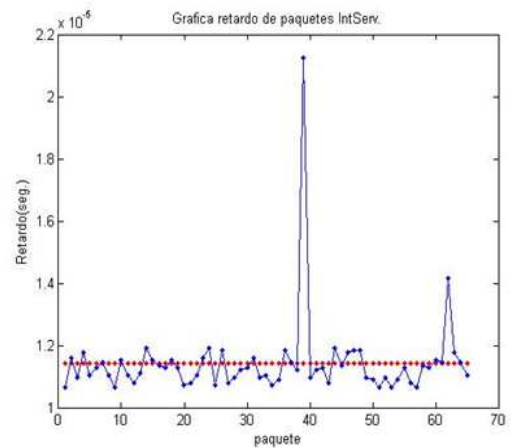
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Número de Paquetes =[50]
	D-MMPP	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
	Cutoff Priority Scheme	Variación de Handoff
		Variación de Handoff
	Clasificación	IntServ
IntServ6		Número de Reservas Fijo
		Incremento de Número de Reservas
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 67* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 67. Retardos de paquetes en clasificación para router inalámbrico IntServ *Cutoff Priority Scheme* y variación de traspaso usando *D-MMPP*

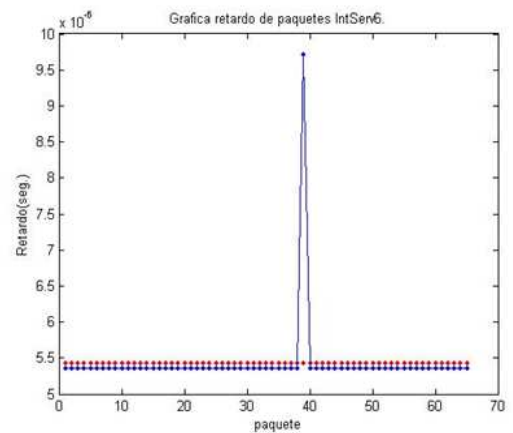
Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Número de Paquetes =[]
	D-MMPP	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Ciclos de Simulación de Proceso =[5]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



La tabla de la *figura 68* corresponde a la configuración del simulador para obtener la gráfica de resultados mostrados en la siguiente figura:

Figura 68. Retardos de paquetes en clasificación para router inalámbrico IntServ6 *Cutoff Priority Scheme* y variación de traspaso usando *D-MMPP*

Proceso	Selección de proceso	opciones
Tráfico	POISSON	Long. Paquete Fija =[]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[]
		Número de Paquetes =[]
	D-MMPP	Long. Paquete Fija =[512 bits]
		Long. Paquete Variable =[] en media
		Número Max. de reservas =[30]
		Ciclos de Simulación de Proceso =[5]
Control de Admisión	Ancho de Banda Fijo	Ancho de Banda Max. =[]
	Ancho de Banda Aleatorio	Ancho de Banda Max. =[]
	New Call Bounding Scheme	Variación de NewCall
		Variación de Handoff
	Cutoff Priority Scheme	Variación de NewCall
		Variación de Handoff
Clasificación	IntServ	Incremento de Número de Reservas
		Número de Reservas Fijo
	IntServ6	Incremento de Número de Reservas
		Número de Reservas Fijo
Planificación	PGPS	-----
	PQ	-----



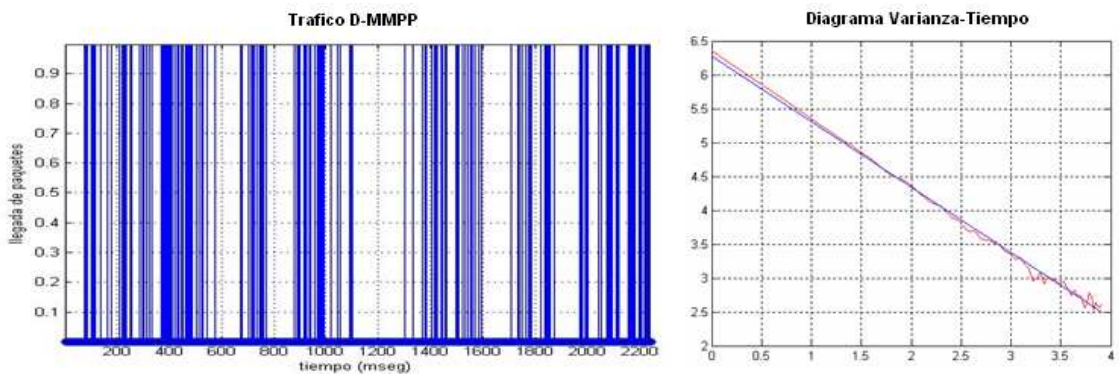
6.10 ANÁLISIS DE RESULTADOS OBTENIDOS DE LA ETAPA DE CLASIFICACIÓN PARA UNA SIMULACIÓN DE RESERV FIJA Y ROUTER INALAMBRICO.

Como se puede observar en las gráficas anteriores se simulan los dos tipos de control de admisión para router inalámbrico *New Call Bounding Scheme* y *Cutoff Priority Scheme* , para los tiempos de retardo de los paquetes en clasificación *IntServ* figuras (53, 55, 57, 59, 61, 63, 65, 67) e *IntServ6* figuras (54, 56, 58, 60, 62, 64, 66, 68); de estas se puede deducir que el comportamiento del tiempo de retardo de clasificación no depende del tipo de CAC seleccionado, ya que el control de admisión se encarga básicamente de simular la entrada y salida de usuarios de la tabla de reservas, haciendo uso de las diferentes probabilidades intrínsecas de cada método, es decir, el tiempo de procesamiento de clasificación no se ve afectado por este cambio, además se puede observar que en estas gráficas *IntServ6* sigue manteniendo un tiempo de retardo menor al de *IntServ*.

El comportamiento observado en las gráficas de *IntServ6* se debe a que para la mayoría de paquetes no había colisión, motivo por el cual el tiempo de retardo es el mismo para ellos, presentándose variaciones solo en caso de presencia de colisiones ver figuras (54, 56, 58, 60, 62, 64, 66, 68); además se puede decir que tanto el comportamiento de *IntServ* como el comportamiento de *IntServ6* es similar al obtenido en las simulaciones de router fijo figuras (49, 50, 51, 52).

6.11 COMPROBACIÓN DE FUNCIONAMIENTO DEL MODELO DE TRÁFICO D-MMPP

Figura 69. Comprobación del modelo de tráfico D-MMPP



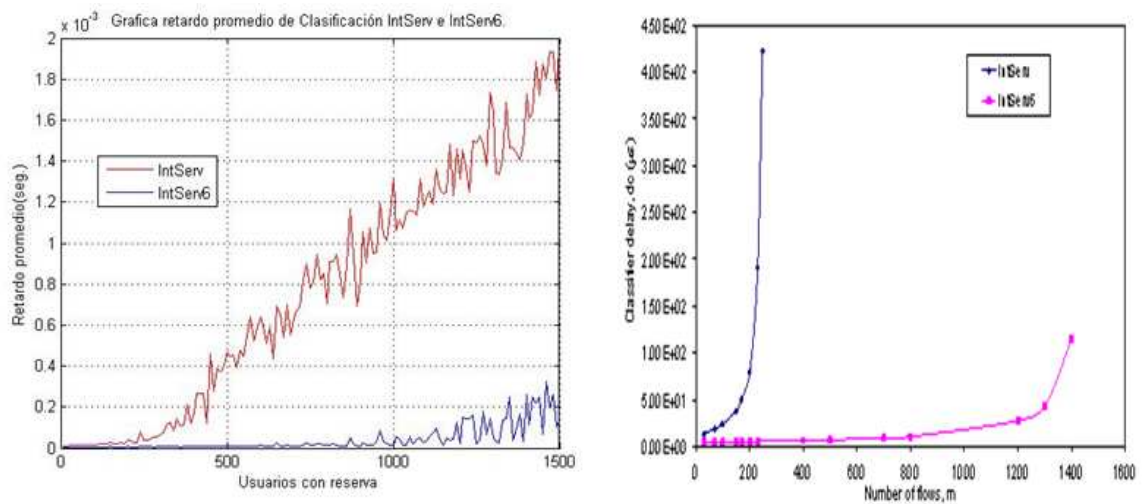
La *figura 67* muestra en la parte izquierda una serie de tiempos de paquetes, que son generadas por el simulador de tráfico D-MMPP, con fin de obtener datos confiables para ser utilizados en el simulador del router. Este generador se comprueba aplicando el método del diagrama varianza-tiempo, donde se estima con suficiente precisión el parámetro de *Hurst*, el cual determina si el tráfico posee características de *autosimilaridad*. El parámetro de *Hurst* para estos paquetes es de $H=0.54965$, que se determina a partir de la pendiente de la gráfica de la *figura 61* parte derecha, lo cual indica que el simulador D-MMPP genera tráfico *autosimilar*, ya que se encuentra en el rango de *autosimilaridad* de $0.5 < H < 1$.

6.12 VERIFICACIÓN DEL MODELO MATEMÁTICO

Comparando el comportamiento gráfico de los resultados obtenidos por el modelo matemático desarrollado en [8], con los obtenidos por el simulador en el presente proyecto, se puede observar que tienen un comportamiento similar tanto para *INTSERV* como para *INTSERV6* (*Figura 68*), ya que los dos tienen un incremento exponencial del tiempo de retardo promedio en clasificación; además este

incremento se empieza a dar para el mismo número de usuarios con reserva aproximadamente (200 reservas para el caso de IntServ y 1200 reservas para el caso IntServ6), una verificación mas profunda de esta similitud se puede observar en [24].

Figura 70. Retardo medio de clasificación IntServ e IntServ6 del simulador y el modelo matemático



7. CONCLUSIONES Y RECOMENDACIONES

VERIFICACIÓN DEL MODELO MATEMÁTICO DE UN ROUTER DE SERVICIOS INTEGRADOS

En esta tesis se realizaron simulaciones del proceso de un router que funciona bajo la arquitectura de servicios integrados (IntServ: Integrated Services) y la arquitectura de servicios integrados sobre IPv6 (*IntServ6*) propuesto en [24]. Además se compararon los resultados obtenidos referentes a los tiempos de retardo en clasificación para los dos tipos de arquitectura, concluyendo que la arquitectura *IntServ6* posee un mejor desempeño que la arquitectura IntServ al prestar un servicio de clasificación más eficiente, esto debido a que el cálculo del Número Hash para el caso de IntServ6 es hecho por el usuario y no por cada router, además, la búsqueda de la reserva requiere de una sola lectura en tabla *Hash* para este tipo de arquitectura.

Se concluye que para IntServ el mayor tiempo de retardo medio de clasificación es debido a la búsqueda de la reserva en la tabla Hash, tabla de resolución de colisiones y el cálculo del número Hash, teniendo en cuenta que al aumentar el número de usuarios con reserva aumentan los tiempos de búsqueda en la tabla Hash, caso contrario para IntServ6, el incremento del retardo medio de clasificación es determinado solo por los tiempos de búsqueda en tabla de resolución de colisiones, ya que la búsqueda en la tabla Hash requiere de una sola instrucción para llevarse a cabo.

Estos resultados se contrastan en [24] con el modelo teórico simulado en [8], obteniendo un comportamiento parecido al presentarse un crecimiento de retardo

para un número de reservas similar, concluyendo que para una arquitectura del tipo *IntServ* el retardo medio de paquetes en el clasificador es aceptable hasta los 200 usuarios con reserva, y para el caso de *IntServ6* es aceptable hasta los 1200 usuarios con reserva, observando el mejor desempeño de *IntServ6* respecto a *IntServ*.

Se desarrollaron dos tipos de generación de tráfico para implementar en la simulación, un modelo *Poisson* y un modelo *D-MMPP*, comprobando el funcionamiento de este último mediante el método *Diagrama Varianza Tiempo*, que calcula el parámetro de Hurst, comprobando que el modelo *D-MMPP* desarrollado posee características autosimilares, aproximándose al comportamiento del tráfico en la red. Para el caso del modelo de *Poisson* se verificó su funcionamiento en [8].

Se desarrollaron dos módulos de planificación, uno que simula el modelo *PGPS* (*Packet-by-packet Generalized Processor Sharing*), que planifica los paquetes limitando los bits transferidos, tomando como referencia el ancho de banda otorgado al usuario; y otro que simula el modelo *PQ* (*Priority Queuing*), que planifica los paquetes teniendo en cuenta cuatro colas de prioridades “alta, media, normal, baja”. Este último, lleva a cabo una planificación injusta, ya que mientras hallan paquetes en las colas de prioridades más altas, los paquetes de las demás colas serán retrasados, motivo por el cual se concluye, que la prestación del servicio de planificación que otorga *PGPS* es mucho más justa que la que presta *PQ*.

También se concluye que tanto para el modelo Planificación *PQ* como para el modelo *PGPS*, los tiempos de retardo medio de Planificación de los paquetes procesados mediante una arquitectura *IntServ*, son un poco menores a los tiempos de retardo medio de Planificación de los paquetes procesados mediante una arquitectura *IntServ6*; estos últimos, al tener menor tiempo de retardo en la

etapa de Clasificación, llegan mas rápido al Planificador, motivo por el cual se incrementa la cola de paquetes a la entrada del mismo y por tanto se incrementa el tiempo de retardo de paquetes en el planificador.

Se implementaron los modelos *New Call Bounding Scheme* y *Cutoff Priority Scheme* desarrollados por Yuguang Fang en [10] para simular el funcionamiento de un router inalámbrico, observando el comportamiento de los retardos de paquetes en la clasificación (variando la intensidad de tráfico de nuevas llamadas y la intensidad de tráfico de llamadas de traspaso o *handoff*) observando que el comportamiento de los tiempos de retardo es similar al obtenido para una simulación de router fijo.

El objetivo de este simulador se centro en la verificación del modelo matemático desarrollado en [8], para lo cual, se llevaron a cabo simulaciones haciendo uso de las mismas condiciones iniciales; obteniendo similitudes en los tiempos de retardo de clasificación calculados, y en el comportamiento de sus respectivas gráficas, concluyendo que la arquitectura IntServ6, presta un mejor servicio que la arquitectura IntServ, y que el modelo matemático entrega datos aproximados a los obtenidos por el simulador de router desarrollado, esta verificación se lleva a cabo mas a fondo en [24].

El simulador desarrollado permite al usuario seleccionar en cada etapa, diferentes esquemas de funcionamiento, permitiéndole configurar el tipo de router que más se acerca a sus necesidades. Esta característica del simulador permite su uso en educación e investigaciones futuras.

El desarrollo de este simulador se deja abierto para que en investigaciones futuras se puedan implementar nuevos esquemas de funcionamiento en cada etapa del router, incrementando así las configuraciones posibles en el simulador;

permitiendo al usuario tener más herramientas para determinar que configuración se acerca al tipo de router que desea simular.

GLOSARIO

ATM Modo de Transferencia Asíncrona o Asynchronous Transfer Mode es una tecnología de telecomunicación desarrollada para hacer frente a la gran demanda de capacidad de transmisión para servicios y aplicaciones.

Backbone se refiere a las principales conexiones troncales de Internet. Está compuesta de un gran número de routers comerciales, gubernamentales, universitarios y otros de gran capacidad interconectados que llevan los datos entre países, continentes y océanos del mundo.

Best Effort Mejor esfuerzo. Servicio más simple en una red. No provee QoS.

Buffer Un buffer de datos es una ubicación de la memoria en una computadora o en un instrumento digital reservada para el almacenamiento temporal de información digital, mientras que está esperando ser procesada.

CAC Call Admission Control. Control de admisión de llamadas.

Cutoff Priority Esquema de CAC o esquema de control de admisión de llamadas.

Datagrama Fragmento de paquete que es enviado con la suficiente información como para que la red pueda simplemente encaminar el fragmento hacia el dispositivo receptor, de manera independiente a los fragmentos restantes. Esto puede provocar una recomposición desordenada o incompleta del paquete en el dispositivo destino. La estructura de un datagrama es: cabecera y datos.

DiffServ Differentiated Services. Servicios Diferenciados

Ethernet Nombre de una tecnología de redes de computadoras de área local (LANs) basada en tramas de datos.

FCFS First come First Serve, primero en llegar, primero en salir.

FIFO First In, First Out, primero en entrar, primero en salir.

Handoff Traspaso.

Hash Número, algoritmo o tabla, descrita o conformada por los cinco campos: identificación de protocolo, puerto de destino y fuente y dirección de destino y fuente o quintupla, que se encuentran en el encabezado de un paquete IP.

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force. Organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, tales como transporte, encaminamiento, seguridad.

IntServ Integrated Services. Servicios Integrados.

Jitter Variación del retardo, diferencia entre el mayor y el menor.

Latencia Retardo entre el tiempo desde cuando la señal en el remitente es generada hasta el tiempo cuando la señal es ejecutada en la salida del receptor.

Leaky bucket y token bucket Algoritmo usado para controlar la tasa a la cual los datos deben ser insertados en la red.

M/M/1 Tipo de cola. Notación de Kendall.

MBONE Red experimental multicast construida en el Internet para audio y videoconferencia.

MPLS MultiProtocol Label Switching. Tecnología para soporte de QoS en Internet fundamentada en la optimización de desempeño.

Multicast Envío de información desde un único emisor a un único receptor.

Multicast Multifusión. Es el envío de la información en una red a múltiples destinos simultáneamente, usando la estrategia más eficiente para el envío de los mensajes sobre cada enlace de la red sólo una vez y creando copias cuando los enlaces en los destinos se dividen.

New Call bounding Esquema de CAC o control de admisión de llamadas.

PGPS Packet-by-Packet Generalized Processor Sharing.

Playback son aplicaciones que primero organizan la información para poderla entregar de forma adecuada y constante.

QoS Quality of Service ó Calidad de Servicio.

Router Dispositivo o software emulador, que determina la dirección a la que han de ser enviados los paquetes que se transmiten, normalmente forma parte de un switch de una red.

RSVP Resource Reservation Setup Protocol.

TCP Transmission Control Protocol. Protocolo de control de transmisión.

UDP User Datagram Protocol. Protocolo de datagrama de usuario.

WFQ Weighted Fair Queuing. Tipo de planificador. Algoritmo.

WWW World Wide Web

BIBLIOGRAFÍA

- [1] Vladimir E. Ordóñez, Andrés F. Correa Montesinos, *Análisis de tráfico IP de la Universidad Pontificia Bolivariana seccional Bucaramanga mediante el modelo MMPP utilizando una herramienta de simulación*, Universidad Pontificia Bolivariana, 2006.
- [2] Alonso Zárate, Luis Gonzaga “*Técnicas de acceso y gestión de recursos para garantizar calidad de servicio en sistemas de comunicaciones móviles basados en CDMA*”, capítulo 6 Tesis Doctoral-UPC, 2001.
- [3] Jhon Jairo Padilla Aguilar, Josep Paradell Aspas, Mónica Karel Huertas, Xavier Hesselbach, *SOPORTE DE QOS SOBRE IPV6 CON INTSERV6*, *Revista Gerencia Tecnológica Informática*, GTI, Número 8, Volumen 4, Fecha de Publicación: 06-05-2005.
- [4] Enrique Hernández Orallo, *Transmisión de datos en Tiempo Real Síntesis de protocolos y redes para transmisión en tiempo real*, Trabajo de Doctorado, Universidad Politécnica de Valencia. 2000.
- [5] Wang, Z., *Internet QoS: Architectures and Mechanisms for Quality of Service*. 1st edition ed. March 15, 2001: Morgan Kaufmann. 256.
- [6] Juan Antonio García, *MBone: Arquitectura y Aplicaciones*, *Red IRIS*, 2007 <https://www.rediris.es/rediris/boletin/43/enfoque3.html>
- [7] Sebastián Andrés Álvarez Moraga-Agustín José González Valenzuela, *Estudio y configuración de calidad de servicio para protocolos IPV4 e IPV6 en una red de fibra óptica WDM*, *Rev. Fac. Ing. - Univ. Tarapacá de Arica, Chile*, Vol. 13 N° 3, 2005, pp. 104-113 2005.
- [8] Álvaro J. Rodríguez Ramírez, *Simulación de la propuesta de calidad de servicio Intsev6*, Tesis de grado, Universidad Pontificia Bolivariana, 2006.
- [9] Danysoft, *Protocolo de Reservación de Recursos: RSVP*, 2006 <http://www.danysoft.info/free/reservarecursos.pdf>

- [10] Fang, Y. and Y. Zhang, *Call admission control schemes and performance analysis in wireless mobile networks*. Vehicular Technology, IEEE Transactions on, 2002. 51(2): p. 371-382.
- [11] Santiago Felici, *“Evolución de mecanismos de calidad de servicio en los routers para servicios multimedia”* Tesis Doctoral “Sistemas y servicios telemáticos” Universidad e Valencia, España, 1999.
- [12] Variable aleatoria, 2007.
<http://www.southlink.com.ar/vap/VARIABLE%20ALEATORIA.htm>
- [13] Jurado, E. Casilari, A. Reyes, A. Díaz-Estrella y F. Sandoval, *Modelado Markoviano de Tráfico Agregado ATM*, Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga. Actas del Congreso JITEL'99, Leganés (Madrid, España), Setiembre, 1999, pp. 193-199
- [14] Ricardo VECCHIO, *Modelización de tráfico auto-similar y evaluación de sus efectos en el tamaño de las colas*, Instituto tecnológico de buenos aires escuela de posgrado tesis de maestría, 2003.
- [15] Oriol López Bassa, *“Caracterización de tráfico TPC con propiedades fractales sobre redes WLAN”*. Trabajo de final de carrera, Universidad Politécnica de Cataluña, 2006.
- [16] Marco Aurelio Alzate Monroy, *Introducción al tráfico autosimilar en redes de comunicaciones*, Universidad Distrital Francisco José de Caldas, revista de ingeniería Vol 6 No. 2 año 2001.
- [17] *Teoría de líneas de esperas*. Sistema de Información de Investigación de Operaciones II, Instituto Tecnológico de Veracruz, 2007.
<http://mx.geocities.com/troyescvm/ModeloColas.htm>
- [18] Ing. Iván G. Andrade D. *Teoría de colas*, Universidad del Azuay. Cuenca, Ecuador, 2007 www.uazuay.edu.ec/estudios/sistemas/operativa2/colas.ppt
- [19] Padilla, J., J. Paradells, and A. Rodriguez, *Supporting QoS over IPv6 wireless networks with IntServ6*. (accepted) PIMRC'06 The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Helsinki -Finland, 2006.

- [20] Padilla, J., J. Paradells, and A. Rodriguez, *Effects of Mobility on Mean Packet Delay over Integrated Services Wireless Networks*. (accepted) WiCOM (WCNM 2006) International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan - China, 2006.
- [21] Michael C. Jeruchim, P.B.K.S.S., *Simulation of Communication Systems, Modeling, Methodology, and Techniques, second edition*. 2000: Kluwer Académico / Plenum Publishers.
- [22] Versión electrónica del manual de la Universidad de Málaga: *Bioestadística: Métodos y Aplicaciones U.D. Bioestadística*. Facultad de Medicina. Universidad de Málaga, 2007. ISBN:847496-653-1
- [23] STALLINGS WILLIAM: *High Speed Networks*. Prentice Hall (eds.). USA 1998. ISBN 0-13-52596657
- [24] (Padilla Aguilar 2008), “*Contribución al soporte de calidad del servicio en redes móviles*” Tesis Doctoral, Universidad Politécnica de Cataluña, Barcelona, España, 2008.
- [25] Néstor Misael Peña Traslaviña, “*Modelos de Trafico en Tiempo Real*”. GEST – Grupo de Electrónica y Sistemas de Telecomunicaciones, Universidad de los Andes, 2006.
- [26] Carlos F. Belaustegui Gotilla, *Cadenas de Markov y Teoría de Colas*. Universidad de Buenos Aires, Argentina, 2003.
<http://www.fi.uba.ar/materias/6615/Material/markov.pdf>
- [27] Germán A. Borray Martínez, JGermán A. Borray Martínez, Juan P. Páez Sánchez, Juan M. Wilches Durán “*Modelado matemático del tráfico de redes de telecomunicaciones de acceso fijo*” Pontificia Universidad Javeriana, Facultad de Ingeniería Departamento de Electrónica, Julio 2001.
- [28] Historia de Internet, 2007.
<http://homepage.mac.com/xelac/albanet/articulos/default.htm>
- [29] David Rincón, Associate Member, IEEE, y Cristina Cano. *Mitigación de la Dependencia a Largo Plazo del Tráfico en Redes WLAN IEEE 802.11* IEEE LATIN AMERICA TRANSACTIONS, VOL. 5, NO. 3, JUNE 2007

- [30] González Sánchez, José Luís, *Protocolo activo para transmisiones garantizadas sobre una arquitectura distribuida y multiagente en redes ATM*, Capitulo 4. Universidad Politécnica de Cataluña, 2001.
- [31] Daniel P. Heyman and David Lucantoni, *Senior Member, IEEE*, "Modeling Multiple IP Traffic Stream With Rate Limits". IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11, NO. 6, DECEMBER 2003.
- [32] Padilla, J., et al. *IntServ6: An approach to Support QoS over IPv6 Networks*. In *10th International Symposium on Computer and Communications*. 2005. Cartagena, España.
- [33] Daniel P. Heyman y David Lucantoni, Modeling IP traffic streams with rate limits. IEEE/ACM Transactions on Networking (TON). Volumen 11, Issue 6, December 2003. Pages: 978-958, ISSN:1063-6692
- [34] Gunter Bolch, Stefan Greiner, Hermann de Meer, Kishor S. Trivedi, *Queuing Networks and Markov chains, Modeling and performance evaluation with computer science applications*. Ed. Wiley-Interscience 2006.

ANEXOS

Anexo A Artículo tesis

Anexo B Manual Usuario

Anexo C CD Simulador MATLAB y documentación en formato digital de la tesis

ANEXO B

MANUAL DE USUARIO

SIMULACIÓN ROUTER DE SERVICIOS INTEGRADOS

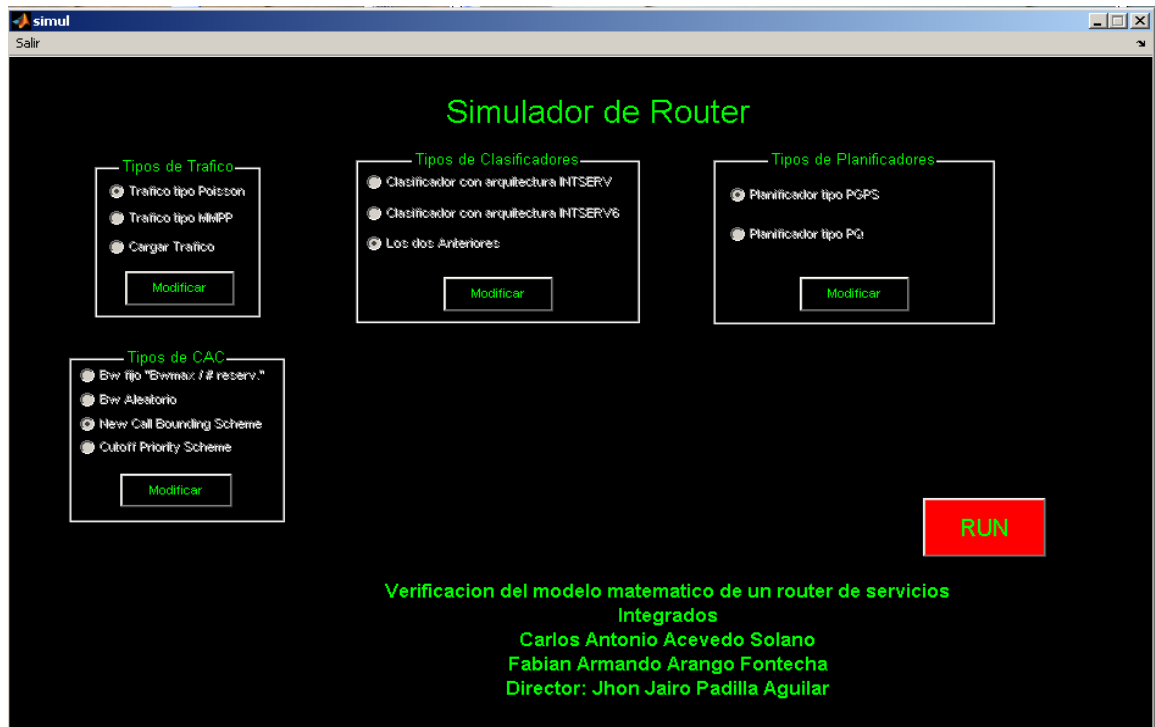
SIMULACIÓN ROUTER DE SERVICIOS INTEGRADOS

Los programas desarrollados en la investigación se realizaron con el software de simulación Matlab. Estos programas son en general muy complejos para que puedan ser utilizados individualmente de forma didáctica y/o para futuras las investigaciones; debido a esto, se ha desarrollado una interfaz grafica del simulador que integra todos los subprogramas del router de *servicios integrados*, esto con el fin de facilitar su uso.

El simulador esta compuesto por cuatro grandes módulos, como se muestra en la *figura 1* que corresponde al menú principal de la interfaz grafica del simulador, ofreciendo al usuario una amplia variedad de opciones que le permiten configurar el tipo de router que más se acerca a sus necesidades. Estos módulos son:

- Generación de tráfico IP
- Módulo de Control de Admisión (CAC)
- Módulo Clasificador
- Módulo Planificador

Figura 1. Menú principal del simulador del router



Para correr la simulación del router se necesita ingresar y seleccionar las condiciones bajo las cuales el usuario desea llevar a cabo la simulación; esta selección se realiza en la interfaz grafica del simulador, sin embargo el simulador tiene predeterminadas unas condiciones iniciales de simulación que son explicadas mas adelante.

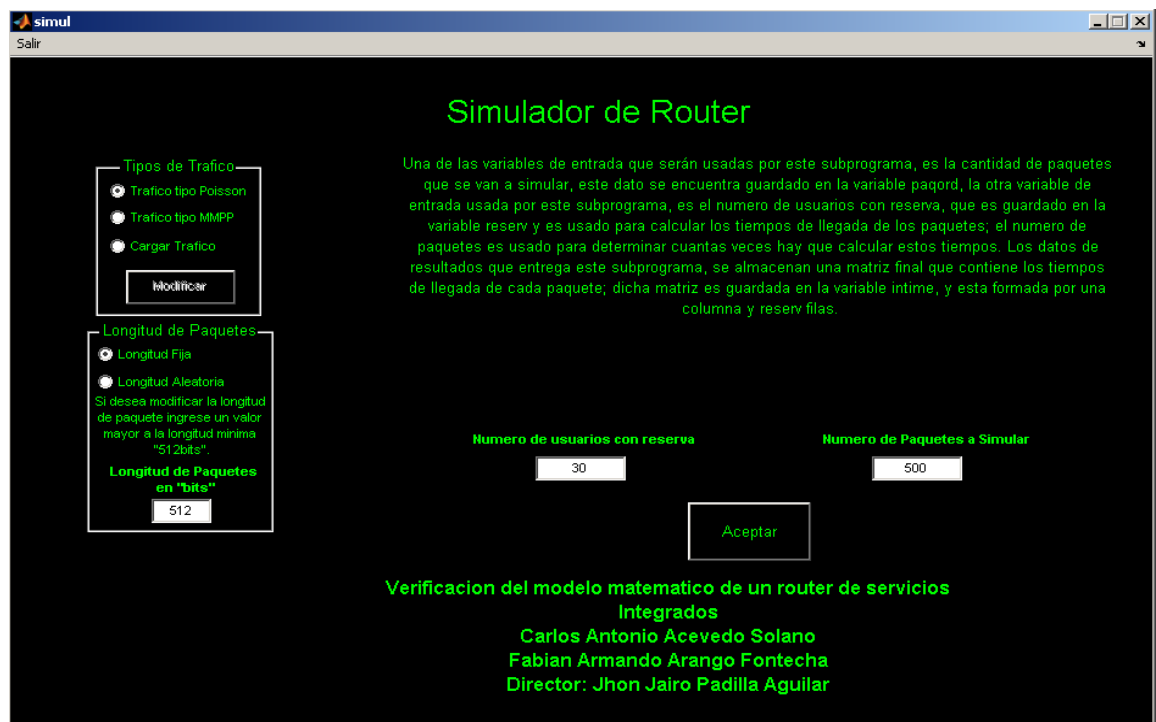
Módulo de tráfico

Este módulo cuenta con dos esquemas de simulación:

- *POISSON*
- *D-MMPP*.

Estos esquemas tienen como opciones a modificar la longitud de los paquetes, ya sea de longitud fija o longitud aleatoria y a su vez se puede ingresar el valor *bits*. Otra opción es la de ingresar el número de usuarios con reserva y la cantidad de paquetes que se desean simular; para el caso de *D-MMPP* se tienen la opción de número de *ciclos* de simulación, al aumentar el número de *ciclos* aumenta el número de paquetes simulados. Ver figura 2

Figura 2. Módulo de Tráfico de Paquetes



Módulo de Clasificación

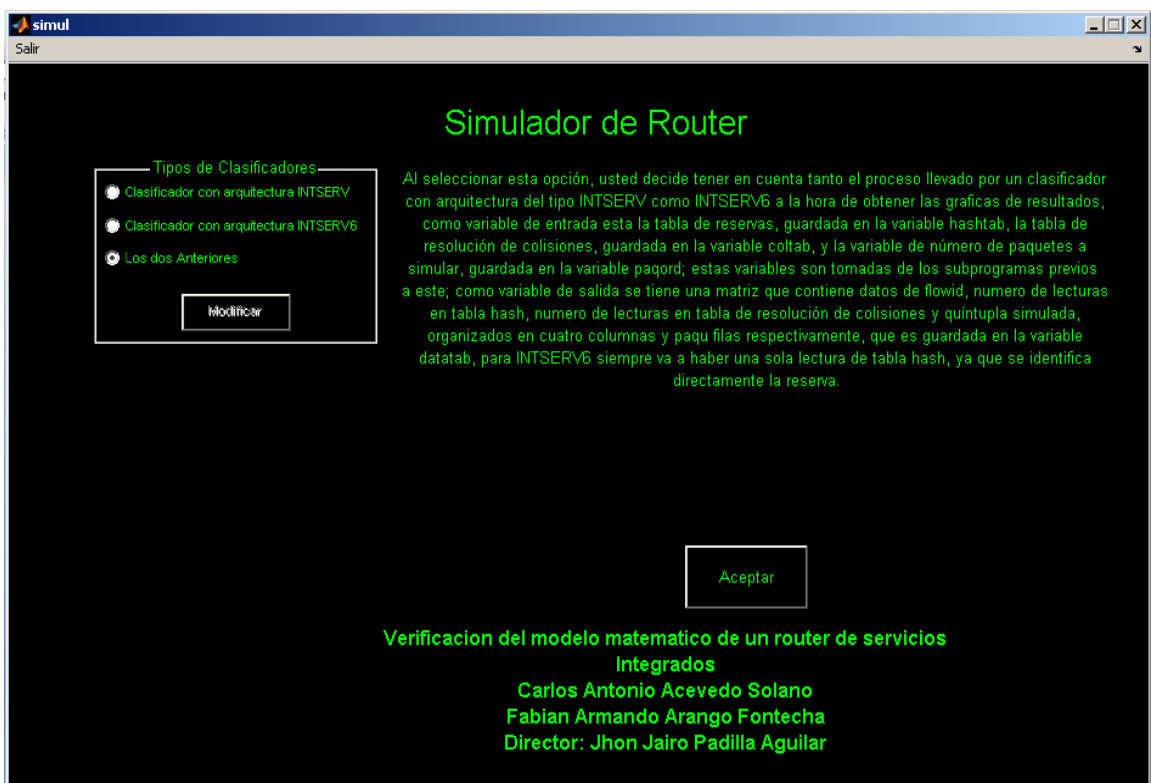
Este módulo cuenta con tres esquemas de simulación:

- Clasificador con arquitectura IntServ

- Clasificador con Arquitectura IntServ6
- Clasificador de las dos arquitecturas(IntServ y IntServ6)

Estos esquemas permiten al usuario determinar que tipo de arquitectura desea observar en las graficas de resultados. *Ver figura 3*

Figura 3. Módulo de Clasificación



Módulo de Planificación

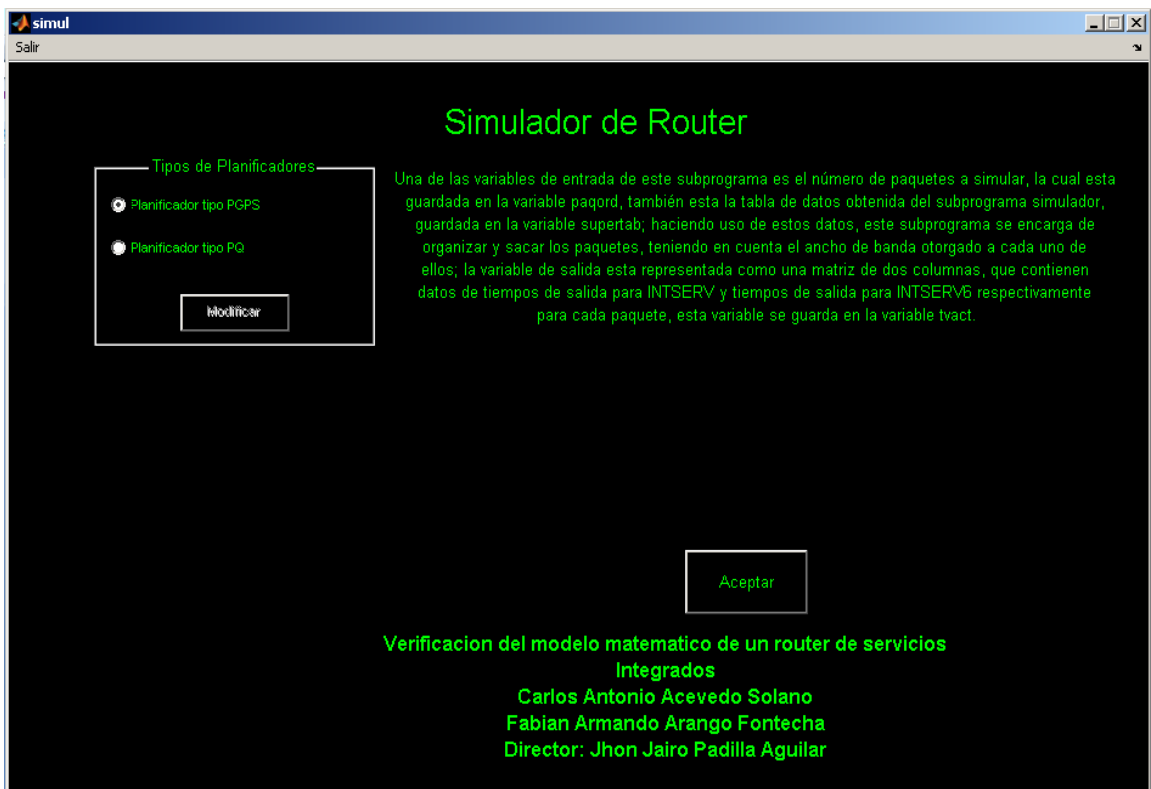
Este módulo cuenta con dos esquemas de simulación:

- Planificador *PQ*

- Planificador *PGPS*

Estos esquemas permiten al usuario determinar que tipo de *Planificación* desea usar en la simulación. Ver figura 4

Figura 4. Módulo de Planificación



Módulo de CAC

Este módulo cuenta con siete esquemas de simulación, cuatro corresponden al tipo de control de admisión que se desean simular, dos correspondientes al tipo de router que se desea simular y otras dos correspondientes al tipo de simulación que se desea llevar a cabo, respectivamente como se muestra a continuación:

- New Call Bounding Scheme
- Cutoff Priority Écheme
- BW Fijo
- BW Aleatorio
- Router Fijo
- Router Inalámbrico
- Número de Reservas Fijo
- Número de Reservas incremental

Estos modelos se pueden seleccionar en la interfaz grafica del simulador y también se ingresa valores como el *Ancho Banda* del canal dado en *Mbps* y el inicio del incremento de reservas se muestra en la *figura 5*.

Figura 5. Módulo de CAC



Variables de los subprogramas del simulador

Antes de correr las simulaciones se deben tener en cuenta una serie de indicaciones que se encuentran como ayuda en el simulador del router, en el cual se muestra cómo son guardadas las variables de entrada y de salida, indicando el nombre del archivo con su respectiva variable y la distribución de almacenamiento de los datos en la matriz, además de algunas indicaciones importantes para que la simulación sea efectiva y no presente ningún tipo de error, estas indicaciones también se encuentran a continuación.

POISSON

Una de las variables de entrada que serán usadas por este subprograma, es la cantidad de paquetes que se van a simular, este dato se encuentra guardado en la variable *paqord*, la otra variable de entrada usada por este subprograma, es el número de usuarios con reserva, que es guardado en la variable *reserv* y es usado para calcular los tiempos de llegada de los paquetes; el número de paquetes es usado para determinar cuantas veces hay que calcular estos tiempos. Los datos de resultados que entrega este subprograma, se almacenan una matriz final que contiene los tiempos de llegada de cada paquete; dicha matriz es guardada en la variable *intime*, y esta formada por una columna y *reserv* filas.

MMPP

Una de las variables de entrada que serán usadas por este subprograma es el número de reservas, que es guardado en la variable *reserv*, la otra variable de entrada es el número de *ciclos*, donde un *ciclo* de simulación equivale a la suma del tiempo de actividad y el tiempo de inactividad, es decir, al aumentar el número de ciclos aumenta el número de paquetes que llegan al simulador, y por condiciones de autosimilitud se recomienda que el número de paquetes sea mayor al número de reservas que se tienen; dicha variable se encuentra guardada en la variable *tsim*; para el número de paquetes a simular, el programa se encarga de generar este dato aleatoriamente a partir de ciertas probabilidades intrínsecas del método, este valor es guardado variable *paqord*, que almacena los paquetes para cada número de reservas simulado respectivamente; la variable de salida correspondiente a los tiempos de llegada de los paquetes es guardada en la variable *intime*, y esta representada como una matriz de una columna y *reserv* filas.

CARGAR ARCHIVO

Verifique que el archivo que desea cargar se encuentre guardado en la carpeta de programas del simulador, y que este representado como una matriz de una columna por *reserv* filas; en cada fila, se debe guardar el tiempo de llegada correspondiente a cada paquete, es decir, al llevar a cabo la simulación, se asumirá un tiempo de llegada de paquete respectivo a su orden de llegada al router, y debe ser guardada en la variable *intime*; además se debe especificar el número de paquetes que se van a simular, este dato debe almacenarse en la variable *paqord* para cada número de reservas simulado respectivamente; así mismo, se debe especificar el número de usuarios con reserva en las opciones de tráfico del simulador, todo esto con el fin de que los demás subprogramas que hacen uso de estas variables, puedan reconocerlas y usarlas sin ninguna dificultad.

BW FIJO

Una de las variables de entrada que serán usadas es el número de reservas, que se encuentra guardado en la variable *reserv*; si desea modificar esta variable, debe dirigirse a las opciones de tráfico y modificar las condiciones iniciales, ya que el número de reservas usado en la etapa de tráfico es el mismo que se utiliza para este subprograma; también está como variable de entrada el ancho de banda máximo, que se encuentra guardado en la variable *bwmax*; este valor corresponde al ancho de banda del canal, que es distribuido equitativamente para todos los usuarios de la forma $bwmax/reserv$, y es el mismo para cada usuario; dichos valores conforman la variable de salida de este subprograma, que está representada como una matriz de una columna por *reserv* filas y es guardada en la variable *bwtab*.

BW ALEATORIO

Este subprograma genera anchos de banda aleatorios para los usuarios con reserva, teniendo en cuenta que la sumatoria de estos sea igual o menor que el ancho de banda máximo, este valor es guardado en la variable *bwmax*; la sumatoria de todos los anchos de banda obtenidos no debe superar el ancho de banda máximo, motivo por el cual este subprograma determina el número de usuarios con reserva, que es guardado en la variable *reserv*; la otra variable de salida de este subprograma contiene los valores de ancho de banda correspondientes a cada usuario y está representada como una matriz de una columna por *reserv* filas, y es guardada en la variable *bwtab*.

NEW CALL BOUNDING

Este subprograma aplica el esquema de Control de Admisión de Llamadas, utilizando en el modelo *New Call Bounding Scheme*. Al seleccionar la opción de *traspaso* varía la intensidad de tráfico de las llamadas de traspaso, de lo contrario se encuentra seleccionada la opción de variación de intensidad de tráfico de nuevas llamas. La variable de salida *admit*, es una matriz de 3 columnas y *reserv* (*numero de reservas*) filas, las columnas están distribuidas de la siguiente manera: en la primera columna se guarda los tiempos de la petición de la reserva de llamadas que se han admitido, la segunda identifica si es nueva llamada con el número uno y si es llamada de traspaso con el número 0 y la tercera columna guarda los tiempos de permanencia en cada estado.

CUTOFF PRIORITY

Este subprograma aplica el esquema de Control de Admisión de Llamadas, utilizando el modelo *Cutoff Priority Scheme*. Al seleccionar la opción de *traspaso* varía la intensidad de tráfico de las llamadas de traspaso, de lo contrario se

encuentra seleccionada la opción de variación de intensidad de tráfico de nuevas llamadas. Las variable de salida *admit*, es una matriz de 3 columnas y *reserv* (*numero de reservas*) filas, las columnas están distribuidas de la siguiente manera: en la primera columna se guarda los tiempos de la petición de la reserva de llamadas que se han admitido, el segundo identifica si es nueva llamada con el número uno y si es llamada de traspaso con el número 0 y la tercera columna guarda los tiempos de permanencia en cada estado.

INTSERV

Al seleccionar esta opción, usted decide tener en cuenta solo el proceso llevado por un clasificador con arquitectura del tipo *INTSERV* a la hora de obtener las graficas de resultados; como variable de entrada esta la tabla de reservas, guardada en la variable *hashtab*, la tabla de resolución de colisiones, guardada en la variable *coltab*, y la variable de número de paquetes a simular, guardada en la variable *paqord*; estas variables son tomadas de los subprogramas previos a este; como variable de salida se tiene una matriz que contiene datos de *flowid*, número de lecturas en tabla *hash*, número de lecturas en tabla de resolución de colisiones y quintupla simulada, organizados en cuatro columnas y *paqu* filas respectivamente, guardada en la variable *datatab*.

INTSERV6

Al seleccionar esta opción, usted decide tener en cuenta solo el proceso llevado por un clasificador con arquitectura del tipo *INTSERV6* a la hora de obtener las graficas de resultados; como variable de entrada esta la tabla de reservas, guardada en la variable *hashtab*, la tabla de resolución de colisiones, guardada en la variable *coltab*, y la variable de número de paquetes a simular, guardada en la variable *paqord*; estas variables son tomadas de los subprogramas previos a este; como variable de salida se tiene una matriz que contiene datos de *flowid*, número

de lecturas en tabla *hash*, número de lecturas en tabla de resolución de colisiones y quintupla simulada, organizados en cuatro columnas y *paqu* filas respectivamente, que es guardada en la variable *datatab*, para *INTSERV6* siempre va a haber una sola lectura de tabla *hash*, ya que se identifica directamente la posición de la reserva.

AMBOS (INTSERV- INTSERV6)

Al seleccionar esta opción, usted decide tener en cuenta tanto el proceso llevado por un clasificador con arquitectura del tipo *INTSERV* como *INTSERV6* a la hora de obtener las graficas de resultados, como variable de entrada esta la tabla de reservas, guardada en la variable *hashtab*, la tabla de resolución de colisiones, guardada en la variable *coltab*, y la variable de número de paquetes a simular, guardada en la variable *paqord*; estas variables son tomadas de los subprogramas previos a este; como variable de salida se tiene una matriz que contiene datos de *flowid*, número de lecturas en tabla *hash*, número de lecturas en tabla de resolución de colisiones y quintupla simulada, organizados en cuatro columnas y *paqu* filas respectivamente, que es guardada en la variable *datatab*, para *INTSERV6* siempre va a haber una sola lectura de tabla *hash*, ya que se identifica directamente la reserva.

Modelo PGPS

Una de las variables de entrada de este subprograma es el número de paquetes a simular, la cual esta guardada en la variable *paqord*, también esta la tabla de datos obtenida del subprograma *simulador*, guardada en la variable *supertab*; haciendo uso de estos datos, este subprograma se encarga de organizar y sacar los paquetes, teniendo en cuenta el ancho de banda otorgado a cada uno de ellos; la variable de salida esta representada como una matriz de dos columnas, que contienen datos de tiempos de salida para *INTSERV* y tiempos de salida para

INTSERV6 respectivamente para cada paquete, esta variable se guarda en la variable *tvact*.

Modelo PQ

Una de las variables de entrada de este subprograma es el número de paquetes a simular, la cual esta guardada en la variable *paqord*, también esta la tabla de datos obtenida del subprograma *simulador*, guardada en la variable *supertab*; haciendo uso de estos datos, este subprograma se encarga de organizar y sacar los paquetes, teniendo en cuenta la prioridad otorgada a cada uno de estos, es decir, mientras hallan paquetes en la cola de prioridad alta, estos van a ser procesados y van a dejar en espera los paquetes de las demás colas y así sucesivamente; motivo por el cual, la variable de salida esta representada como una matriz de cuatro columnas, tiempos de salida para INTSERV, tiempos de salida para INTSERV6 y prioridad de paquetes respectivamente para cada paquete, esta variable se guarda en la variable *tvact*.