

# DISEÑO DE ISSIM Y EVALUACIÓN DE ALGORITMOS DE ENRUTAMIENTO CON CALIDAD DE SERVICIO

Néstor Misael Peña\*, Hollman. A. Díaz\*\*.

## RESUMEN:

*ISSIM es un simulador por eventos discretos diseñado para evaluar algoritmos de enrutamiento con y sin capacidad de calidad de servicio en el contexto de la arquitectura actual de Internet, Mejor Esfuerzo, y la arquitectura de Servicios Integrados. ISSIM es una herramienta que permite construir las topologías de simulación en forma gráfica y los resultados son entregados en un formato compatible con una hoja de cálculo. Este nuevo simulador fue diseñado para poder comparar los tres algoritmos de enrutamiento, SPF (Shortest Path First) y QOSPF (Quality of Service Shortest Path First) precálculo y en demanda, bajo las mismas condiciones. Entre los resultados obtenidos se encuentran la determinación de la complejidad de las variantes con calidad de servicio del algoritmo de Dijkstra (precálculo y en demanda), el impacto en tiempo de procesamiento y en ancho de banda consumido.*

## ABSTRACT:

*ISSIM is a discrete event simulator designed to evaluate routing algorithms with and without routing quality of service capabilities in Best Effort and Integrated Services architectures. ISSIM is a tool that allows to build network topologies in a graphic way and results are delivered in a spreadsheet compatible format. A new simulator was designed because of necessity of comparing the three routing algorithms, SPF and QOSPF precalculus and calculus on demand, under the same conditions. Among the found results are the complexities of the Dijkstra algorithm and its variants with Quality of Service (precalculus and calculus on demand), the impacts in the process time and used bandwidth.*

## PALABRAS CLAVE

Calidad de Servicio, Servicios Integrados, Protocolos de enrutamiento, QOSPF, Simulación.

## 1 INTRODUCCIÓN

Debido al gigantesco crecimiento de Internet, las debilidades del protocolo IP con su servicio de mejor esfuerzo se están haciendo evidentes. La solución obvia para tratar con estos problemas es incrementar el ancho de banda para evitar la congestión en los enlaces. Sin embargo, no sólo ha aumentado la cantidad de tráfico transportado por la red, también han cam-

biado sus características. Ahora hay nuevas aplicaciones, muchas basadas en IP, cuyos requerimientos operacionales son muy diferentes a los de aplicaciones como transferencia de archivos y correo electrónico [1].

Existen varias propuestas para poder suplir la necesidad de calidad de servicio en las redes IP. Entre ellas se encuentran las arquitecturas de Servicios Integrados y Servicios Diferencia-

\* PhD. Profesor Asociado, Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes; [npena@uniandes.edu.co](mailto:npena@uniandes.edu.co)

\*\* Magíster en Ingeniería Electrónica y de Computadores, Universidad de los Andes, [hodiaz@uniandes.edu.co](mailto:hodiaz@uniandes.edu.co)

dos y MPLS [2], [3]. Aunque se cree que Servicios Diferenciados provee una mejor solución desde el punto de vista de escalabilidad, actualmente sólo Servicios Integrados ofrece una solución integral desde la reserva de recursos hasta el manejo de flujos en redes de área extendida [4].

Por esto se escogió la arquitectura de Servicios Integrados como el modelo en el simulador. Además, ya que OSPF (*Open Shortest Path First*, [5]) es uno de los algoritmos de enrutamiento intrasistema autónomo más usado, se tomó como base sobre la cual se han evaluado algunas de las extensiones propuestas en [6].

Actualmente se tiene acceso a varios simuladores, entre ellos se destaca MaRS por su capacidad en el enrutamiento [4]. Sin embargo, MaRS puede simular solamente el algoritmo de Bellman-Ford y SPF [7]. Esto deja un vacío para el análisis del enrutamiento con calidad de servicio, que de acuerdo a [2] tiene el potencial de ser la pieza perdida en el esfuerzo de garantizar verdadera calidad de servicio en redes IP.

Este vacío es llenado en parte por QRS [4], el cual es un trabajo basado en la modificación de MaRS. QRS puede evaluar un algoritmo de enrutamiento, basado en SPF, con capacidad de calidad de servicio usando el modelo de cálculo por demanda. Sin embargo, tanto [4] como [6] concuerdan en decir que no existe aún claridad acerca de cuál modelo es mejor: Precálculo que consume grandes recursos en intervalos separados de tiempo o cálculo por demanda cuya carga computacional es menor pero se hace más seguido.

Se requería una herramienta que permitiera evaluar los algoritmos de enrutamiento por precálculo y en demanda bajo las mismas condiciones, para que las medidas puedan ser comparadas. Para satisfacer este requerimiento se diseñó este nuevo simulador.

En la segunda parte del artículo se describe el diseño del simulador ISSIM y la arquitectura de Servicios Integrados usada como modelo. En la tercera parte se muestran los resultados obtenidos en la evaluación de los tres algoritmos de enrutamiento. Por último, se presenta una serie de conclusiones del trabajo realizado.

## 2 DISEÑO DE ISSIM

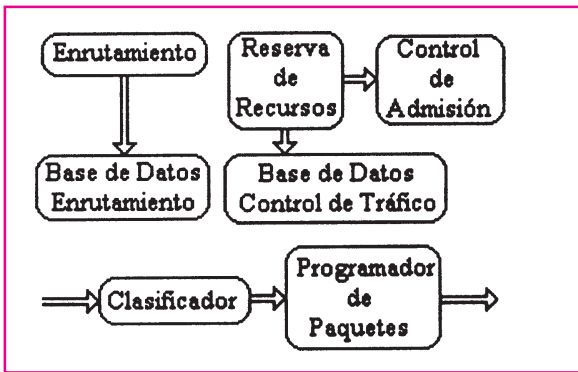
ISSIM es un simulador por eventos discretos enfocado principalmente a la parte de enrutamiento en redes IP de conmutación de paquetes. Soporta dos tipos de servicio: El servicio de Mejor Esfuerzo y el servicio de Carga Controlada [8] en el marco de la arquitectura de Servicios Integrados [9].

Con él se evaluaron, bajo las mismas condiciones, los algoritmos de enrutamiento de tipo SPF sin y con calidad de servicio en las variantes de precálculo y cálculo por demanda. Además, se buscó diseñar una herramienta amigable al usuario que permita desarrollar futuros trabajos en áreas como control de admisión, reserva de recursos, disciplinas de cola y descarte de paquetes.

### 2.1 ARQUITECTURA DE SERVICIOS INTEGRADOS

De acuerdo con [9], la arquitectura de Servicios Integrados es la disposición de cuatro componentes para que, trabajando en conjunto, pueda suministrarse servicios de Carga Controlada o Servicio Garantizado. Los tres primeros componentes: Programador de paquetes, clasificador y control de admisión constituyen una función del enrutador conocida como control de tráfico. El otro componente es la reserva de recursos. El enrutamiento no forma parte del modelo de la arquitectura y evaluar su desempeño en ella es el interés de este trabajo.

En la Figura 1 se muestra la relación existente entre los diferentes componentes dentro de un enrutador. Esta figura sirvió como base para la implementación de los nodos en el simulador.



**Figura 1.** Modelo de referencia de enrutadores en Servicios Integrados [9].

## 2.2 ESTRUCTURA DE ISSIM

ISSIM se compone básicamente de cinco grandes partes:

**Motor de simulación:** Es el encargado del manejo de los eventos y paquetes dentro de la red. Se usó como modelo base SIMLIB [10]. También obtiene las estadísticas de las variables.

**Interfaz gráfica:** Provee al usuario la facilidad de hacer y verificar la red diseñada y alterar los parámetros de ésta y de la simulación.

**Enrutadores:** Siguen el modelo de la Figura 1 y las recomendaciones de [11]. Para la reserva de recursos se usó una versión simplificada (sólo soporta reserva FF) del protocolo RSVP (*ReSerVation Protocol*) [12]. El programador de paquetes tiene tres disciplinas de cola: FIFO, por prioridades y CBQ (*Class Based Queueing*), como se describen en [2] y el descarte de paquetes se hace por fin de cola [2] y por algoritmo RED (*Random Early Detection*) [13]. Para el control de admisión se implementaron las opciones de aceptar todo, control por suma simple y por suma medida [2], [14]. El clasificador se basa en la dirección destino para seleccionar la interfaz de salida y en la prioridad del paquete para asignar la cola. En lo referente a enrutamiento se soportan tres protocolos, todos ellos basados en el algoritmo de Dijkstra: SPF, QOSPF por precálculo y en demanda.

**Fuentes generadoras de tráfico:** Generan el tráfico en la red y las solicitudes de reserva de recursos. Se aplicaron las recomendaciones de [2] y [15], entre otras, al implementar las fuen-

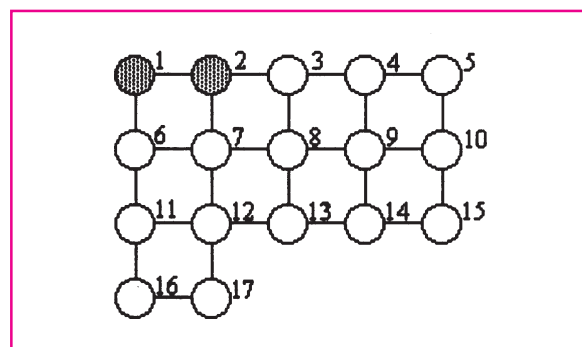
tes. Pueden operar como fuentes de tasa de salida constante o variable. y usan UDP como protocolo de transporte, es decir, son no orientadas a conexión.

**Generador de reportes:** Escribe en un archivo csv las estadísticas generadas durante la simulación. Incluye la media, la varianza, el número de las muestras, los valores máximos y mínimos, el reporte consolidado de varias iteraciones y los valores de los intervalos de confianza.

## 3 EVALUACIÓN DE LOS ALGORITMOS DE ENRUTAMIENTO

Antes de realizar los experimentos se hicieron seis grupos de simulaciones, cada uno de 30 iteraciones, para verificar el funcionamiento de ISSIM. La comparación se hizo contra Comnet III [16] y contra valores teóricos. Los resultados obtenidos arrojaron diferencias menores a 0.11% para redes con parámetros determinísticos. Para otras, con variables aleatorias, las diferencias fueron menores al 1.2%.

El primer experimento consistió en hallar el tiempo medio de procesamiento de los enrutadores y su dependencia respecto al número de ellos. Se usó, entre otras, la topología de la Figura 2.



**Figura 2.** Topología de red usada para hallar la complejidad de los algoritmos de enrutamiento. En los nodos sombreados se colocaron fuentes de tráfico.

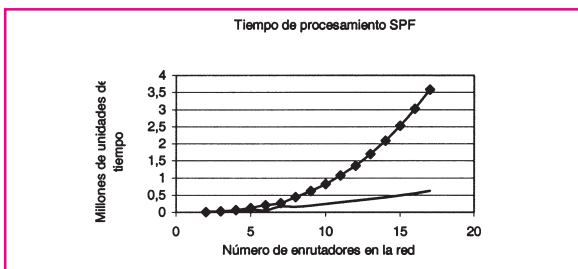
En ella se tienen dos fuentes de tráfico (no se muestran), sólo la que está conectada al enrutador 1 está activa, la otra sólo funciona

como sumidero. Todos los enlaces son de 2048kbps con 1ms de tiempo de propagación.

Dado que el tiempo de procesamiento con SPF y QOSPF depende de la cantidad de LSA (*Link State Advertisement*) procesados, en este experimento se procuró que para cada red este número permaneciera constante. Los LSA se generan de varias formas: periódicamente (en este caso cada 100ms), cuando cambia la topología de la red por caídas o subidas de enlaces o enrutadores y, para el caso de enrutamiento con calidad de servicio, cuando el ancho de banda disponible de un enlace cambia por encima de cierto porcentaje umbral. Para que sólo los LSA periódicos fueran generados, no se incluyeron eventos de caídas de enrutadores o enlaces y la fuente de tráfico generaba datos a una tasa de 20kbps, es decir, cerca del 1% del ancho de banda de los enlaces y el umbral de actualización se colocó en 20%.

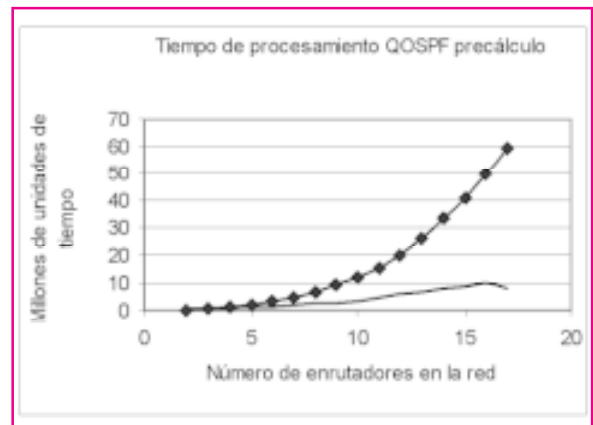
El experimento, con una duración de tiempo a simular de 10s, se repitió 30 veces para cada algoritmo de enrutamiento y luego el último enrutador era borrado hasta que sólo quedaron dos.

En la Figura 3 se observan como marcas sobre la curva los valores obtenidos para el tiempo medio de procesamiento en cada enrutador con SPF. La curva corresponde a una regresión cuadrática,  $R^2=0.9986$ . La línea de abajo corresponde al cociente entre la diferencia de los tiempos para  $n$  y  $n+1$  enrutadores y la diferencia entre el número de enrutadores, lo que es una aproximación a la derivada. Las unidades de tiempo (UT), que pueden ser ms o ns, dependen del procesador del enrutador.



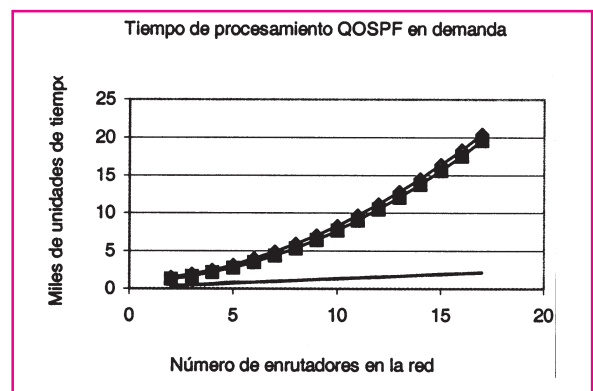
**Figura 3.** Curva de tiempo medio de procesamiento en la red con el algoritmo SPF y  $Dt/Dn$  contra el número de enrutadores en la red.

La Figura 4 muestra los resultados para el algoritmo QOSPF en precálculo.



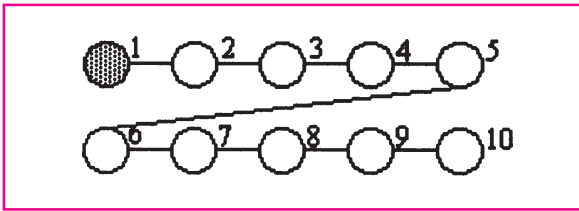
**Figura 4.** Curva de tiempo medio de procesamiento en la red con el algoritmo QOSPF en precálculo y  $Dt/Dn$  contra el número de enrutadores en la red.

Para el algoritmo QOSPF en demanda, el promedio del tiempo de procesamiento se obtuvo para los enrutadores que recibían solicitudes (enrutador 1 y enrutador 2) y los resultados se presentan en la Figura 5.



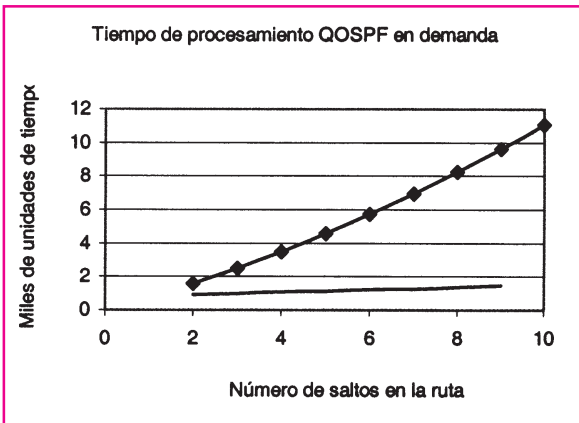
**Figura 5.** Curva de tiempo medio de procesamiento en los enrutadores 1 y 2 con el algoritmo QOSPF en demanda y  $Dt/Dn$  contra el número de enrutadores en la red.

Para determinar la relación entre el tiempo medio de procesamiento (QOSPF en demanda) en cada enrutador de la red y el número de saltos se empleó la topología de la Figura 6. En ella existe una fuente de tráfico en el enrutador 1 y su sumidero se ubicó en los enrutadores 2 al 10. Los parámetros de la fuente y de la simulación fueron los mismos que en el



**Figura 6.** Topología de red usada para hallar la relación entre tiempo de procesamiento y número de saltos. En el nodo sombreado hay una fuente de tráfico, su sumidero varió en posición.

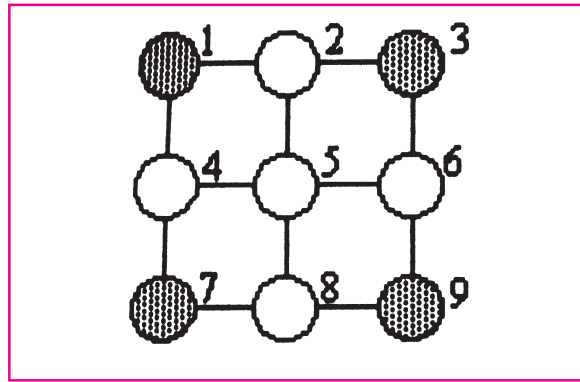
experimento anterior. Los resultados se presentan en la Figura 7.



**Figura 7.** Curva de tiempo medio de procesamiento en la red con el algoritmo QOSPF precálculo y Dt/Dn contra el número de saltos en la ruta.

Se observa que el tiempo de procesamiento medio en la red para el algoritmo QOSPF en demanda tiene una dependencia cuadrática respecto al número de saltos de la ruta. La regresión cuadrática da un valor de  $R^2=1$ .

Un tercer experimento fue desarrollado para determinar el impacto de los algoritmos de enrutamiento con calidad de servicio en el uso del ancho de banda, los tiempos de retraso y la carga de procesamiento. Para tal fin se usó la topología de la Figura 8. En ella los enrutadores sombreados tienen conectadas dos fuentes de tráfico que envían paquetes a las fuentes ubicadas en el vértice opuesto. Aquí todas las fuentes funcionan como origen y sumidero.



**Figura 8.** Topología de red usada para hallar el impacto de los algoritmos de enrutamiento en la red. En los nodos sombreados hay dos fuentes/sumideros que envían datos a las ubicadas en el vértice opuesto.

En este caso también se consideraron enlaces de 2048kbps con 1ms de tiempo de propagación. Las fuentes tienen una tasa de salida de 550kbps y solicitan un ancho de banda de 600kbps. La actualización de LSA se hacía cada 500ms o cada vez que el ancho de banda disponible cambiaba en 20%, éste se medía con un filtro EWMA (*Exponential Weighted Moving Average*) que muestreaba el enlace cada 20ms y tenía un peso de 0.2.

En la primera parte del experimento, las fuentes se encendían o apagaban de acuerdo a una distribución exponencial con media 1s. La simulación se hizo para un periodo de 10s y se repitió 30 veces. En la segunda parte, el tiempo medio de encendido y de apagado era de 100ms también siguiendo una distribución exponencial.

En las tablas 1 y 2 se puede ver la utilización media de todos los enlaces por parte del tráfico de control (enrutamiento y reserva) y el tráfico de los paquetes. Se aprecia que prácticamente la utilización por los paquetes de datos no depende de los tiempos de encendido/apagado de las fuentes. Lo que sí cambia es la utilización del tráfico de control que se dispara por las continuas generaciones de LSA en los algoritmos con calidad de servicio y por los mensajes RSVP.

**Tabla 1.** Utilización media de todos los enlaces de la red para un tiempo medio de encendido y apagado de 1s.

	Utilización Control	Utilización Carga	Total
SPF	0.007106	0.142910	0.15
QOSPF precálculo	0.070306	0.124109	0.194
QOSPF en demanda	0.069931	0.129437	0.199

**Tabla 2.** Utilización media de los todos enlaces de la red para un tiempo medio de encendido y apagado de 100ms.

	Utilización Control	Utilización Carga	Total
SPF	0.007111	0.158733	0.165
QOSPF precálculo	0.08599	0.119232	0.205
QOSPF en demanda	0.08474	0.120241	0.205

En las tablas 3 y 4 se encuentran la cantidad de LSA generados y el tiempo de procesamiento para cada algoritmo de enrutamiento y para cada tasa de encendido/apagado. Como se dijo antes, los dos algoritmos con calidad de servicio generan gran cantidad de LSA, es muy similar tanto para precálculo como en cálculo por demanda.

**Tabla 3.** Cantidad de LSA procesados y tiempo medio de procesamiento en la red para un tiempo medio de encendido y apagado de 1s.

	LSA procesados	Tiempo procesamiento Unidades de Tiempo
SPF	158.5	129587.6
QOSPF precálculo	993.9	11844298.7
QOSPF en demanda	988.1	19628.2

**Tabla 4.** Cantidad de LSA procesados y tiempo medio de procesamiento en la red para un tiempo medio de encendido y apagado de 100ms.

	LSA procesados	Tiempo procesamiento Unidades de Tiempo
SPF	158.6	129662.5
QOSPF precálculo	1192.5	13898027.2
QOSPF en demanda	1175.2	163586.3

Para el algoritmo sin calidad de servicio, los LSA procesados no varían pues ellos son gene-

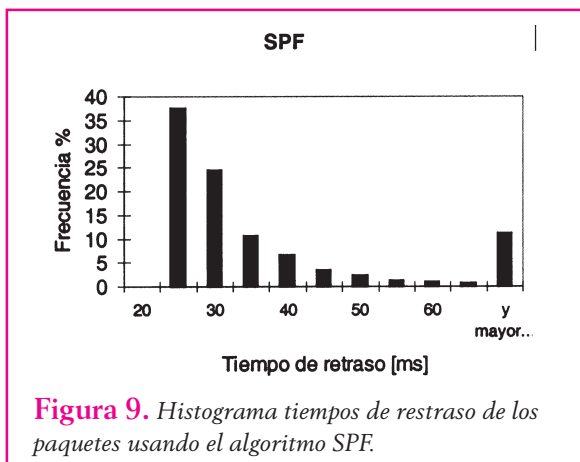
rados solamente por un mecanismo periódico. Para los otros algoritmos, la cantidad de LSA depende en parte del tiempo de encendido o apagado de la fuente, se incrementa con la cantidad de sesiones que se abren y se cierran. Pero esta cantidad varía más con el tráfico presente en la red como lo muestra la tabla 5 la cual se obtuvo apagando cuatro de las fuentes (una por cada enrutador de los vértices) y repitiendo el experimento con el que se halló la tabla 3.

**Tabla 5.** Cantidad de LSA procesados y tiempo medio de procesamiento en la red para un tiempo medio de encendido y apagado de 1s y sólo cuatro fuentes encendidas.

	LSA procesados	Tiempo procesamiento Unidades de Tiempo
SPF	158.9	129791.0
QOSPF precálculo	531.6	6421878.3
QOSPF en demanda	688.5	10317.6

Se aprecia que la generación de LSA en los algoritmos con calidad de servicio está ligada a los niveles de tráfico en la red y a la forma en que este tráfico se distribuye en el tiempo. El tiempo de procesamiento de SPF y QOSPF por precálculo no depende directamente de los tiempos de encendido y apagado de las fuentes sino de la cantidad de LSA a procesar la cual, sí se ve afectada por este parámetro. Para QOSPF en demanda el tiempo de procesamiento, tablas 3 y 4, crece más de ocho veces, la misma proporción en que crece la tasa de solicitudes de las fuentes que pasó de 0.476solicitudes/s a 3.94solicitudes/s.

En la Figuras 9 y 10 puede observarse el comportamiento del tiempo medio de retraso de los paquetes. Los dos algoritmos con calidad de servicio presentan menores tiempos medios de retraso y menor dispersión debido a que hacen una mejor distribución del tráfico sobre los enlaces. Este tiempo es menor a pesar de tener una utilización media más alta. SPF sólo hace uso de una ruta, sin importar cuán carga- da esté, en algunos casos la utilización de un enlace fue mayor al 60%.



tividad de ellos. Esto repercute en menores tiempos de retraso y en una menor variabilidad de estos, características deseables en aplicaciones de tiempo real. Esto aplica para los dos algoritmos QOSPF.

Dado que ambos algoritmos con calidad de servicio presentan características similares en lo que respecta a generación de tráfico de control, mejor utilización de los enlaces y menores tiempos de retraso promedio de los paquetes, el punto clave para su diferenciación está en sus tiempos medios de procesamiento.

Aunque en los dos, estos tiempos tienen dependencia cuadrática del número de enrutadores, la tasa de crecimiento de QOSPF por precálculo es mucho mayor que la del algoritmo en demanda. El tiempo de procesamiento del primero depende también de la cantidad de LSA procesados y la generación de LSA periódicos crece linealmente con el número de enrutadores en la red.

Por otro lado, ya que el tiempo de procesamiento de QOSPF en demanda crece linealmente con el número de peticiones de rutas, podría pensarse que para una alta tasa de éstas, el algoritmo por precálculo tendría menos tiempo de procesamiento. Pero debe tenerse en cuenta que a mayor cantidad de cambios en el ancho de banda disponible en la red, ocasionado por las fuentes de tráfico, mayor será la cantidad de LSA producidos y por tanto también aumentará el tiempo de proceso del algoritmo por precálculo.

En resumen, QOSPF en demanda se muestra como la mejor alternativa para la implementación de enrutamiento con calidad de servicio dado sus menores tiempos de procesamiento.

Durante el desarrollo de las simulaciones, se notó que la operación conjunta del algoritmo de reserva RSVP con cualquiera de los algoritmos de enrutamiento puede llegar a

crear bucles que se mantienen indefinidamente si la fuente de tráfico continúa realizando refrescos de mensajes *Path*. Este comportamiento se presenta con más frecuencia en los algoritmos con calidad de servicio debido a los frecuentes cambios en el dígrafo que representa la red debido a las variaciones en las utilidades de los enlaces. La especificación de RSVP [12] no trata este problema aunque sí consideran que puede presentarse. En ISSIM se adicionó la detección de bucles de mensajes RSVP y se proveyó una solución consistente en que el enrutador que encuentre un mensaje en bucle lo elimine y genere un mensaje de error que viaje por el ciclo y borre la reserva, con el propósito de que en el próximo refresco, los dígrafos de todos los enrutadores hayan convergido y se encuentre un ruta sin ciclos.

## 5 REFERENCIAS

- [1] Qos Forum. 1.999a. *The Need for QoS*. [www.qosforum.com](http://www.qosforum.com).
- [2] Huston, G. 2.000. *Internet Performance Survival Guide*. John Wiley & Sons.
- [3] Qos Forum. 1.999b. *Qos protocols and architectures*. [www.qosforum.com](http://www.qosforum.com).
- [4] Zhang, P., Kantola, R. Ma, Z. 2.000. *Design and Implementation of QRS*. [www.tct.hut.fi/~pgzhang/papers](http://www.tct.hut.fi/~pgzhang/papers)
- [5] Moy, J. 1.998. *OSPF Version 2, RFC 2328*. [www.ietf.org](http://www.ietf.org)
- [6] Apostolopoulos, G. Williams, D. 1.999. *QoS Routing Mechanisms and OSPF Extensions, RFC 2676*. [www.ietf.org](http://www.ietf.org)
- [7] Alaettinoglu, C., Shankar, A., Matta, I. 1.994. *Design and Implementation of MaRS: A Routing Testbed*. [www.isi.edu/~cengiz/publications/](http://www.isi.edu/~cengiz/publications/).
- [8] Wroclawski J. 1.997. *Specification of the Controlled-Load Network Element Service, RFC 2211*. [www.ietf.org](http://www.ietf.org)
- [9] Clark, D., Shenker S. 1.994. *Integrated Services in the Internet Architecture: an Overview, RFC 1633*. [www.ietf.org](http://www.ietf.org)



- [10] Law A., Kelton D. 2.000. *Simulation Modelling and Analysis*. McGraw Hill.
- [11] Shenker S., Wroclawski J. 1.997. *General Characterization Parameters for Integrated Service Network Elements*, RFC 2215. [www.ietf.org](http://www.ietf.org)
- [12] Braden R., Zhang L., Berson S., Herzog S., Jamin S. 1.997. *Resource ReSerVation Protocol (RSVP Version 1 Functional Specification*, RFC 2205. [www.ietf.org](http://www.ietf.org)
- [13] Floyd, S., Jacobson V. 1.993. *Random Early Detection Gateways for Congestion Avoidance*. IEEE/ACM Transactions on Networking. Volume: 1 Issue: 4, pg: 397 - 413
- [14] Jamin S., Shenker S., Danzig P. 1.997. *Comparison of Measurement Admission Control Algorithms for Controlled-Load Service*. IEEE/ACM Transactions on Networking. Volume: 5 Issue: 1, pg: 56 -70
- [15] Floyd, S., Paxson V. 1.995. *Wide-Area Traffic: The Failure of Poisson Modeling*. IEEE/ACM Transactions on Networking. Volume: 3 Issue: 3, pg: 226 - 244
- [16] CACI. 1.996. *COMNET III Reference Guide*.
- [17] Stallings, W. 2.000. *Comunicaciones y Redes de computadores*. Prentice Hall.