

**CALIDAD DE SERVICIO EN LA INTERNET:  
Protocolo de Reserva de Recursos - RSVP**

Ing. Daniel Díaz Ataucuri  
[ddiaz@dit.upm.es](mailto:ddiaz@dit.upm.es)

*Universidad Nacional Mayor de San Marcos  
Universidad Politécnica de Madrid (Pre-Doctorado)  
Instituto Nacional de Investigación y Capacitación (INICTEL)*

**Resumen :** Desde que el Profesor Leonard Kleinrock<sup>1</sup>, al inicio de la década de los 60, propone la técnica de conmutación por paquetes dando inicio de lo que más tarde sería la red de redes Internet, a la fecha nuevas aplicaciones han surgido, creando el desafío de cómo adaptar la Internet a estos nuevos cambios y necesidades. De allí, que en la actualidad, todos los esfuerzos de investigación están centrados en mejorar la calidad de su servicio (**QoS**, Quality of Service). Aquí surgen dos enfoques de solución. Una que plantea un nuevo modelo de Internet, donde los routers decidirán las reservas de recursos, Internet de Servicios Integrados **IntServ**, y otra que asigna a cada datagrama diferentes prioridades para que sean tratadas adecuadamente en los routers de la red, Internet de Servicios Diferenciados **DiffServ**. Es en la IntServ, donde se define el protocolo **RSVP** (Resource **ReSerVation Protocol**) para solucionar y reservar recursos en cada uno de los nodos de la red. En este artículo, se analiza el funcionamiento del protocolo RSVP, su estado actual, principales inconvenientes de su utilización en la Internet y las recientes investigaciones para solucionarlos.

**Abstract :** Since the early 1960s when the professor Leonard Kleinrock, introduced the new technique of packet switching thus creating what would eventually be the w.w.w., to date new applications have appeared; resulting in a new challenge: How may these new internet technologies be adaptable to those new changes and requirements? Accordingly, currently all the research efforts are focused on the goal of improving the internet's Quality of Service (**QoS**). One approach proposes a new model of internet where the routers will rule resource reservation, integrated services internet (**IntServ**), and another (**Approach**) by means of which different priorities may be assigned to each datagram so that such priorities may be suitably handled by network's routers, differentiated services internet (**DiffServ**). In intserv the RSVP (**Resource Reservation Protocol**) is defined with the goal of requesting and reserving resources in each node of the network. In this paper, the RSVP Protocol Performance is discussed; similar analysis holds for the present protocol state, major troubles regarding its use in internet and recent researchs oriented towards its solution.

**Palabras Claves :** Internet, Protocolo, Calidad de Servicio, Internet de Servicios Integrados, RSVP.

---

<sup>1</sup> El Dr. Leonard Kleinrock en la propuesta de su tesis doctoral "Information Flow in Large Communication Nets" del 31 de mayo de 1961, plantea la conmutación de paquetes como alternativa a la conmutación de circuitos. Tesis en <http://milennium.cs.ucla.edu/LK/Bip/REPORT/PhD/index.html>

## I. INTRODUCCION

La Internet actual, es una red que ofrece un servicio no confiable (la entrega no se garantiza), sin conexión (cada datagrama es tratado de manera independiente) y de entrega con el mejor esfuerzo o *best effort* (la red trata a todos los datos de igual manera y hace todo lo posible en entregar los datos, al menos que los recursos estén agotados o el enlace falle). Con la aparición de nuevas aplicaciones sobre la Internet como voz e imagen en tiempo real, trabajo compartido en directo, reporte de diagnósticos médicos en vivo, telefonía, comando y control de sistemas remotos, entre otros, la Internet actual se ve inadecuada y es necesario que ofrezca una mejor **Calidad de Servicio (QoS)**. El concepto de QoS está relacionado con la planificación de los elementos de la red de acuerdo con la demanda de las aplicaciones, para ofrecer un mínimo nivel de garantía que satisfaga los requerimientos de tráfico: mínimo retraso de envío de los datos, mínima variación de los retrasos, ancho de banda adecuado para el envío satisfactorio de los datos, entre otros. Dicho de otra manera, QoS se refiere a la manera en que se clasifican los paquetes de datos con el único propósito de que sean tratados adecuadamente respecto a otros.

Además del servicio *best effort* que la Internet actual ofrece, existen otros enfoques para mejorar la QoS. La primera, utilizando **Reserva de Recursos (Servicios Integrados, IntServ)**, donde las reservas de recursos son requeridas por las aplicaciones receptoras (*receiver*) de acuerdo a ciertas políticas, aquí se define el Protocolo de Reserva de Recursos **RSVP (Resource ReSerVation Protocol)** para suministrar la señalización requerida, ofreciendo dos tipos de servicios: *Servicios Garantizados* y *Servicios de Carga Controlada*. La segunda, utilizando **Prioridad (Servicios Diferenciados, DiffServ)** donde el tráfico que circula por la red es clasificado de acuerdo a niveles de prioridad. Para ello se suministran marcas dentro de las cabeceras de los datagramas IP para definir prioridades de tráfico. Decidir cual de los dos enfoques anteriores es el más adecuado para la Internet dependerá de muchos factores. Por ejemplo, una aplicación que requiere determinada QoS puede obtenerla incluso con el servicio convencional *best effort*. La disponibilidad de contar con la tecnología que soporte *reserva de recursos* o *prioridades* es otro aspecto a considerar. Finalmente, otra consideración relacionado con la QoS, es la seguridad y autenticación para los usuarios que soliciten de la red mejor QoS, ya que esta estará asociado con un mayor costo. Es decir, a mejor calidad, mayor costo.

## II. FACTORES RELACIONADOS EN LA CALIDAD DE SERVICIO

Las aplicaciones tradicionales, como correo electrónico, web, ftp, etc, son transportadas por la Internet actual con una clase de servicio muy pobre y plano denominado *best effort*. Con la aparición de nuevas aplicaciones de tiempo real es necesario mejorar la calidad de servicio. Una primera forma de enfocar la QoS es desde el punto de vista *cuantitativo*; saber como un usuario final de una aplicación percibe la red. Dicho de otra manera, por la satisfacción que siente un usuario al finalizar una sesión en la red. El otro enfoque es desde el punto de vista *cuantitativo*, de aquellos parámetros que están relacionados directamente con la mejora de la calidad de servicio y que al ser mejorados la red puede ofrecer una mejor calidad de servicio para aplicaciones que hoy en día son de gran importancia; videoconferencia, telefonía, control. Estos parámetros son: el ancho de banda, el retardo, la variación de retardo o *jitter* y la sensibilidad.

### 2.1 Ancho de banda

Es la capacidad de transmisión de una línea de comunicación expresada en bits por segundos (bps). Indica la capacidad máxima teórica de una conexión, pero debido a diversos factores como la presencia de varios flujos compartiendo uno o varios enlaces causa su disminución.

### 2.2 Retardo

Es el tiempo que un dato tarda en atravesar la red desde el transmisor al receptor. En aplicaciones en tiempo real, como voz, retardos muy elevados degradarán el servicio, haciendo prácticamente inservible. Otro

concepto relacionado con el retardo es la *latencia*, que se refiere a la cantidad de tiempo en que permanece un dato dentro de un router o conmutador.

### 2.3 Variación de retardo o jitter

Representa la distorsión de una señal cuando es propagado a través de la red, donde la señal varía desde su referencia original. Una variación de flujo o *jitter* de 2 milisegundos significa que los datos podrán llegar hasta 1 milisegundo antes o después respecto a su retardo promedio. Algunas aplicaciones de tiempo real pueden ser sensibles al retardo y variación de retardo.

### 2.4 Sensibilidad

Este parámetro indica la cantidad de paquetes perdidos de manera consecutiva que se puede tolerar

## III. SERVICIOS INTEGRADOS

En la actualidad, la tecnología permite disponer de estaciones de trabajo equipados con los últimos procesadores, técnicas de codificación para señales de audio y vídeo, programas de aplicación multimedia y transmisión multicast en la Internet. Aplicaciones en tiempo real es hoy en día uno de mayor demanda. A partir de 1994, la comunidad de Internet inicia los estudios para proponer una nueva arquitectura que modifique a la actual Internet y que satisfaga a las nuevas aplicaciones de tiempo real. En esta arquitectura, tanto los usuarios como los administradores de la red, deberán de disponer del ancho de banda de los enlaces para asignar adecuadamente a las diferentes aplicaciones tanto convencional como de tiempo real. Es aquí donde se define en [BRA94] el término **Servicios Integrados (IS)** para un modelo de servicio de la Internet que incluye tanto servicio de mejor esfuerzo o *best effort* y servicio de tiempo real. El modelo de servicio fundamental de la actual Internet, el servicio de envío con el *mejor esfuerzo*, ha ido cambiando desde sus inicios; ahora con los servicios integrados lo que se pretende es extender la arquitectura original, ya que proponer una nueva arquitectura sería una meta casi imposible. La arquitectura extendida está constituida por dos partes: el modelo de servicios entendidos (*extended service model*) llamado modelo IS y la referencia de la estructura de implantación (*reference implementation framework*). La razón principal de estas dos partes, es separar el comportamiento del servicio integrado de la manera como se implanta este servicio, ya que éste último puede ir cambiando según se establezca el modelo.

### 3.1 Modelo Servicios Integrados

El modelo de servicios integrados está diseñado para trabajar tanto en entornos unicast como multicast e incluye dos tipos de clases de tráfico: **carga garantizado** que soporta flujo de tráfico de tiempo real que requiere un límite cuantificable de retraso y **carga controlada** que es una aproximación a un servicio del *mejor esfuerzo* sobre una red no congestionada. Un aspecto básico de este modelo es que los recursos deben ser explícitamente gestionados a fin de satisfacer los requerimientos de las aplicaciones; de esta manera la *reserva de recursos* y el *control de admisión* son los elementos claves del servicio. Los servicios en tiempo real se sustentan en la garantía de los servicios, y la garantía muy difícilmente se consigue sin reservar recursos. Un aspecto que debe ser resaltado, es el de la prioridad. Asignar un nivel de prioridad alta a los tráfico de tiempo real podría ser adecuado. Pero se debe tener en cuenta que prioridad es un mecanismo de implantación, no es un modelo de servicio. El éxito de la reserva de recursos reside en que los router se encuentren disponibles para reservar recursos. Esto representa un cambio fundamental en el modelo de la Internet ya que en su inicio ésta fue establecida bajo el concepto de potenciar a los sistemas finales.

El modelo de servicio integrado propugna una red única tanto para aplicaciones de tiempo real como aquellas que no requieren tiempo real, descartando el uso de una red exclusiva para tráfico de tiempo real. Más aun, establece el uso del protocolo IP para los datos de tiempo real. El uso de reserva de recursos, permitirá acceder a los servicios de la Internet de manera privilegiada, siendo necesario imponer políticas y controles

administrativos, originando requerimientos de autenticación tanto de los usuarios, quien hace las reservas de recursos, como de los datagramas que usan los recursos reservados. Respecto al modelo de reserva, la arquitectura de servicios integrados establece la creación de un protocolo que reserve recursos, para crear y mantener el estado de reserva en los hosts extremos y en los routers a lo largo de la trayectoria del flujo. Una aplicación debe especificar la QoS deseada usando una lista de parámetros denominado **flowspec**. A la fecha se ha definido el protocolo **RSVP (Resource Reservation Protocol)** como parte integrante de la arquitectura de servicios integrados. El funcionamiento del protocolo RSVP se puede resumir indicando que cada una de las aplicaciones en el receptor envía un requerimiento de reserva de recursos a la red y ésta podría o bien aceptar o rechazar el pedido. En realidad el proceso de requerimiento de reserva es más complejo; ya que es válido tanto para entornos unicast como multicast y las aplicaciones pueden requerir diferentes niveles de QoS.

### 3.2 Implantación de Servicios Integrados

En la Internet actual, el protocolo IP (*Internet Protocol*) trata a todos los datagramas de manera igual, de tal manera que cada datagrama recibe la misma QoS y son típicamente enviados usando una cola FIFO (primero en entrar primero en salir). En general, ante la llegada de un datagrama, un router debe aceptar o rechazar. En caso de aceptar al datagrama, debe seleccionar una ruta adecuada por donde enviará este datagrama hacia su destino final. Adicionalmente un router debe reordenar el datagrama con respecto a otros datagramas que están esperando ser enviados. Para los sistemas integrados, los routers deben de asignar una adecuada QoS para cada flujo y su implantación requiere de tres componentes: planificador de paquetes (*packet scheduler*), clasificador de paquetes (*packet scheduler*) y control de admisión (*admission control*). En la figura 1 se ilustra estos componentes.

- **Planificador de Paquetes (*packet scheduler*).**- De una manera sencilla, un planificador reordena la cola de salida para la transmisión. Es decir, gestiona el envío de diferentes flujos de datagramas usando un grupo de colas y temporizadores.
- **Clasificador de paquetes.**- Cada datagrama debe ser relacionado con alguna clase de tal manera que todos los datagramas de la misma clase tendrán el mismo tratamiento por parte del planificador de paquetes. Esta relación es realizada por el *clasificador de paquetes*.
- **Control de Admisión.**- Ejecuta los algoritmos de decisión que un router (o un host) usa para determinar si a un nuevo flujo se puede otorgar un requerimiento de QoS sin alterar los requerimientos ya existentes.

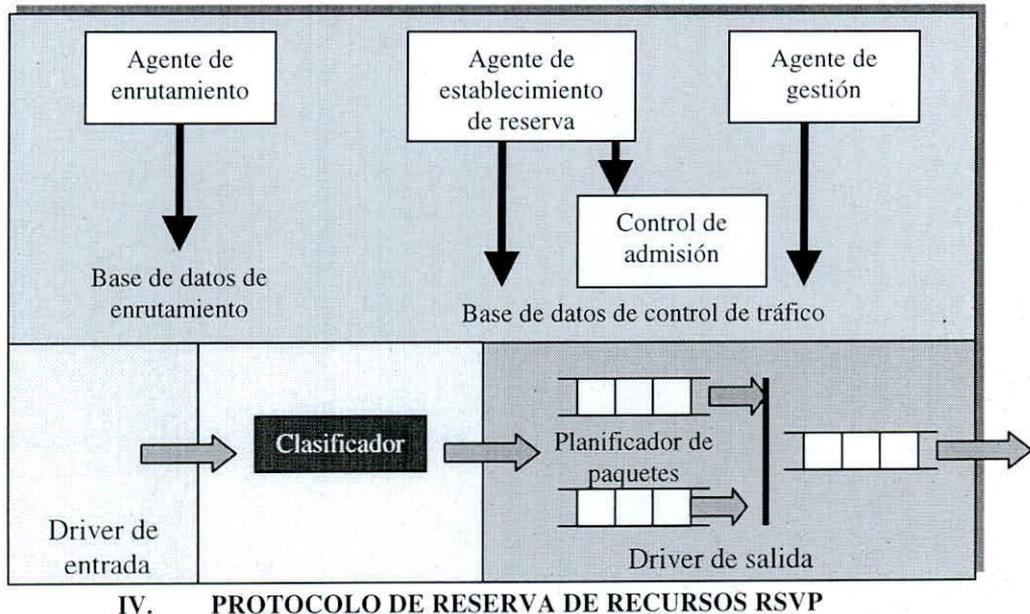


Figura 1. Implantación del Modelo de Referencia para routers

#### IV. PROTOCOLO DE RESERVA DE RECURSOS RSVP

El Protocolo de Reserva de Recurso, **RSVP** (*Resource ReSerVation Protocol*), fue desarrollado en el Instituto de Ciencias de la Información (*ISI, Information Science Institute*) de la Universidad del Sur de California (*USC, University of Souther California*) y en el centro de investigación de Xerox Palo Alto. RSVP permite que los hosts receptores reserven recursos de la red, tanto para aplicaciones unicast como multicast. RSVP opera con el protocolo IPv4 como con IPv6 y se encuentra situado encima de la capa de Internet, dentro de la estructura del protocolo TCP/IP, ocupando el lugar de los protocolos de transporte; pero RSVP no transporta datos ni realiza enrutamiento de los datagramas, solo reserva recursos de la red. Además, RSVP reserva recursos para flujo de datos simplex, es decir, reserva en una sola dirección.

##### 4.1 Visión del protocolo RSVP

El protocolo RSVP, definido en la RFC 2205 [BRA97], se encuentra implantado tanto en los hosts como en los routers, a lo largo de la trayectoria de un datagrama. El protocolo RSVP es usado por un host receptor para requerir calidad de servicio (QoS) específico de la red en nombre de una aplicación para un flujo de datos en particular, de allí que se dice que el protocolo RSVP usa requerimiento de reserva orientado a receptor. Los routers utilizan RSVP para enviar los requerimientos de calidad de servicio a todos los nodos a lo largo de la trayectoria del flujo y para establecer y mantener el estado que suministre el servicio requerido. RSVP lleva el requerimiento de reserva de recursos solicitado por cada host receptor a través de la red visitando cada nodo de la red, transportando **mensajes de requerimiento de reserva** o simplemente **mensajes de reserva** (*reservation messages*). Antes de que el protocolo RSVP envíe los *mensajes de reserva*, el host transmisor debe de enviar al host receptor **mensajes de trayectoria** (*path messages*) con la misma dirección IP de origen y destino que la aplicación misma. Los *mensajes de trayectoria* irán almacenando en cada router, durante su trayectoria al host receptor, la dirección IP del router anterior, de tal manera que cada uno de los routers que forman la trayectoria del flujo de datos contiene la dirección IP del router previo. De esta manera, el host receptor, conoce el camino por donde la aplicación llegará y los *mensajes de reserva* podrán ir de host a host, en el sentido inverso de la trayectoria del flujo de datos, realizando las reservas de recursos correspondiente. En la figura 2 se ilustra el uso de estos mensajes.

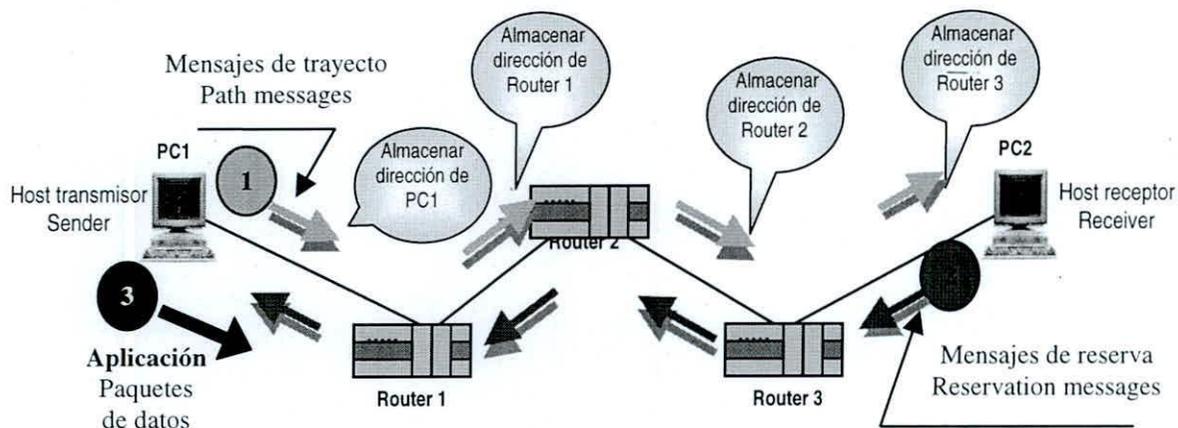


Figura 2. Mensajes de trayecto y de reserva

Por otro lado, es común definir que los *mensajes de trayectoria* son enviados flujo abajo o *downstream* y los *mensajes de reserva* flujo arriba o *upstream*.

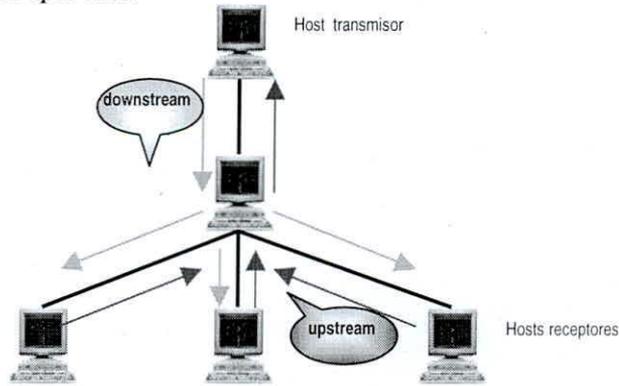


Figura 3. Flujos *downstream* y *upstream*

La calidad de servicio es implantada por un conjunto de mecanismos denominado *control de tráfico* (*Traffic Control*). Estos mecanismos son el **clasificador de paquetes** (*packet classifier*), **control de admisión** (*admission control*) y el **planificador de paquetes** (*packet scheduler*). La figura 4 ilustra el protocolo RSVP en los hosts y en los routers. Para realizar una reserva en un host o router, el proceso RSVP (llamado también *demonio RSVP*) se comunica con los módulos *control de admisión* y *control de política* (*policy control*). Control de admisión determina si el host o router tiene los suficientes recursos disponibles para suministrar la calidad de servicio requerida. Control de política determina si el usuario tiene los permisos administrativos para realizar la reserva. Si el nodo tiene los recursos disponibles para entregar la QoS adecuada y el usuario tiene los permisos para solicitar la reserva, el proceso RSVP establece los parámetros en un *clasificador de paquetes* y *planificador de paquetes* para obtener la QoS deseada. El *clasificador de paquetes* determina la clase de QoS para cada paquete y el *planificador de paquetes* ordena la transmisión de los paquetes para conseguir la QoS prometida para cada uno de los flujos. En caso de no satisfacer los requerimientos en el control de admisión o control de política, el proceso RSVP retorna una notificación de error a la aplicación que originó el requerimiento.

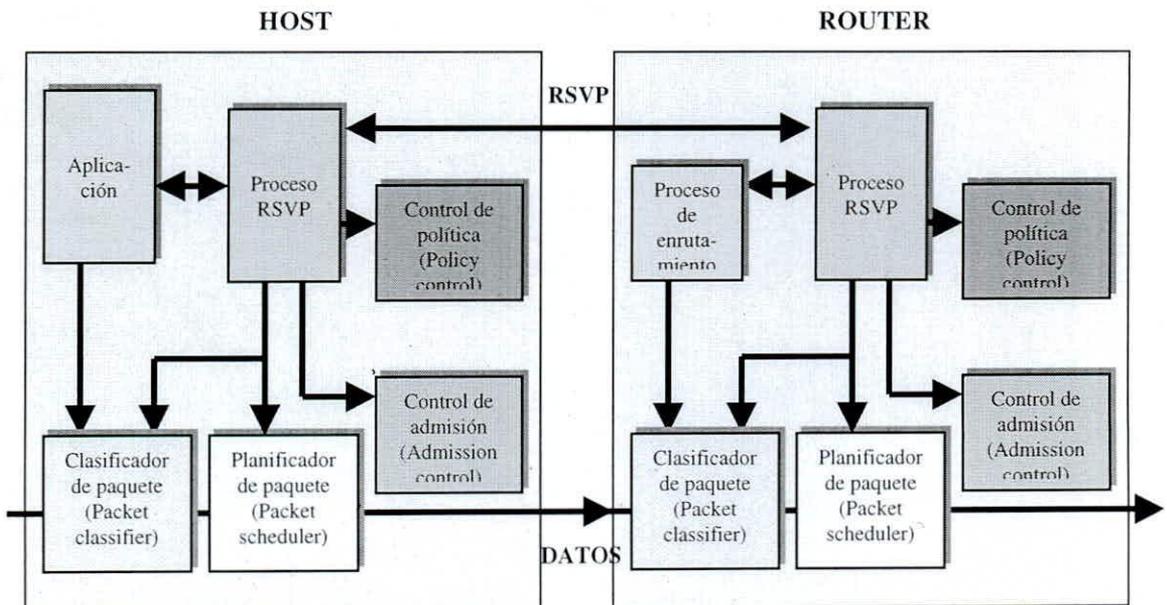


Figura 4. RSVP en hosts y routers

RSVP es escalable<sup>(1)</sup> a grandes grupos multicast debido a que es un protocolo que utiliza requerimiento de reserva orientado a receptor. En entornos multicasts, las reservas para un solo receptor no necesitan viajar hasta alcanzar el origen de la transmisión multicast, en su lugar solo basta que viaje hasta que alcance una reserva adecuada dentro de la estructura multicast.

#### 4.2 Soft state

Desde que la red Internet es una red dinámica, puede suceder que un router en la trayectoria del flujo de datos salga fuera de servicio, el protocolo RSVP deberá de conocer la nueva trayectoria para realizar la reserva de recursos. O puede suceder que un nuevo y más eficiente router en la red sea puesto en funcionamiento; en este caso también se establecerá una nueva ruta para la trayectoria de los flujos de datos a su destino. Aquí, el protocolo RSVP también deberá conocer este nuevo router y la nueva trayectoria para actualizar el requerimiento de una QoS adecuada. Para solucionar estos cambios en la red, RSVP establece el **soft state**.

Un *soft state* se refiere a un estado en los routers y los hosts extremos que pueden ser actualizados por ciertos mensajes RSVP. Ante la ausencia de mensajes de refresco, el estado es automáticamente borrado después de transcurrir un tiempo de haber sido establecido (*time out*). Dicho de otra manera, RSVP utiliza el *soft state* para gestionar el estado de reserva en los routers y hosts y es creado y periódicamente refrescado por los *mensajes de trayectoria y de reserva*.

El estado en cada nodo de la red puede ser borrado si no llega ningún mensaje de refresco antes que expire el *tiempo establecido de borrado (cleanup timeout)* o cuando directamente llega a cada nodo un mensaje de borrado (*teardown message*). Cuando un router cambia, el siguiente *mensaje de trayectoria* iniciará el estado de trayectoria en la nueva ruta, y futuros *mensajes de reserva* establecerán reservas para esta nueva ruta, y los estados en los segmentos no usados de la ruta serán borrados automáticamente después de un tiempo adecuado (*time out*).

#### 4.3 Característica de RSVP

El protocolo RSVP presenta las siguientes características:

- RSVP está diseñado para funcionar tanto con el protocolo IPv4 como con IPv6.
- RSVP realiza reserva de recursos tanto para aplicaciones unicast como multicast, adaptándose dinámicamente a los cambios de usuarios en el grupo multicast (algunos usuarios dejan y otros se unen a la aplicación multicast), así como a los cambios de routers en la red (algunos routers serán desconectados de la red y otros con mejores características adicionados a la red).
- RSVP es un protocolo simplex, es decir solo se establece reserva de recursos en una sola dirección de los flujos de datos. Para una comunicación *full dúplex*, se deberá establecer reserva de recursos para cada dirección del flujo de datos. De esta manera un host puede ser transmisor (sender) y receptor (receiver) en una determinada comunicación.
- RSVP es orientado a receptor, es decir son los receptores los que establecen y mantienen las reservas de los recursos de la red para una QoS deseada.
- RSVP introduce el concepto de *soft state* en los routers y hosts para soportar cambios dinámicos y adaptabilidad automática a los cambios de enrutamiento.
- RSVP no es un protocolo de transporte de datos ni de enrutamiento.
- RSVP suministra varios estilos o modelos de reserva para una variedad de aplicaciones.
- RSVP es transparente para routers que no soportan RSVP.

---

(1) RSVP presenta otro problema de escalabilidad relacionado con la cantidad de mensajes generados para reservar recursos, sobrecargando la red. Esto hace difícil, por el momento, que se utilice en grandes redes como Internet.

#### 4.4 Estilos de reserva

En la actualidad se han definido 03 estilos de reserva: *Wildcard-Filter (WF)*, *Fixed-Filter (FF)* y *Shared-Explicit (SE)*.

##### 4.4.1 Estilo Wildcard-Filter, WF

El estilo WF reserva recursos para un tipo general de flujo, sin especificar los flujos precisamente. Por ejemplo, se puede desear que todo flujo que va hacia el puerto 1234 utilice el recurso reservado Q. Una reserva de estilo WF se puede representar como:  $WF (* \{Q\})$ . Donde el carácter \* representa a todos los flujos y Q representa al *flowspec*(lista de parámetros).

##### 4.4.2 Estilo Fixed-Filter, FF

El estilo FF reserva recurso para un flujo en particular y no los comparte con ningún otro flujo de la misma sesión. Por ejemplo, una aplicación de videoconferencia que requiere un mínimo de ancho de banda deberá utilizar una reserva FF. Simbólicamente, reservas del estilo FF se pueden representar como:  $FF ( S \{Q\} )$

Donde S es el transmisor seleccionado y Q es su correspondiente *flowspec*. RSVP permite múltiples reservas de estilo FF en un mismo tiempo, usando una lista de flujos con su correspondiente *flowspec*, siguiendo la notación:  $FF ( S1 \{Q1\} , S2 \{Q2\} , \dots )$

##### 4.4.3 Estilo Shared-Explicit, SE

El estilo SE reserva recurso para varios flujos especificados. Es decir, reservas SE crean una simple reserva que es compartida por un grupo de flujos seleccionados. A diferencia de la reserva de estilo WF, el estilo SE permite a los receptores explícitamente especificar el grupo de flujos a ser incluidos en la reserva de recursos. Por ejemplo, 05 personas pueden participar en una audio conferencia, pero sólo una llamada a la vez puede producirse. En lugar de realizar 5 reservas de 64Kbytes cada uno, una simple reserva de 64Kbytes puede ser hecha para ser compartida por los 5 usuarios. Se puede representar un requerimiento de reserva estilo SE indicando una lista de los flujos (S1, S2, ..... ) que soportarán una calidad de servicio especificado por *flowspec* en Q, de la manera siguiente:  $SE ( ( S1 , S2 , \dots ) \{Q\} )$

#### 4.5 Mensajes generados en el Protocolo RSVP

Para el funcionamiento del protocolo RSVP, existen un conjunto de tipos de mensajes definidos: mensaje de requerimiento de reserva (*Resv*), mensaje de trayectoria (*Path*), mensaje de borrado (*PathTear* y *ResvTear*), mensaje de error (*PathErr* y *ResvErr*) y mensaje de confirmación de reserva (*ResvConf*). De estos, los dos primeros son indispensables para el funcionamiento del protocolo RSVP. En la figura 5 se ilustra la dirección de transmisión de cada uno de ellos. Actualmente, se está definiendo un grupo de mensajes para solucionar algunos problemas que presenta RSVP al momento de ser aplicado a redes como Internet.

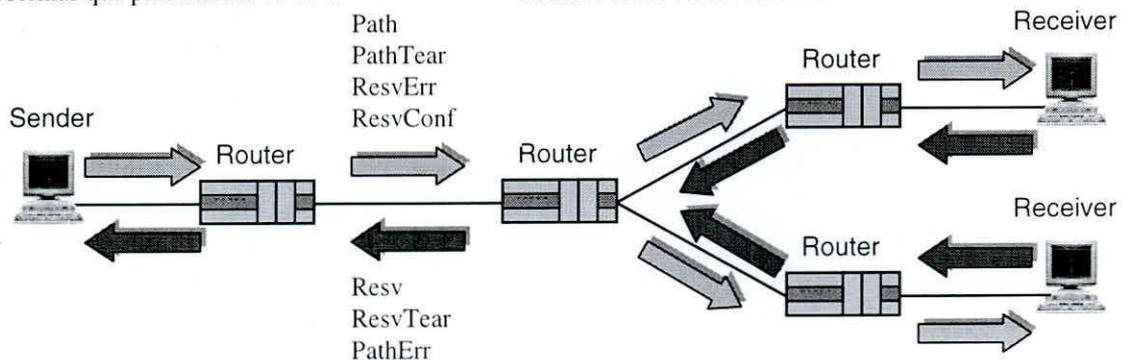


Figura 5. Dirección de los mensajes RSVP

#### 4.5.1. Mensajes de requerimiento de reserva (*Resv*) y de trayectoria (*Path*)

Cada uno de los hosts receptores envía mensajes de requerimiento de reserva (***Resv***) flujo arriba o *upstream* hacia los transmisores. Los mensajes *Resv* deben de seguir exactamente la trayectoria inversa que seguirán los flujos de datos. Estos mensajes *Resv* crean y mantienen el **estado de reserva** (*reservation state*) en cada uno de los nodos que define la trayectoria del flujo de datos y deben de ser enviados a todos los host transmisores.

Al inicio, cada uno de los hosts transmisores enviará mensajes de trayectoria (***Path***) flujo abajo o *downstream* hacia los hosts receptores a lo largo de rutas unicast o multicast definido por los protocolos de enrutamiento, atravesando routers y siguiendo la trayectoria que usará el flujo de datos. Los mensajes *Path* almacenarán el **estado de trayectoria** (*path state*) en cada uno de los nodos de la trayectoria que seguirá el flujo de datos. El **estado de trayectoria** debe de incluir por lo menos la dirección unicast IP del nodo del salto previo (*previous hop*), el cual será usado para enrutar mensajes *Resv* en la dirección inversa. Es de esta manera como los receptores conocerán la trayectoria futura del flujo de datos necesario para realizar las reservas de recursos, a través del envío de los mensajes *Resv*.

#### 4.5.2. Mensajes de borrado (*teardown message*)

Los mensajes de borrado (***teardown message***) eliminan los *estado de reserva* y *estado de trayectoria*, en el nodo donde es recibido, de manera inmediata. Existen dos tipos de mensajes de borrado: ***PathTear*** y ***ResvTear***. Los mensajes *PathTear* viajan hacia todos los receptores flujo abajo desde su punto de inicio y borra todos los *estados de trayectoria* así como los *estados de reserva* dependientes, a lo largo del trayecto. Los mensajes *ResvTear* borran los *estados de reserva* y viajan hacia todos los transmisores, flujo arriba desde su punto de inicio. Los mensajes de borrado pueden ser iniciados en el host transmisor o host receptor como en los routers. En los routers como consecuencia de haber transcurrido un tiempo establecido (*time out*) sin que lleguen los mensajes de refresco (*soft state*). Si un mensaje de requerimiento de borrado es perdido, no causará falla en el protocolo debido a que el estado no usado después de un tiempo estimado (*time out*) será borrado.

#### 4.5.3. Mensajes de error

Existen dos tipos de mensajes de error, ***PathErr*** y ***ResvErr***. Los mensajes *PathErr* son simples, ya que si un error es encontrado mientras se procesa un mensaje *Path* para crear un *estado de trayectoria* (*path state*) en algún nodo, un mensaje *PathErr* será enviado a los transmisores que originaron este mensaje *Path*. Los mensajes *Path* sólo dan un reporte a las aplicaciones en el transmisor del tipo de error sucedido y la dirección IP del nodo donde sucedió el error (especificado en el objeto `ERROR_SPEC`). Por otro lado, un mensaje *ResvErr* reporta errores en el procesamiento de un mensaje *Resv* o cuando existe una ruptura espontánea de una reserva y el manejo de los mensajes *ResvErr* es algo más complejo. Ya que el requerimiento de falla puede ser el resultado de la unión de un número de requerimientos, el mensaje *ResvErr* debe ser reportado a todos los receptores responsables. La unión de requerimientos diferentes puede crear un problema denominado "***killer reservation***", en el cual un requerimiento puede negar servicio a otro.

#### 4.5.4. Mensaje de Requerimiento de Confirmación, *ResvConf*

Mensajes ***ResvConf*** son enviados a los receptores como un indicador de que ha sido aceptado un requerimiento de reserva. Cuando un receptor *Ri* desea confirmar su requerimiento de reserva, debe de incluir en su mensaje *Resv* un objeto de requerimiento de confirmación (`RESV_CONFIRM`) conteniendo la dirección unicast IP del receptor *Ri*. Se deberá notar que la recepción de un mensaje *ResvConf* no garantiza que la reserva se ha realizado adecuadamente, solo indica una gran probabilidad que esto haya sucedido.

#### 4.6 Formato del protocolo RSVP

Como es conocido, un datagrama IPv6 dispone de una cabecera básica de 40 bytes, cabeceras opcionales de longitud variable y un campo de carga útil donde se aloja el protocolo de la capa superior. Es en este último lugar, dentro del datagrama IPv6, donde los mensajes RSVP son alojados para ser transportados por la red de redes. Además, tanto la cabecera básica IPv6 como las cabeceras opcionales, en el caso de estar presentes, disponen de un campo de 8 bits denominado *cabecera siguiente*, que indica el tipo de cabecera que sigue. Si este campo toma el valor de **46**, indica que la siguiente cabecera es en realidad un mensaje RSVP y está localizado en la parte final del datagrama IPv6. En la figura 6 se indica la ubicación de un mensaje RSVP dentro de un datagrama IPv6, asumiendo la presencia de cabeceras opcionales.

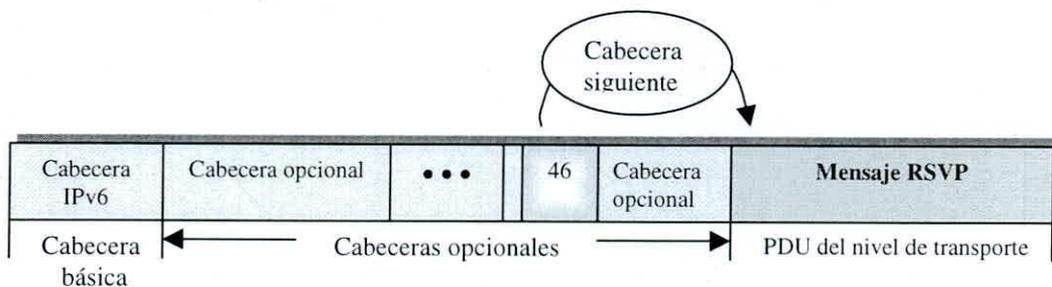


Figura 6. Ubicación de un mensaje RSVP

Un mensaje RSVP presenta una *cabecera RSVP* común, seguido por un conjunto de *objetos RSVP*, cada uno de longitud variable. En los objetos RSVP se especifica por ejemplo, la sesión tanto en IPv4 como IPv6, la dirección IP del router previo, frecuencia de refresco de los mensajes *Path* y *Resv*, estilo de reserva (WF, FF o SE), entre otros. En la figura 7 se ilustra con detalle el formato de un mensaje RSVP dentro de un datagrama IPv6, sin ninguna cabecera opcional.

##### 4.6.1 Cabecera RSVP

Cada mensaje RSVP presenta una cabecera común conformado por campos definidos y por definir o en reserva.

- El campo *Versión*, de 4 bits, indica la versión del protocolo RSVP; actualmente en 1.
- El campo *Indicador* o *flags*, de 4 bits, aun no está definido y está reservado para futuros usos<sup>[1]</sup>.
- El campo *Tipo de mensaje* o *Msg Type*, de 8 bits, identifica el tipo de mensaje RSVP que es enviado en el datagrama IPv6, estos pueden ser: mensaje de trayectoria *Path*, mensaje de requerimiento de reserva *Resv*, mensaje de error de trayectoria *PathErr*, mensaje de error de reserva *ResvErr*, mensaje de borrado de trayectoria *PathTear*, mensaje de borrado de reserva *ResvTear* y mensaje de confirmación de reserva *ResvConf*. Los valores que toma este campo son indicados en la Tabla 1.

[1] Al momento de escribir este artículo se está proponiendo algunos indicadores en este campo para solucionar el problema de la escalabilidad, de tal manera que RSVP sea posible en redes como la Internet.

Tabla 1 Campo *Tipo RSVP* (\*)

Campo <i>Tipo RSVP</i>	Tipo de mensaje
1	Path
2	Resv
3	Path Err
4	Path Tear
5	ResvTear
6	ResvConf
7	

(\*) Existen otros campos en plena definición

- El campo *Chequeo de suma RSVP* o *RSVP Checksum*, de 16 bits, representa el complemento a uno de la suma del mensaje, con el campo *Chequeo de suma RSVP* colocado inicialmente a cero para el propósito del cómputo de este campo. Si este campo es colocado a cero, significa que ningún chequeo de suma fue calculado y enviado.

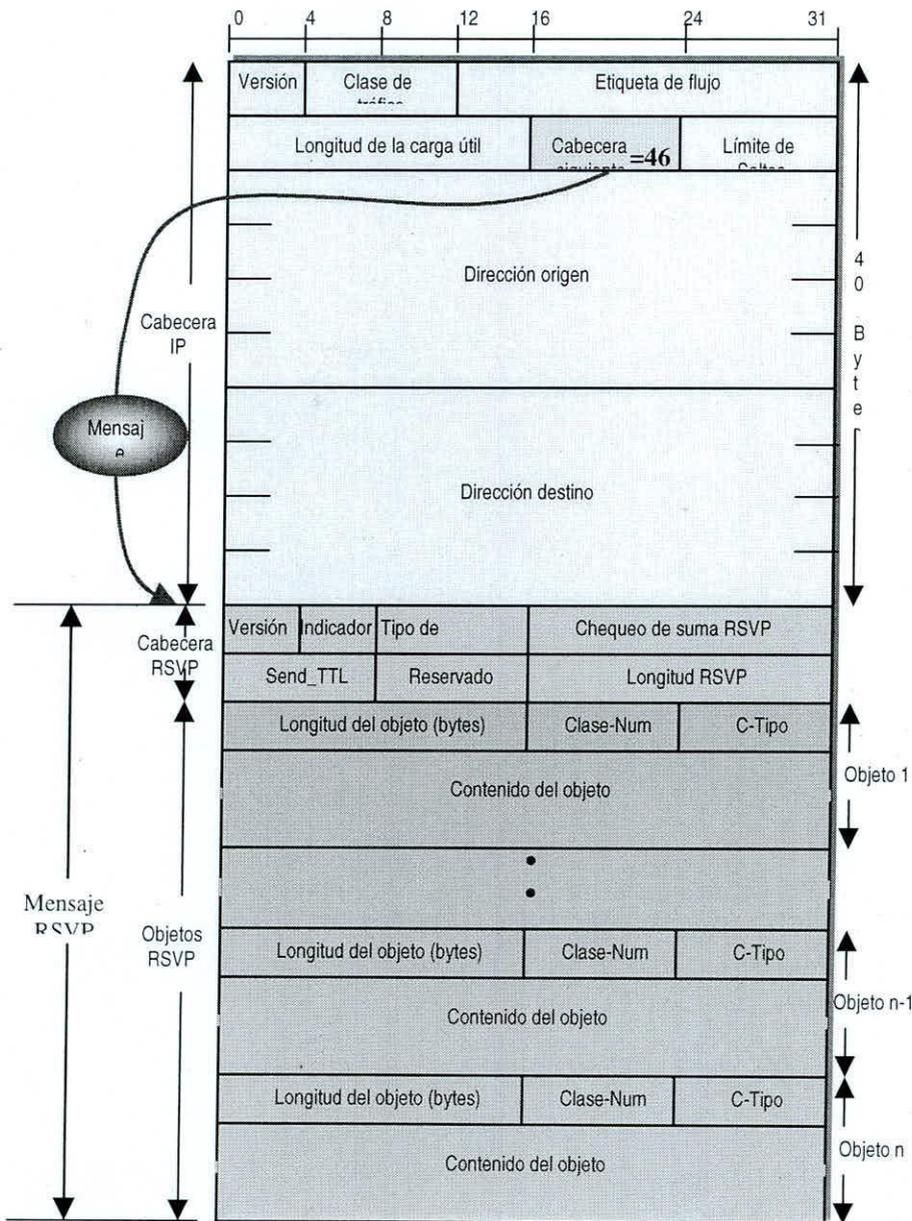


Figura 7. Formato de un mensaje RSVP

- El campo *Send\_TTL*, de 8 bits, es el valor IP TTL con el cual el mensaje fue enviado. El protocolo RSVP puede detectar saltos que no soportan RSVP comparando el valor IP TTL con el cual un mensaje *Path* fue enviado con el valor TTL recibido.
- Campo *Reservado*, de 8 bits, será utilizado para futuras aplicaciones.
- El campo *Longitud RSVP o RSVP Length*, de 16 bits, contiene la longitud total, expresada en bytes, del mensaje RSVP; es decir, incluyendo tanto la cabecera RSVP como los objetos de longitud variable que siguen a esta cabecera.

Los siguientes bytes del mensaje RSVP corresponden a los objetos, quienes llevarán los requerimientos necesarios para establecer la reserva de recursos deseada. Para cada uno de los tipos de mensajes definidos en RSVP, existe un conjunto de objetos asociados.

#### 4.6.2 Objetos RSVP

Cada uno de los objetos presenta una cabecera de 32 bits de longitud, con 3 campos definidos: *Longitud del objeto* o *Length*, *Número de clase* o *Class-Num* y *Tipo de clase* o *C-Type*, y una región de longitud variable que contiene el objeto en sí. El campo *Longitud del objeto* o *Length*, de 16 bits, indica la longitud total del objeto expresado en bytes y debe ser un múltiplo de 4. El mínimo valor asignado a este campo es 4 (sólo la cabecera del objeto) y el máximo valor es de 65532 bytes (máxima longitud del contenido del objeto de 65528 bytes). Los campos *Número de clase* o *Class-Num* y *Tipo de clase* o *C-Type*, ambos de 8 bits, identifican los diferentes tipos de objetos. Estos dos campos deben de ser usados en forma conjunta para identificar a cada uno de los objetos. Cada uno de los objetos tiene un nombre y actualmente se tiene definido los siguientes tipos de objetos: NULL, SESSION, RSVP\_HOP, INTEGRITY, TIME\_VALUES, ERROR\_SPEC, SCOPE, STYLE, FLOWSPEC, FILTER\_SPEC, SENDER\_TEMPLATE, SENDER\_TSPEC, ADSPEC, POLICY\_DATA y RSV\_CONFIRM.

El análisis de cada uno de estos objetos escapa al presente artículo. Como referencia se muestra sólo los objetos STYLE y TIME\_VALUES.

- **STYLE.**- El campo *Class-Num* estará siempre en 8 y el campo *C\_Type* en 1. En este objeto se define el estilo de reserva. En la figura 8 se ilustra este objeto. El campo *Indicadores*, de 8 bits, está aun por definir. El campo *vector de opción*, de 24 bits, define el estilo de reserva. Los valores de éste vector de detallan en la figura 9. El objeto STYLE es requerido por el mensaje *Resv*.

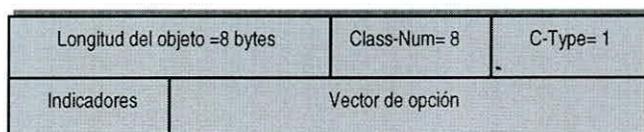


Figura 8 Objeto STYLE

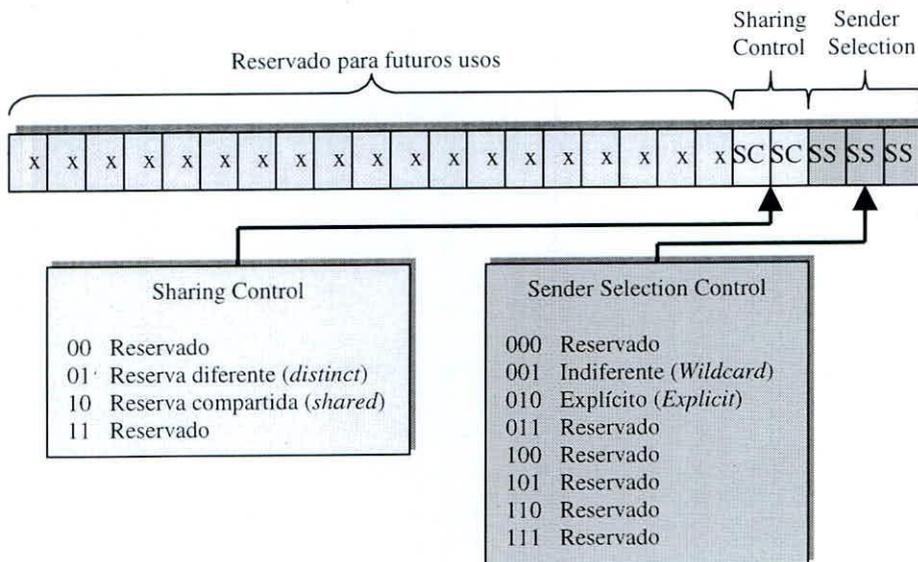


Figura 9. Vector de opción

- **TIME\_VALUES.**- El campo *Class-Num* estará siempre en 5 y el campo *C\_Type* en 1, como se ilustra en la figura 10. Contiene el valor para el periodo de refresco *R* usado por el que crea el mensaje *Path* o *Resv*. El valor del periodo de refresco está expresado en milisegundos.

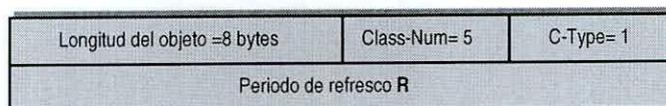


Figura 10 Objeto TIME\_VALUES

## V. OBSERVACIONES DEL PROTOCOLO RSVP

Aunque del protocolo RSVP es presentado para obtener una mejor reserva de recursos en la Internet, su uso en gran escala presenta algunas observaciones. Existen tres factores, indicados en la RFC 2208 [MAN97], que hacen que RSVP todavía se encuentre en fase de consolidación: escalabilidad, seguridad y políticas de control.

### 5.1 Escalabilidad

Por cada sesión (grupo de flujos de datos con un destino en particular) establecida en la red, en cada uno de los routers que soportan RSVP se deberán crear y mantener el estado de reserva (*reservation state*) y el estado de trayectoria (*path state*), a través del envío de los mensajes *Resv* y *Path* respectivamente. En una red, cuando más sesiones se establezcan, más mensajes se irán generando y transmitiendo; es decir, cuando más aplicaciones soliciten reservar recursos de la red más mensajes propios del protocolo RSVP se irán generando, sobrecargando la red. Más aun, en aplicaciones bidireccionales, como voz sobre IP (VoIP), se deberán reservar recursos para cada uno de los sentidos de envío de los datos, ya que RSVP sólo reserva recursos en un sólo sentido. Por ejemplo, si un Proveedor de Servicios de Internet (ISP) ofrece telefonía sobre IP y en un momento determinado

existen 1000 llamadas cursadas (2000 flujos) circulando a través de los routers, se deberá mantener el estado en cada uno de los routers para los 2000 flujos, a parte de los flujos correspondiente a los datos de voz.

### 5.2 Seguridad

El RSVP establece estados distribuidos a lo largo de los routers y hosts en una red, para reservar recursos e implementar la Internet de servicios integrados. Esto origina, que diferentes usuarios puedan acceder a la red de manera preferencial para solicitar reservas de recursos. Un requisito que se impone en RSVP es que cualquier tipo de modificación en la solicitud de reserva deba ser evitada y de suceder, detectada inmediatamente. Esto es, la reserva de recursos en la red no sólo dependerá de los recursos disponibles de la red, sino también del cumplimiento satisfactorio de todos los mecanismos de protección en la transmisión de los mensajes RSVP entre nodos vecinos; evitando de esta manera el robo de servicios en la red. Aunque el requisito de seguridad no es exclusivo de RSVP, aquí se hace muy necesario ya que la calidad solicitada por los diferentes usuarios estará relacionado a un costo. Una solución propuesta a este problema es la definición del objeto INTEGRITY<sup>[1]</sup> que serán transportados por los mensajes RSVP. Este objeto es utilizado para encriptar los datos que sirven de autenticación el nodo origen y para verificar el contenido de los mensajes RSVP.

### 5.3 Políticas de Control

Una vez establecido el protocolo de reserva de recursos para establecer diferentes servicios, el problema de política de control está relacionado en quien puede o no realizar las reservas. A la fecha, la especificación del protocolo RSVP define mecanismos para el transporte de información de políticas a lo largo de las reservas. Pero estas especificaciones no definen las políticas en sí. Para abordar este tema, se ha definido el objeto POLICY\_DATA para decidir si una reserva es administrativamente permitida. Lo que se trata es de evitar que cada empresa tenga que definir y usar sus propios mecanismos de políticas.

## VI. ASPECTOS RECIENTES DEL PROTOCOLO RSVP

El problema de escalabilidad de RSVP en los routers es hoy en día un aspecto muy crítico ya que RSVP establece un estado por cada flujo en cada uno de los routers de una red. Obviamente, el problema es aun más crítico en el centro de una gran red, como es el caso de la Internet, donde la densidad de flujo es muy alta. Actualmente, se propone agrupar los mensajes RSVP en dos tipos: **mensajes Trigger** o **de disparo** (trigger messages) y **mensajes Refresh** o **de refresco** (refresh messages). Los mensajes Trigger son generados por un host o router debido a un cambio de estado; es decir establecen estados o llevan cualquier tipo de información que no han sido transmitidas previamente. Estos mensajes Trigger incluyen mensajes de aviso de nuevos estados, un cambio de ruta que alteró la trayectoria de reserva o una modificación de reserva por un router flujo abajo. Los *mensajes Refresh* actualizan estados previamente establecidos y contienen casi los mismos objetos y la misma información como el mensaje que fue previamente transmitido, la excepción está relacionado con la seguridad e indicadores que permiten diferenciar los mensajes *Refresh*. Por ejemplo, si el *demonio RSVP* de un router cae y re-inicia, este router perderá toda la información de estado RSVP. Sin embargo, desde que sus routers vecinos envían copias de la información de los estados periódicamente, el router puede recuperar los estados perdidos dentro de un intervalo de refresco. Sólo los mensajes *Path* y *Resv* pueden formar los mensajes *Refresh*.

En [BER99-2] se proponen algunas soluciones al problema de la sobrecarga y confiabilidad de la red, que presenta el Protocolo RSVP en la generación de mensajes de refrescos. En primer lugar, es necesario agrupar los diferentes mensajes en uno nuevo denominado *mensaje de paquetes RSVP (RSVP Bundle Message)* para reducir los mensajes de refrescos. Tres nuevos objetos son definidos: `MESSAGES_` para reducir el

[1] En <http://www.ietf.org/internet-drafts/draft-ietf-rsvp-md5-08.txt> se encuentra los primeros aspectos de seguridad.

procesamiento de los mensajes de refrescos y permitir a los receptores una fácil identificación de un mensaje que no cambia, MESSAGE\_ID ACK para soportar acuse de recibo y confiabilidad en el envío de mensajes RSVP y MESSAGE\_ID NACK para detectar los mensajes perdidos. Además, se define un mensaje *resumen de refresco* para habilitar el estado de refresco sin transmitir el mensaje de refresco íntegramente, mientras se mantenga la disponibilidad de RSVP de indicar cuando un estado es perdido o cuando los siguientes saltos cambian. A la fecha, todas estas soluciones están en pleno análisis y se espera que con las pruebas que se realicen en los laboratorios, se consoliden este nuevo enfoque para mejorar la escalabilidad del protocolo RSVP.

Otro aspecto a solucionar es el uso del protocolo RSVP con IPv6 haciendo uso del concepto de etiqueta de flujo. En IPv6, se define [DEE98] un campo de 20 bits denominado *etiqueta de flujo* que permite asignar a cada datagrama una etiqueta para que los diferentes routers de la red puedan tratarlos adecuadamente. El protocolo RSVP puede utilizar esta etiqueta de flujo para ofrecer mayor calidad de servicio bajo redes IPv6. A la fecha existe una incongruencia entre el campo definido en el protocolo IPv6 (20 bits) y el campo definido en uno de los objetos que maneja la etiqueta de flujo (de 24 bits). En el draft [BER99-1] se plantea una solución.

## VII. CONCLUSIONES

La Internet actual es una red de "mejor esfuerzo" (*best effort*) y la presencia de nuevas aplicaciones en tiempo real como videoconferencia y telefonía en IP está obligando a la Internet actual que ofrezca un servicio más óptimo que el *best effort*. La IETF tiene como meta mejorar la arquitectura de servicios de la Internet, desde una red de *best effort* a una red de *servicios integrados (IntServ)*. En esta última arquitectura, las aplicaciones deberán ser capaces de requerir diferentes clases de servicios y reservar recursos tanto en los routers como en los hosts, utilizando reserva de recursos entre los extremos. En la *IntServ*, se ha definido dos clases de servicios: *Servicios Garantizados (GS, Guaranteed Service)* y *Servicios de Carga Controlada (CLS, Controlled Load Service)*.

El protocolo encargado de transmitir los requerimientos de reserva desde las aplicaciones hacia los routers, es RSVP. En RSVP se define un conjunto de mensajes que transportan objetos donde se especifica tanto la calidad de servicio que una aplicación en el transmisor desea esperar, como los requerimientos solicitados por las aplicaciones en los receptores. El proceso de reserva de recursos se inicia cuando una aplicación en el transmisor envía un mensaje *Path* a los receptores, llevando información del tráfico que el transmisor desea e información de los diferentes routers. Cuando los mensajes *Path* llegan al receptor, la aplicación en los receptores los interpreta y genera los mensajes *Resv* que son enviados hacia los transmisores y en cada uno de los routers, en el trayecto hacia el destino, reservará los recursos con la QoS deseada. Todo el proceso de reserva termina cuando el mensaje *Resv* llega a las aplicaciones en los transmisores.

Los problemas de escalabilidad, seguridad y control de políticas que presenta RSVP están siendo abordadas en la actualidad<sup>[1]</sup>. Estos inconvenientes, y otros que puedan surgir, son más manejables en redes Intranet o en limitados Proveedores de Servicio de Internet (ISP) con aplicaciones de tiempo real. El siguiente salto será, con todos estos problemas resueltos, el uso de RSVP en la Internet.

Se espera, en un futuro cercano, con la puesta en funcionamiento del protocolo RSVP, se logre mejorar la calidad de servicio de la Internet tanto para entornos unicast como multicast. Permitiendo que aplicaciones, como videoconferencias y voz sobre IP, sean transmitidas con la calidad necesaria, haciendo posible el uso masivo de la Internet en áreas como la medicina (tele-medicina) y la educación (tele-educación).

[1] Las nuevas propuestas en plena discusión sobre el protocolo RSVP, así como de otros temas telemáticos, se encuentran disponibles en draft. Estos pueden ser obtenidos en <http://www.ietf.org/internet-drafts/>

Las propuestas ya aceptadas por consenso, se encuentran disponibles en RFC (Request For Comment). Los RFC pueden ser obtenidos en <ftp://ftp.dit.upm.es/mirrors/ftp.ripe.net/rfc/> o en <ftp://ftp.isi.edu/in-notes/>.

### VIII. REFERENCIAS BIBLIOGRAFICAS

- [BER99-1] S. Berson, "RSVP and Integrated Services with IPv6 Flow Labels", Internet Draft, June 1999, draft-berson-rsvp-ipv6-fl-00.txt
- [BER99-2] L. Berger, G. D. Gan, G. Swallow, P. Pan, "RSVP Refresh Reduction Extensions", Internet Draft, October 1999, draft-ietf-rsvp-refresh-reduct-01.txt
- [BER97] L. Berger, T. O'Malley, "RSVP Extensions for IPSEC Data Flows", IETF Standards Track RFC 2207, September 1997.
- [BRA94] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", IETF Informational RFC 1633, June 1994.
- [BRA97] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", IETF Standards Track RFC 2205, September 1997.
- [DEE98] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", IETF Standards Track RFC 2460, December 1998.
- [DES99] E. Desmet, G. Gastaud, G. Petit, "Calidad de Servicio en la Internet", Revista de telecomunicaciones de Alcatel, 2do trimestre de 1999.
- [MAN97] A. Mankin, F. Baker, S. Brander, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang, "Resource Reservation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment", IETF Informational RFC 2208, September 1997.
- [MET99-1] C. Metz, "IP QoS: Traveling in First Class on the Internet", IEEE Internet Computing, March-April 1999.
- [MET99-2] C. Metz, "RSVP: General-Purpose Signaling for IP", IEEE Internet Computing, May-Jun 1999.
- [SHE97-1] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", IETF Standards Track RFC 2212, September 1997.
- [SHE97-2] S. Shenker, J. Wroclawski, "General Characterisation Parameters for Integrated Service Network Elements" IETF Standards Track RFC 2215, September 1997.
- [SHE97-3] S. Shenker, J. Wroclawski "Network Element Service Specification Template", IETF Informational RFC 2216, September 1997.
- [WAN99] L. Wang, A. Terzis, L. Zhang, "A New Proposal for RSVP Refreshes", 7th International Conference on Network Protocols (ICNP'99), October 1999. <http://irl.cs.ucla.edu/papers/ICNP99.ps>
- [WHI97] P. White, "RSVP and Integrated Services in the Internet: A Tutorial", IEEE Communications Magazine, May 1997.
- [WRO97-1] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", IETF Standards Track RFC 2210, September 1997.
- [WRO97-2] J. Wroclawski, "Specification of the Controlled-Load Network Element Service", IETF Standards Track RFC 2211, September 1997.