

---

# Calidad de servicio en redes interconectadas

---

PID\_00268540

Enric López i Rocafiguera  
Pere Barberán Agut

---

Tiempo mínimo de dedicación recomendado: 3 horas

---



**Pere Barberán**

Ingeniero de Telecomunicaciones por la Universidad Politécnica de Cataluña. Profesor de la Escuela Universitaria Politécnica de Mataró donde forma parte del Área de Redes y Servicios. De 2005 a 2010 ha sido director del Departamento de Telecomunicaciones y Arquitectura de Computadores. Actualmente responsable del laboratorio de *networking* TCM NetLab en la Fundación Tecnocampus Mataró-Maresme.

**Enric López Rocafiguera**

Ingeniero de Telecomunicaciones, en la especialidad de Comunicaciones por la Escuela Técnica Superior de Ingenieros de Telecomunicación de Barcelona, de la Universidad Politécnica de Cataluña. Profesor de Redes de comunicaciones y Redes de computadores, en las carreras de Ingeniería Técnica de Telecomunicaciones, Ingeniería Técnica Industrial e Ingeniería Técnica Informática en la Escuela Politécnica Superior (EPS) de la Universidad de Vic (UVic). Miembro del grupo de la UVic. Profesor del máster de Tecnologías de la información y la comunicación en la empresa. Ha sido jefe del Departamento de Teoría de la Señal y Comunicaciones, y miembro del Consejo de Dirección del EPS de la UVic.

La revisión de este recurso de aprendizaje UOC ha sido coordinada por el profesor: Ferran Adelantado Freixer (2019)

Segunda edición: septiembre 2019

© Enric López i Rocafiguera, Pere Barberán Agut

Todos los derechos reservados

© de esta edición, FUOC, 2019

Av. Tibidabo, 39-43, 08035 Barcelona

Realización editorial: FUOC

*Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares del copyright.*

# Índice

<b>Introducción</b> .....	5
<b>Objetivos</b> .....	6
<b>1. Conceptos básicos</b> .....	7
1.1. Calidad de servicio .....	9
1.2. Tráfico en Internet .....	10
1.3. Multidifusión .....	11
<b>2. Mecanismos de planificación y control</b> .....	12
2.1. Planificación .....	12
2.1.1. Colas FIFO ( <i>first-in first-out</i> ) .....	12
2.1.2. Colas con prioridad .....	13
2.1.3. Colas equitativas (FQ y WFQ) .....	14
2.2. Control del flujo de paquetes .....	14
2.2.1. Algoritmo de regulación por goteo ( <i>leaky-bucket</i> ) .....	15
2.2.2. Regulación por testigos ( <i>token-bucket</i> ) .....	16
<b>3. Mecanismos de control del tráfico</b> .....	18
3.1. Servicios integrados ( <i>IntServ</i> ) .....	18
3.1.1. Reserva de recursos .....	19
3.1.2. Servicios .....	19
3.2. El protocolo de reserva de recursos (RSVP) .....	21
3.2.1. Procedimiento del protocolo RSVP .....	21
3.2.2. Estilos de reservas .....	23
3.2.3. Características de RSVP .....	26
3.3. Servicios diferenciados ( <i>DiffServ</i> ) .....	26
3.3.1. Arquitectura .....	27
3.3.2. Acondicionamiento del tráfico .....	28
3.3.3. Comportamiento por salto (PHB) .....	29
<b>Resumen</b> .....	31
<b>Actividades</b> .....	33
<b>Ejercicios de autoevaluación</b> .....	33
<b>Solucionario</b> .....	34
<b>Glosario</b> .....	34
<b>Bibliografía</b> .....	35



## Introducción

A medida que crecen las redes privadas e Internet, aumenta el volumen de tráfico y la cantidad de información que ha de transmitirse mediante la red. Añadido a este aumento de comunicaciones, éstas cada vez están formadas por más gráficos y aplicaciones en tiempo real (audio y vídeo) con una gran cantidad de información que provocan que la carga todavía aumente más. Eso crea la necesidad de gestionar correctamente el tráfico y controlar la congestión.

Las únicas redes pensadas para transmitir tráfico de diferentes aplicaciones mediante la interconexión de redes WAN son las redes basadas en el protocolo IP. Pero estas redes no están diseñadas para transmitir aplicaciones multimedia y multidifusión en tiempo real.

Hay que poder transmitir esta gran variedad de tráfico, con mucha variedad de necesidades referentes a calidad de servicio (QoS), dentro de la arquitectura TCP/IP.

En este módulo didáctico se quiere dar una visión de los mecanismos que proporcionan un entorno de trabajo para los servicios actuales y futuros de redes interconectadas.

La arquitectura de servicios integrados (*IntServ*) proporciona un marco para el desarrollo de protocolos para tratar el tráfico multimedia y multidifusión, como por ejemplo el RSVP.

La arquitectura de servicios diferenciados (*DiffServ*) proporciona una herramienta simple para ofrecer unos servicios de red particulares en función del tipo de tráfico.

## Objetivos

Los materiales didácticos de este módulo os deben permitir alcanzar los siguientes objetivos:

- 1.** Entender la problemática de la transmisión de gran cantidad y variedad de tráfico mediante la interconexión de redes privadas e Internet para alcanzar una determinada calidad de servicio.
- 2.** Conocer los mecanismos de planificación y control del tráfico por parte de los encaminadores mediante los diferentes modelos de colas y de cubetas.
- 3.** Entender la aportación de los servicios integrados (*IntServ*) a la gestión del tráfico multimedia.
- 4.** Entender el funcionamiento de la reserva de recursos del protocolo RSVP (*resource reservation protocol*).
- 5.** Conocer los servicios que ofrece la arquitectura de servicios diferenciados (*DiffServ*) según el tipo de tráfico.

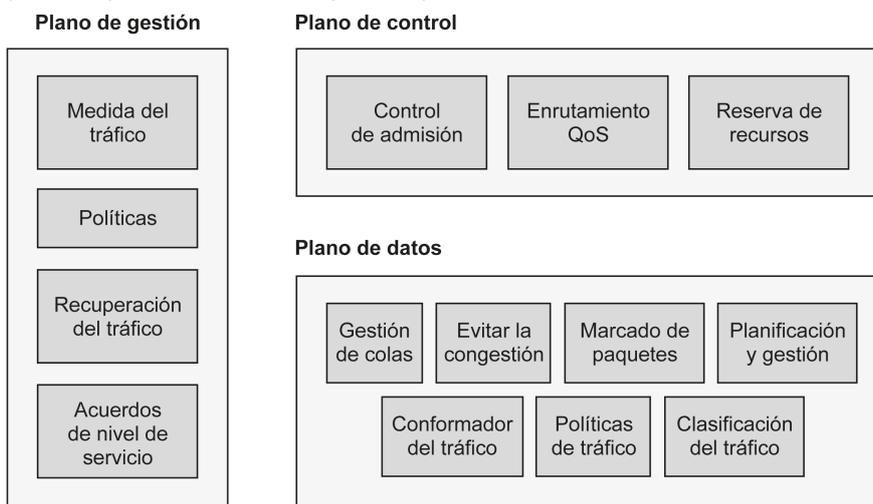
## 1. Conceptos básicos

Internet proporciona un servicio de mejor esfuerzo (*best effort*) a todas sus aplicaciones. Este servicio consiste en mover cada paquete del origen al destino tan rápidamente como sea posible, sin fijar ningún retardo máximo entre extremos, ni sobre la pérdida de paquetes en un flujo de datos; es decir, no promete ningún tipo de calidad de servicio en las aplicaciones. Cada aplicación tendrá el rendimiento que la red le pueda proporcionar en cada momento. Todos los paquetes son tratados del mismo modo, tanto si son sensibles al retardo, como el audio y el vídeo, como si no lo son.

En la arquitectura Internet se pueden añadir varios componentes que intentarán proteger las diferentes aplicaciones de la congestión que se pueda producir y así dar una calidad de servicio que Internet no puede dar por sí sola.

Antes de empezar a describir los diversos componentes referentes a la calidad de servicio, es útil tener una imagen de la arquitectura de referencia que relaciona los distintos elementos. Este marco de referencia ha sido desarrollado por la ITU-T (International Telecommunication Union) y forma parte de las recomendaciones Y.1291 (*An Architectural Framework for support of quality of service in packet networks*). La figura siguiente muestra los diferentes elementos, que se organizan en tres planos: datos, control y gestión.

Figura 1. Arquitectura de referencia para el soporte QoS



A continuación describiremos estos elementos, a pesar de que trataremos algún otro con más detalle posteriormente:

- **Plano de datos:** este plano incluye todos los mecanismos que operan directamente en los flujos de datos.

- Gestión de colas: los algoritmos de gestión de colas gestionan la longitud de las colas de paquetes y descartan ciertos paquetes cuando sea necesario. La gestión activa de las colas tiene como objetivo prioritario evitar la congestión.
  - Planificación y gestión de colas: hay diferentes modelos de colas que permiten gestionar la asignación de la capacidad de transmisión entre flujos de datos.
  - Evitar la congestión: su objetivo es tratar de mantener la carga de la red bajo ciertos umbrales que permitan que pueda operar a unos niveles de rendimiento aceptables.
  - Marcado de paquetes: implica dos funciones. Por un lado, los paquetes pueden ser marcados en la red de acceso para indicar que necesitan recibir algún tipo de calidad de servicio por los nodos intermedios. Por otro lado, los paquetes pueden ser marcados para indicar que se pueden descartar en caso de que haya congestión (como vimos por ejemplo en Frame Relay).
  - Clasificación del tráfico: esta clasificación se puede hacer por paquete o por flujo. Así, todo el tráfico asignado a un flujo recibe un tratamiento similar.
  - Conformador del tráfico: controla la tasa y el volumen de un flujo de tráfico transmitido a la red.
- **Plano de control:** se ocupa de la creación y gestión de los caminos mediante los cuales se envían los flujos de datos. Incluye el control de admisión, enrutamiento QoS y la reserva de recursos.
    - Control de admisión: determina qué tráfico de usuario puede entrar en la red. Viene determinado por las necesidades del flujo y si la red se puede comprometer a cumplir estas necesidades.
    - Enrutamiento QoS: determina el camino dentro de la red que se acomoda a las necesidades de calidad de servicio que necesita un flujo.
    - Reserva de recursos: es el mecanismo que hace la reserva de recursos dentro de la red para garantizar las demandas de un flujo concreto.
- **Plano de gestión:** este plano contiene los mecanismos que afectan tanto al plano de datos como al plano de control. Incluye los acuerdos de nivel de servicio (SLA), recuperación del tráfico, medida del tráfico y políticas de tráfico.
    - El SLA representa el acuerdo entre el usuario y el proveedor de servicio, en el que se especifican los niveles de disponibilidad, rendimiento, etc., del servicio.
    - Recuperación del tráfico: se refiere a la respuesta de la red ante fallos.
    - Medida del tráfico: se ocupa de monitorizar las propiedades dinámicas de los flujos de tráfico mediante métricas, como por ejemplo la tasa de datos y la pérdida de paquetes de datos.

## 1.1. Calidad de servicio

Podemos definir la calidad de servicio (QoS) como el conjunto de requisitos que ha de tener un determinado flujo de datos.

Las principales características de un flujo de datos son estas:

- **Fiabilidad en la transmisión, y que no haya pérdidas de paquetes:** esta característica es necesaria para garantizar que el flujo de datos se envía de origen a destino de manera segura. La falta de fiabilidad significa la pérdida de paquetes o reconocimientos, lo que implica retransmisiones. La fiabilidad no tiene la misma importancia para todas las aplicaciones. Así, por ejemplo, la fiabilidad es mucho más importante para la transmisión de ficheros o correo electrónico que para el audio.
- **Retardo en la transmisión:** de la misma manera que en el caso anterior, algunas aplicaciones pueden tolerar mejor o peor este retardo. Aplicaciones como la telefonía, la videoconferencia o similares necesitan un retardo mínimo, mientras que la transferencia de ficheros o el correo son más tolerantes.
- **Distorsión debida a la variación del retardo en un mismo flujo (*jitter*):** por ejemplo, si enviamos cuatro paquetes en los instantes 0, 1, 2, 3 y llegan en los instantes 10, 11, 12 y 13, tenemos un mismo retardo de 10 unidades de tiempo. En cambio, si llegan en los instantes 10, 12, 13 y 17, tenemos diferentes retardos. Para aplicaciones como el audio y el vídeo, el primer caso es aceptable, mientras que el segundo no lo es. Para este tipo de aplicación la variación de retardo debe estar acotada.
- **Ancho de banda o velocidad de transmisión:** diferentes aplicaciones tienen necesidades de ancho de banda distintas.

### Las características de un flujo de datos

La transmisión del correo electrónico necesita más fiabilidad que la telefonía; o la transferencia de archivos permite más retardo que cualquier aplicación en tiempo real, como la telefonía. La velocidad de transmisión de vídeo ha de ser mucho mayor que la de transmisión de un texto.

A continuación, se muestra una tabla con diferentes aplicaciones y sus necesidades:

Aplicación	Fiabilidad	Retardo	Variación del retardo	Ancho de banda
FTP	Alta	Bajo	Baja	Media
HTTP	Alta	Medio	Baja	Media
Audio bajo demanda	Baja	Bajo	Alta	Media
Vídeo bajo demanda	Baja	Bajo	Alta	Alta
VoIP	Baja	Alto	Alta	Baja
Vídeo IP	Baja	Alto	Alta	Alta

Según estas características, los flujos se pueden clasificar en grupos o clases, tal como hace ATM.

 Sobre la clasificación de los flujos en grupos o clases podéis ver el módulo "WAN".

La calidad de servicio está estrechamente ligada al control de la congestión. Cualquier técnica que mejore la congestión mejorará la calidad de servicio. Las principales técnicas para mejorar la calidad de servicio son las siguientes:

- La planificación para tratar los flujos de entrada y salida adecuadamente.
- El control del tráfico.
- La reserva de recursos.
- El control de admisión de los diferentes flujos.

El término QoS en las redes conmutadas como *frame relay* o ATM se trata de forma diferente. En el caso de redes de conmutación de paquetes, hablábamos de congestión cuando se enviaban más paquetes a la red de los que era capaz de tratar, y explicábamos los mecanismos de control. En el caso del *frame relay* se utilizaban cuatro informaciones diferentes para poder controlar el tráfico: la velocidad de acceso, la medida de la ráfaga comprometida ( $B_c$ ) y la de la ráfaga en exceso ( $B_e$ ), y la tasa de información comprometida (CIR). ATM utilizaba las clases de servicios y los atributos con el fin de gestionar la QoS.

 Sobre el uso del término QoS en las redes conmutadas como *frame relay* o ATM, podéis ver el módulo "WAN".

La congestión en las redes, o interconexión de redes, sucede, entre otras cosas, porque en los encaminadores hay unos *buffers* de entrada y de salida que almacenan el paquete antes y después de ser procesado. En función de las diferencias entre la tasa de llegada, la tasa de procesamiento y la tasa de salida, estos *buffers* estarán más o menos llenos y se producirán unos retardos más o menos grandes. El control de la congestión en estos casos consistirá en la gestión de las diferentes tasas y de las colas de entrada y salida, en función de los diferentes flujos, tal como veremos seguidamente.

Hay dos formas básicas de control de la congestión:

- **Bucle abierto.** Consiste en aplicar los mecanismos adecuados para prevenir la congestión antes de que se produzca. Algunos de estos mecanismos son las políticas de admisión, las de descarte, la retransmisión, la confirmación, etc.
- **Bucle cerrado.** Consiste en aplicar mecanismos para eliminar la congestión una vez se ha producido.

## 1.2. Tráfico en Internet

El tráfico en una red, o en diferentes redes interconectadas, se puede dividir en dos categorías con unas necesidades muy diferentes:

- **Tráfico elástico.** Es el tráfico que se puede ajustar a cambios en el retardo y en el rendimiento de la red y continuar satisfaciendo las necesidades de las aplicaciones. Es el tipo de tráfico que tradicionalmente circula a través de las redes basadas en TCP/IP y que intenta utilizar tanta capacidad como haya disponible. Algunas de estas aplicaciones son el correo electrónico, la transferencia de ficheros, la gestión de red, el acceso web, etc. Son aplicaciones, todas ellas, con necesidades muy diferentes.
- **Tráfico inelástico.** Es el tráfico que no se adapta fácilmente a los cambios en el retardo y en el rendimiento en una red y que a veces no consigue adaptarse. Es lo que denominamos tráfico en tiempo real. Es un tráfico que necesita un rendimiento mínimo, no puede sufrir retardos ni variación de estos retardos y no tolera la pérdida de paquetes. Eso introduce dos requisitos nuevos en la arquitectura de interconexión de redes:
  - Han de poder obtenerse los medios preferentes para aplicaciones con requisitos más exigentes. Eso supone que antes de la transmisión se han de poder indicar las necesidades de recursos y, posteriormente, se han de poder reservar.
  - Es necesario seguir atendiendo el tráfico elástico simultáneamente. El tráfico inelástico, en caso de congestión, no reducirá su carga y eso puede provocar que no haya recursos para el tráfico elástico. Por ello será necesario un protocolo que reserve recursos para que el tráfico elástico se pueda continuar transmitiendo.

### 1.3. Multidifusión

Una complejidad añadida es la multidifusión, que consiste en enviar una misma información de un determinado origen a un grupo de destinos. En la actualidad existen varias aplicaciones que se realizan por multidifusión, como las aplicaciones multimedia, la teleconferencia, el acceso a bases de datos, el trabajo en grupo, etc.

La multidifusión es más compleja que la difusión en un solo destino, entre otras cosas porque se puede producir la duplicación de paquetes, que a su vez puede causar confusión y un incremento de la carga en la red. Hay diferentes algoritmos de multidifusión que intentan minimizar el número de paquetes transmitidos por la red, como es el IGMP (*Internet group management protocol*). Estos mecanismos se basan en la determinación del camino de menor coste entre el origen de la multidifusión y todos los miembros del grupo de multidifusión, de manera que se crea un árbol de expansión. Posteriormente, la fuente transmite un único paquete por medio del árbol y los diferentes encaminadores crean réplicas de este paquete en los puntos de bifurcación del árbol.

#### Aplicaciones del IGMP

El IGMP es el protocolo que utilizan los encaminadores para intercambiar información entre los miembros de un grupo sobre una LAN.

## 2. Mecanismos de planificación y control

### 2.1. Planificación

Con el fin de proporcionar garantías de calidad de servicio para las diferentes aplicaciones multimedia se deben conocer los mecanismos de colas que se utilizan.

En cada puerto de salida de un encaminador hay una cola. El mecanismo de funcionamiento consiste en que, en el momento en el que llega un paquete a un encaminador, se envía a uno de los puertos de salida y se coloca al final de la cola correspondiente. Mientras haya paquetes en la cola de salida del encaminador, éste transmitirá el paquete más antiguo.

Por otra parte, también es importante la **política de eliminación de paquetes** que nos determina qué pasa si la cola está llena al llegar un nuevo paquete. El encaminador o bien puede desestimar este paquete o bien puede eliminar otros paquetes que están en la cola para que quepan los paquetes nuevos que llegan.

Hay diferentes modelos de colas utilizados para planificar los enlaces: las colas FIFO, las colas con prioridad o las colas equitativas.

Los algoritmos que permiten asignar recursos entre los paquetes de un flujo y entre flujos que compiten se denominan algoritmos de planificación. Hay tres tipos de recursos que potencialmente pueden ser reservados para diferentes flujos:

- ancho de banda
- espacio en la cola
- ciclos de CPU

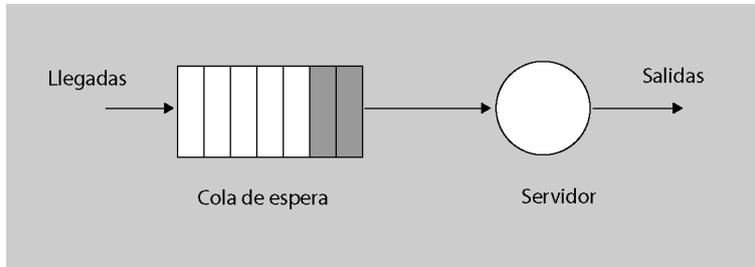
Estos algoritmos de planificación asignan ancho de banda y otros recursos del conmutador y determinan cuál de los paquetes puestos en las colas debe ser enviado a continuación.

#### 2.1.1. Colas FIFO (*first-in first-out*)

El modelo de cola para la disciplina de planificación de enlace, primero en entrar-primero en salir (FIFO), es el que se ha utilizado tradicionalmente en los dispositivos de encaminamiento en cada uno de los puertos de salida.

En cada puerto de salida hay una cola simple y los paquetes que llegan han de ubicarse detrás de la cola y esperar su turno para ser transmitidos. La forma de vaciado es la de elegir el paquete más antiguo en la cola. Gráficamente lo podemos representar con la figura 2.

Figura 2. Modelo de disciplina de planificación de enlace FIFO



#### Representación del modelo FIFO

Al modelo FIFO lo podemos representar con una pila, donde por un extremo van llegando los paquetes hacia la cola y por el otro extremo se van sirviendo, tomando el paquete más antiguo en la cola. Todos los paquetes tienen el mismo tratamiento.

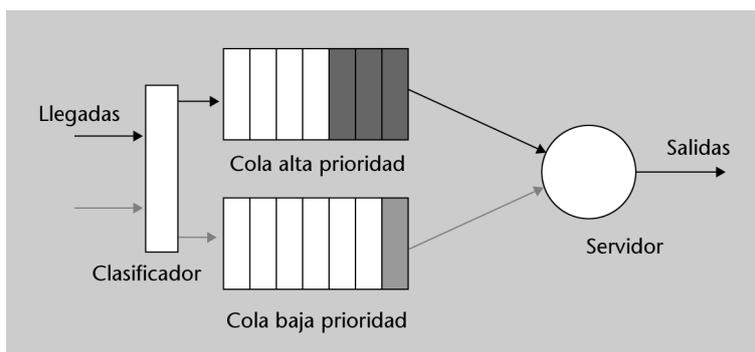
Este mecanismo tiene el inconveniente de no observar las prioridades de los paquetes de flujos sensibles a los retardos, ni tampoco la dimensión de los paquetes.

### 2.1.2. Colas con prioridad

Las colas con prioridad intentan solucionar alguno de los inconvenientes de las colas FIFO. Los paquetes que llegan al enlace de salida se clasifican en **clases de prioridad**. Esta clase de prioridad se obtiene de la cabecera del paquete, de la dirección IP del origen o del destino, o de otros criterios.

Cada clase de prioridad tiene su cola independiente. Cuando se deba transmitir un paquete se elegirá uno de la clase más alta de una cola no vacía. Cada cola trabajará a partir del criterio anterior FIFO para elegir el paquete que ha de transmitir. En la figura 3 podéis ver la representación gráfica de las colas con prioridad.

Figura 3. Modelo de disciplina de planificación de enlace con prioridades



#### Representación del modelo de colas con prioridad

En la representación del modelo de colas con prioridad, se distinguen dos prioridades: prioridad alta y prioridad baja. Los flujos de paquetes que llegan son clasificados según la prioridad y posteriormente son colocados en la cola correspondiente. El servidor envía a la salida un paquete de la cola de alta prioridad (en caso de que no esté vacía). Una vez enviados los paquetes de alta prioridad se pasa a servir los de baja prioridad.

El inconveniente que tiene este modelo es que mientras haya paquetes de prioridad alta no se podrán servir los paquetes de prioridad baja.

### 2.1.3. Colas equitativas (FQ y WFQ)

Otro sistema que soluciona algunos inconvenientes de los sistemas con colas FIFO es el sistema de colas equitativas o *fair queuing* (FQ). Este sistema consiste en hacer que los paquetes también se ordenen en clases, como en el caso anterior. Puede ser, incluso, una clase para cada flujo de paquetes.

La disciplina de planificación de enlace consiste en que cada paquete que llega se coloca en la cola correspondiente. Las colas se tratan con la disciplina de colas *round robin*, en la que se toma un paquete de cada cola no vacía por turnos. Si una cola está vacía se salta a la siguiente.

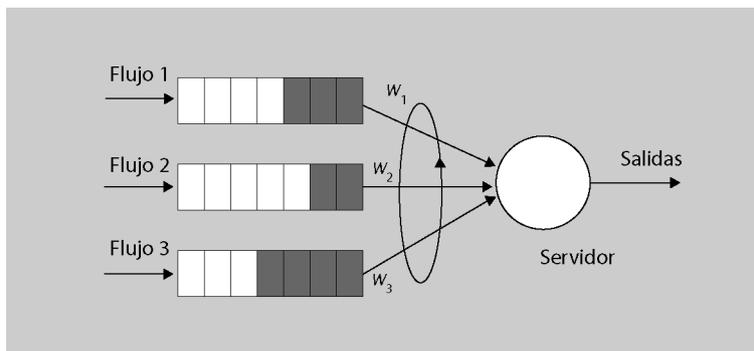
Este sistema es equitativo, ya que cada flujo consigue enviar exactamente un paquete en cada ciclo.

Algunos fabricantes implementan una mejora de las colas equitativas, conocida con el nombre de sistema de colas equitativas ponderadas o *weighted fair queuing* (WFQ). Los paquetes que llegan se clasifican en clases y se colocan en la cola correspondiente a su clase. Va dando servicio a las clases de forma circular, al igual que con *round robin*. El sistema WFQ asigna más capacidad en las colas más ocupadas, pero sin dejar de atender las colas menos ocupadas. WFQ ajusta la disciplina de atención de las colas mediante la ponderación del servicio de las clases con pesos ( $w_i$ ) en función de la cantidad de tráfico, o en función de la cantidad de servicio solicitado por el tráfico de cada flujo. En la figura 4 podéis ver la representación gráfica de las colas ponderadas.

#### Round robin

El *round robin* es una disciplina de planificación que consiste en servir los diferentes flujos de datos de forma secuencial, mediante turnos.

Figura 4. Modelo de disciplina de planificación de enlace equitativo ponderado



## 2.2. Control del flujo de paquetes

Un aspecto importante que hay que tener en cuenta es el control del flujo de paquetes. Este control se puede realizar mediante tres parámetros:

1) **Tasa media.** Puede interesar limitar la velocidad media a la que se pueden enviar los paquetes de un flujo de datos a la red en un determinado intervalo de tiempo grande.

2) **Tasa pico.** Permite limitar el número de paquetes enviados en un intervalo más corto de tiempo.

3) **Dimensión de impulso.** Es el límite de paquetes que se pueden enviar a la red instantáneamente, en un intervalo muy corto. Físicamente no tiene mucho sentido, pero es una abstracción útil en determinados casos.

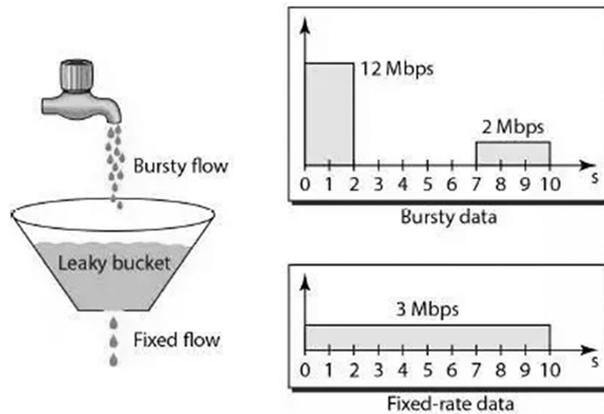
Este control se puede realizar a partir de la especificación del tráfico utilizando el algoritmo de regulación por goteo.

### 2.2.1. Algoritmo de regulación por goteo (*leaky-bucket*)

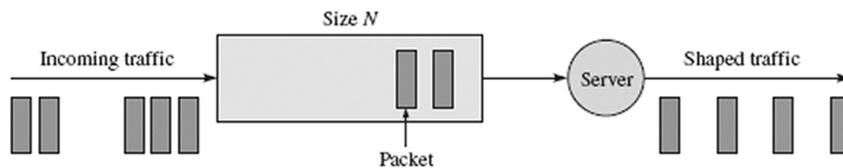
Podemos comparar el modelo del algoritmo de regulación por goteo con un cubo con un agujero en la parte inferior. El agua del cubo saldrá a una velocidad constante, independientemente de la velocidad a que se introduzca agua en el cubo. Del mismo modo, un tráfico a ráfagas se puede suavizar mediante un sistema como este. Las ráfagas de paquetes que llegan se convierten en un tráfico con una tasa fija obtenida haciendo la media de la tasa de llegada de los datos. Si el cubo está lleno se pueden perder datos, al igual que si un cubo está lleno de agua, el agua que queremos añadir se derramará y se perderá.

El algoritmo de regulación por goteo transforma el tráfico a ráfagas en tráfico con tasa constante. En caso de que la cola esté llena, se descarta el tráfico que llega.

En la figura siguiente se ve la aplicación de la regulación por goteo, también conocida por el término inglés *leaky bucket*, y sus efectos. El *host* envía el tráfico a ráfagas. Inicialmente, envía una ráfaga a 12 Mbps durante dos segundos, que equivale a un total de 24 Mbits. A continuación, hace una pausa de 5 segundos y, finalmente, envía una segunda ráfaga a 2 Mbps durante 3 segundos, que significa un total de 6 Mbits. Lo que hace el algoritmo de regulación por goteo es suavizar el tráfico entrante, enviando a la salida una tasa constante de 3 Mbps durante un periodo de 10 segundos.

Figura 5. Mecanismo de regulación por goteo o *leaky bucket*

En la figura siguiente se muestra cómo se implementaría un nodo que aplique el algoritmo de *leaky bucket*.

Figura 6. Implementación de *leaky bucket*

Este algoritmo es bastante restrictivo, dado que no tiene en cuenta el *host* que no envía paquetes durante un periodo de tiempo. En esta situación podemos encontrar con que el cubo (cola) esté vacío, pero el mecanismo solo calcula la tasa media mientras hay actividad y no en los periodos de inactividad.

### Ejercicio

Un *host* A tiene que enviar 19,5 Mbytes y los envía con ráfagas que se transmiten a 6 Mbps. La tasa máxima del conmutador de salida es de 4 Mbps. Si la línea de entrada al conmutador utiliza el algoritmo de *leaky bucket*, queremos saber cuál tiene que ser la capacidad de la cola para no descartar ningún dato.

### Solución:

Tiempo total que el *host* tarda en enviar los datos hacia el conmutador:

$$T_{\text{total host}} = (19,5 \text{ Mbytes} \times 8 \text{ bits/byte}) / 6 \text{ Mbps} = 26 \text{ segundos}$$

El conmutador transmite a una tasa constante de 4 Mbps. Por lo tanto, en 26 segundos habrá conmutado 13 Mbytes (es decir,  $4 \text{ Mbps} \times 26 \text{ s} / 8 \text{ bits/byte} = 13 \text{ Mbytes}$ ).

Si el *host* ha enviado 19,5 Mbytes, la cola deberá ser de 6,5 Mbytes.

### 2.2.2. Regulación por testigos (*token-bucket*)

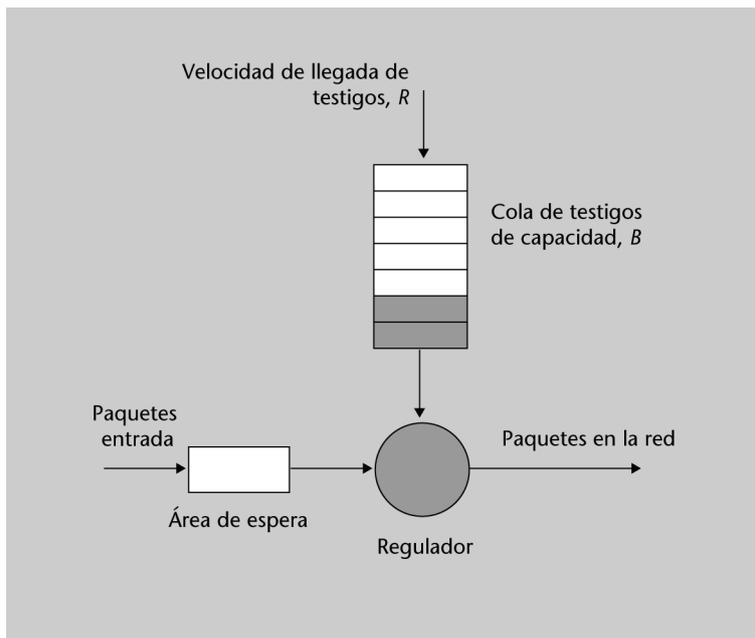
El algoritmo de regulación por goteo se puede aplicar a testigos en lugar de aplicarlo directamente a los datos. Por lo tanto, lo podemos asociar al concepto de un cubo de testigos, o fichas, asociados a los datos. Se basa en dos parámetros:

- Velocidad de llegada de los testigos ( $R$ ), o tasa media de datos admitida por un flujo determinado en un periodo de tiempo largo.
- Capacidad del regulador ( $B$ ), que nos indicará el número de testigos que puede almacenar.

Veamos cómo lo aplicamos al control de flujo. Primero hemos de suponer que para que un paquete se pueda transmitir en la red debe eliminar un testigo del regulador. Si el regulador está vacío de testigos, el paquete debe esperar a que llegue un testigo. En cambio, si está lleno con los  $B$  testigos permitirá que haya una ráfaga instantánea de hasta  $B$  paquetes. La capacidad del regulador ( $B$ ) nos limitará la dimensión del impulso.

El número máximo de paquetes que puede pasar a la red en cualquier intervalo de tiempo  $T$  es  $R \cdot T + B$ . El encaminador fija la velocidad de llegada de los testigos en el regulador ( $R$ ) y así fija la tasa media de transmisión de los paquetes.

Figura 7. Modelo de control de flujo con regulador por testigos



#### Esquema del modelo del regulador con testigos

En la representación del modelo del regulador con testigos el regulador está modelado con una cola de capacidad  $B$ . El regulador tiene la función de eliminar un testigo por cada paquete que envíe a la red. El control del flujo de paquetes que se enviarán a la red depende directamente de la tasa de llegada de los testigos ( $R$ ) y de la capacidad del regulador ( $B$ ).

### 3. Mecanismos de control del tráfico

Se han diseñado dos mecanismos para mejorar la calidad de servicio en Internet: los servicios integrados y los servicios diferenciados. Ambos modelos enfatizan el uso de la calidad de servicio desde el punto de vista de la red (IP), aunque también se pueden utilizar en cuanto a enlace de datos.

El protocolo IP se diseñó para ofrecer el mismo nivel de servicio a todos los usuarios. Eso no garantiza el mínimo de servicio a las aplicaciones en tiempo real. Cuando una aplicación necesita ancho de banda extra, puede afectar a otras aplicaciones y producir congestión.

Los servicios integrados son un modelo de QoS basado en flujo; eso quiere decir que un usuario necesita crear un flujo, parecido a un circuito virtual, desde el origen hasta el destino, e informar a todos los encaminadores de sus necesidades de recursos.

#### 3.1. Servicios integrados (*IntServ*)

Los servicios integrados o *integrated services* son un modelo de QoS basado en flujo diseñado por IP.

IP es un protocolo no orientado a conexión que, por lo tanto, utiliza la técnica de conmutación de paquetes basada en datagramas. Para implementar *IntServ* sobre IP es necesario un protocolo de señalización que permita hacer la reserva de los recursos; este protocolo es el RSVP (*resource reservation protocol*).

La arquitectura *IntServ* fue desarrollada por el IETF (Internet Engineering Task Force) con el fin de proporcionar garantías de calidad de servicio a las diferentes aplicaciones de manera individualizada.

*IntServ* necesita, por una parte, saber qué recursos hay reservados y cuáles están utilizando los flujos en un momento determinado. Por otra parte, para poder dar una QoS a un nuevo flujo también necesita poder reservar los recursos necesarios en cada uno de los encaminadores por los cuales ha de pasar para ir del origen al destino. !

#### IETF

El Internet Engineering Task Force (IETF) es un organismo que desarrolla estándares de protocolos por Internet.

### 3.1.1. Reserva de recursos

Con el fin de poder realizar una nueva comunicación, hay que hacer la reserva de recursos para el nuevo flujo. Para poder realizar la reserva, es necesario que cada encaminador determine cuáles son los recursos locales que son necesarios, teniendo en cuenta los recursos que ya están ocupados por otras comunicaciones, y comprobando si dispone de los recursos suficientes para satisfacer la QoS de la nueva sesión sin afectar a las sesiones abiertas. El procedimiento es el siguiente:

1) **Especificación del flujo.** Para que un encaminador pueda determinar si dispone de suficientes recursos para un nuevo flujo, hay que caracterizar la cantidad de tráfico que comportará este flujo y cuáles son sus necesidades de QoS. Para eso utiliza dos parámetros, Rspec y Tspec:

- Rspec. Especifica los recursos que necesita reservar el flujo (ancho de banda, *buffers*, etc.) para una determinada QoS.
- Tspec. Define las características de tráfico del flujo.

2) **Señalización.** Los dos parámetros, Rspec y Tspec, se deben enviar a los encaminadores para que reserven los recursos necesarios. El protocolo que hace esta función en Internet es el RSVP.

3) **Admisión de la llamada.** Cuando un encaminador recibe una petición de reserva de recursos para una QoS de un nuevo flujo a partir de los parámetros Rspec y Tspec, puede decidir si admitirla o denegarla. Dependerá de la especificación del tráfico, del tipo de servicio y de los recursos que ya tiene comprometidos y de los que todavía dispone.

 El protocolo de reserva de recursos (RSVP) se trata en el subapartado 3.2 de este módulo.

### 3.1.2. Servicios

*IntServ* define dos clases de servicios: servicios garantizados y servicios de carga controlada.

La arquitectura *IntServ* clasifica los principales servicios en los siguientes tipos:

- **Servicios garantizados.** Están pensados para aplicaciones en tiempo real que necesitan un retardo máximo garantizado en la transmisión entre los extremos y no aceptan la pérdida de paquetes. Estos servicios proporcionan una tasa de transmisión de datos, un límite de retardo en la transmisión y garantizan que los paquetes no se perderán en las colas de los encaminadores.

El retardo vendrá dado por el retardo de propagación y por el retardo en los encaminadores. El retardo en la propagación dependerá de la distancia de la transmisión, y en general será pequeño. El retardo en los encaminadores dependerá de la longitud de las colas y de la tasa de transmisión. Si aproximamos el tráfico a un encaminador con el modelo del regulador por goteo con una tasa de *llegada*  $R$  y una capacidad  $B$ , el retardo máximo que puede tener un paquete en la cola es de  $B/R$ , que corresponderá al retardo que experimentará un paquete que encuentra la cola con sólo una posición vacía, y, por lo tanto, para poder ser enviado deberá *esperar* que los  $B - 1$  paquetes anteriores sean transmitidos a la velocidad media  $R$ .

Cada aplicación indicará con el parámetro  $T_{\text{spec}}$  cuál es el tráfico que espera y el encaminador determinará si lo puede garantizar. Este es el servicio más exigente ofrecido por *IntServ*.

- Servicios de carga controlada. Este servicio ofrece a la sesión una calidad de servicio muy parecida a la QoS que recibiría el mismo flujo de datos con un servicio de mejor esfuerzo sin carga. Asegura que un porcentaje alto de paquetes no sufrirá prácticamente retardo ni será desestimado, y, por lo tanto, que llegará al destino correctamente.

El servicio de carga controlada garantiza que la red reservará suficientes recursos para que cualquier aplicación que reciba este servicio no tenga que competir por los recursos con otras aplicaciones en tiempo real.

Este servicio está dirigido a aplicaciones multimedia en tiempo real por Internet que funcionan correctamente si la red está descargada y dejan de hacerlo en caso de congestión. Por ejemplo, si se producen pequeños retardos en la voz, ésta puede ser adaptada con la modificación de los períodos de silencio.

Los principales inconvenientes de la utilización de servicios integrados que pueden atascar la implementación en Internet son los siguientes:

- **La escalabilidad.** Los encaminadores deben tener información de cada flujo. Actualmente, Internet está creciendo mucho, tanto con respecto a usuarios como con respecto a carga transmitida; eso puede ser un problema teniendo en cuenta la gran cantidad de flujos que se generan.
- **Limitación del tipo de servicio.** El modelo *IntServ* sólo permite dos tipos de servicios, el servicio garantizado y el de carga controlada, y algunas aplicaciones podrían necesitar otros servicios.

### 3.2. El protocolo de reserva de recursos (RSVP)

En los apartados anteriores hemos visto que cualquier aplicación necesita una reserva de recursos con el fin de prevenir la congestión y poder obtener la QoS deseada. En el modelo *IntServ*, esta reserva de recursos se emplea para un flujo; por lo tanto, si se quiere utilizar el modelo *IntServ* al nivel IP, hay que crear un flujo basándose en una red de circuitos virtuales.

#### RSVP

RSVP son las iniciales de *resource reservation protocol*, que es un protocolo para reservar recursos para *IntServ*.

Una red basada en circuitos virtuales necesita un sistema de señalización con el fin de configurar correctamente el circuito virtual antes del envío de datos.

El **protocolo de reserva de recursos (RSVP)** es el protocolo que permite al protocolo IP hacer la reserva de los recursos necesarios (ancho de banda y *buffers* en los encaminadores) para crear un flujo.

El protocolo RSVP es un protocolo independiente, que se puede utilizar en otros modelos diferentes a *IntServ*, pero que ha de estar implementado en el origen, en el destino y en los encaminadores.

Este protocolo es un sistema de señalización diseñado por multidifusión, como es el caso del tráfico multimedia. En todo caso, también se puede utilizar para enviar a un único destino, ya que se puede considerar un caso particular de multidifusión un grupo formado por un solo miembro.

#### 3.2.1. Procedimiento del protocolo RSVP

El protocolo RSVP es un protocolo orientado al receptor, es decir, quien hace la reserva es el receptor y no el emisor. El motivo por el cual es el receptor quien hace la reserva es que es posible que, en el caso de multidifusión, diferentes miembros del grupo soliciten diferentes requisitos de QoS, o también es posible que, si hay diferentes emisores transmitiendo a un grupo multidifusión, los miembros del grupo quieran seleccionar recibir sólo una parte de las transmisiones. Por estos motivos es preferible que la reserva sea realizada por los receptores, de acuerdo con sus necesidades.

El protocolo RSVP consta de diferentes tipos de mensajes:

- Mensaje *Path*. Los receptores son los que hacen la reserva en un flujo, pero antes de hacer la reserva no saben qué camino seguirán los paquetes. Primero hay que establecer el camino. Por eso se utilizan los mensajes *Path*. Los mensajes *Path* se envían desde el emisor hacia todas las destinaciones mediante árbol de distribución. Este mensaje contiene la información ne-

cesaria para que los receptores conozcan el camino. Cuando un camino se divide se crea una réplica del mensaje.

- Mensaje *Resv*. Cuando el receptor de un grupo de multidifusión recibe un mensaje *Path*, envía un mensaje *Resv* hacia el emisor mediante el árbol de distribución, y hace la reserva de recursos en los encaminadores que soportan el protocolo RSVP. Estos mensajes se fusionan en los encaminadores para que en el árbol fluya sólo un único mensaje por los diferentes tramos de la red.

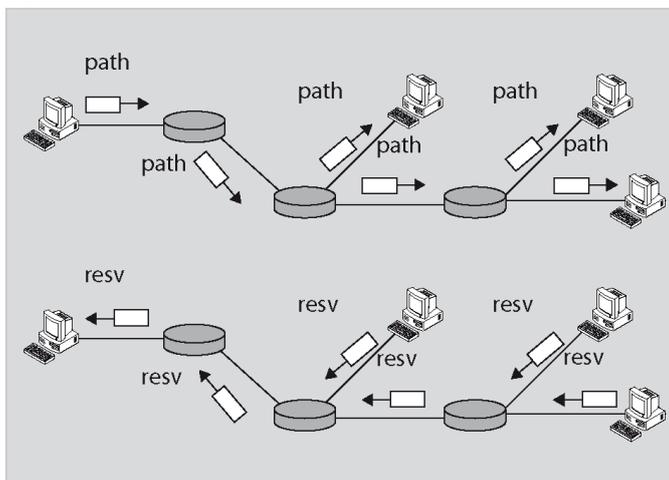
#### Árbol de distribución

El árbol de distribución es la estructura de enlaces que se crea desde el origen hasta los diferentes receptores de la información.

#### Ejemplo de envío de mensajes para establecer un camino

En la figura 8 podéis ver un ejemplo de cómo se realiza la reserva de recursos. Inicialmente está el envío de los mensajes *Path* del emisor en los diferentes receptores para poder establecer el camino. Una vez recibido este mensaje, los receptores envían el mensaje *Resv* con el fin de reservar los recursos necesarios a los encaminadores intermedios.

Figura 8. Envío de mensajes para establecer un camino

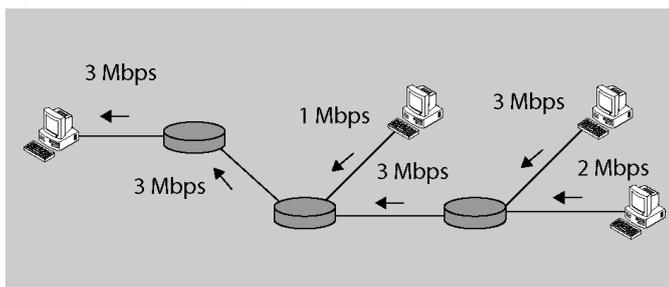


#### Ejemplo de fusión de los flujos en RSVP

En el protocolo RSVP los recursos no se reservan para cada receptor en un flujo diferente, sino que se fusionan las reservas. Eso lo podemos ver en este ejemplo.

Los receptores tienen diferentes necesidades de reserva de recursos. Estas reservas se fusionan eligiendo la mayor de las reservas que se hagan a cada encaminador.

Figura 9. Fusión de los flujos en RSVP



### 3.2.2. Estilos de reservas

Cuando hay más de un flujo, el encaminador debe hacer la reserva que convenga para todos los flujos. El protocolo RSVP define tres tipos de estilos de reservas, en función del tipo de reserva y de la selección del emisor del flujo. La reserva de recursos o bien puede ser compartida por diversos flujos o bien puede ser para cada flujo de forma independiente. Por otra parte, el receptor puede indicar una lista de emisores (explícita) o seleccionarlos todos implícitamente sin especificar nada (comodín).

Selección del emisor	Tipo de reserva	
	Independiente	Compartida
Explícita	Estilo de filtro fijo (FF)	Estilo explícito compartido (SE)
Comodín		Estilo de filtro comodín (WF)

Los tres estilos de reserva obtenidos de la combinación de las opciones son los siguientes:

1) **Estilo de filtro comodín (WF, *wildcard filter*)**. En este caso, el encaminador crea una única reserva que han de compartir todos los emisores que tengan esta dirección. La reserva de la capacidad se basa en la mayor petición de recursos. Este estilo se utiliza si los flujos de los diferentes emisores no se dan al mismo tiempo, como puede ser una multiconferencia, en la que sólo habla una persona y se puede compartir la capacidad para todos los emisores.

2) **Estilo de filtro fijo (FF, *fixed filter*)**. En este estilo, el encaminador crea una reserva diferente para cada emisor o flujo y proporciona una lista explícita de emisores. Por lo tanto, puede haber tantas reservas diferentes como número de flujos haya. Los recursos reservados por un enlace determinado son la suma de los recursos para todos los emisores. Este estilo se utiliza si los flujos de los diferentes emisores suceden al mismo tiempo, como es el caso de la distribución de vídeo desde diferentes emisores. Este estilo permite recibir las señales de vídeo en diferentes receptores compartiendo los recursos, es decir, un mismo flujo puede ser compartido por diferentes receptores; en cambio, no permite compartir recursos entre flujos generados por diferentes emisores.

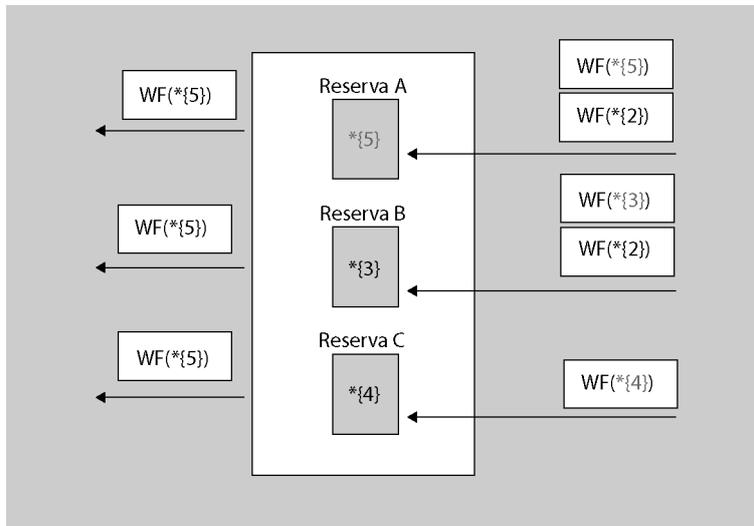
3) **Estilo explícito compartido (SE, *shared explicit*)**. En este caso, el encaminador crea una única reserva que comparten los flujos de una determinada lista explícita de emisores. Este estilo es adecuado para aplicaciones de multidifusión con varios flujos con poca probabilidad de que se transmitan simultáneamente.

Veamos los diferentes estilos de reservas con un ejemplo de reserva en un encaminador.

### Ejemplo de estilo de filtro comodín

Supongamos que, en el encaminador, las solicitudes de reserva de cinco receptores llegan por los tres puertos de salida (derecha), tal y como muestra la figura 10. En este caso se trata de una reserva con filtro comodín y la nomenclatura es WF (\*{Q}), donde \* indica el estilo comodín y Q indica la petición de recursos. El encaminador calcula la reserva para cada puerto de salida, con una capacidad correspondiente a la fusión de las peticiones solicitadas en el puerto y seleccionando el valor de mayor demanda. Por ejemplo, por el puerto A obtiene una reserva de 5, en el puerto B una reserva de 3, y en el C de 4. De todas formas, el encaminador hará una petición de reserva hacia el encaminador siguiente, en dirección al emisor, fusionando todas las peticiones de reservas y enviando el mayor valor, en este caso 5, por todos los puertos de entrada.

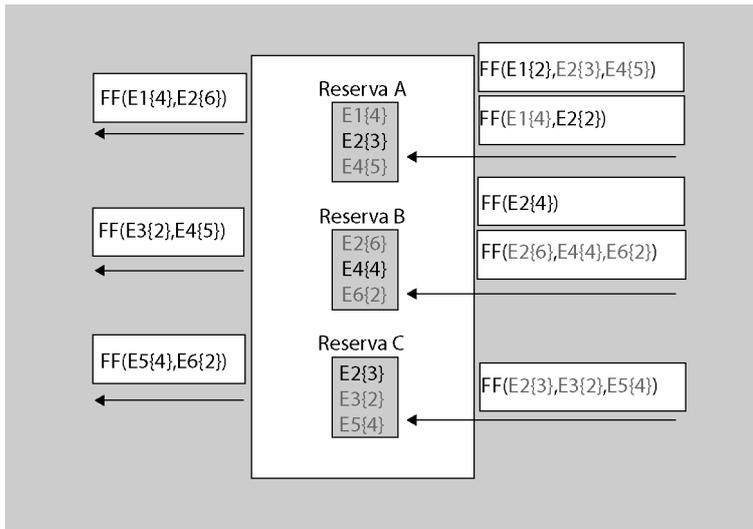
Figura 10. Ejemplo de reserva en un encaminador con estilo de filtro comodín, WF



### Ejemplo de estilo de filtro fijo

Igualmente, en el ejemplo anterior se hacen cinco peticiones diferentes de recursos desde los tres puertos de salida de un encaminador. En este caso, se trata de una reserva con filtro fijo y la nomenclatura es FF (E1{Q1}, E2 {Q2}...), donde  $E_i$  indica el emisor y  $Q_i$  indica la petición de recursos para este emisor  $E_i$ . El encaminador calcula la reserva de recursos para cada puerto de salida y para cada emisor fusionando las peticiones. Por ejemplo, por el puerto A hace una reserva de 4 para el emisor E1, de 3 para el E2 y de 5 para el E4. La petición de reserva que hará el encaminador será del valor obtenido de la fusión de las diferentes reservas por un emisor concreto por el puerto correspondiente al del emisor. En este caso vemos que por el primer puerto de entrada se hace la reserva correspondiente a los emisores E1 y E2 con unos valores de 4 y 6, respectivamente.

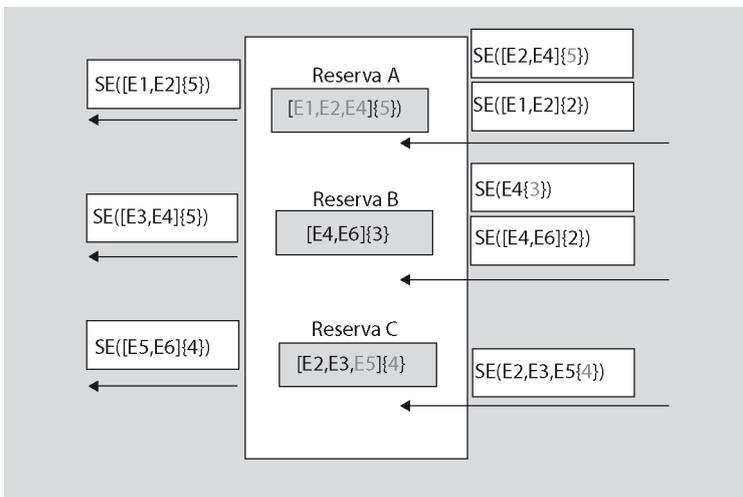
Figura 11. Ejemplo de reserva en un encaminador con estilo de filtro fijo, FF



### Ejemplo de estilo de filtro explícito compartido

Las peticiones de recursos especifican explícitamente el emisor, o emisores, a los cuales está dirigida la reserva, pero ésta es única. En este caso la nomenclatura es SE(E1, E2... {Q}), donde  $E_i$  indica el emisor y Q indica la petición de recursos. El encaminador calcula la reserva para cada puerto fusionando los valores y hace una petición de reserva en los puertos de entrada, obtenida otra vez a partir de la fusión de los valores de reserva correspondientes a los emisores de aquel puerto. Por ejemplo, para el primer puerto de entrada, que encamina hasta los emisores E1 y E2, hará una reserva de 5, que es el valor obtenido de fusionar todas las peticiones de recursos para alguno de estos emisores.

Figura 12. Ejemplo de reserva en un encaminador con estilo de filtro explícito compartido, SE



Es necesario que la información de la reserva almacenada en los nodos para cada flujo se refresque periódicamente. Este hecho se conoce con el nombre de estado tipo *soft*, a diferencia del estado tipo *hard* de las tablas de encaminamiento que utilizan otros protocolos con circuitos virtuales, como son ATM o *frame relay*, en los que toda la información se mantiene hasta ser borrada. El intervalo de refresco es de 30 segundos por defecto. Este refresco permite la adaptación dinámica a posibles cambios de flujo.

### 3.2.3. Características de RSVP

Finalmente, enumeraremos, a modo de resumen, las principales características que especifica el RFC 2205 para el protocolo RSVP:

- Realiza reservas tanto por unidifusión como por multidifusión basándose en los requerimientos individuales de cada miembro del grupo multidifusión y adaptándose a cualquier cambio.
- Las reservas son por flujos unidireccionales.
- El receptor del flujo de datos es el que realiza y mantiene la reserva de recursos.
- Las reservas en los encaminadores no son permanentes y son responsabilidad de los usuarios finales.
- Especifica el mecanismo de reserva de los recursos por parte de los grupos de multidifusión en los encaminadores intermedios.
- Como las reservas son independientes del protocolo RSVP se pueden utilizar encaminadores que no utilicen el protocolo RSVP y que sigan la técnica del mayor esfuerzo.
- RSVP puede utilizar información de la cabecera de los paquetes de los protocolos IPv4 y IPv6.

#### RFC

El *request for comment* (RFC) es una recomendación publicada y numerada que todavía no ha pasado a ser un estándar de Internet.

### 3.3. Servicios diferenciados (*DiffServ*)

Los servicios diferenciados o *differentiated services* son un modelo de QoS basado en clases y pensado para IP.

Los servicios diferenciados se desarrollaron con el fin de solucionar los principales problemas para realizar la reserva de recursos por flujos como en los servicios integrados:

- Escalabilidad. La reserva de recursos en los encaminadores por parte de RSVP puede suponer una sobrecarga importante en los encaminadores principales.
- Modelos de servicio flexible. *IntServ* define unas clases de servicios predefinidos, pero no nos permite definir servicios cualitativamente diferentes.

Estos problemas han llevado a definir los servicios diferenciados (*DiffServ*) por parte de IETF.

Por una parte, los encaminadores no han de almacenar la información sobre los flujos. Los terminales definen el tipo de servicio que necesitan cada vez que envían un paquete. Eso permite solucionar el problema de la escalabilidad.

Por otra parte, se ha cambiado el servicio por flujo a un servicio por clase. El encaminador encamina el paquete en función de la clase de servicio que se define en el paquete y no del flujo. Se pueden definir diferentes tipos de clases en función de las necesidades de las aplicaciones. Eso resuelve el problema de la limitación del tipo de servicio.

### 3.3.1. Arquitectura

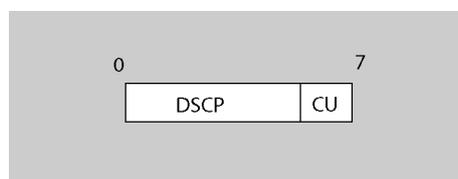
La arquitectura *DiffServ* consta de dos conjuntos de elementos funcionales:

- Acondicionamiento de tráfico. El origen del tráfico marca los paquetes que genera indicando la clase de tráfico a la cual pertenecen. Cada clase de paquetes recibirá un determinado servicio por parte de la red.
- Comportamiento por salto. Es el mecanismo que utiliza el encaminador con el fin de enviar un paquete recibido según la clase a la que está asociado el paquete. Está basado en la clase de tráfico a la cual pertenecen los paquetes. Si dos paquetes de flujos diferentes de la misma clase de tráfico precisan enviarse por el mismo enlace, el encaminador no los distingue y los envía conjuntamente.

En *DiffServ* cada paquete contiene un campo, llamado DS, para indicar la clase de tráfico, y se encuentra o bien en el campo *tipo de servicio* (TOS) de la cabecera de IPv 4, o bien en el campo *clase de tráfico* de la cabecera de IPv 6. El contenido de este campo lo fijan los extremos de entrada en la red (por los terminales o por el primer encaminador). Este campo DS está dividido en dos subcampos:

- DSCP (punto de código de servicio diferenciado). Es el código DS, que son 6 bits que nos permiten definir hasta 64 clases de tráfico diferentes. Definen el comportamiento por salto (PHB).
- CU. Es un campo de 2 bits que no se utiliza.

Figura 13. Formato del campo DS utilizado para definir el comportamiento por salto en los encaminadores intermedios



### 3.3.2. Acondicionamiento del tráfico

El acondicionamiento del tráfico se realiza en el extremo de la red, en el primer encaminador *DiffServ* por el cual pasa el tráfico y que está directamente conectado con el emisor. Es en los encaminadores de los extremos donde está la mayor complejidad a la hora de marcar los paquetes.

La función de acondicionamiento de tráfico consta de cinco elementos:

1) **Clasificador.** Al llegar un paquete a un encaminador, lo primero que éste hace es clasificarlo. Esta clasificación se basa en la información de la cabecera: código DS, dirección origen, dirección destino, puertos, etc.

2) **Contador.** Comprueba si el flujo de entrada coincide con el perfil de tráfico negociado. Determina si una clase de flujo de paquetes determinada cumple un determinado nivel de servicio garantizado por aquella clase o se excede.

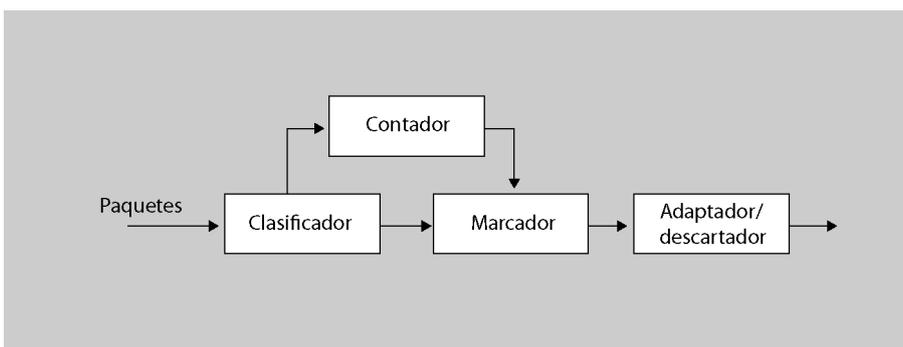
3) **Marcador.** Puede remarcar un paquete en caso de que exceda el perfil. Por ejemplo, si para una clase de servicio se garantiza un determinado rendimiento, cualquier paquete que lo sobrepase se puede remarcar para ser tratado con el servicio de mejor esfuerzo.

4) **Adaptador.** Utiliza la información proveniente del contador para adaptar o ajustar el tráfico si se excede el perfil negociado.

5) **Descartador.** Descarta los paquetes si el flujo de paquetes excede de manera importante el perfil negociado.

Podemos ver la relación que existe entre los diferentes elementos en la figura 14. Cuando llega un flujo de paquetes se clasifica y, posteriormente, el contador mide el volumen del flujo para verificar si cumple el perfil contratado. Los paquetes que sobrepasen el perfil se remarcarán y, finalmente, el adaptador ajustará el tráfico según la información del contador y descartará paquetes en caso de necesidad.

Figura 14. Elementos de un acondicionador de tráfico



### 3.3.3. Comportamiento por salto (PHB)

El modelo *DiffServ* define comportamientos por salto (PHB, *per-hop behaviour*) para cada encaminador que recibe un paquete. El comportamiento por salto permite obtener diferentes clases de tráfico y define diferentes rendimientos observables externamente entre clases. No indica los medios que se deben utilizar para no obtener el rendimiento, sino la medida de este rendimiento.

Actualmente se han definido dos PHB asociados a dos servicios diferenciados específicos:

- PHB de reenvío urgente (EF, *expedited forwarding*). Es un mecanismo que permite una calidad de servicio superior, con baja tasa de pérdidas, baja latencia, baja fluctuación del retardo y ancho de banda asegurado. Por los extremos es parecido a una conexión punto a punto, o a una conexión virtual entre el origen y el destino. Para que se pueda producir este servicio y no haya grandes retardos, ni fluctuaciones, ni pérdidas en las colas de los encaminadores hay que sobredimensionarlos aislando de alguna manera las diferentes clases de tráfico. Se debe asegurar una determinada tasa mínima con los recursos disponibles en cada encaminador independientemente del tráfico producido por otras clases que lleguen a este encaminador. Los encaminadores intermedios han de tener una tasa mínima de salida superior a la tasa máxima de entrada que han fijado los encaminadores de los extremos para que no se produzcan retardos ni pérdidas.
- PHB de reenvío asegurado (AF, *assured forwarding*). Debe permitir un servicio mejor que el de mejor esfuerzo sin necesidad de hacer una reserva de recursos en la red ni una diferenciación entre los flujos de los diferentes usuarios. Es un sistema más complejo que el reenvío urgente. En este caso la entrega del paquete es muy segura, siempre que el tráfico no sobrepase el perfil de tráfico del encaminador.

El mecanismo consiste en hacer que el tráfico se divida en cuatro clases de tráfico con un perfil diferenciado (ancho de banda, *buffers*); en los encaminadores de los extremos se marcan los paquetes de los usuarios en función de si cumplen el perfil de su clase de tráfico en tres etiquetas de preferencia por ser descartado en caso de congestión. En caso de congestión, el encaminador sólo deberá mirar en cada paquete esta preferencia para ser descartado, y proceder a descartarlos según el orden indicado en las etiquetas.

Conceptualmente, este método es muy simple. Los encaminadores internos realizan muy poco trabajo y muy simple, pueden tratar de manera separada el tráfico de las cuatro clases y proceder, si hace falta, a descartar paquetes para cada clase. En cambio, los encaminadores extremos deben

realizar todo el etiquetado de los paquetes en función del tipo de tráfico que se envíe y del nivel de servicio de las diferentes clases.

AF define clases usando el valor DSCP. AF tiene cuatro clases: desde el AF1x al AF4x ( $x$  toma un valor entre 1 y 3), donde la clase AF4x se considera más importante que la clase AF1x. Dentro de cada clase hay tres niveles de probabilidad de descarte. En función de la política de la red, se pueden configurar los paquetes para un PHB en función de los requisitos de rendimiento, retardo, pérdida, etc.

AF PHB define un método mediante el cual las clases de tráfico tienen diferentes garantías de reenvío.

En la tabla siguiente se muestran los códigos DSCP para especificar la clase AF con la probabilidad de descarte correspondiente. Los bits 0, 1 y 2 definen la clase; los bits 3 y 4 especifican la probabilidad de descarte; el bit 5 está siempre a 0.

Tabla 1. Codificación DSCP para las diferentes clases AF

<b>Probabilidad de descarte</b>	<b>Clase 1</b>	<b>Clase 2</b>	<b>Clase 3</b>	<b>Clase 4</b>
Baja	001010 AF11	010010 AF21	011010 AF31	100010 AF41
Media	001100 AF12	010100 AF22	011100 AF32	100100 AF42
Alta	001110 AF13	010110 AF23	011110 AF33	100110 AF43

## Resumen

Actualmente, Internet se utiliza cada vez más para transmitir aplicaciones multimedia, de transmisión de audio y vídeo en tiempo real. Estas aplicaciones son muy sensibles a los retardos, pero no tanto a pequeñas pérdidas.

La técnica del *best effort* que utiliza Internet no permite solucionar las dificultades de la transmisión multimedia con una cierta calidad; por ello, hemos tratado algunas técnicas para mejorar la calidad en el servicio de las transmisiones multimedia mediante las redes.

Hemos analizado diferentes mecanismos de planificación y control que permiten obtener una QoS. Hemos tratado los diferentes modelos de colas (FIFO, con prioridad, *round robin* o ponderadas) y de mecanismos de control de flujo (cubo agujereado, cubo de testigos).

Posteriormente, hemos pasado a definir nuevos estándares de QoS en Internet. Un primer estándar es *IntSer*, que nos define dos servicios: el servicio de calidad garantizada y el de carga controlada. Con el fin de llegar a ofrecer estos servicios hay que poder hacer una reserva de recursos. El protocolo que permite esta reserva de recursos y la creación del flujo de datos es el RSVP. *IntServ* es un modelo de QoS basado en flujo diseñado por IP; su principal inconveniente es que no es escalable, es decir, que los encaminadores no permiten reservar recursos si el número de flujos va creciendo de manera continuada a causa del incremento del uso de Internet.

Un último estándar es la arquitectura *DiffServ*, que está basada en la clasificación de los paquetes en un número pequeño de clases basado en IP. Los encaminadores se encargan de transmitirlos en función de estas clases mediante el comportamiento por salto (PHB).



## Actividades

1. Describid cuáles son los atributos que nos permiten definir un flujo de datos.
2. Indica las principales características del tráfico elástico e inelástico. Buscad ejemplos de aplicaciones de cada uno de ellos.
3. Apuntad cuáles son las principales diferencias entre las disciplinas de planificación de enlace FIFO y WFQ.
4. Referid cuáles son las cuatro técnicas que permiten mejorar la QoS.
5. Señalad alguno de los problemas asociados al modelo *IntServ* y a RSVP.
6. Buscad cuáles son los posibles valores del campo punto de código de Servicio Diferenciado (DSCP) y a qué clases de paquetes indican.
7. Buscad a través de Internet los RFC 2597 y RFC 2598 que nos describen los dos tipos de comportamientos por salto, PHB, que se pueden asociar a dos servicios diferenciados. Comentad las diferencias principales.

## Ejercicios de autoevaluación

1. El modelo de control de flujo con regulador por testigos (*token-bucket*) es un modelo creado para evitar la congestión antes de que se produzca, y, por lo tanto, limita la cantidad de datos que puede transmitir un determinado flujo. Suponiendo que la capacidad del regulador es de  $B$  bits, la tasa de llegada de testigos es de  $R$  bits/s y la máxima velocidad de transferencia de salida, tal y como se muestra en la figura 7. Obtened:
  - a) La longitud máxima de la ráfaga que soporta al modelo,  $S$ .
  - b) El tiempo máximo necesario para transmitirla,  $T_S$ .
  - c) Calculad los parámetros anteriores en el caso de que  $B = 250$  kbits/s,  $R = 2$  Mbits/s y  $M = 25$  Mbits/s.
2. Haced una tabla comparativa entre los dos modelos de QoS: *IntServ* y *DiffServ*.

## Solucionario

1. Los testigos se acumulan en la pila del regulador a una velocidad  $R$  hasta una cantidad de  $B$  testigos. Los datos llegan al regulador y si encuentran testigo lo toman y se transmiten a una velocidad  $M$ . Si no hay testigos estos datos se descartan.

a) El tiempo que tardan en acabarse los testigos del regulador, suponiendo que inicialmente está lleno con  $B$  testigos, es  $B/M$ .

Los nuevos testigos que llegan en el tiempo anterior es  $(B/M)R$ .

La cantidad máxima de testigos que se pueden utilizar para regular la transmisión, que es igual a la longitud máxima de la ráfaga que se puede transmitir, es:

$$S = B + \frac{B}{M}R$$

b) Si transmitimos  $S$  bits a una velocidad de  $M$  bits/s, el tiempo que durará su transmisión es:

$$T_s = \frac{S}{M}$$

c) Según los datos del enunciado:

$$S = 250 \text{ Kbits} + \frac{250 \text{ Kbits}}{25 \text{ Mbits/s}} \cdot 2 \text{ Mbits/s} = 250 \text{ K} + 20 \text{ K} = 270 \text{ Kbits}$$

$$T_s = \frac{S}{M} = \frac{270 \text{ Kbits}}{25 \text{ Mbits/s}} = 10,8 \text{ ms}$$

2. Las principales características que diferencian al modelo *IntServ* del modelo *DiffServ* se muestran en siguiente tabla.

	<b><i>IntServ</i></b>	<b><i>DiffServ</i></b>
<b>Aplicación del servicio</b>	Por flujo individual	Conjunto de flujos (clases)
<b>Recursos en los encaminadores</b>	Por cada flujo	Por el conjunto de flujos
<b>Clasificación del tráfico</b>	Con diferentes cabeceras	Información campo DS
<b>Control de admisión</b>	Es necesario (para cada flujo nuevo)	Es necesario para poder diferenciar las clases
<b>Protocolo de señalización</b>	Es necesario reservar recursos (RSVP)	No es necesario reservar recursos
<b>Gestión del servicio</b>	<i>End-to-end</i>	Local ( <i>per-hop</i> )
<b>Ámbito del servicio</b>	Monodifusión y multidifusión	Cualquiera
<b>Escalabilidad</b>	Limitada por el número de flujos	Limitada por el número de clases de servicio
<b>Funcionamiento de la red</b>	Basado en las características de los flujos y de los requerimientos de QoS	Basado en la utilización de clases
<b>Gestión de la red</b>	Semejante a las redes de conmutación de circuitos	Semejante a las redes IP

## Glosario

**calidad de servicio**  $f$  Propiedades de una red que definen el grado de satisfacción de los usuarios en cuanto a su rendimiento.

sigla: QoS

**comportamiento por salto**  $m$  Tratamiento que hacen los encaminadores en los paquetes en lo referente a las especificaciones de los servicios diferenciados.

sigla: PHB

**cola FIFO** *f* Modelo de cola para la planificación de enlace que consiste en hacer que el primer elemento que entra sea el primero en salir.

**DiffServ** Véase servicios diferenciados

**encaminador** *m* Dispositivo de interconexión que conecta varios enlaces utilizando los protocolos a nivel 3.

**interconexión de redes** *f* Comunicación de dos dispositivos mediante varios tipos de redes.

**Internet** *f* Conjunto de redes privadas y públicas conectadas mediante encaminadores basadas en el protocolo TCP/IP.

**IntServ** Véase servicios integrados

**multidifusión** *f* Transmisión que llega a diferentes destinos simultáneamente.

**paquete** *m* Grupo de bits que llevan datos e información de control a nivel 3 del OSI.

**PHB** Véase comportamiento por salto

**protocolo de reserva de recursos** *m* Protocolo de reserva de recursos que se utiliza en los servicios integrados.  
sigla: RSVP

**protocolo IP** *m* Protocolo de Internet que proporciona un servicio no orientado a conexión mediante la interconexión de redes.

**QoS** Véase calidad de servicio

**RSVP** *m* Véase protocolo de reserva de recursos.

**servicios diferenciados** *m* Grupo de servicios en que el tráfico se diferencia en clases, cada una de ellas con unas necesidades diferentes.  
en *DiffServ*

**servicios integrados** *m* Servicios que se ofrecen a los flujos de información a partir de la reserva previa de los recursos para dar una QoS determinada.  
en *IntServ*

**sistema de colas equitativo ponderado** *m* Disciplina de colas en que se da salida a un elemento de cada cola de manera rotativa y al mismo tiempo ponderada con unos pesos.  
sigla: WFQ

**WFQ** Véase sistema de colas equitativo ponderado

## Bibliografía

**Forouzan, B. A.** (2007). *Transmisión de datos y redes de comunicaciones* (4.<sup>a</sup> ed.). Madrid: McGraw-Hill.

**Kurose, J. F.; Ross, K. W.** (2004). *Redes de computadores. Un enfoque descendente basado en Internet* (2.<sup>a</sup> ed.). Madrid: Pearson Educación.

**Stallings, W.** (2004). *Comunicaciones y redes de computadores* (7.<sup>a</sup> ed.). Madrid: Pearson Educación.

**Stallings, W.** (2004). *Redes e Internet de alta velocidad. Rendimiento y calidad de servicio* (2.<sup>a</sup> ed.). Madrid: Pearson/Prentice Hall.

**Stallings, W.** (2014). *Data and Computer Communication* (10.<sup>a</sup> ed.). Pearson Education.

