CAPÍTULO 7

CONTROL DE CONGESTIÓN Y DEL TRÁFICO

7.1. Control de congestión
7.2. Calidad de servicio y control del tráfico
7.3. QoS garantizada

Una vez estudiados los problemas de encaminamiento e interconexión de redes, a continuación se abordan otros de los aspectos fundamentales en el diseño de redes. Concretamente, centrados en el contexto de las redes de conmutación de paquetes, el presente capítulo tiene un doble objetivo, en primer lugar se estudia el problema del control de congestión y posteriormente se incluye una introducción al control del tráfico, estudiando el problema de la provisión de calidad de servicio.

7.1. Control de congestión

7.1.1. Causas y efectos de la congestión

Por lo general cualquier red, independientemente de su naturaleza, se despliega con unos recursos limitados. En particular, para redes de conmutación de paquetes (estudiadas en el Capítulo 6), estas limitaciones se concretan en líneas de transmisión con una capacidad finita y en *routers* con capacidad de almacenamiento igualmente finita (*buffers* limitados).

Por cuestiones de rentabilidad y optimización del uso de las infraestructuras de red, el número de usuarios o fuentes de información puede potencialmente implicar una carga ofrecida (o tráfico generado) que temporalmente implique una demanda superior a lo que la red puede cursar.

De este modo, el fenómeno de la congestión aparece cuando el tráfico generado es superior a los recursos disponibles. De hecho, el usuario (o las aplicaciones) ante tales circunstancias percibirán una respuesta variable de la red, es decir un retardo o un ancho de banda efectivo variables. En

general, cuando la red sufre episodios de congestión, el usuario percibe un servicio con calidad inferior a la esperada, lo que se manifiesta en mayores retardos —debido a mayores tiempos de espera en los *buffers* (colas) de los *routers*— e incluso en pérdidas de paquetes —dependiendo de la severidad del episodio— debido al desbordamiento (*overflow*) de las colas de los mismos.

Para evitar la congestión se pueden adoptar diferentes estrategias paliativas. Una de ellas consiste en sobredimensionar los recursos, es decir, tanto la capacidad de las líneas como la capacidad de almacenamiento (memoria) de los *routers*. La adopción de esta estrategia está justificada por las siguientes consideraciones:

1. *Precio:* téngase en cuenta que el coste de la memoria (y especialmente del ancho de banda) es cada vez menor, lo que implica que cada vez sea menos costoso sobredimensionar los recursos.
2. *Facilidad en la gestión:* una red sobredimensionada es más sencilla de gestionar. Por contra, una red ajustada en recursos implica la necesidad de herramientas eficaces de monitorización y control que supervisen la red, para en su caso disparar las alarmas y adoptar las técnicas reactivas que se precisen. Además, una red sobredimensionada no necesita personal especialmente formado para su gestión y control, lo cual siempre implica un coste menor.
3. *Estrategia:* el operador que sobredimensiona recursos estará posicionado estratégicamente para atender potenciales crecimientos de la demanda por la aparición de nuevas aplicaciones o necesidades de sus usuarios.

En este sentido, es significativa la evolución histórica que ha seguido el despliegue de la red de conmutación de paquetes dominante: Internet. Su evolución se ha caracterizado por una oscilación pendular (Figura 7.1), con sobredimensionamientos alternativos entre la red troncal y la de acceso. Cuando el usuario accedía con tecnologías de modulación básica sobre el canal telefónico (Red Telefónica Básica, RTB, o Red Telefónica Conmutada, RTC), la red troncal estaba sobredimensionada adoptando tecnologías de retransmisión de tramas (*frame relay*). A partir de esta situación, la red de

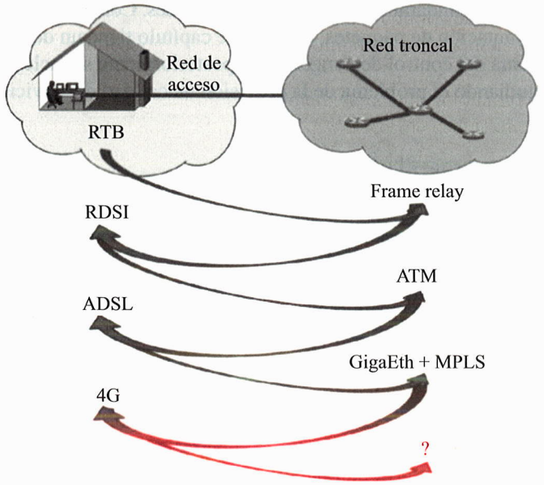


Figura 7.1. Evolución temporal del sobredimensionado en Internet.

acceso evolucionó y se desplegó la Red de Servicios Integrados (RDSI), lo que significó un sobredimensionamiento del acceso. La siguiente evolución implicó el sobredimensionamiento de la red troncal con el despliegue de la tecnología ATM. Posteriormente, se sobredimensionó nuevamente el acceso, siendo en este caso ADSL la tecnología dominante. La siguiente evolución fue el despliegue de Gigabit Ethernet y MPLS como tecnologías dominantes en la red troncal, mientras que en la actualidad asistimos al nuevo sobredimensionado de la red de acceso con el despliegue de la siguiente generación de redes celulares móviles, la así denominada 4G, junto con el despliegue de la red de acceso de fibra óptica.

Aunque el sobredimensionado es una aproximación plausible por las razones anteriormente esgrimidas, no es la mejor solución desde el punto de vista de ingeniería, que siempre debe tener por objetivo optimizar el uso de los recursos disponibles maximizando el retorno económico obtenido. Sobredimensionar no lo consigue. Además, la evolución histórica (como ya se ha comentado) demuestra que ante un crecimiento en los recursos disponibles, siempre le sigue la aparición de nuevas aplicaciones que generan demandas mayores de los usuarios, lo que tiende sin remedio a dejar exhausto ese crecimiento.

Por otro lado, sobredimensionar indiscriminadamente puede tener un efecto contraproducente —en circunstancias particulares— en el desempeño de la red en su conjunto, como así se puede comprobar en el siguiente ejemplo tomado del manual de Welz (ver referencias bibliográficas al final del presente capítulo).

Sea una topología sencilla como la de la Figura 7.2, en la que se dispone de 2 *routers* unidos por una línea de transmisión (lo que supone un cuello de botella) dimensionado con una capacidad de 300 kbps, mientras que las líneas de acceso tienen una capacidad nominal de 100 kbps, 100 kbps, 100 kbps y 1 Mbps respectivamente, para los usuarios 1, 2, 3 y 4.

Si suponemos que los usuarios 1 y 2 generan tráficos (a la máxima capacidad de su línea de acceso) dirigidos respectivamente a los usuarios 3 y 4, la carga nunca superará la disponibilidad de los recursos, y concretamente los usuarios 3 y 4 recibirían en el tiempo un tráfico con una velocidad igual al tráfico generado (100 kbps).

Sin embargo, si la línea de acceso del usuario 1 se sobredimensiona a 1 Mbps y el usuario 1 hace un uso exhaustivo de esta capacidad, generando tráfico a la capacidad máxima, mientras que el usuario 3 seguirá recibiendo un caudal de 100 kbps (por la limitación impuesta en su línea de acceso), el usuario 4 irá recibiendo paulatinamente una velocidad de transmisión menor, llegando incluso a casi la inanición, ya que el tráfico entre las fuentes 1 y 3 monopolizará los recursos en el *router* de entrada.

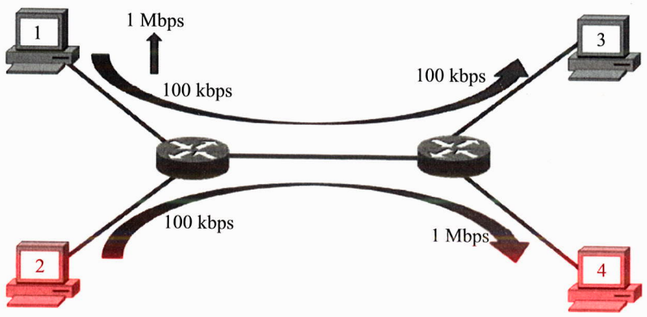


Figura 7.2. Ejemplo de sobredimensionamiento.

La verificación simultánea de estos requisitos no es fácil. A continuación se estudian diferentes familias de planificadores. Téngase en cuenta que el objetivo final es dotar al *router* de un procedimiento que pueda proporcionar QoS para ciertas clases de tráfico frente a otras, para las que se espera un tratamiento no diferenciado y sin garantías.

Ley de conservación en los planificadores conservativos

Antes de adentrarnos en diferentes estrategias de planificación, en este apartado se estudia la ley de conservación enunciada por Kleinrock en 1975, la cual se verifica en toda estrategia de planificación conservativa.

Un planificador es conservativo si y solo si se queda en reposo (es decir, no genera tráfico por la línea de salida) únicamente cuando no hay paquetes esperando en ninguna de sus colas de entrada.

Por lo general, todos los planificadores comerciales son conservativos, ya que aunque las estrategias no conservativas tendrían su justificación para eliminar *jitter* (pues pueden suavizar el patrón del tráfico cursado), no se suelen usar debido a que la mitigación del *jitter* se puede hacer en el destino final (siempre con la penalización de aumentar el retardo).

La ley de conservación coincide con lo que intuitivamente cabe esperar (en estrategias conservativas): reducir el retardo medio para una clase es siempre a costa de aumentar el retardo de otras.

Podemos enunciar más formalmente la ley de conservación como: *la suma de los retardos medios pesada por la utilización media de la línea de salida es una constante, con independencia de la estrategia adoptada por el planificador.*

En particular, sean N flujos de entrada en un planificador, tal que la razón de llegadas para la clase n con $(1 \leq n \leq N)$ es λ_n. Si suponemos que el flujo n es servido con una tasa μ_n (razón de salida), la utilización del enlace de salida por parte de este flujo será $\rho_n = \lambda_n \mu_n$. Entonces, si el tiempo de espera para el flujo n es q_i, de acuerdo con la ley de conservación se cumple que,

$$\sum_{i=1}^{N} \rho_i q_i = \text{cte} \tag{7.15}$$

Esta ley se cumple con independencia del planificador en concreto y viene a decir que es imposible reducir el retardo de todas las clases respecto al retardo que tendrían en un planificador FIFO.

Planificador FIFO

Los planificadores FIFO se caracterizan por gestionar el tráfico con una política que respeta el orden de llegada. Por tanto, son el punto de partida en nuestro estudio ya que corresponden con la estrategia de planificación que se adopta en los *routers* convencionales de conmutación de paquetes que no proporcionan QoS. Se caracterizan por ser no equitativos, ya que son susceptibles de tener bloqueos por parte de usuarios malintencionados, siempre y cuando no haya una supervisión previa.

A pesar de no proporcionar QoS, si un gestor de cola altera las probabilidades de disparo de cada una de las clases (ver algoritmo RED del Apartado 7.3.4) podría proporcionar un cierto tratamiento diferencial por clases de tráfico usando este tipo de planificadores. Para estos planificadores, una vez que un paquete es encolado, el retardo estará acotado por K/C, siendo K el tamaño del *buffer* y C la capacidad de la línea de salida.

Planificador con colas de prioridad

En este caso se trata de un planificador para el que se arbitran una serie de colas diferenciadas, cada una correspondiente a una prioridad distinta (Figura 7.17).

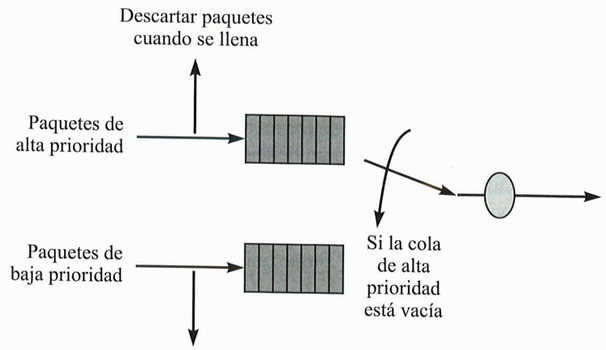


Figura 7.17. Planificador con colas de prioridad.

La política de selección de la cola a servir consiste en cursar tráfico de la cola de máxima prioridad siempre que tenga tráfico en espera. Solo en el caso de que esta cola quede servida, el planificador servirá al tráfico de la cola de menor prioridad. Aunque en la figura se muestra el caso particular de solo 2 colas, el procedimiento se puede generalizar a tantas colas como clases se necesiten, ordenadas por prioridad.

Eligiendo adecuadamente el tamaño de los *buffers* se pueden gestionar distintas probabilidades de pérdidas por clase. Se distinguen dos variantes, dependiendo de si se adopta una estrategia de prioridad con derecho o sin derecho preferente. En este último caso, el paquete que esté siendo servido no se interrumpe al llegar tráfico de mayor prioridad.

En cuanto al retardo, estos planificadores solo pueden garantizar un retardo acotado para la cola de máxima prioridad, ya que para las otras clases no estará garantizado de ninguna manera, pues en este caso (si suponemos un planificador sin derecho preferente) el retardo para un paquete que llega a una cola sin prioridad sería el acumulado de los siguientes términos:

— Tiempo en servir el paquete actualmente en servicio (acotado).
— Tiempo en servir todos los paquetes de mayor prioridad (acotado).
— Tiempo en servir todos los paquetes de mayor prioridad que lleguen mientras se está a la espera de ser servido.

Para el caso de planificadores con derecho preferente habría que sumar a los términos anteriores el tiempo empleado en servir las interrupciones causadas por llegar paquetes de mayor prioridad, mientras se está siendo servido.

En cualquier caso, esta estrategia no es adecuada para la provisión de QoS ya que no garantiza ancho de banda para las colas de menor prioridad. Para estas colas no puede garantizar retardo acotado ni consigue un reparto *max_min* (véase el Apartado 7.1.5). Además, los planificadores con colas de prioridad no permiten diferenciar entre usuarios o clases con la misma prioridad, aunque en este caso un gestor de colas podría gestionar diferentes probabilidades de descarte.

El planificador ideal GPS

Hemos comentado que uno de los requisitos de todo planificador es garantizar la imparcialidad (pesada o no), y en particular es deseable poder realizar un reparto *max_min*.

El planificador GPS («Generalised Processor Sharing») es una estrategia de planificación ideal conservativa que proporciona un reparto *max_min*. Su funcionamiento consiste en asignar cada clase a una cola diferente y servir en cada turno una cantidad infinitesimal de tráfico para cada una de las colas, siempre que la cola tenga tráfico pendiente de ser cursado. En otro caso, saltar a la cola siguiente con un retardo ideal cero y así seguir procediendo de forma cíclica visitando todas las clases. Evidentemente, este es un planificador ideal no realizable, pero tiene como ventaja que en cualquier intervalo finito de tiempo cada cola es visitada al menos una vez. Es más, el GPS consigue de forma inherente un reparto *max_min*, ya que al ir sirviendo de forma cíclica todas las colas, conseguirá satisfacer primero a las de menor demanda, repartiendo el exceso de capacidad de forma alícuota entre las otras colas restantes.

Cabe plantearse cómo sería un GPS en el que quisiéramos dar un peso diferente a cada una de las clases. La respuesta es bien sencilla: tan solo bastaría (idealmente) con servir en cada turno una cantidad (infinitesimal) para cada cola proporcional a su peso. Una posible visualización de su funcionamiento sería servir un número de bits en cada turno igual al peso de la cola en cuestión.

Por razones obvias, el GPS implica un comportamiento ideal sin ningún interés práctico, ya que es imposible servir una cantidad infinitesimal de información por cada cola, o incluso si se sirviera bit a bit, esto implicaría romper con la unidad de información: el paquete. No obstante, sí puede servir como guía en el diseño de cualquier planificador, ya que si se aproxima su comportamiento asintóticamente, estaremos realizando un reparto *max_min* (pesado o no), lo cual es uno de los requisitos u objetivos marcados para el diseño de la estrategia de planificación.

En lo que sigue se van a ir proponiendo diferentes estrategias de planificación realizables que aproximen o emulen el comportamiento del GPS, toda vez que este asegura un reparto *max_min*.

Planificador RR («Round Robin»)

Este planificador consiste en visitar cíclicamente cada una de las colas, y servir en cada turno un paquete de aquellas colas que tengan al menos un paquete pendiente, saltando a la siguiente siempre que se encuentre una cola inactiva o vacía.

Es fácil comprobar que este planificador es equitativo siempre y cuando los paquetes para todas las colas tengan un tamaño igual. Además, por su propia naturaleza protege a unos flujos frente a otros, pues evita los posibles bloqueos que pudieran ocurrir si una clase fuera muy ambiciosa, ya que, a lo sumo el planificador servirá un solo paquete de esta clase en cada ciclo o turno.

El planificador RR aproxima al GPS de forma realizable (es decir, procesando paquete a paquete sin romper la unidad de información) siempre y cuando todos los paquetes tengan igual longitud y las colas tengan igual peso.

Además, sería posible construir un RR pesado (siempre y cuando todos los paquetes tuvieran igual longitud), en cuyo caso se serviría un número de paquetes para cada clase y para cada ciclo proporcional a su peso.

La exigencia de que todos los paquetes sean de igual longitud es demasiado restrictiva, ya que en las redes desplegadas en el mundo real (por ejemplo en Internet) esta exigencia no es verificable. Lo que sí se podría es exigir al menos conocer a priori el tamaño medio de los paquetes para cada una de las clases.

Planificador DRR («Deficit Round Robin»)

Este planificador de baja complejidad computacional es una propuesta que consigue aproximar al GPS (pesado o no) sin exigir la asunción de la hipótesis de conocer a priori el tamaño medio de los paquetes para cada una de las clases a servir.

Su funcionamiento se basa en la definición de dos variables, *quantum* (Q) y *déficit* (D). En cada ciclo el planificador visita a cada una de las colas establecidas y agrega al déficit D de cada cola, inicialmente igual a cero para todas ellas, una cantidad de bits igual a Q. Una cola será elegible para ser servida si, y solo si, el paquete pendiente tiene una longitud menor o igual que el D acumulado para esa cola (incluyendo el Q recientemente acumulado). Si el planificador visita una cola vacía con un déficit no cero, este se pone a valor cero para evitar que siga creciendo.

Como norma general, si Q es igual al mayor tamaño de paquete esperado, se estará asegurando que se sirve un paquete por ciclo de cada cola no vacía.

Para adoptar una estrategia pesada, bastaría con asignar un Q diferente a cada una de las colas, tal que fuera proporcional a su peso.

Familia de planificadores WFQ («Weighted Fair Queueing»)

Esta familia de planificadores tiene como característica principal la aproximación asintótica de forma realizable y real al GPS (es decir, consigue un reparto *max_min*) sin suposiciones de tamaño infinitesimal para paquetes de longitud variable y sin exigir un conocimiento del tamaño medio a priori. Es más, bajo ciertas circunstancias que se explicarán más adelante WFQ puede garantizar requisitos de QoS (ancho de banda y retardo) extremo a extremo para una clase o clases dadas, con independencia del comportamiento de las otras.

Al igual que en los casos anteriores, el funcionamiento de esta familia de planificadores se basa en arbitrar una cola diferente para cada una de las clases a servir. El objetivo fundamental consiste en emular (de forma virtual) al planificador GPS, tal que los paquetes sean servidos como lo haría dicho planificador ideal.

El planificador WFQ asigna a cada paquete recibido una etiqueta (F) denominada *finish number*, definida como el número de ciclos (*rounds*) que el planificador GPS ideal (bit a bit) necesitaría para servir a ese paquete en cuestión. Idealmente la etiqueta F coincidirá con el instante de tiempo en que el último bit del paquete será servido por el GPS virtual. Nótese que F es una función real que depende del instante de tiempo de llegada del paquete, de la historia o llegadas previas al planificador y de la longitud del paquete actual. Una vez que se asigne una etiqueta a un paquete esta no cambiará.

Cada vez que el planificador real tiene que tomar una decisión, simplemente servirá la cola con el paquete con menor *finish number F*. De esta manera, el WFQ sirve los paquetes con el mismo orden con el que el planificador emulado (virtual) GPS los serviría. No hay que confundir el planificador real que va sirviendo paquete a paquete a la capacidad de la línea, con el emulado (virtual) que sirve bit a bit por turnos a las diferentes colas de forma no realizable e irreal.

Más formalmente, sea $R(t)$ una función real del tiempo, *round number*, cuyo valor para el instante de tiempo t coincide con el número de ciclos o *rounds* que el planificador virtual GPS habrá dado hasta el instante t. Nótese que $R(t)$ es una función real que depende de la distribución de las llegadas de los paquetes a las colas. Además, tendrá una evolución temporal con una pendiente que dependerá del número de colas activas. En el límite, si hubiera solo una cola activa, $R(t)$ crecería a la velocidad impuesta por la capacidad de la línea de salida.

Se define una cola como *activa* en el instante t, si la mayor de las etiquetas de cualquiera de los paquetes en esa cola (o del paquete actualmente servido caso de que pertenezca a esa cola) es mayor que el $R(t)$ actual.

Si un paquete k, llega a una cola inactiva i, en el instante de tiempo t, siendo $L(i, k, t)$ la longitud de dicho paquete, la etiqueta asignada (esto es, el número de ciclos necesarios para que el planificador GPS sirviera ese paquete) será

$$F(i, k, t) = R(t) + L(i, k, t) \tag{7.16}$$

Si por el contrario llega a una cola activa, la etiqueta será

$$F(i, k, t) = F(i, k-1, t) + L(i, k, t) \tag{7.17}$$

Obsérvese que F es una función real que depende inversamente del número de conexiones activas. Llegados a este punto nos preguntamos ¿qué modificación habría que hacer para obtener un planificador WFQ pesado? Pues tan solo habría que etiquetar cada paquete recibido con el número de *rounds* (o ciclos) que un GPS pesado tendría que realizar para servirlo. Para ello, con tener en cuenta simplemente que un GPS pesado sirve en cada ciclo un número de bits proporcional a su peso, se necesitarían por tanto $L(i, k, t)/w_i$ turnos para servir el paquete, siendo w_i el peso de la cola i. Nótese que $R(t)$ en el caso del planificador pesado crecerá en el tiempo con una pendiente proporcional a $1/\Sigma_j w_j$ (siendo j el índice que recorre todas las colas activas en ese instante).

Por tanto, de forma más compacta, considerando las ecuaciones (7.16) y (7.17), podemos decir que para un WFQ las etiquetas se calculan según la siguiente expresión:

$$F(i, k, t) = \text{máx}\,\{F(i, k-1, t), R(t)\} + \frac{L(i, k, t)}{w_i} \tag{7.18}$$

Por último, es interesante notar que una de las debilidades de este tipo de planificadores es su alto coste computacional, pues el cálculo y la gestión de $R(t)$ (realizada cada vez que llega o se sirve un paquete) es computacionalmente muy demandante. Alternativamente, en la literatura se han propuesto otras variantes de la estrategia anterior como por ejemplo el planificador SCFQ («Self-Clocked Fair Queuing») y el WF^2Q («Worst-case Fair Weighted Fair Queuing»).

7.3. QoS garantizada

Llegados a este punto, en este apartado concluimos enunciando dos resultados fundamentales para la provisión de QoS en redes de paquetes.

El primero de ellos está relacionado con la garantía en la provisión de ancho de banda. En concreto, dado un planificador WFQ operando sobre una línea de salida caracterizada por una capacidad de R bps, este planificador asegura que el tráfico o clase asociada a la cola i con peso w_i para $i = 1, ..., N$ recibirá en las condiciones de peor caso (es decir, cuando las otras colas están no vacías) una tasa, T, dada por

$$T = \left(\frac{w_i}{W}\right) \cdot R \text{ [bps]} \tag{7.19}$$

siendo

$$W = \sum_{i=1}^{N} w_i \tag{7.20}$$

El segundo de los resultados está relacionado con la acotación del retardo extremo a extremo para una clase de tráfico dada. En concreto, si suponemos que:

— el tráfico está conformado mediante un cubo de permisos, caracterizado por una velocidad media igual a r Bps (velocidad con la que son generados los permisos) y un tamaño de b Bytes (correspondiente a la máxima duración de la ráfaga permitida para ese tráfico),

— R_j es la capacidad de la línea de salida para el *router j*, con $j = \{1, ..., H\}$, y

— si cada *router* adopta un planificador WFQ para ese tráfico conformado, con el peso w_i tal que para esa clase garantice una tasa $T \geq r$,

en ese caso, de acuerdo con el teorema de Parekh y Gallager establecido en 1992, el retardo extremo a extremo (sin incluir el tiempo de propagación), estará acotado para esa clase de tráfico y bajo esas condiciones por

$$D \leqslant \frac{b}{T} + \frac{(H-1)m}{T} + \sum_{j=1}^{H} \frac{M}{R_j} \tag{7.21}$$

donde m es el tamaño máximo del paquete para esa clase, M el tamaño máximo del paquete para cualquier paquete en la red, y H, como se indicó previamente, el número de *routers* (dotados con un planificador WFQ) para alcanzar el destino.

Este resultado es fundamental porque impone unas condiciones que permiten garantizar un retardo extremo a extremo acotado, con independencia de la naturaleza y características de los otros tráficos en la red.

Los tres términos de la expresión anterior pueden tener una interpretación más o menos intuitiva. Obsérvese que el primero está relacionado con el tiempo necesario para que la ráfaga de mayor duración posible (b) —para esa clase— sea servida con una tasa T. El segundo término es un factor de corrección que modela el hecho de que el paquete (salto a salto) llegue justo después de que el planificador virtual GPS hubiese servido la cola. Finalmente, el último término corresponde a que el paquete llegue a un planificador que esté ocupado sirviendo otra cola, luego en las condiciones de peor caso implica el tiempo necesario para servir al paquete de longitud mayor en la red (M) servido a la máxima capacidad de la línea de salida en el *router j*, dada por R_j, para los H planificadores en la ruta.

RESUMEN

Para concluir este capítulo y antes proponer los ejercicios recomendados, resumimos los principales objetivos e ideas fundamentales que han conformado su elaboración.

El capítulo tiene dos objetivos principales, por un lado estudiar el problema de la congestión y por otro estudiar el control del tráfico para la provisión de calidad de servicio.

Para ello, tras la identificación de las principales causas (relacionadas con la escasez de recursos frente al tráfico generado) y efectos (mayores retardos y pérdidas de paquetes) de la congestión, se justifica por qué el sobredimensionamiento no es una estrategia aconsejable para evitar la congestión. Se incide en la idea de que con un algoritmo de control de congestión no siempre es posible maximizar el rendimiento sin menoscabar la equidad. Para ello se estudia el escenario del *parking lot* como ejemplo significativo.

Se estudian diferentes variantes de esquemas que se pueden adoptar para el control de la congestión, comentando las ventajas e inconvenientes de las estrategias en bucle cerrado frente a las de bucle abierto, las de la notificación implícita frente a la explícita, así como los esquemas que controlan la tasa frente a los basados en ventana, haciendo hincapié para estos último en la importancia del producto $BW \times RTT$.

Adoptando una serie de simplificaciones se caracterizan y estudian las cuatro posibles variantes o tipos de respuesta para un sistema que adopte un ajuste lineal de la velocidad, concluyendo que, por cuestiones de eficacia, imparcialidad y estabilidad, es preferible una respuesta AIMD frente a otras.

En relación a la imparcialidad se definen las métricas más significativas. En particular se definen el índice de Jain, la métrica del producto, el criterio *max_min* y la equidad proporcional.

Seguidamente se estudia la calidad de servicio. Para ello se identifican a grandes rasgos los bloques funcionales constitutivos de una arquitectura genérica de un *router* de conmutación de paquetes que no provee QoS. A partir de este se propone una arquitectura que sí incluye los bloques funcionales necesarios para la provisión de QoS. En particular, se identifica y estudia el primer bloque necesario para hacer un tratamiento diferenciado de los distintos tráficos mediante el uso de un clasificador. A continuación se explica el bloque del control de admisión, supervisión y marcado. En este caso se explican las metáforas del cubo con escape y del cubo de permisos, resaltando las diferencias en cuanto a su uso para el control de admisión frente a su uso como conformadores de tráfico.

Tras estudiar los procedimientos de gestión activa de colas, como último componente arquitectónico —y no menos importante— de un *router* con QoS se estudian los planificadores. Se adopta una aproximación de complejidad ascendente, presentando los planificadores FIFO, con colas de prioridad, GPS, RR, DRR y WFQ.

El capítulo concluye enunciando los dos resultados fundamentales para la provisión de QoS en relación a la familia de planificadores WFQ: la posibilidad de garantizar un ancho de banda mínimo y la posibilidad (bajo ciertas hipótesis) de poder acotar paquete a paquete el retardo extremo a extremo, con independencia de la naturaleza de los otros tráficos presentes en la red.

EJERCICIOS

1. Considerando la red de la Figura 7.4, correspondiente al escenario *parking lot*, demuestre que un reparto equitativo no consigue la máxima eficacia. Suponiendo un reparto equitativo de la capacidad a la salida de cada *router*, ¿cuál sería el *throughput* total si $n_i = 1$ para ($i = 1, ..., I$)? Compruebe que para el caso $I = 2$, $n_0 = n_1 = 1$, $n_2 = 9$, si x_0 es tal que maximiza el *throughput*, asignar recursos equitativamente a x_i no es una buena solución dado que otros repartos para x_i harían que algunas fuentes puedan obtener más tasa sin que se penalice a otras; es decir, hay mejores repartos que el equitativo.

2. En una red en la que se adopta un protocolo de la capa 2 salto a salto para el control del flujo, ¿sería necesario adoptar un protocolo de control de congestión?

3. Suponiendo una red no síncrona con dos orígenes o fuentes de información como la mostrada en la Figura 7.6 (caracterizada por un comportamiento binario frente a episodios de congestión dado por la expresión (7.3)), explique mediante un gráfico cómo sería la evolución de la velocidad para una respuesta AIMD en la que una de las fuentes es notificada por parte de la red con un frecuencia doble que la otra.

4. Dada una línea de transmisión de capacidad 300 Mbps y 10 fuentes de información con demandas o velocidades de transmisión iguales a {23, 24, 27, 30, 35, 38, 45, 50, 55 y 64}Mbps. ¿Cómo sería un reparto que verifique el criterio *max_min*? ¿Cómo sería el reparto *max_min* si cada una de las fuentes tuviera pesos dados respectivamente por {2,5, 4, 0,5, 1, 4, 1, 2, 0,5, 1, 2,5}?

5. Muestre cómo sería el diagrama de flujo y pseudocódigo de un cubo con escape.

6. Suponga un algoritmo de supervisión con un cubo con escape donde todos los paquetes tienen igual tamaño. Si la velocidad de pico es 1.000 paquetes por segundo y se supone que el cubo tiene una capacidad máxima de 2 paquetes, determine si las siguientes llegadas de paquetes en los instantes {10,0, 11,0, 11,2, 11,5, 12,0} ms son cumplidores o no.

7. Suponga un algoritmo de supervisión con un cubo de testigos. Si se quiere supervisar una velocidad media igual a 100 paquetes por segundo y se admiten hasta ráfagas de 10 paquetes consecutivos, determine si la secuencia de llegadas en los instantes {100,0, 100,1, 100,2, 100,3, 100,4, 100,5, 110,0, 120,1, 120,2, 120,3, 120,4, 120,5, 120,6} ms cumplen con ese perfil de tráfico.

8. Si se utilizara el cubo de testigos anterior para conformar tráfico, ¿cuál sería el número máximo de paquetes generados en un intervalo de 10 segundos?

9. Un ordenador tiene que transmitir 19,5 MB generados con una velocidad de ráfaga de 6 Mbps. Si la velocidad de trasmisión máxima a través de los *routers* de la red es 4 Mbps y el tráfico generado es conformado con un cubo con escape, ¿qué capacidad debe tener el cubo para que no se pierda información alguna?

10. Sea un planificador con razón de salida constante para la clase n dada por $\mu_n = 0{,}1$ ms/paq. Sean dos flujos $f1$ y $f2$ con tasas de salida $\lambda_1 = \lambda_2 = 10$ paq/s y $q_1 = q_2 = 0{,}1$ ms. Si se cambian las característica del flujo 1, tal que la razón entrada aumenta a $\lambda_1 = 15$ paq/s, ¿qué implicaciones tendrá en el flujo 2?

11. Sean 3 clases de tráfico {A, B, C}, con pesos respectivos {0,5, 0,75, 1} y con longitudes medias por paquete de {50, 500, 1.500} bytes. Explique cómo estas clases serían servidas por un planificador RR pesado.

12. Dados 3 flujos A, B, C con paquetes de tamaños {600, 1.000 y 1.400} bytes, explique cómo serían servidos por un planificador DRR si el *quantum* fuera 1.000 bytes. ¿Y si se asignaran pesos, respectivamente, de 3, 2 y 1?

13. En el instante 0 llegan 3 paquetes con longitudes 200, 100 y 300 bits a las líneas A, B y C de un planificador FQ. En el instante $t = 2$ s llega 1 paquete de longitud 300 a la línea A, y en el instante $t = 3$ s llega un paquete de 100 bits a la B. Si la velocidad de transmisión de la línea es 150 bps:

    a) ¿En qué instante de tiempo serán servidos los paquetes?
    b) ¿Cuál es el «*round number*» para cada paquete cuando es servido?
    c) Si un paquete de longitud 20 bits llega a la línea C en el instante $t = 4$ s, ¿cuál será su «*finish number*»? Comente el resultado.
    d) Repita el ejercicio si las líneas A, B y C tuvieran, respectivamente, pesos 2, 5 y 7.

BIBLIOGRAFÍA

M. Welz: «Network Congestion Control. Managing Internet Traffic». John Wiley & Sons, 2005. ISBN: 978-0-470-02528-4.

G. Armitage , Quality of Service in IP Networks, Macmillan Technical Pub., 2000. ISBN: 1578701899.

S. Keshav: «*An engineering approach to computer networking*». Addison-Wesley. 1997 ISBN: 0201634422.

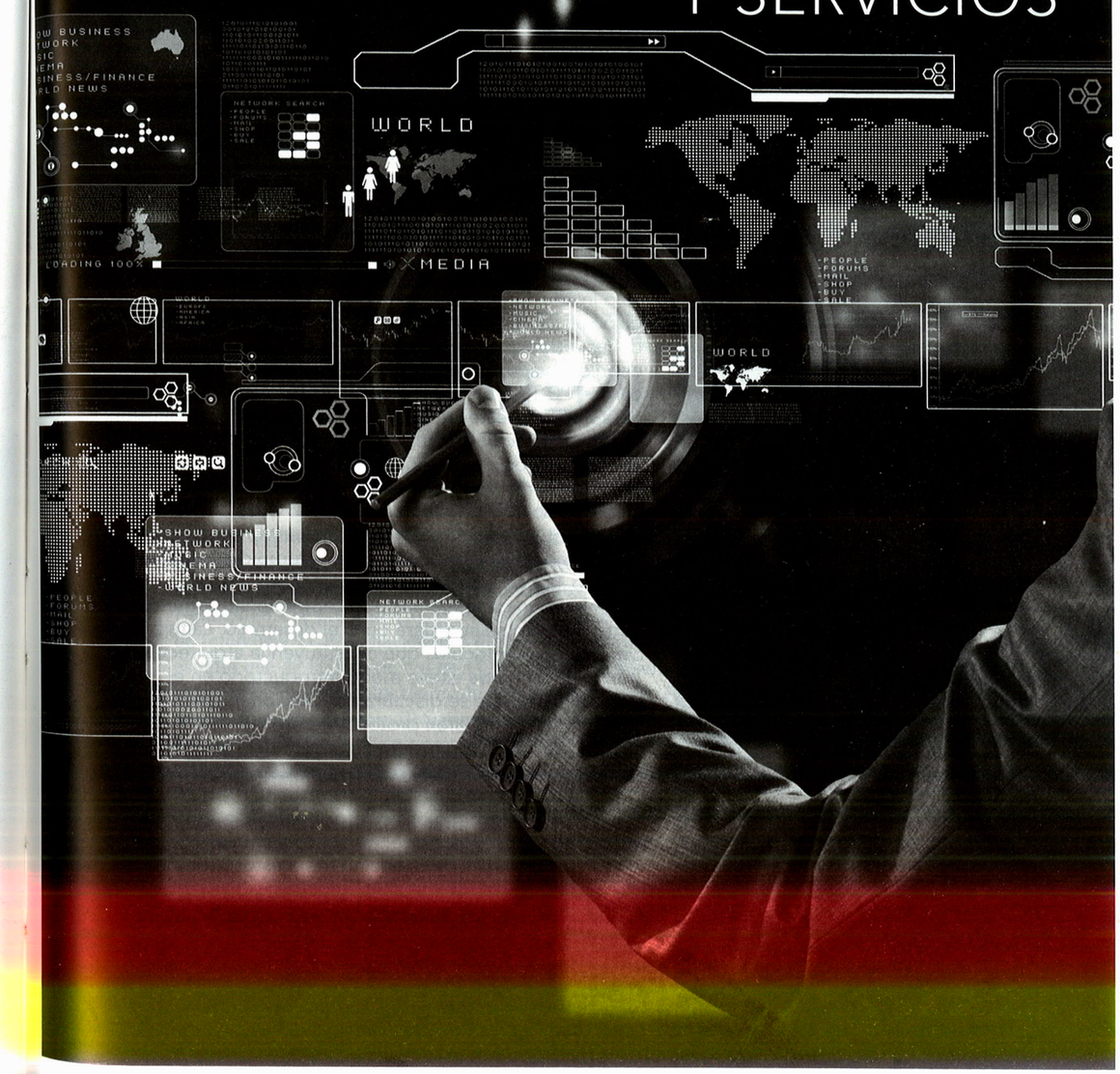
S. Jha, M. Hassan: Engineering Internet QoS, Artech House 2002. ISBN 1580535666.

W. Stallings, Redes e Internet de alta velocidad. Rendimiento y calidad de servicio, Pearson, 2004. ISBN: 842053921X.

J. F. Kurose, K. W. Ross, «Computer Networking. A top-Down Approach» Prentice Hall, Ed. 6.ª 2012 ISBN: 0132856204.

J. Postel. RFC 792: «Internet Control Message Protocol», 1981.

J. Heinanen, R. Guerin. RFC 2697: «A Single Rate Three Color Marker», 1999.
J. Heinanen, R. Guerin. RFC 2698: «A Two Rate Three Color Marker», 1999.
B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang. RFC 2309: «Recommendations on Queue Management and Congestion Avoidance in the Internet», 1998.

PARTE II

INTERNET, ARQUITECTURA TCP/IP, PROTOCOLOS Y SERVICIOS