

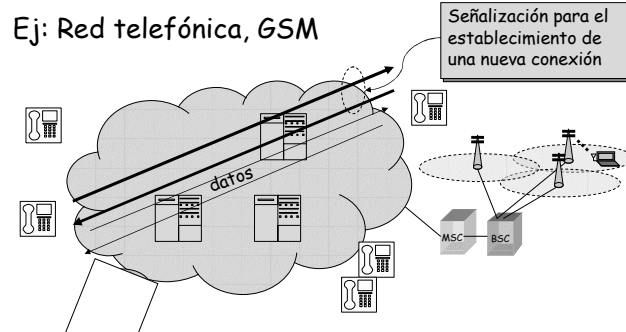
Mecanismos para el soporte a QoS en redes de paquetes

Jorge García Vidal

PIAM, Jorge García Vidal 2004

Redes con soporte a QoS

- Ej: Red telefónica, GSM

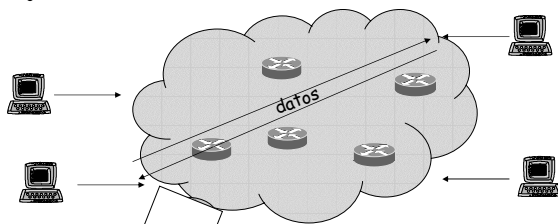


Durante el establecimiento de la conexión, la red reserva los recursos necesarios para atender la nueva llamada (ej: busca slots temporales libres en tramas TDM). Las conexiones no necesitan de una descripción adicional: son siempre de 64 Kbps

PIAM, Jorge García Vidal 2004

Redes con soporte a QoS

- Ej: Internet, GPRS

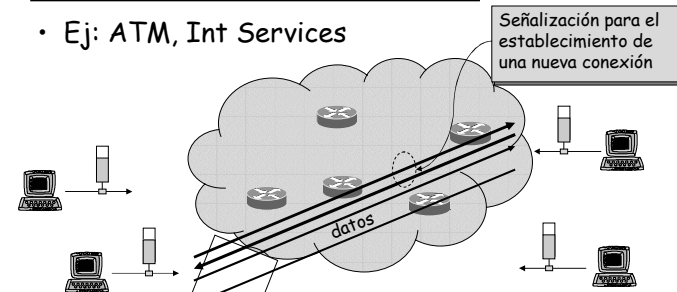


La red NO determina si hay recursos suficientes como para atender la nueva llamada. Como consecuencia, no se puede garantizar ni un retardo acotado ni una tasa de errores por debajo de un cierto umbral. La red es NO orientada a la conexión.

PIAM, Jorge García Vidal 2004

Redes con soporte a QoS

- Ej: ATM, Int Services

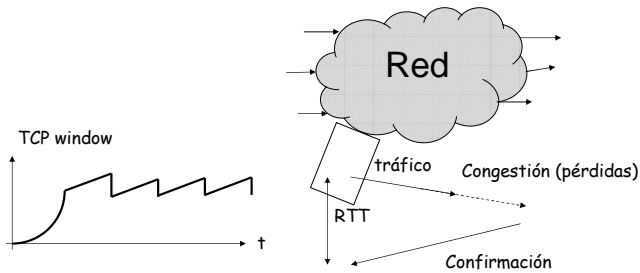


Durante el establecimiento de la conexión, la red determina si hay recursos suficientes como para atender la nueva llamada (ej: la utilización de los enlaces está por debajo de un cierto umbral). Las conexiones necesitan dar una descripción adicional. Ej: Conexión a tiempo real a 64 Kbps. Conexión de datos a 2 Mbps, etc... La red controla que el tráfico generado corresponde al que ha sido declarado durante el establecimiento de la conexión.

PIAM, Jorge García Vidal 2004

Control de congestión en TCP

- Para el tráfico "best effort" podemos usar un control de congestión reactivo: TCP



PIAM, Jorge García Vidal 2004

Control de congestión para tráfico en tiempo real

- Para tráfico en tiempo real, el control de TCP no es adecuado.
- Podemos controlar la congestión evitando que las fuentes inyecten demasiado tráfico en la red.

PIAM, Jorge García Vidal 2004

Redes con soporte a QoS

- La conexión pregunta si hay recursos suficientes, dando una descripción de sus características (CAC, Control de Admisión de Comunicaciones)
- Si la nueva conexión es admitida, la red controla que los parámetros del tráfico generado corresponden a la descripción dada (UPC, Función de Policía o de Control de Parámetros de Usuario)
- Por lo tanto, ya no tenemos una red puramente No Orientada a la Conexión...

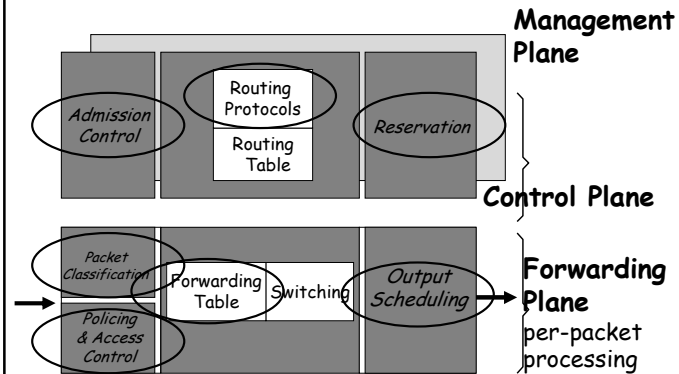
PIAM, Jorge García Vidal 2004

Redes con soporte a QoS

- Una vez el nuevo flujo es admitido, los routers deben establecer una disciplina de servicio que garantice el diferente uso de recursos para cada flujo. Por lo tanto los flujos deben ser clasificados en cada router.
- El algoritmo de encaminamiento puede también ser utilizado para escoger el mejor camino, de forma que se cumplan las restricciones impuestas por la nueva conexión

PIAM, Jorge García Vidal 2004

Redes con soporte a QoS



PIAM, Jorge Garcia Vidal 2004

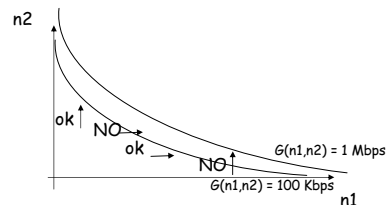
Control de admisión de conexiones: CAC

- Supongamos un link de capacidad C , y $N-1$ flujos ya establecidos. Un nuevo flujo será admitido si:
 - **Suma:** $\sum_{i=1, N-1} r_i + r_N < C$, en donde r_i es el ancho de banda de pico declarado por cada fuente.
 - **Medidas:** Si $\sum_{i=1, N-1} m_i + r_N < C$, en donde m_i es el ancho de banda que medimos como realmente utilizado por cada fuente.
 - **Región de aceptación:** Clasificamos los flujos en s clases. Los flujos de una clase comparten las mismas características. Supongamos que hay n_i flujos de la clase i y que la nueva conexión es de clase j . La nueva conexión es aceptada si $G(n_1, n_2, \dots, n_j+1, \dots, n_s) < C$. $G()$ define la región de aceptación.
- Existen otros métodos...

PIAM, Jorge Garcia Vidal 2004

CAC

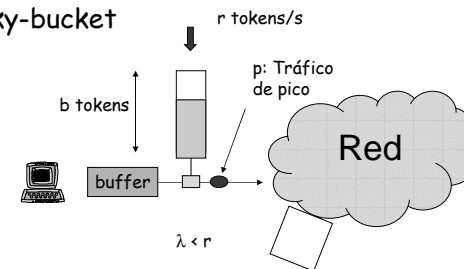
- Ejemplo:
 - $C = 10$ Mbps
 - n_1 : conexiones audio
 - n_2 : conexiones vídeo



PIAM, Jorge Garcia Vidal 2004

Función de policía: Leaky-Bucket, Token Bucket Filter

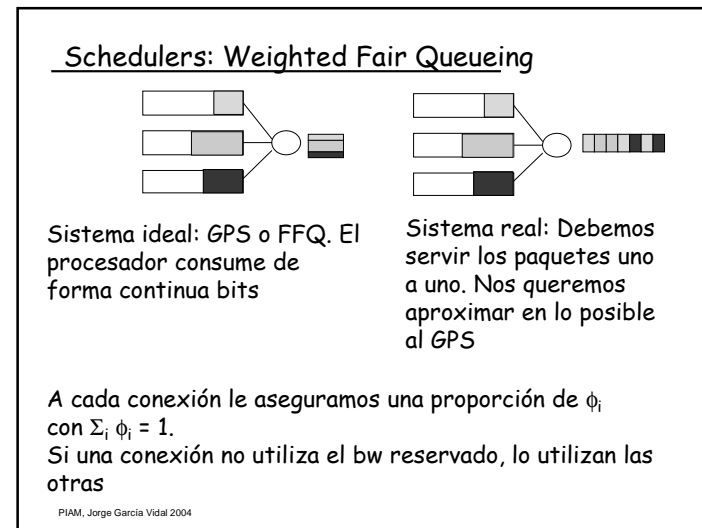
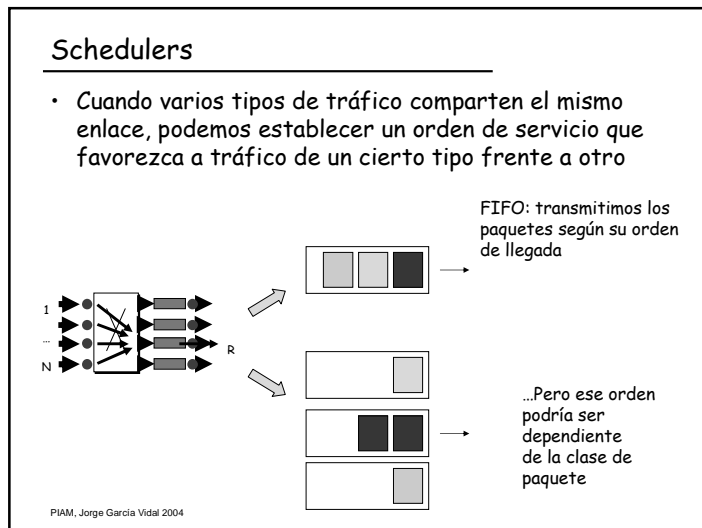
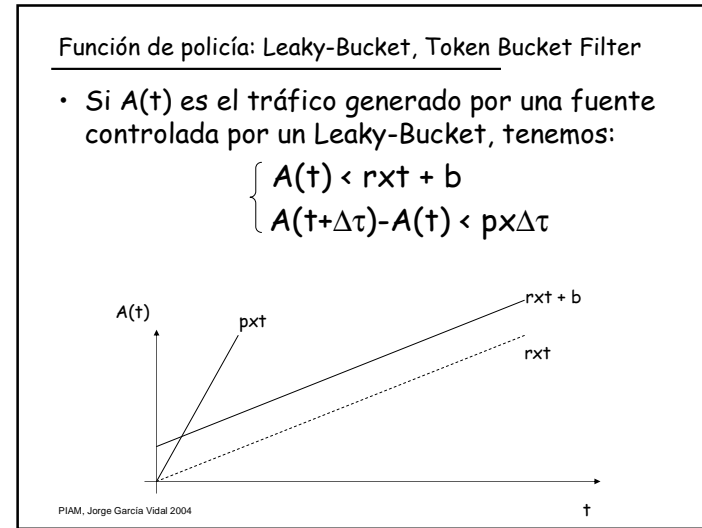
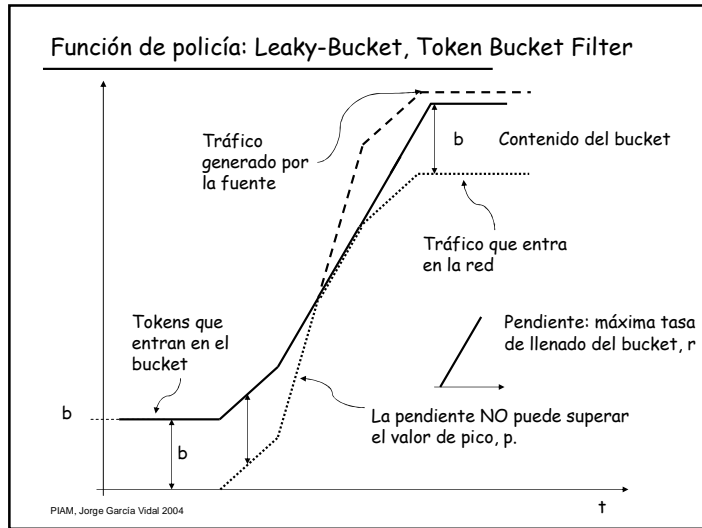
- Leaky-bucket



El bucket se llena a razón de r tokens/s hasta un máximo de b tokens. Cuando la fuente transmite un paquete de L bits, debe sacar del bucket L tokens. Si no los hay, el paquete no puede ser transmitido o bien es marcado como de baja prioridad.

El tráfico medio λ debe ser menor o igual a r . Admitimos desviaciones temporales (bursts) por encima de la media (parámetro b)

PIAM, Jorge Garcia Vidal 2004

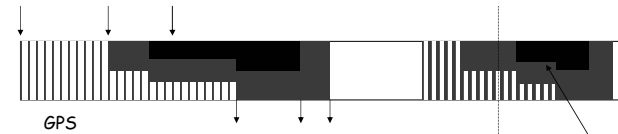


Schedulers: Weighted Fair Queueing

- Idea: Para un sistema que se parezca lo más posible al ideal, calculamos el orden de salida de los paquetes. El sistema real los irá sirviendo en dicho orden.
- Es decir: Al acabar un servicio escogemos el paquete que acabaría antes de ser servido en un sistema ideal (es decir, un GPS) *en el que no hubiera más llegadas*

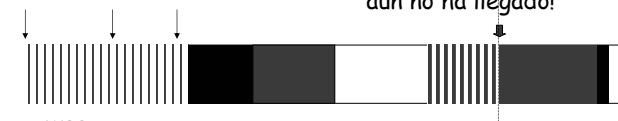
PIAM, Jorge Garcia Vidal 2004

Schedulers: WFQ



El orden de salida en el sistema ideal determina el orden de entrada al servicio en el sistema real

En este punto el paquete aún no ha llegado!



WFQ

PIAM, Jorge Garcia Vidal 2004

Resultado de Parek&Gallager

- En una red con servidores GPS se pueden establecer cotas del retardo si el tráfico de entrada está controlado por Leaky Buckets (b,r) como

retardo máximo = b/r
máximo backlog = b

PIAM, Jorge Garcia Vidal 2004

P&G

En WFQ tenemos aprox. una diferencia con PGS igual al tiempo de transmisión de un paquete, así que

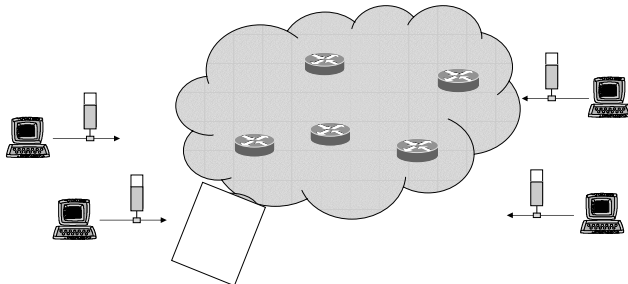
$$D^* \text{ en WFQ} < b/r + (K-1)L_{\max}/r + \sum_i L_{\max}/R_i$$

En donde R_i es la velocidad de transmisión en cada enlace

PIAM, Jorge Garcia Vidal 2004

P&G

- Conclusión: Si el tráfico que entra en la red esta limitado por Leaky-Buckets y los routers implementan schedulers WFQ, el retardo está acotado:



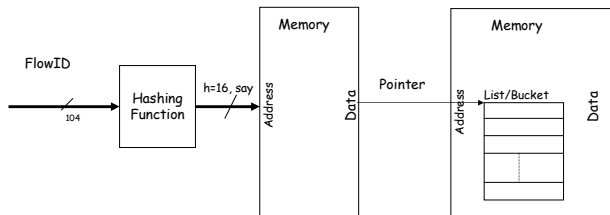
PIAM, Jorge Garcia Vidal 2004

Clasificación de flujos

- Si queremos dar un servicio distinto a cada flujo, debemos tener una forma de clasificación.
- La clasificación se realiza en el forwarding path, por lo que debe realizarse a alta velocidades.
- Típicamente el flujo se identifica por ($@IP_{origen}, @IP_{destino}, Port_{origen}, Port_{destino}, Protocolo$), es decir por 104 bits
- Típicamente se usan funciones de hash para reducir el número de bits a utilizar

PIAM, Jorge Garcia Vidal 2004

Clasificación de flujos



PIAM, Jorge Garcia Vidal 2004

Clasificación de flujos

Si la función de Hash da como resultado h bits ($N = 2^h$), y tenemos M flujos guardados, la probabilidad de colisión en el caso ideal es:

$$P\{\text{Colisión}\} = 1 - [1 - ((N-1)/N)^M] \times N/M \approx M/2N = M/2^{h+1} \quad \text{si } N \gg M \gg 1$$

Ejemplo: $M = 1024, h = 16, P\{\text{Colisión}\} = 0.008$ (1% aprox)
 $M = 64536, h = 16, P\{\text{Colisión}\} = 0.373$ (37% aprox)

Ejemplo:
 Para tener una $P\{\text{Colisión}\} < 1\%$ cuando $M = 100.000$, deberíamos usar
 Un valor de $h \approx \log_2(M/P\{\text{Colisión}\}) - 1 \approx 3 \times \log_{10}\{10^5/10^{-2}\} - 1 = 20$,
 Por lo que la memoria intermedia debería tener un tamaño de $1 M \times n$

PIAM, Jorge Garcia Vidal 2004