

upna

Universidad Pública de Navarra  
Nafarroako Unibertsitate Publikoa

**Tecnologías Avanzadas de Red**  
*Área de Ingeniería Telemática*

# WFQ: Cotas

# Cotas (*bounds*) en WFQ

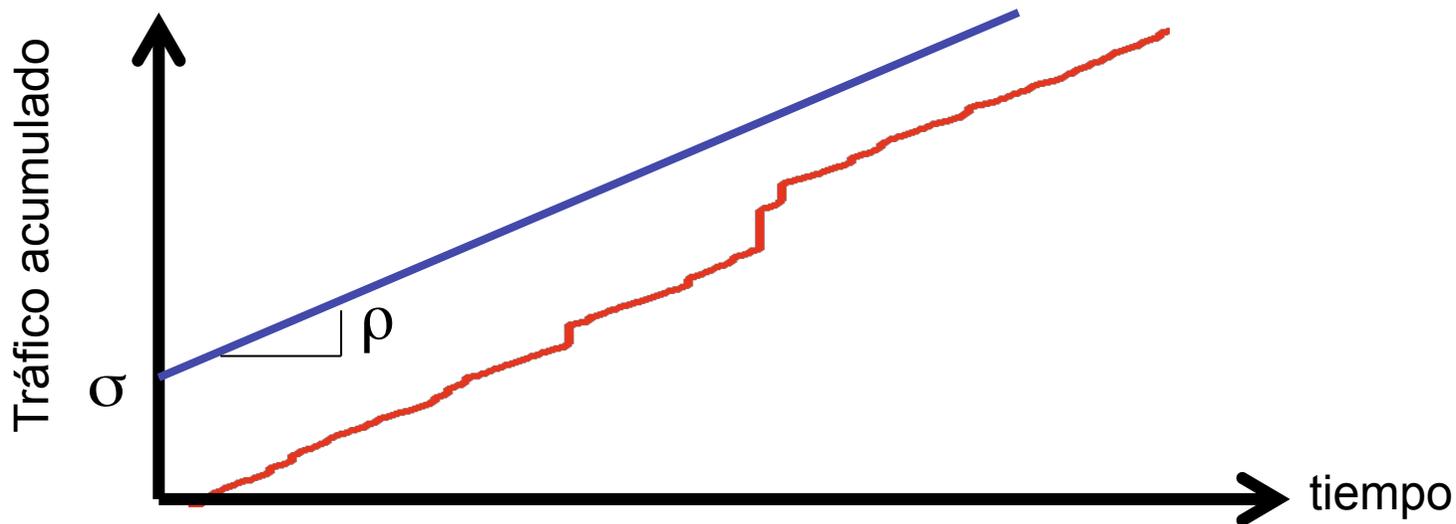
- WFQ garantiza reparto weighted max-min fair
- Eso quiere decir que cada flujo recibe una asignación proporcional a su peso

$$c_i = C \frac{\phi(i)}{\sum \phi(i)}$$

- Además pone una cota al retardo máximo
- ¿Cómo? (...)

# Cotas (*bounds*) en WFQ

- Supongamos un flujo con una restricción “sigma-ro” ( $\sigma, \rho$ ) :
  - En un intervalo  $t$  llegan como mucho  $\sigma + \rho t$  bits
  - Es la salida de un *token bucket*
  - *Linear Bounded Arrival Process* (LBAP)
- Un flujo  $i$  con restricción ( $\sigma(i), \rho(i)$ )
- El resto del tráfico puede no estar conformado
- (...)



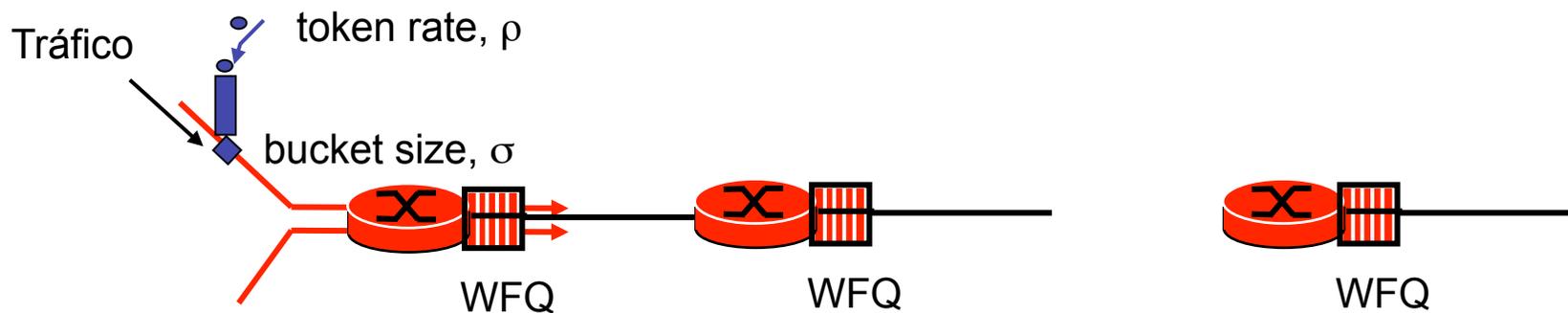
# Cotas (*bounds*) en WFQ

- Camino con  $K$  saltos (todos WFQ)
- Se le ha asignado una tasa  $g(i,k)$  en cada uno:

$$g(i,k) = r(k) \frac{\phi(i,k)}{\sum_j \phi(j,k)}$$

$r(k)$  link rate en enlace  $k$

- (...)



# Cotas (*bounds*) en WFQ

## Ejemplo

- Flujo LBAP con parámetros (16 KiBytes, 150 Kbps)
- $K = 10$  saltos, todos a 45 Mbps, retardo de propagación total de 30ms
- Máximo tamaño de paquete de 8 KiBytes
- Queremos un retardo extremo-a-extremo máximo de 100 ms
- Entonces retardo máximo de  $100 - 30 = 70$  ms
  - (...)

# Cotas (*bounds*) en WFQ

- Con planificadores WFQ en todo el camino
- Y que el flujo esté conformado por un leaky bucket
- Podemos garantizar un retardo máximo extremo a extremo
- No se imponen restricciones al resto de flujos en la red
- Solo hay que seleccionar los valores adecuados de reserva de BW en los enlaces

$$D^*(i) \leq \frac{\sigma(i)}{g(i)} + \sum_{k=1}^{K-1} \frac{P_{\max}(i)}{g(i,k)} + \sum_{k=1}^K \frac{P_{\max}}{r(k)}$$

upna

Universidad Pública de Navarra  
Nafarroako Unibertsitate Publikoa

**Tecnologías Avanzadas de Red**  
*Área de Ingeniería Telemática*

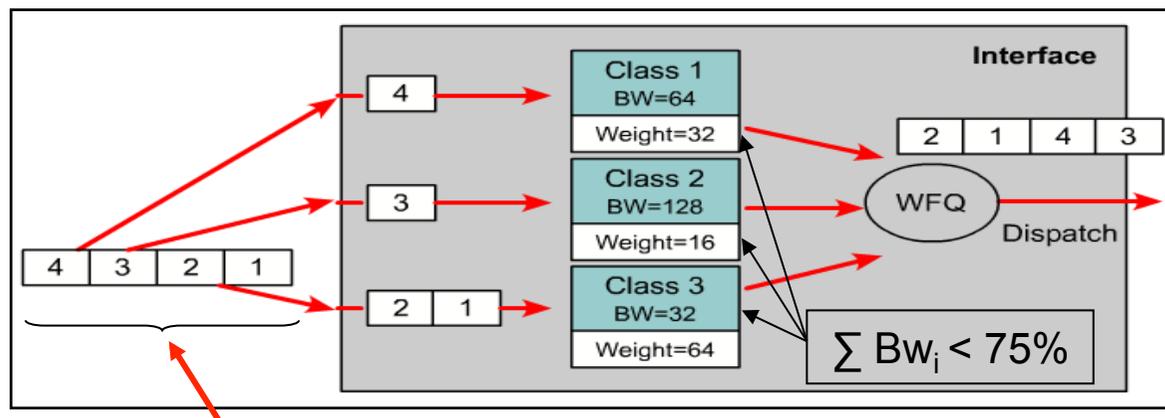
# WFQ en la práctica

# Típica implementación de WFQ

- Cada flujo es una “conversación” reconocida por info. de layer 3 (direcciones IP, precedencia) y de layer 4 (puertos)
- Pesos en función de los bits de precedencia (parte del TOS) de los paquetes
- No requiere configuración
- No escala (una cola por conversación)
- Alternativa: (...)

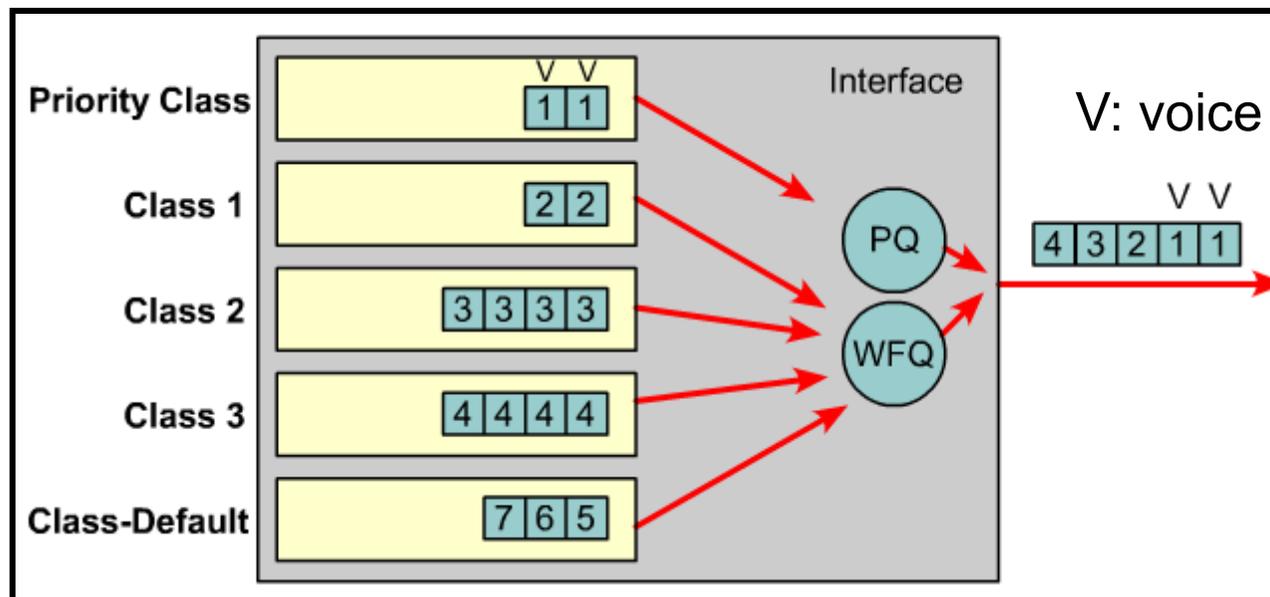
# Típica implementación de WFQ

- Cada flujo es una “conversación” reconocida por info. de layer 3 (direcciones IP, precedencia) y de layer 4 (puertos)
- Pesos en función de los bits de precedencia (parte del TOS) de los paquetes
- No requiere configuración
- No escala (una cola por conversación)
- CBWFQ
  - *Class Based WFQ*
  - Especificar los filtros (clases) que determinan los paquetes que van a cada cola (una por clase, no por flujo)
  - Especificar peso para cada cola



# Low Latency Queueing (LLQ)

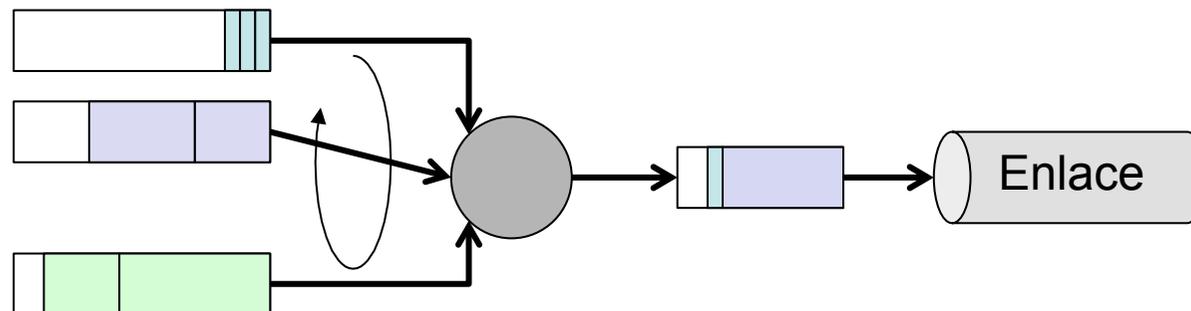
- Añade una PQ (*Priority Queue*) a CBWFQ = PQ-CBWFQ = LLQ
- Recomendable para tráfico multimedia (VoIP): bajo retardo y jitter.
- Aunque WFQ puede acotar retardo lo habitual para tráfico con requisitos estrictos de bajo retardo es usar una cola de prioridad
- Se puede configurar junto al resto de colas CBWFQ como una cola más asociada a una clase determinada.



# Retardos a la salida y fragmentación

# Cola en el interfaz

- Normalmente el planificador no envía los paquetes directamente al enlace
- Los envía a una cola del interfaz hardware
- Esta cola permite maximizar el throughput en el interfaz
- Es una cola FIFO: “interface FIFO” o “transmit ring buffer” o “tx-ring”
- Entre el scheduler y la cola se emplea un control de flujo para que no se desborde el tx-ring
- (...)

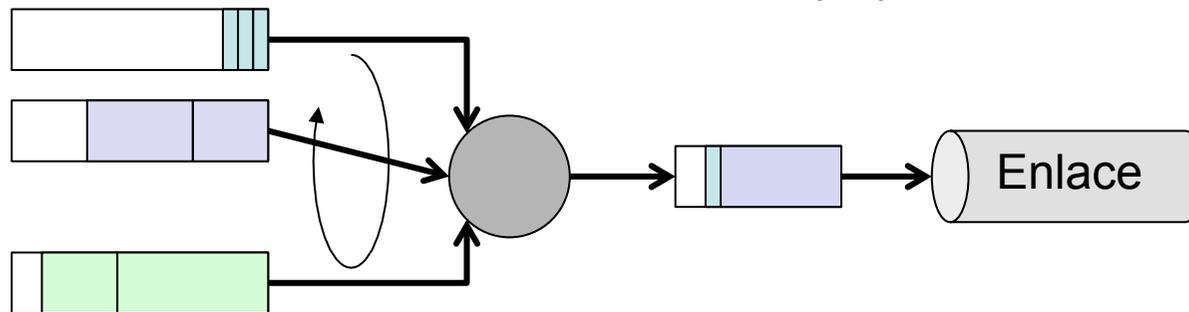


# Cola en el interfaz

- Incluso un paquete con prioridad se encola ahí
- El tamaño suele estar en torno a 1-2 MTUs
- Puede introducir un retardo significativo en enlaces de muy baja velocidad
- Básicamente es el problema de que llegue un paquete de alta prioridad cuando se está transmitiendo uno de baja prioridad

$$D^*(i) \leq \frac{\sigma(i)}{g(i)} + \sum_{k=1}^K \frac{P_{\max}(i)}{g(i,k)} + \sum_{k=1}^K \frac{P_{\max}}{r(k)}$$

WFQ: Efecto de encontrar paquete de otra clase delante



# Link Fragment and Interleaving (LFI)

## Solución:

- Fragmentar los paquetes de datos
- Ej.: límite fragmentos “de 10ms”
- Es decir, tamaño de un paquete igual a máximo que se pueda enviar en 10 ms
- Insertar paquete de VoIP entre estos paquetes
- Asegura un retraso mucho menor
- Los paquetes VoIP no deben fragmentarse

	64Bytes	1024Bytes	1500Bytes
128Kbps	4 ms	64 ms	93 ms
256Kbps	2 ms	32 ms	46 ms
512Kbps	1 ms	16 ms	23 ms
768Kbps	0.64 ms	10.2 ms	15 ms
1Mbps	0.51 ms	8.19 ms	12 ms

