



# Planificación de Paquetes

---

Jhon Jairo Padilla Aguilar  
PhD. Student

Jhon Jairo Padilla A.  
Calidad del servicio en Internet



# Introducción

- El planificador de paquetes es el responsable de asegurar la asignación de recursos a flujos individuales
- El planificador decide qué paquetes deberían tomar los recursos.
- *Es un despachador.* Mantiene información de cuántos paquetes ha enviado cada flujo y los compara con la cantidad de recursos que ha reservado cada flujo. Los paquetes de un flujo son enviados sólo cuando el flujo no ha usado todos los recursos reservados.



# Requerimientos básicos

---

Jhon Jairo Padilla A.  
Calidad del servicio en Internet



# Aislamiento y Distribución

---

- El propósito de un planificador es permitir compartir un recurso común de forma controlada
- Hay dos aspectos importantes a tener en cuenta:
  - **Aislamiento de flujos:** El caso extremo sería el de la conmutación de circuitos (los recursos reservados se subutilizan).
  - **Distribución de recursos:** El manejo más eficiente de un recurso se logra con multiplexación estadística, usada en conmutación de paquetes.
- Hay un compromiso entre estos dos aspectos
- Los algoritmos de distribución justa (fair queuing) logran un balance.



# Límites de retardo

---

- Hay dos tipos de límites de retardo:
  - *Determinísticos* (Ej: retardo del peor caso en Intserv). Requieren fuertes garantías de reservas.
  - *Estadísticos* (Ej: retardos debido al crecimiento de las colas por un tiempo máximo). Permiten multiplexación estadística.



# Asignación de ancho de banda

---

- Cuando no hay suficientes recursos para satisfacer las demandas de tráfico, debe asignarse de forma justa el ancho de banda para todos los flujos en competencia.
- Una política de distribución justa de recursos (criterio de justicia) que ha sido ampliamente considerada en la literatura es llamada *max-min fair sharing*.



# Criterio de Justicia: Max-min fair share

---

- Trata de maximizar la mínima parte asignada a un flujo cuya demanda no ha sido completamente satisfecha.
- Principios usados:
  - Los recursos se distribuyen en función de la demanda de los usuarios
  - A un usuario no se le asignan más recursos de los que ha solicitado
  - Los usuarios que no han satisfecho sus demandas, obtienen igual distribución de los recursos



# Max-min Fair Share

---

- Si la demanda de un usuario no ha sido satisfecha, se trata de darle una mayor parte de los recursos en la siguiente repartición (por lo menos se le asigna un mínimo)

$$FS = \frac{Rec.totales - suma(recursos asignados)}{Núm. Total usuarios con necesidad}$$





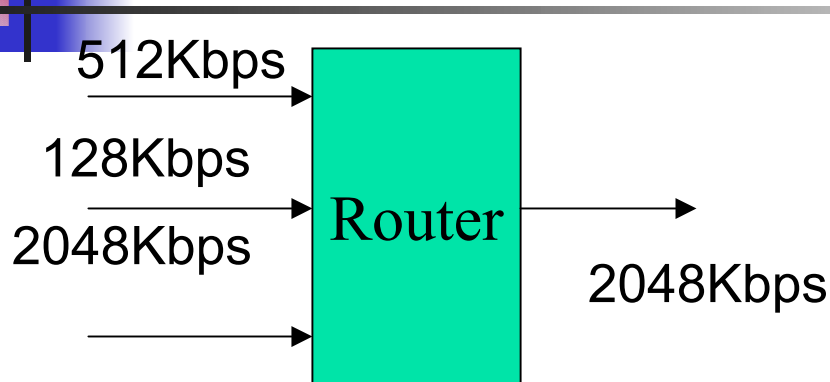
# Max-min Fair Share

---

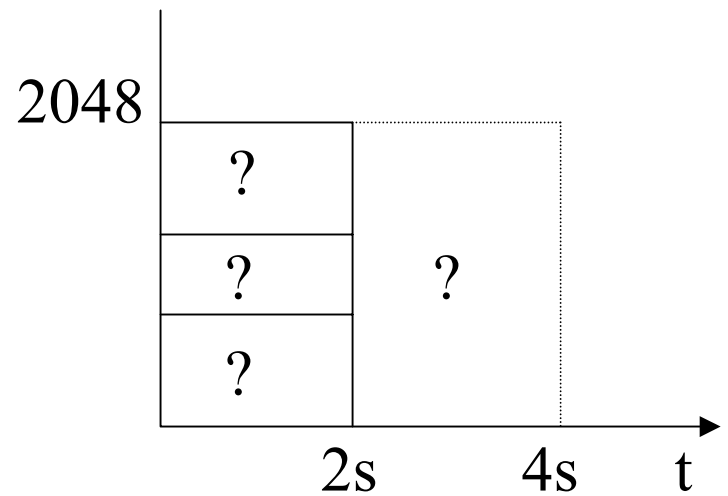
## ■ Cálculo:

1. Calcular el FS inicial (recursos asignados=0)
2. Para los flujos que tienen una demanda igual o menor que el FS, se asigna la demanda real del flujo
3. Los flujos satisfechos se remueven y se sustraen los recursos asignados de la capacidad total
4. Repetir pasos 2 y 3 para el resto de los flujos con el FS actual ( $\text{capac. Restante} / \text{num. Flujos restantes}$ ) hasta que las demandas restantes sean mayores que el FS actual. Todas las demandas restantes tomarán el actual FS.

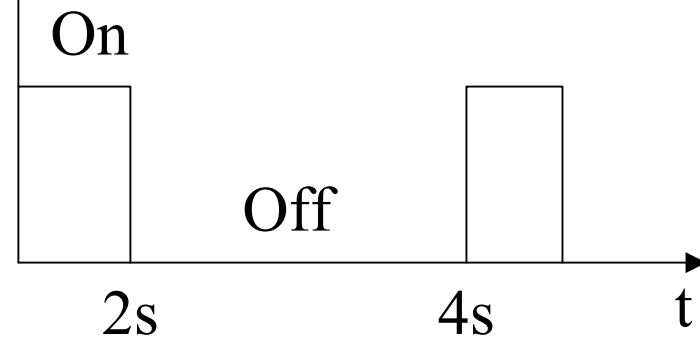
# Ejercicio Max-min FS



BW enlace salida (Kbps)



Comportamiento fuentes





# Max-min Fair Share

---

- Todos los flujos con demandas menores que FS serán satisfechos, mientras que los recursos no utilizados de estos flujos son asignados de acuerdo a los mismos principios para los flujos restantes con demandas mayores que FS.
- Se asume que todos los flujos tienen igual derecho a los recursos.



# Principios de diseño

---

Jhon Jairo Padilla A.  
Calidad del servicio en Internet



# Tipo de disciplina

---

- Hay dos tipos de disciplinas:
  - **Conservativas:** El planificador estará ocupado mientras haya paquetes en cola. (Ej: FCFS)
  - **No conservativas:** El planificador puede estar desocupado aunque haya paquetes en cola. Esto sucede porque los paquetes no cumplen con alguna condición para la transmisión (p.ej. espaciamiento temporal constante a la salida para reducir el jitter)



# Prioridad simple

---

- Hay un número de niveles de prioridad
- La prioridad más alta siempre tiene precedencia sobre la más baja.
- Cuando en el enlace de salida hay disponibilidad, un planificador de prioridad siempre selecciona paquetes de la cola de mayor prioridad.
- Los paquetes con prioridades bajas son transmitidos sólo cuando no hay paquetes en las colas de prioridades altas.



# Prioridad simple

---

- Ventajas:
  - Fácil de entender
  - Implementación directa
- Desventajas:
  - Grandes volúmenes de paquetes con alta prioridad devoran los recursos. Los paquetes de baja prioridad podrían no ser transmitidos nunca



# Categorías

---

- Los algoritmos de planificación pueden ser clasificados en general en 3 categorías:
  - Fair Queuing (distribución justa)
  - Deadline based (basado en plazos)
  - Rate based (basado en la tasa)





# Aproximación Fair-queuing

---

- El ancho de banda asignado a un flujo es representado por un número real, a menudo denominado *peso*.
- Tratan de asignar un BW proporcional a los flujos activos o represados con base en sus pesos.
- Si un flujo no consume todo el BW asignado, el BW restante es asignado a flujos represados en proporción a sus pesos



# Aproximación Fair-queuing

---

- Se le garantiza a un flujo que puede usar su BW asignado y podría usar más.
- Es capaz de proveer un límite de retardo E2E. Por esto se usa en IntServ.



# Esquemas Deadline-based

---

- Planifica paquetes usando el principio EDF (earliest deadline first).
- A cada paquete se le asigna un plazo.
- El planificador transmite los paquetes en el orden de los plazos.
- Con esta aproximación se desacoplan el retardo y el ancho de banda (son independientes).
- EJ: Un flujo con BW pequeño puede obtener también un retardo pequeño



# Esquemas Deadline-based

---

- El control de admisión es más complejo:
  - Primero, se verifica que no se exceda la capacidad del enlace
  - Segundo, se debe verificar que los plazos no se hayan vencido.
- La distribución del BW podría no ser justa (dos procesos con igual BW y retardo podrían no recibir el mismo nivel de servicio).



# Planificación basada en la tasa

---

- Tienen dos componentes:
  - Un regulador (suaviza el tráfico, puede ser token bucket)
  - Un planificador (Selecciona el paquete para transmisión, puede ser tipo Fair queuing, FCFS o EDF)



# Weighted Fair Queuing

---

Jhon Jairo Padilla A.  
Calidad del servicio en Internet



# Introducción

---

- WFQ es una clase de algoritmos de planificación
- WFQ soporta asignación de BW y límites de retardo
- Fue propuesto hace más de 10 años
- Tiene variantes con diferentes compromisos entre complejidad y precisión
- Ha sido implementado ampliamente en routers para soportar QoS



# Modelo de Fluidos

---

- WFQ es explicado a menudo como un modelo de fluidos.
- En el modelo de fluidos se asume que:
  - El tráfico es infinitamente divisible
  - Un nodo puede servir múltiples flujos simultáneamente
- En una red real:
  - Los paquetes son procesados uno a la vez
  - Por tanto, el tamaño del paquete afecta el sistema de colas





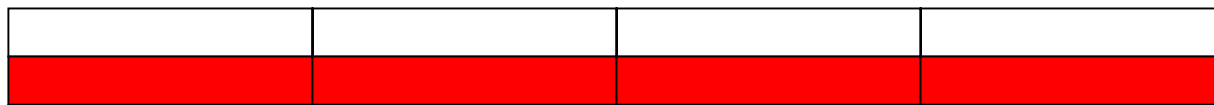
# Modelo de fluidos

---

- Ventajas del modelo de fluidos:
  - Permite visualizar la operación de las disciplinas de colas más fácilmente
  - Es sencillo derivar resultados para un sistema paquetizado de un modelo de fluidos

# Modelo de fluidos vs. Modelo paquetizado

Modelo de fluidos



Modelo Paquetizado



# GPS: Generalized Processor Sharing

---

- GPS es un algoritmo tipo fair-queuing ideal
- Está basado en el modelo de fluidos
- Provee una distribución de recursos justa con el criterio *max-min fair share*
- Cuando una conexión tiene datos en la cola, se dice que está en un estado *backlogged* (espera)



# GPS

---

- Supuestos:
  - N flujos siendo servidos por un servidor
  - R: Tasa de servicio del servidor
  - A un flujo i-ésimo se le asigna un peso  $\phi_i$
  - $S(i, \tau, t)$ : Cantidad de datos servidos para el flujo i durante el intervalo  $(\tau, t)$



# GPS

---

- En el intervalo  $(\tau, t)$  el flujo  $i$  recibe una parte justa mínima proporcional a su peso:

- $$R_i = \frac{\phi_i}{\sum_j^V(\phi_j)} \times R$$

- Donde  $V$  es el conjunto de flujos en espera durante el intervalo
- Los flujos que no están en espera (envían menos de su parte asignada) reciben servicios sin ninguna cola.



# GPS

---

- El BW restante es repartido a los flujos en espera en proporción a sus pesos
- Se garantiza la mínima parte a cada flujo y el ancho de banda sobrante es repartido proporcionalmente a los flujos en espera



# Características de GPS

---

- Provee una repartición justa del BW
- Puede garantizar un límite de retardo de  $(b/r)$ , si la fuente de tráfico es restringida con un token bucket con tamaño de ráfaga  $b$  y tasa de tokens:

$$r \geq \frac{\phi_i}{\sum_j V(\phi_j)} \times R$$



# Ejercicio:

---

- Se tienen 3 flujos  $\phi_1, \phi_2, \phi_3$
- El peso de los flujos es 50,25 y 25 respectivamente
- De  $t_0$  a  $t_1$ : se presentan los 3 flujos
- De  $t_1$  a  $t_2$  se presentan sólo  $\phi_2$  y  $\phi_3$
- Calcular la distribución del BW mediante GPS





# WFQ

---

- WFQ (Weighed Fair Queuing) es la aproximación paquetizada de la planificación GPS
- WFQ también se conoce como PGPS (Packet-by-packet generalized processor sharing)
- WFQ trata de emular un sistema GPS calculando el tiempo de salida de un paquete (*finish time*) en un sistema GPS. Esta marca de tiempo se usa para planificar los paquetes.



# WFQ

---

- El *finish time* no es el tiempo de salida real de un paquete, sino que es un número que representa el orden de los paquetes en el sistema.
- Si un paquete A termina la transmisión antes que un paquete B en un sistema GPS, el paquete A será seleccionado para transmisión antes que el paquete B en el sistema WFQ.

# Cálculo de las marcas de tiempo en WFQ

## ■ Caso:

- Todos los flujos están en espera todo el tiempo (siempre enviando a tasas más altas que su BW reservado)
- $F_i^k$ : marca de tiempo del paquete  $k$  del flujo  $i$

$$F_i^k = F_i^{k-1} + \left( \frac{L_i^k}{\phi_i} \right)$$

← Tiempo para Txtr el paq. actual

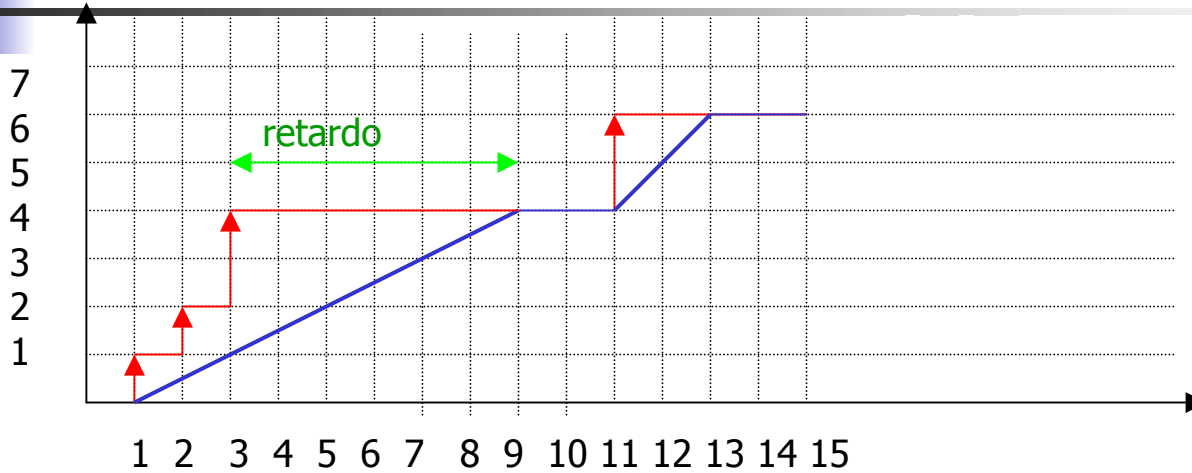
# Cálculo de las marcas de tiempo en WFQ

- **Caso:** Los flujos podrían cambiar de espera a no-espera y viceversa, por lo que podría haber tiempo no usado entre dos paquetes
- Marca de tiempo:

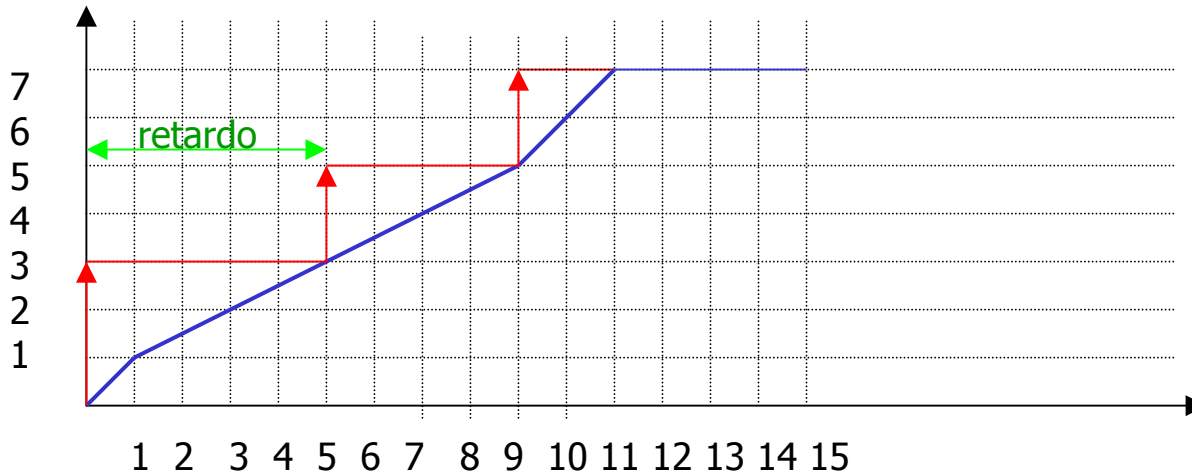
$$F_i^k = \max[F_i^{k-1}, V(t)] + \frac{L_i^k}{\phi_i}$$

$V(t)$  es una función lineal del tiempo real  $t$  en cada uno de los intervalos de tiempo

# Curvas Llegada-salida



Flujo 1



Flujo 2



# Ejercicio

---

- Realice una tabla con las marcas de tiempo para PGPS si se tienen 11 flujos con peso  $\phi_1=0.5$  y  $\phi_2$  a  $\phi_{11}$  con peso 0.05. Suponga una longitud de paquete de 1. Calcule las marcas de tiempo para  $t=1,2,3,\text{etc...}$



# Worst-Case Fair WFQ

---

- Es una extensión de WFQ
- En WF<sup>2</sup>Q el paquete seleccionado para transmisión es el que tenga el tiempo de salida más corto de los paquetes que ya han iniciado servicio en el sistema GPS.
- Tiene el mismo límite de retardo para los paquetes que WFQ, pero mejora la justicia.
- WFQ no es tan justo, pues si un flujo deja de transmitir, después se le dará demasiado BW.



# Hierarchical WFQ

---

- Un enlace puede ser subdividido mediante WFQ y luego sus divisiones pueden ser subdivididas usando nuevamente WFQ.
- *Ejemplo:* Una empresa con 10 divisiones que comparten un enlace. A su vez cada división comparte su parte del enlace entre sus aplicaciones





# Variantes de WFQ

---

Jhon Jairo Padilla A.  
Calidad del servicio en Internet



# Introducción

---

- WFQ tiene propiedades que la hacen una disciplina ideal para soportar QoS en Internet.
- *Desventaja:* Es compleja e impractica a altas velocidades.
- Se han propuesto variantes de WFQ que son más sencillas y usadas en muchas implementaciones: SCFQ, WRR, DRR entre otras.



# SCFQ

---

- Self-clocking Fair Queuing
- SCFQ propone una forma simple de aproximar el tiempo de salida en un sistema GPS correspondiente.
- SCFQ usa el tiempo de salida del paquete actualmente en servicio como el  $V(t)$ .
- Esto alarga el retardo de peor caso, además de que este retardo aumenta linealmente con el número de flujos en el sistema.



# Round Robin (RR)

---

- Sirve paquetes de cada cola en orden cíclico.
- Cada flujo tiene el mismo peso y los paquetes son del mismo tamaño
- Sólo se sirven las colas llenas.



# WRR

---

- Weighted Round Robin
- Transmite un número de paquetes por cada flujo que es proporcional a su peso.
- Ej: flujo A (peso=1), flujo B (peso=2); se transmite 1 paquete de A y dos paquetes de B en su respectivo turno.
- Funciona bien cuando el tamaño de paquete es fijo o se conoce el tamaño medio de paquete para un flujo. De otra forma, podría no ser justo.



# DRR

---

- Deficit Round Robin
- Cada flujo tiene un contador de déficit, el cual se fija en cero al inicio.
- El planificador sirve un número fijo de bits (llamado quantum) para cada flujo.
- Si en un turno no se transmite toda la cantidad de bits del quantum, el sobrante se guarda en el contador de déficit y se suma al quantum del flujo en el siguiente turno (se le guarda lo que se le debe).
- El quantum puede ser asignado de acuerdo al peso de cada flujo para emular un WFQ.