

# Temas 22 y 23

## Control de congestión y flujo

Eytan Modiano

# CONTROL DE FLUJO

---

LIDS

- **Control de flujo: mecanismo extremo a extremo para regular el tráfico entre el origen y el destino**
- **Control de congestión: mecanismo utilizado por la red para limitar la congestión**
- **En realidad no se pueden separar y nos referiremos a ambos como control de flujo**
- **En cualquier caso, ambos se refieren a mecanismos destinados a limitar la cantidad de tráfico que entra en la red:**
  - **En ocasiones, la carga es mayor de lo que la red puede soportar**

# SIN CONTROL DE FLUJO

---

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

*LIDS*

- **Cuando se produce una sobrecarga:**
  - Se forman colas
  - Se eliminan paquetes
  - Desde el origen es necesario retransmitir los mensajes
  - Aumenta la congestión => inestabilidad
- **El control de flujo previene la inestabilidad de la red haciendo que los paquetes esperen fuera de la red, en vez de formando colas dentro de ella:**
  - Evita un gasto de los recursos de la red
  - Evita "desastres"

# OBJETIVOS DEL CONTROL DE FLUJO

---

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

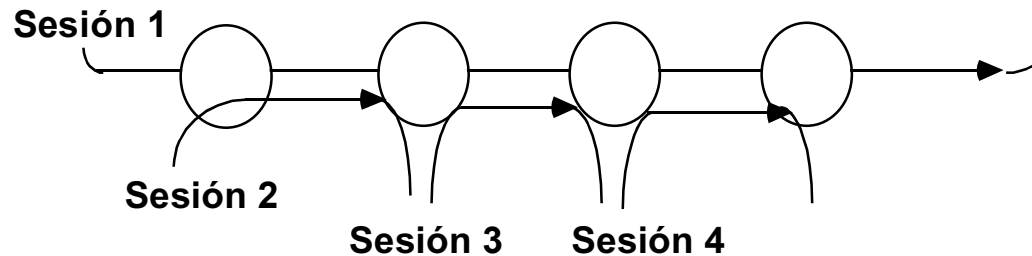
*LIDS*

- **Maximizar la tasa de transferencia de la red**
- **Reducir los retardos de la red**
- **Mantener los parámetros de calidad de servicio**
  - **Equidad, retardo, etc.**
- **Intercambio entre equidad, retardo, tasa de transferencia...**

# EQUIDAD

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

LIDS

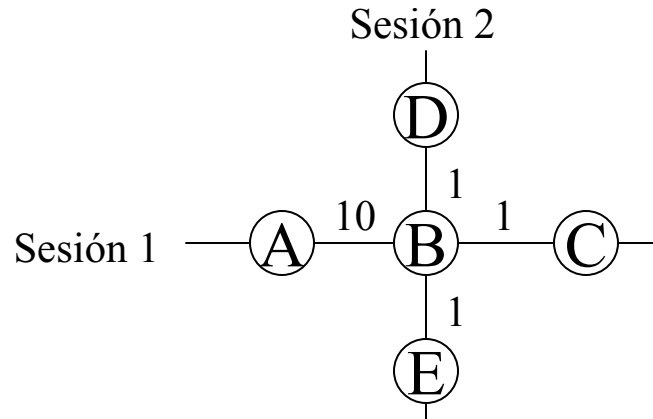


- **Si las capacidades del enlace son cada una 1 unidad, entonces:**
  - La tasa de transferencia máxima se logra asignando una unidad a la sesión corta y cero unidades a la larga; una tasa de transferencia total de 3 unidades
  - Un concepto de equidad daría a cada usuario  $1/2$  unidad; una tasa de transferencia total de 2 unidades
  - Alternativamente, la acción de asignar recursos iguales a cada sesión daría  $3/4$  a cada usuario de un solo enlace y  $1/4$  de unidad a la sesión larga

# EQUIDAD

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

LIDS

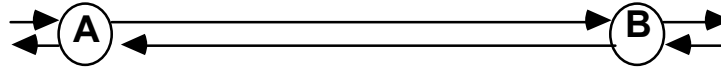


- Cola limitada en el nodo B
- Ambas sesiones están claramente limitadas a 1 unidad de tráfico
- Sin control de flujo, la sesión 1 puede dominar la cola del nodo B:
  - Dado que llegan 10 paquetes de la sesión 1 por cada paquete de la sesión 2, 10/11 de los paquetes de la cola pertenecerán a la sesión 1

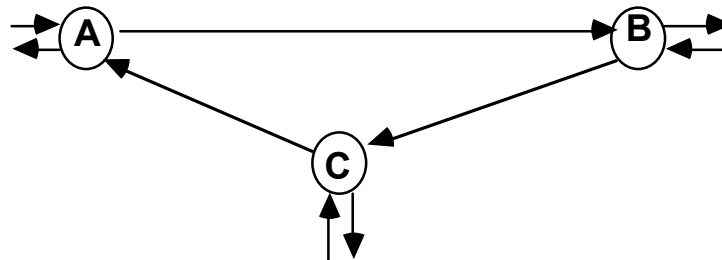
# ATASCO POR DESBORDAMIENTO DE *BUFFER*

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

LIDS



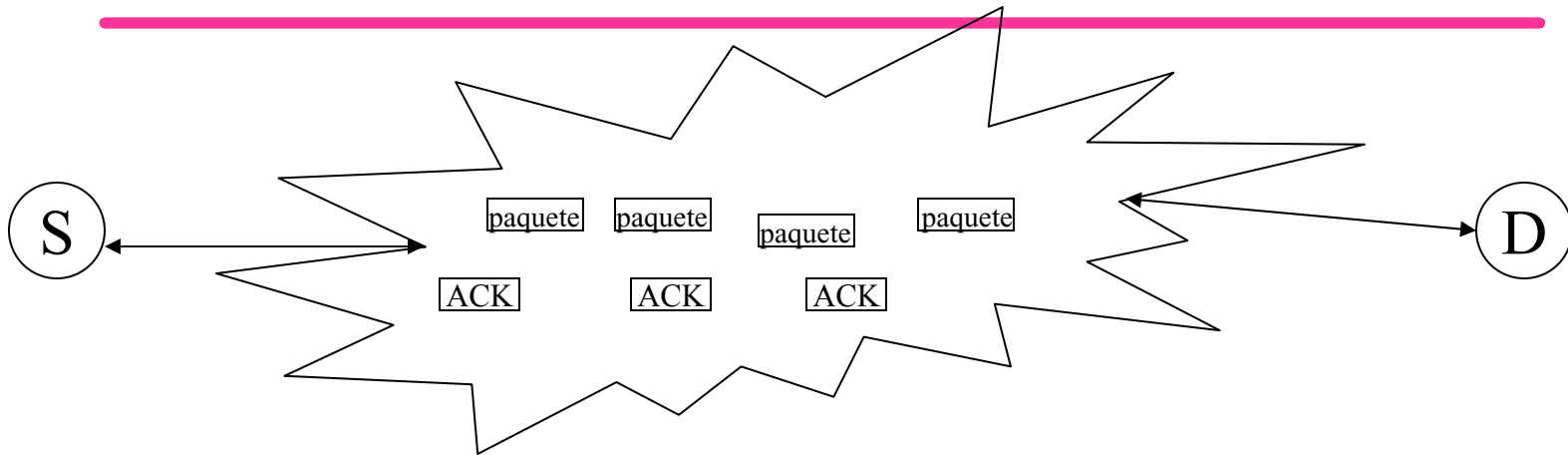
- Si las colas en A se llenan con el tráfico que se dirige a B y viceversa, entonces A no puede aceptar ningún tráfico que venga desde B y viceversa, dando lugar a un atasco:
  - A no puede aceptar ningún tráfico que venga desde B
  - B no puede aceptar ningún tráfico que venga desde A
- A puede estar llena de tráfico de B, B de tráfico de C y C de tráfico de A



# CONTROL DE FLUJO DE LA VENTANA

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

LIDS



- **Similar al ARQ basado en ventanas:**
  - Ventana extremo a extremo para cada sesión,  $Wsd$
  - Cada paquete es confirmado por el receptor
  - Numero total de paquetes sin confirmación (ACK)  $\leq Wsd$
- ⇒ El tamaño de la ventana es un límite superior para el número total de paquetes y confirmaciones en la red
- ⇒ Límite de la cantidad de cola necesaria en el interior de la red

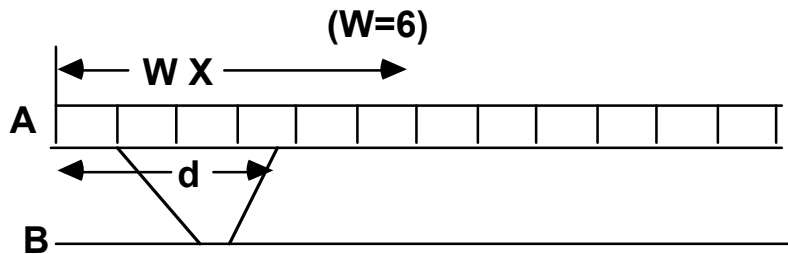
# VENTANAS EXTREMO A EXTREMO

- Sean  $x$  el tiempo esperado de transmisión de un paquete,  $W$  el tamaño de la ventana y  $d$  el retardo total de un paquete para una vuelta de recorrido:
  - Lo ideal sería que el control de flujo sólo estuviese activo en los momentos de congestión:

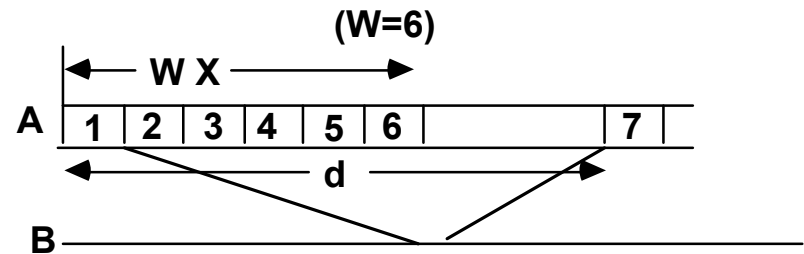
Por lo tanto,  $Wx$  debería ser elevado con relación al retardo total para una vuelta de recorrido  $d$  en ausencia de congestión

Si  $d \leq Wx$ , control de flujo inactivo y tasa de la sesión  $r = 1/x$

Si  $d > Wx$ , control de flujo activo y tasa de la sesión  $r = W/d$  paquetes por segundo

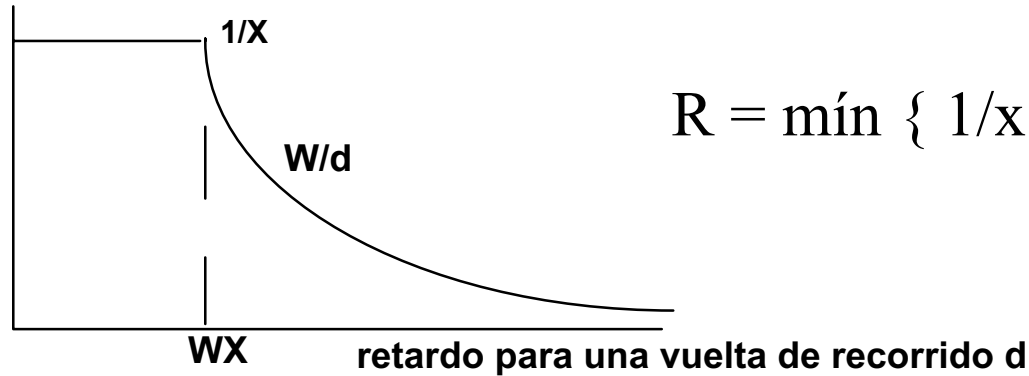


Control de flujo inactivo



Control de flujo activo

# Comportamiento de las ventanas extremo a extremo



$$R = \text{mín} \{ 1/x, W/d \} \text{ paquetes/segundo}$$

- A medida que  $d$  aumenta, se activa el control de flujo y limita la tasa de transmisión
- A medida que disminuye la congestión, disminuirá  $d$  y aumentará  $r$
- El control de flujo tiene la capacidad de estabilizar los retardos de la red

# Elección del tamaño de la ventana

- Sin congestión, la ventana debería ser lo suficientemente grande para permitir la transmisión a una tasa total de  $1/x$  paquetes por segundo
  - $d'$  = retardo en una vuelta de recorrido cuando no hay colas
  - $N$  = número de nodos que hay a lo largo de la ruta
  - $D_p$  = retardo de propagación a lo largo de la ruta
- ⇒  $d' = 2Nx + 2 D_p$  (retardo de enviar un paquete y un ACK por  $N$  enlaces)
- ⇒  $Wx > d' \Rightarrow W > 2N + D_p/x$
- Cuando  $D_p < x$ ,  $W \sim 2N$  (el tamaño de la ventana es independiente del retardo de propagación)
- Cuando  $D_p \gg Nx$ ,  $W \sim 2D_p/x$  (el tamaño de la ventana es independiente la longitud de la ruta)

# Impacto por congestión

---

LIDS

- Sin congestión  $d = d'$  y el control de flujo está inactivo
- Con congestión  $d > d'$  y se activa el control de flujo
  
- Problema: cuando  $d'$  es elevado (ej.:  $D_p$  es grande) el retardo en cola es menor que el retardo de propagación  $y$ , por tanto, resulta difícil controlar la congestión:
  - $\Rightarrow$  el retardo en cola aumentado tiene un pequeño impacto sobre  $d$  y, por tanto, un pequeño impacto sobre la tasa  $r$

# PROBLEMAS CON VENTANAS

---

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

LIDS

- **El tamaño de la ventana debe variar con el nivel de congestión**
- **Es difícil garantizar unos retardos o una tasa de datos para una sesión**
- **Para sesiones de alta velocidad en redes de alta velocidad, las ventanas deben ser muy grandes:**
  - **Ej.: para 1 Gbps a través del país, cada ventana debe exceder de 60Mb**
  - **El control de flujo de la ventana se vuelve ineficaz**
  - **Las ventanas grandes requieren hacer mucha cola en la red**
- **Las sesiones de rutas largas con ventanas grandes reciben un mejor trato que las sesiones de rutas cortas. En un punto de congestión, la ventana grande llena la cola y acapara el servicio (a menos que se utilice un servicio de peticiones)**

# VENTANAS NODO A NODO



- **Ventana ( $w$ ) separada para cada enlace a lo largo de la ruta de las sesiones**
  - Cola de tamaño  $w$  en cada nodo
- **Un *bit* de confirmación (ACK) vuelve a un enlace cuando se entrega un paquete al siguiente enlace:**
  - $\Rightarrow$  la cola nunca se desborda
- **Si un enlace se congestiona, los paquetes permanecen en cola y los ACKs no vuelven al enlace anterior, ya que éste se podría congestionar a su vez y dejar de enviar ACKs (presión hacia atrás):**
  - Las colas se llenarán en nodos sucesivos:
    - En condiciones de congestión, los paquetes se expanden uniformemente por la ruta en lugar de acumularse en el punto de congestión
- **En redes de alta velocidad esto aún requiere ventanas grandes y, por tanto, grandes colas en cada nodo**

# CONTROL DE FLUJO BASADO EN LA TASA

Para ver esta imagen son necesarios el QuickTime™ y un descompresor del formato GIF

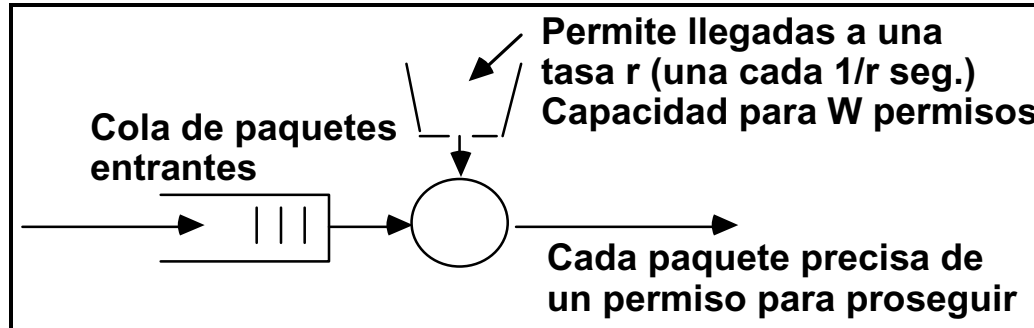
LIDS

- El control de flujo de la ventana no puede garantizar una tasa o un retardo
  - Requiere ventanas grandes para enlaces de alto (retardo \* tasa)
  - Los esquemas de control de tasa ofrecen al usuario una tasa garantizada y alguna capacidad limitada para exceder dicha tasa:
    - Implementación estricta: para una tasa de  $r$  paquetes por segundo permite exactamente un paquete cada  $1/r$  segundos  
=> TDMA => ineficaz para el tráfico a ráfagas
    - Implementación menos estricta: permite  $W$  paquetes cada  $W/r$  segundos  
La tasa media permanece igual, pero se permiten ráfagas de hasta  $W$  paquetes
- Por lo general se implementa utilizando un esquema de "leaky bucket" (caldero agujereado)

# CONTROL DE TASA *LEAKY BUCKET*

Para ver esta imagen son necesarios el QuickTime™ y un descompresor del formato JPEG

LIDS



- El caldero (*bucket*) de la sesión contiene  $W$  permisos:
  - Para entrar en la red, el paquete debe obtener primero un permiso
  - El caldero recibe nuevos permisos a una tasa de uno cada  $1/r$  segundos
- Cuando el caldero está lleno, una ráfaga de hasta  $W$  paquetes puede entrar en la red:
  - El parámetro  $W$  especifica **how bursty the source can be**
    - $W$  pequeño => control de tasa estricto
    - Un  $W$  grande permite ráfagas más largas
  - $r$  especifica la tasa máxima a largo plazo
- Una sesión inactiva ganará permisos, pudiendo usarlos más tarde

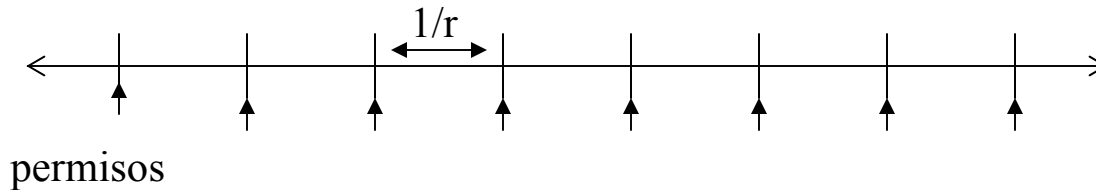
# Control de flujo *leaky bucket*

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

LIDS

- ***Leaky bucket* es un mecanismo para modelar el tráfico**
- **Los esquemas de control de flujo pueden ajustar los valores  $W$  y  $r$  en respuesta a la congestión:**
  - **Ej.: las redes ATM utilizan células RM (gestión de recursos) para indicar a los orígenes que ajusten sus tasas en función de la congestión**

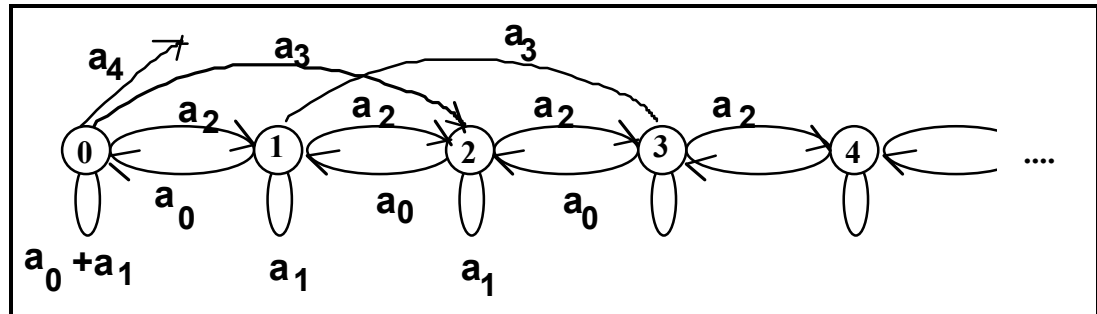
# ANÁLISIS DE COLAS DE *LEAKY BUCKET*



- **Sistema dividido en *slots* de tiempo con un cambio de estado cada  $1/r$  seg.:**
  - Un permiso llega al inicio del *slot* y si el caldero está lleno se descarta
  - Los paquetes llegan de acuerdo a un proceso de Poisson de tasa  $\lambda$
  - $a_i = \text{Prob}(i \text{ llegadas}) = (\lambda/r)^i e^{-\lambda/r} / i!$
  
  - $P$  = número de paquetes que esperan por un permiso en la cola
  - $B$  = número de permisos en cola
  - $W$  = tamaño del caldero
  
- **Estado del sistema:  $K = W + P - B$** 
  - El estado representa el “déficit de permisos” y es igual al número de permisos necesarios para rellenar el caldero
    - Estado 0 => caldero lleno de permisos
    - Estado  $W$  => ningún permiso en cola
    - Estado  $W + j$  =>  $j$  paquetes esperando por un permiso

# ANÁLISIS DE COLAS (continuación)

Cadena de Markov:



- **Obsérve que esto es lo mismo que una M/D/1 con servicio dividido en *slots*:**
  - En estado estacionario, la tasa de llegada de paquetes es igual a la tasa de llegada de permisos (cuando el caldero está lleno los permisos se eliminan; los permisos no llegan en estado "0" cuando no llegan paquetes)

$$\Rightarrow \lambda = (1 - P(0))a_0 r, \quad \Rightarrow P(0) = (r - \lambda) / (a_0 r)$$

- **A continuación, a partir de ecuaciones de equilibrio globales:**
  - $P(0) [1 - a_0 - a_1] = a_0 P(1)$
  - $P(1) = [(1 - a_0 - a_1) / a_0] P(0) \Rightarrow$  se puede resolver para  $P(1)$  en términos de  $P(0)$
  - $P(1) [1 - a_1] = a_2 P(0) + a_0 P(2) \Rightarrow$  se obtiene  $P(2)$  en términos de  $P(1)$
  - Se resuelve recursivamente para todo  $P(i)$

- **Retardo medio para obtener un permiso =** 
$$T = \left[ \sum_{j=W+1}^{\infty} (j - W) P(j) \right] \frac{1}{r}$$

# Elegir un valor para $r$

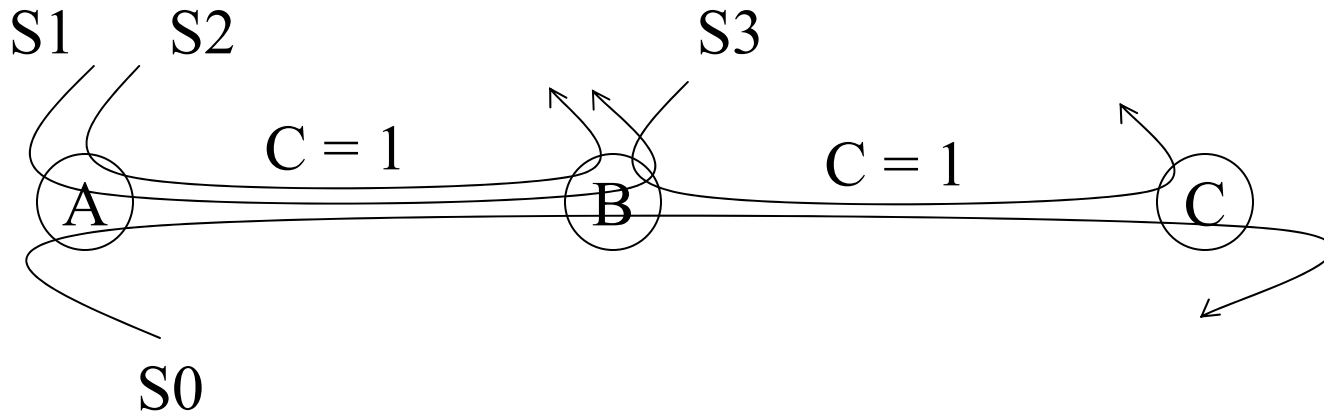
---

LIDS

- **¿Cómo decidimos la tasa que se asigna a una sesión?**
- **Enfoques:**
  1. **Enrutamiento óptimo y control de flujo:**
    - Intercambio entre el retardo y la tasa de transferencia
  2. **Equidad máx-mín:**
    - Asignación equitativa de recursos
  3. **Basada en contrato:**
    - Tasa negociada por un precio (ej.: tasa garantizada, etc.)

# Equidad máx-mín

- Se tratan todas las sesiones por igual
- Ejemplo:



- Las sesiones S0, S1 y S2 comparten el enlace AB y cada una de ellas obtiene un reparto equitativo de  $1/3$
- Las sesiones S3 y S0 comparten el enlace BC pero, dado que la sesión S0 está limitada a  $1/3$  por el enlace AB, a la sesión S3 se le puede asignar una tasa de  $2/3$

# Noción máx-mín

---

LIDS

- La idea que subyace tras la equidad máx-mín es asignar a cada sesión la mayor tasa posible teniendo en cuenta que el aumento de la tasa de una sesión no debería ir en detrimento de otra cuya tasa asignada no sea mayor que la dada: sesión cuya tasa se va a incrementar:
  - Es decir, si aumentar la tasa de una sesión va en detrimento de otra que ya posee una tasa inferior, no se hace
- Dado un conjunto de peticiones de sesiones  $P$  y un conjunto asociado de tasas  $R_p$ ,  $R_p$  es equitativo máx-mín si:
  - Para cada sesión  $p$ , no se puede incrementar  $r_p$  sin disminuir  $r_{p'}$  para alguna sesión  $p'$  para la que  $r_{p'} \leq r_p$

# Definición de equidad máx-mín

- Establecemos  $r_p$  como la tasa asignada a la sesión  $p$  y observamos un enlace con capacidad  $C_a$

- El flujo del enlace  $a$  viene dado por: 
$$F_a = \sum_{p \text{ que atraviesa el enlace } a} r_p$$

- Un vector tasa  $R$  es posible si:

- $R_p \geq 0$  para todo  $p$  en  $P$  (todas las peticiones de sesión) y
- $F_a \leq C_a$  para todo  $a$  en  $A$  (donde  $A$  es el conjunto de enlaces)

- $R$  es equitativo máx-mín si es posible y:

Para todo  $p$ , si existe un  $R^1$  posible tal que  $r_p < r_p^1$

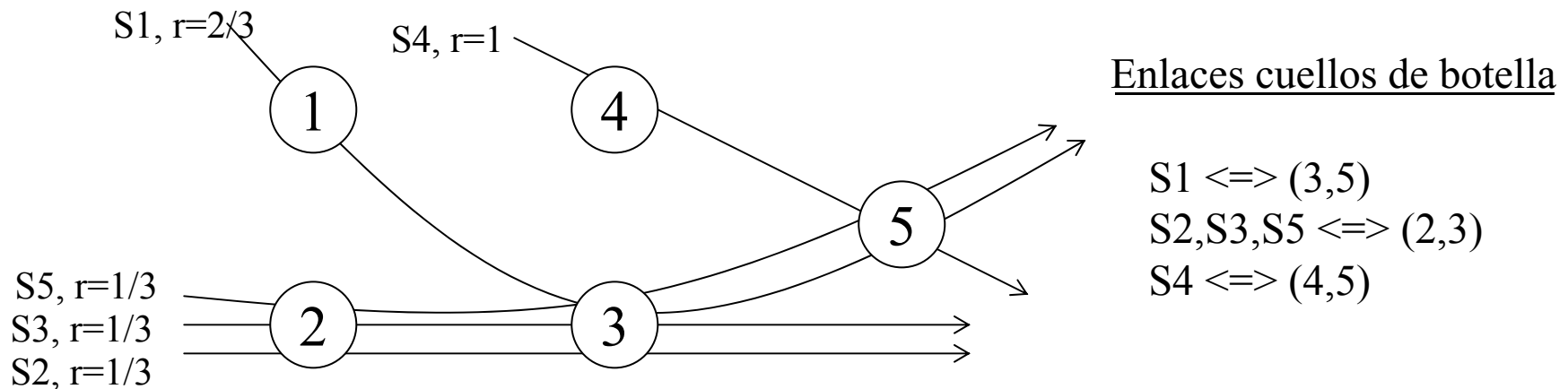
Entonces, existe una sesión  $p'$  tal que  $r_{p'} > r_{p'}^1$  y  $r_{p'} \leq r_p$

- En otras palabras, **solo se puede incrementar la tasa de una ruta disminuyendo la de otra a la que no se le haya asignado previamente más capacidad**

# Enlace cuello de botella (*bottleneck*)

LIDS

- Dado un vector tasa  $R$ , un enlace 'a' es un cuello de botella para la sesión  $p$  si:
  - $F_a = C_a$  y  $r_p \geq r_{p'}$ , para todas las sesiones  $p'$  que atraviesan el enlace 'a'
  - Obsérvese que todas las otras sesiones deben tener algún otro enlace que actúe como cuello de botella, de lo contrario, podría aumentar su tasa en el enlace 'a'
- Propuesta: un vector tasa posible  $R$  es equitativo máx-mín si y sólo si: cada sesión tiene un enlace cuello de botella con respecto a  $R$
- Ejemplo ( $C=1$  para todos los enlaces)



# Algoritmo de equidad máx-mín

- **Empezar todas las sesiones con tasa cero**
- **Incrementar las tasa de todas las sesiones por igual en una pequeña cantidad  $\delta$ :**
  - **Continuar incrementando hasta que algún enlace alcanza la capacidad ( $F_a = C_a$ )**
    - Todas las sesiones que comparten ese enlace tienen tasas iguales
    - El enlace es un cuello de botella con respecto a esas sesiones
    - Dejar de aumentar las tasas de esas sesiones (esa es su asignación máx-mín)
  - **Continuar incrementando la tasa de todas las otras sesiones que todavía no hayan llegado a un enlace cuello de botella**
    - Hasta encontrar otro enlace cuello de botella
  - **El algoritmo termina cuando todas las sesiones tienen un enlace cuello de botella**
- **En la práctica, las sesiones no se conocen de antemano y no resulta práctico calcular las tasas antes**

# Servicio GPS (*Generalized Processor Sharing*) (colas equitativas **con** AKA)

---

Para ver esta imagen son necesarios el QuickTime™ y un descompresor GIF

LIDS

- **Se sirve la sesión por orden de peticiones:**
  - Si las sesiones tienen siempre un paquete para enviar, todas ellas comparten el enlace por igual
  - Si algunas de las sesiones están vacías, las otras comparten la capacidad por igual
- El *processor sharing* generalmente se refiere a un modelo “fluido” en el que las tasas de la sesión se pueden afinar de modo arbitrario
- El GPS es un enfoque basado en paquetes en el que los paquetes de cada sesión se sirven por orden de petición