

---

## **Temas 17 y 18**

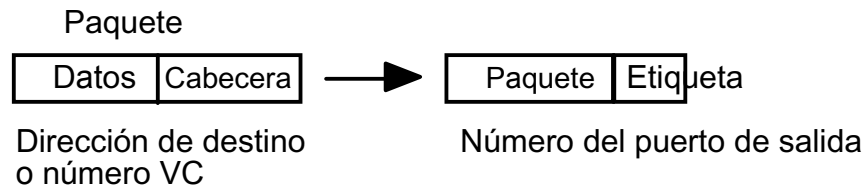
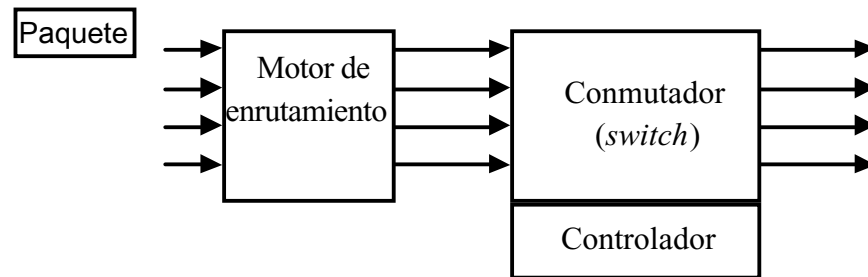
# **Conmutación rápida de paquetes**

**Eytan Modiano**

**Instituto Tecnológico de Massachusetts**

# Conmutadores de paquetes

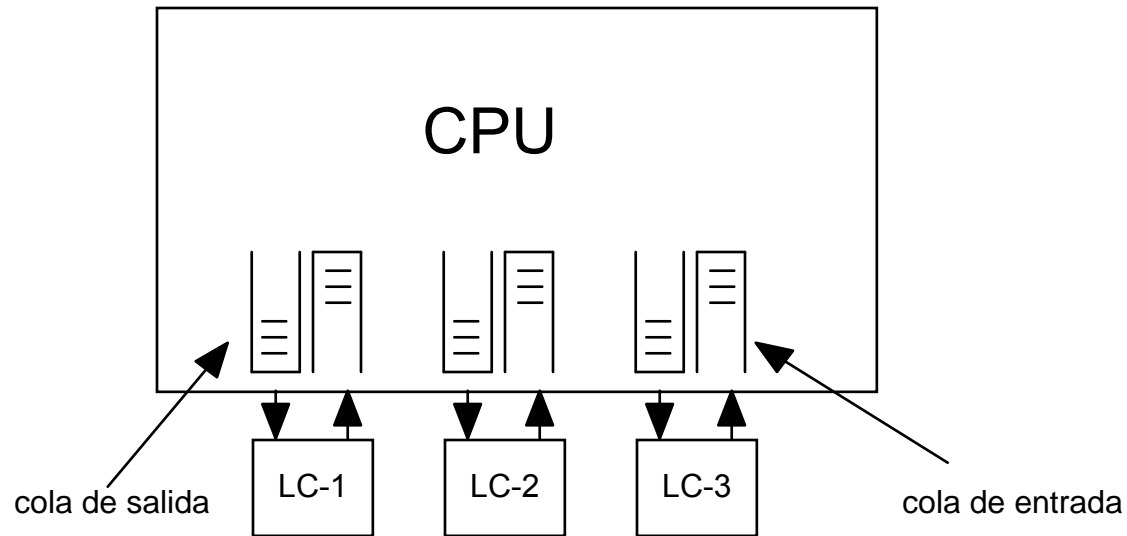
---



- Un conmutador de paquetes consta de un motor de enrutamiento (que consulta la tabla), un gestor y la estructura del conmutador en sí.
- El motor de enrutamiento consulta la dirección del paquete en una tabla de enrutamiento y determina a qué puerto de salida debe enviar el paquete:
  - El paquete se etiqueta con el número de puerto
  - El conmutador utiliza la etiqueta para enviar el paquete al puerto de salida adecuado

# Conmutadores de primera generación

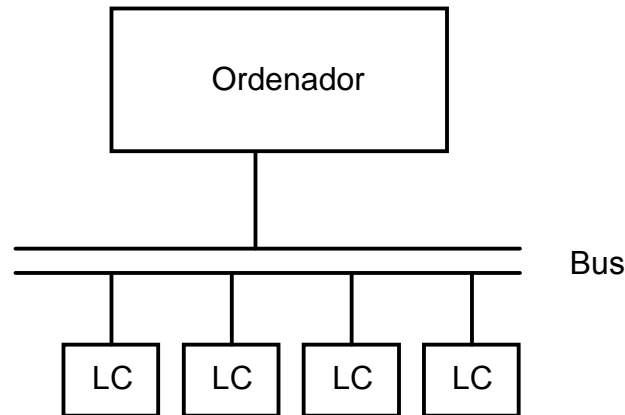
---



- **Odenador con tarjetas de líneas múltiples:**
  - La CPU sondea las tarjetas de líneas
  - La CPU procesa los paquetes
- **Es sencillo, pero el rendimiento está limitado por la velocidad del procesador y la del bus**
- **Ejemplos: puentes *Ethernet* y routers *low-end***

# Conmutadores de segunda generación

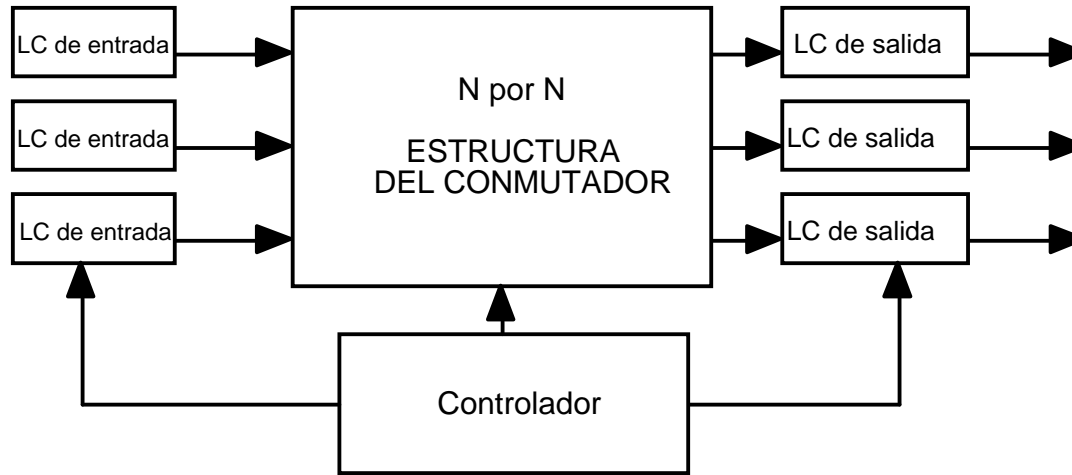
---



- **Ahora, la mayoría del proceso tiene lugar en las tarjetas de línea:**
  - Consulta de la tabla de enrutamiento, etc.
  - Las tarjetas de línea almacenan en cola los paquetes
  - La tarjeta de línea envía los paquetes al puerto de salida adecuado
- **Ventajas:** la CPU y la memoria principal ya no son el cuello de botella
- **Desventajas:** el rendimiento está limitado por la velocidad del bus:
  - El bus BW debe tener N veces la velocidad de la LC (N puertos)
- **Ejemplo:** *router* CISCO 7500

# Conmutadores de tercera generación

---



- **Se sustituye el bus compartido por una estructura de conmutador**
- **El rendimiento depende de la estructura del conmutador, pero potencialmente puede aliviar el cuello de botella del bus**

# Arquitecturas de conmutadores

---

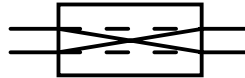
- Cola distribuida
- Cola de salida
- Cola de entrada

# Cola distribuida

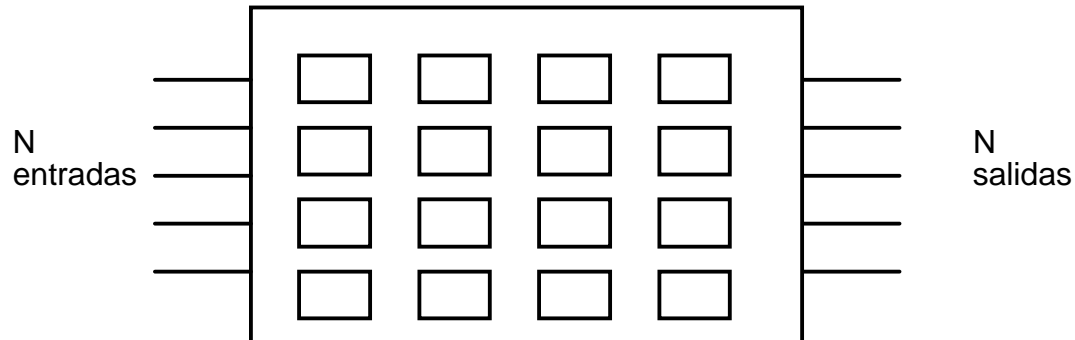
---

- **Arquitectura modular:**

El módulo principal es un *switch 2x2* , que puede estar en posición ON u OFF



- **Colas del conmutador:** ninguna, a la entrada o a la salida de cada módulo. La estructura del conmutador consiste en muchos módulos 2x2

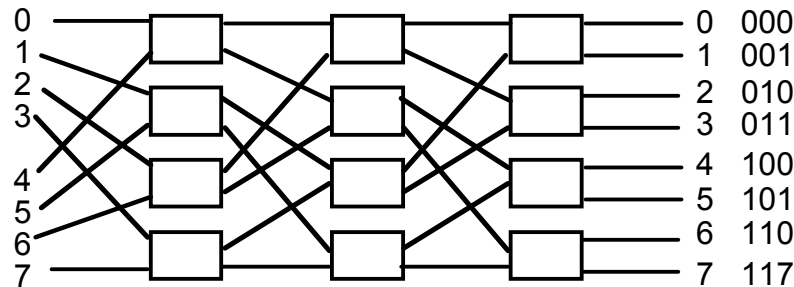


# Redes de interconexión

---

- N entradas
- $\text{Log}(N)$  etapas con  $N/2$  módulos por etapa

Ejemplo: Omega (red del tipo *shuffle / exchange*)



- Obsérvese que el orden de las entradas en una etapa es una mezcla aleatoria de las de la etapa anterior: (0,4,1,5,2,6,3 y 7)
- Se puede extender fácilmente a más etapas
- Se puede obtener cualquier salida a partir de cualquier entrada con una configuración adecuada del conmutador:
  - No se pueden completar todas las rutas simultáneamente
  - Hay exactamente una ruta entre cada par SD
  - Red de autoenrutamiento o autoencaminamiento (*self-routing*)

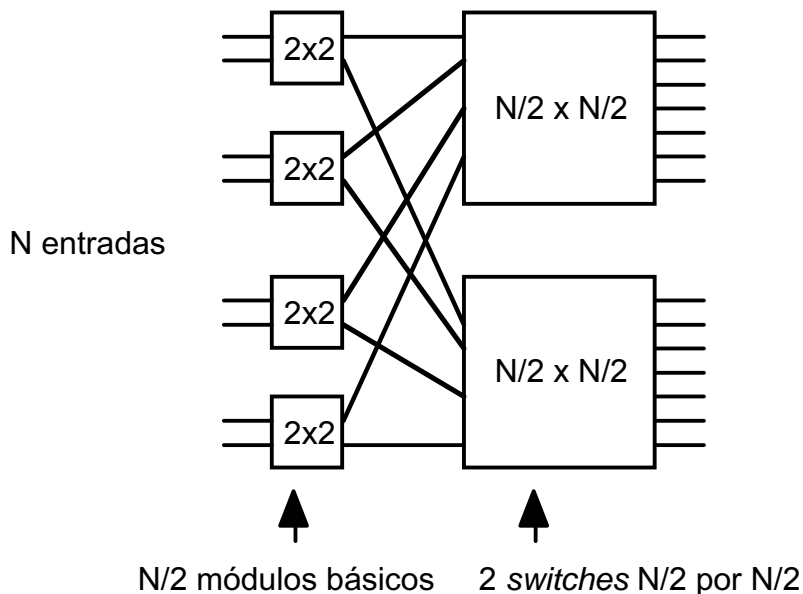
# Autoenrutamiento

---

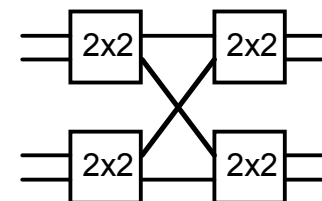
- Se emplea una etiqueta: una secuencia de  $n$  *bits* con un *bit* por cada etapa de la red:
  - Ej.: Etiqueta =  $b_3b_2b_1$
- El módulo de la etapa  $i$  mira el *bit*  $i$  de la etiqueta ( $b_i$ ) y envía el paquete hacia adelante si  $b_i=0$  y hacia atrás si  $b_i=1$
- En la red omega, la etiqueta para un puerto de destino con la dirección binaria  $abc$  es  $cba$ :
  - Ejemplo: salida 100 => etiqueta = 001
  - Obsérvese que, independientemente del puerto de entrada, la etiqueta 001 llevará a la salida 100

# Red de línea base

- Otro ejemplo de una red de interconexión multietapa
- Construida utilizando el módulo básico de *switch* 2x2
- Construcción recursiva:
  - Construir un *switch* N por N utilizando dos *switches* N/2 por N/2 y una nueva etapa de N/2 módulos básicos (2x2)
  - El *switch* N por N tiene  $\text{Log}_2(N)$  etapas, cada una con N/2 módulos básicos (2x2)



ejemplo de *switch* 4 x 4



# Contención

---

- Puede que dos paquetes quieran utilizar el mismo enlace a la vez (el mismo puerto de salida de un módulo)
- Efecto *hot spot*
- Solución: *Buffering* (almacenamiento en cola)

# Análisis de la tasa de transferencia de las redes de interconexión

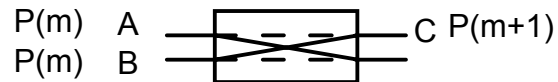
---

- Suponer que no hay almacenamiento en cola en los *switches*
- Si dos paquetes quieren utilizar el mismo puerto, se descarta uno de ellos
- Suponer que el *switch* tiene  $m$  etapas
- Tiempo de transmisión de paquetes = 1 *slot* (entre etapas)
- Nueva llegada de paquetes en las entradas, a cada *slot*:
  - Análisis de saturación (para una tasa de transferencia máxima)
  - Distribución de destinos uniforme, independiente de un paquete a otro

# Tasa de transferencia de la interconexión (continuación)

---

- Sea  $P(m)$  la probabilidad de que un paquete se transmita por el enlace en la etapa  $m$ :

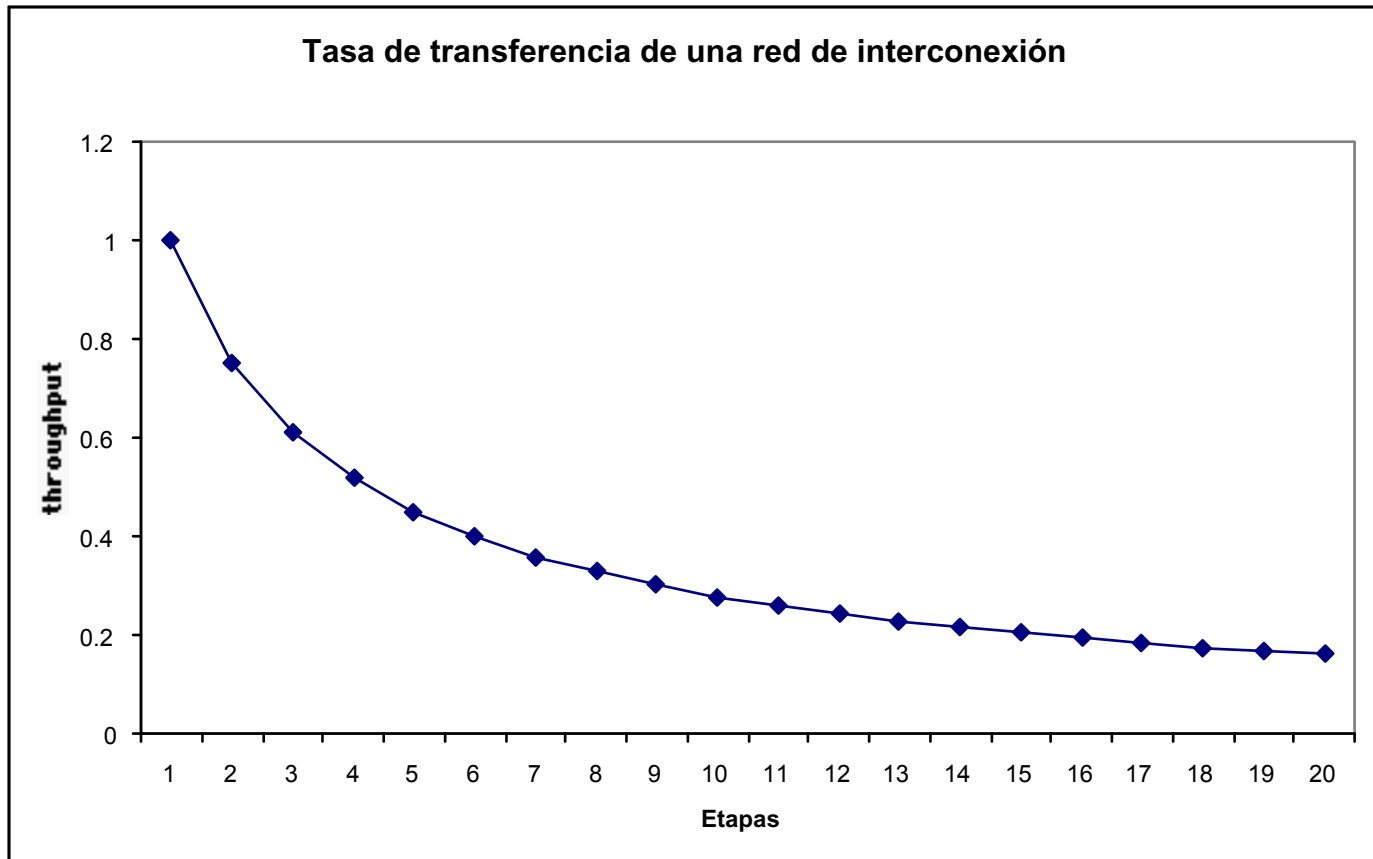


- $P(0) = 1$
- $P(m+1) = 1 - P(\text{ningún paquete en el enlace en la etapa } m+1 \text{ (enlace c)})$   
 $= 1 - P(\text{ninguna de las entradas en la etapa } m+1 \text{ elige esta salida})$
- Cada entrada tiene un paquete con probabilidad  $P(m)$  y ese paquete elegirá el enlace con probabilidad  $1/2$ . Así:

$$P(m + 1) = 1 - \left(1 - \frac{1}{2} P(m)\right)^2$$

- Ahora podemos resolverla recursivamente para  $P(m)$
- Para una red de  $m$  etapas, la tasa de transferencia (por enlace de salida) es igual a  $P(m)$ , que es la probabilidad de que haya un paquete en la salida

# Tasa de transferencia de la interconexión (continuación)



- **La tasa de transferencia se puede mejorar significativamente añadiendo colas en las etapas:**
  - Las colas aumentan el retardo
  - Relación entre el retardo y la tasa de transferencia

# Ventajas y desventajas de una arquitectura multietapa

---

- **Ventajas:**
  - Modular
  - Escalable
  - El bus (enlaces) sólo ha de ser igual de rápido que las tarjetas de línea
- **Desventajas:**
  - Retardos por pasar a través de las etapas:  
    Es posible atajar cuando las colas están vacías
  - Tasa de transferencia reducida, debido al bloqueo interno
- **Alternativas: colas externas a la estructura del conmutador:**
  - Colas de salida
  - Colas de entrada

# Arquitectura de la cola de salida

---



- **Tan pronto como llega un paquete, se transfiere a la cola de salida apropiada**
- **Suponer un sistema con *slots* (conmutador de celdas)**
- **Durante cada *slot* la estructura del conmutador transfiere un paquete de cada entrada (en caso de estar disponible) a la salida adecuada:**
  - **Ha de ser capaz de transmitir N paquetes por *slot***
  - **La velocidad del bus ha de ser N veces la tasa de la línea**
  - **Sin almacenamiento en cola en las entradas:**
    - Almacenar en cola como mucho un paquete en la entrada para un *slot***

# Análisis de colas

---

- Si las llegadas externas a cada entrada son de Poisson (tasa media  $\bar{A}$ ), cada cola de salida actúa como una cola M/D/1
  - La duración de un paquete es igual a un *slot*  $\bar{X} = \overline{X^2} = 1$

- El promedio de paquetes en cada salida viene dado por la fórmula M/G/1:

$$N_Q = \frac{2\bar{A} - (\bar{A})^2}{2(1 - \bar{A})}$$

- Obsérvese que el único retardo se debe a las colas de las salidas y ninguno es debido a la estructura del conmutador

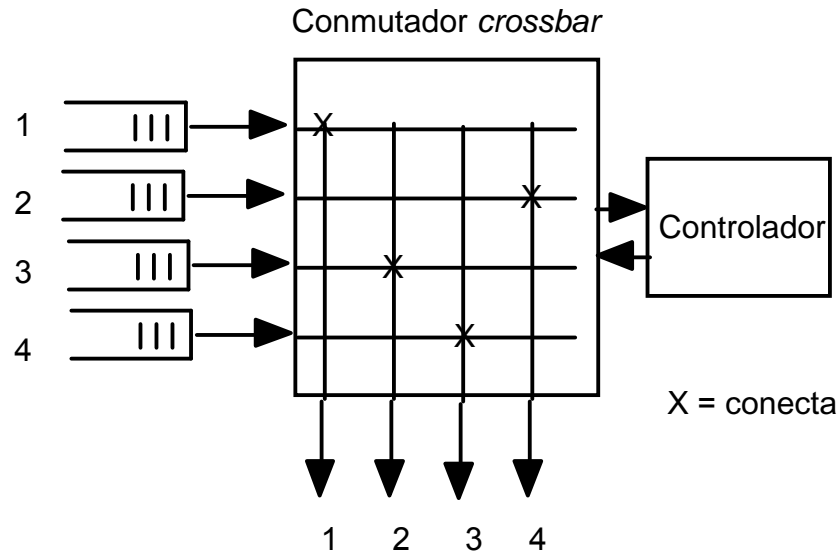
# Ventajas y desventajas de la arquitectura de las colas de salida

---

- **Ventajas:** no hay retardo ni bloqueo dentro del *switch*
- **Desventajas:**
  - La velocidad del bus debe ser  $N$  veces la de la línea  
Impone un límite práctico sobre el tamaño y la capacidad del *switch*
- **Colas de salida compartidas:** las colas de salida se implementan en la memoria compartida mediante un listado común
  - Requiere menos memoria (debido al multiplexado estadístico)
  - La memoria debe ser rápida

# Arquitectura de las colas de entrada

- Los paquetes hacen cola en la entrada en vez de en la salida:
  - No es necesario que la estructura del conmutador sea tan rápida



- Durante cada *slot*, el controlador establece las conexiones *crossbar* para transmitir los paquetes desde la entrada hasta las salidas:
  - Máximo un paquete desde cada entrada
  - Máximo un paquete hacia cada salida
- Bloqueo de la cabeza de la línea (HOL) –cuando los paquetes que encabezan dos o más colas de entrada se destinan a la misma salida, solo se puede transmitir uno de ellos y los otros se bloquean

# Análisis de la tasa de transferencia de *switches* con colas de entrada

---

- El bloqueo de HOL limita la tasa de transferencia, porque algunas entradas (y, consecuentemente, salidas) se mantienen vacías durante un *slot* incluso cuando tienen en su cola otro paquete para enviar
- Examinar un *switch* NxN y suponer, una vez más, que las entradas están saturadas (siempre tienen un paquete para enviar)
- Tráfico uniforme => cada paquete está destinado a cada salida con la misma probabilidad ( $1/N$ )
- A continuación, examinar sólo los paquetes que encabezan cada cola (¡hay N!)

# Análisis de la tasa de transferencia (continuación)

---

- Sea  $Q_m^i$  el número de paquetes HOL destinados al nodo  $i$  al final del *slot* número  $m$ :

$$Q_m^i = \text{máx}(0, Q_{m-1}^i + A_m^i - 1)$$

- Donde:

$A_m^i$  = número de nuevos mensajes en la HOL dirigidos al nodo  $i$  que alcanzan la cabeza de la línea (HOL) durante el *slot*  $m$ . Ahora:

$$P(A_m^i = l) = \binom{C_{m-1}}{l} (1/N)^l (1 - 1/N)^{C_{m-1} - l}$$

- Donde:

$C_{m-1}$  = número de mensajes en la HOL que partieron durante el *slot*  $m-1$  = número de nuevas llegadas a la cabeza de la línea (HOL)

- A medida que  $N$  tiende a infinito,  $A_m^i$  se convierte en Poisson de tasa  $C/N$ , donde  $C$  es el promedio de salidas por *slot*

# Análisis de la tasa de transferencia (continuación)

---

- En estado estacionario,  $Q_i$  actúa como una M/D/1 de tasa  $\bar{A}$  y, al igual que antes:

$$\bar{Q}^i = \frac{2\bar{A} - (\bar{A})^2}{2(1 - \bar{A})}$$

- Obsérvese, sin embargo, que el número total de paquetes que se dirigen a las salidas es  $N$  (número de paquetes en la HOL). Así:

$$\sum_{i=1}^N Q^i = N \quad \Rightarrow \quad \bar{Q}^i = \frac{2\bar{A} - (\bar{A})^2}{2(1 - \bar{A})} = 1$$

Ahora podemos resolverla utilizando la ecuación cuadrática y obtenemos:

$$\bar{A} = \textit{utilización} = 2 - \sqrt{2} \approx 0.58$$

# Resumen de *switches* con colas de entrada

---

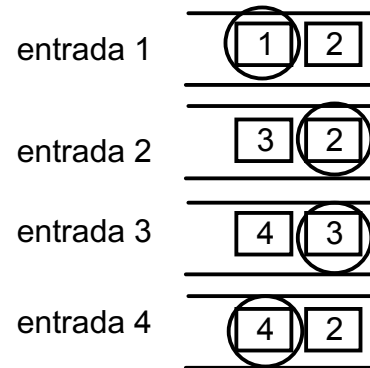
- La tasa de transferencia máxima de un *switch* con colas de entrada está limitada a un 58% por el bloqueo de la HOL (para un tamaño N):
  - Suponer un tráfico uniforme y un servicio FCFS
- Ventajas de las colas de entrada:
  - Sencillez
  - Tasa del bus = tasa de la línea
- Desventajas: limitación de la tasa de transferencia

# Cómo superar el bloqueo de la HOL

---

- Si las entradas pueden transmitir paquetes que no encabezan sus colas, la tasa de transferencia se puede mejorar enormemente (sin FCFS)

**Ejemplo:**



- ¿Cómo decide el controlador a qué salida transferir cada entrada?

# Matriz de paquetes pendientes (*backlog*)

---

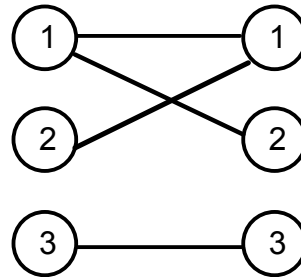
		salida		
		1	2	3
entrada	1	3	3	0
	2	2	0	0
	3	0	0	2

- Cada entrada de la matriz de paquetes pendientes representa el número de paquetes en la cola de la entrada  $i$  destinados a la salida  $j$
- Durante cada *slot* el controlador puede transmitir como máximo un paquete desde cada entrada hasta cada salida:
  - El controlador debe elegir un paquete (como máximo) de cada fila y columna de la matriz de paquetes pendientes
  - Esto se puede hacer resolviendo un algoritmo de *matching* en grafo bipartito
  - El grafo bipartito consiste en  $N$  nodos que representan las entradas y  $N$  nodos que representan las salidas

# Representación del grafo bipartito

---

- En el grafo, un arco o enlace une una entrada a una salida si hay un paquete en la matriz de paquetes pendientes esperando para ser transmitido desde esa entrada a esa salida:
  - El grafo bipartito para la anterior matriz de paquetes pendientes es:



- **Definición:** un *matching* es un conjunto de arcos en el que no hay dos arcos que compartan un nodo:
  - Encontrar un *matching* en el grafo bipartito equivale a encontrar un conjunto de paquetes en el que no haya dos paquetes que compartan una fila o columna en la matriz de paquetes pendientes
- **Definición:** un *matching* máximo es un *matching* con el mayor número posible de arcos:
  - Encontrar el *matching* máximo equivale a encontrar el mayor conjunto de paquetes que se pueden transmitir simultáneamente

# Matching Máximo

---

- Existen algoritmos para encontrar el *matching* máximo
- Los algoritmos más conocidos tienen  $O(N^{2.5})$  operaciones:
  - Demasiado largos para un tamaño  $N$
- Alternativas:
  - Soluciones subóptimas
  - *Matching* maximal: un *matching* que no se puede aumentar para una matriz de paquetes pendientes dada
  - Para el ejemplo anterior:
    - (1-1,3-3) es maximal
    - (2-1,1-2,3-3) es máximo
- Cuestión: el número de arcos en un *matching* maximal  $\geq 1/2$  el número de arcos en un *matching* máximo

# **Cómo alcanzar una tasa de transferencia del 100% en un *switch* con colas de entrada**

---

- **Encontrar un *matching* máximo durante cada *slot* de tiempo no elimina los efectos del bloqueo de la HOL:**
  - Debe ver más allá de un *slot* cada vez al gestionar
- **Definición: un grafo bipartito con peso es un grafo bipartito con los costes asociados a los arcos**
- **Definición: un *matching* de peso máximo es un *matching* con los pesos máximos de los arcos**
- **Teorema: un controlador que elige durante cada *slot* de tiempo el *matching* de peso máximo en el que el peso del enlace (i,j) es igual a la longitud de la cola (i,j), logra una total utilización (una tasa de transferencia del 100%):**
  - **Demostración: ver “Achieving 100% throughput in an input queued switch” de N. McKeown, *et. al.*, IEEE Transactions on Communications, Ag. 1999.**