Paradigmas de conmutación

Area de Ingeniería Telemática
http://www.tlm.unavarra.es

Arquitectura de Redes, Sistemas y Servicios
Grado en Ingeniería en Tecnologías de Telecomunicación, 2º

Temario

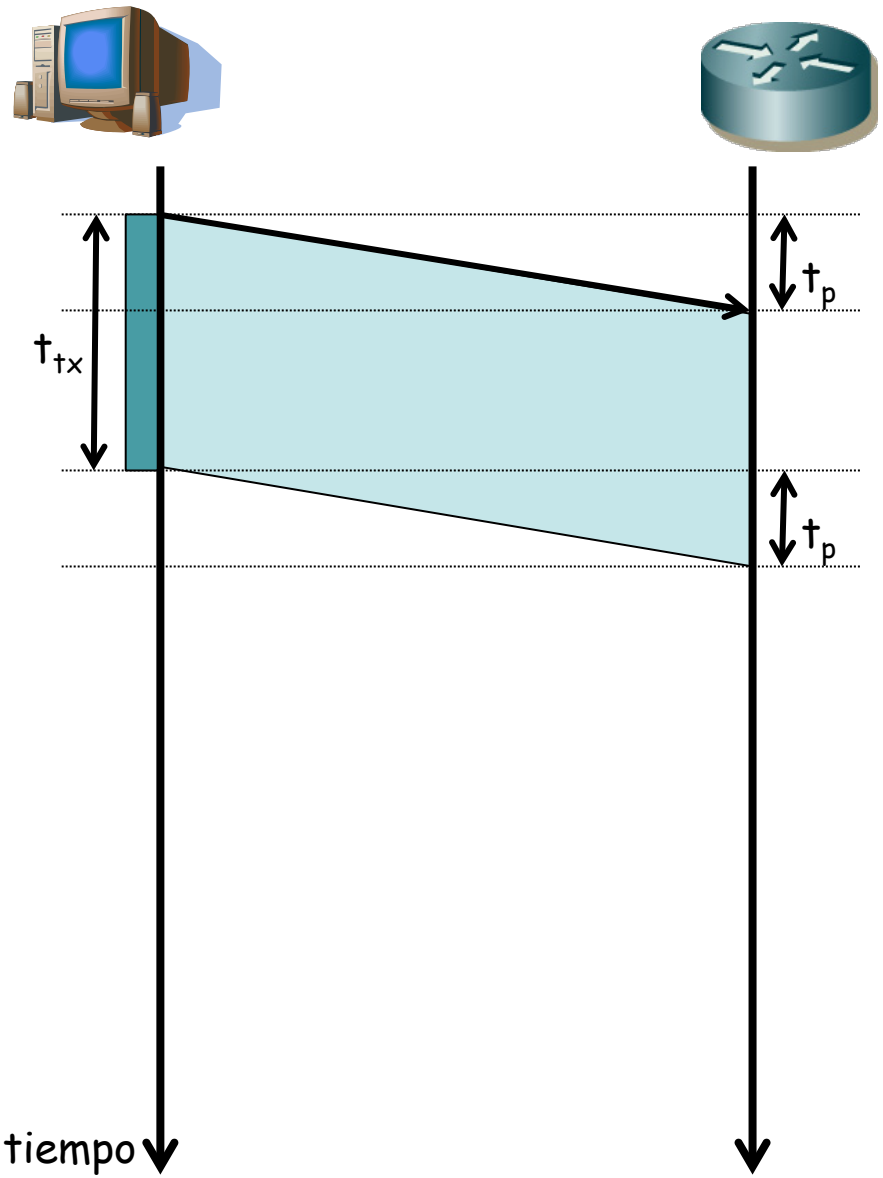
0. Introducción
1. **Arquitecturas de conmutación y protocolos**
   - Elementos, protocolos y arquitecturas de protocolos
   - Arquitecturas OSI y TCP/IP
   - Servicios, interfaces, funcionalidades
   - Conmutación de circuitos y de paquetes
   - **Retardos de** transmisión, propagación, **procesado, cola**
   - **Variación del retardo, pérdidas y throughput**
2. Introducción a las tecnologías de red
3. Control de acceso al medio
4. Conmutación de circuitos
5. Encaminamiento
6. Transporte fiable
7. Programación para redes y servicios

Objetivos

- Diferenciar y saber trabajar con **retardos** de propagación y transmisión en redes con almacenamiento y reenvío
- Comprender el origen y comportamiento general del retardo en cola
- Saber que existe la **variación del retardo** en redes de conmutación de paquetes, a qué se debe y qué efectos tiene
- Conocer la existencia y los motivos de las **pérdidas** en redes de conmutación de paquetes
- Entender qué es un cuello de botella

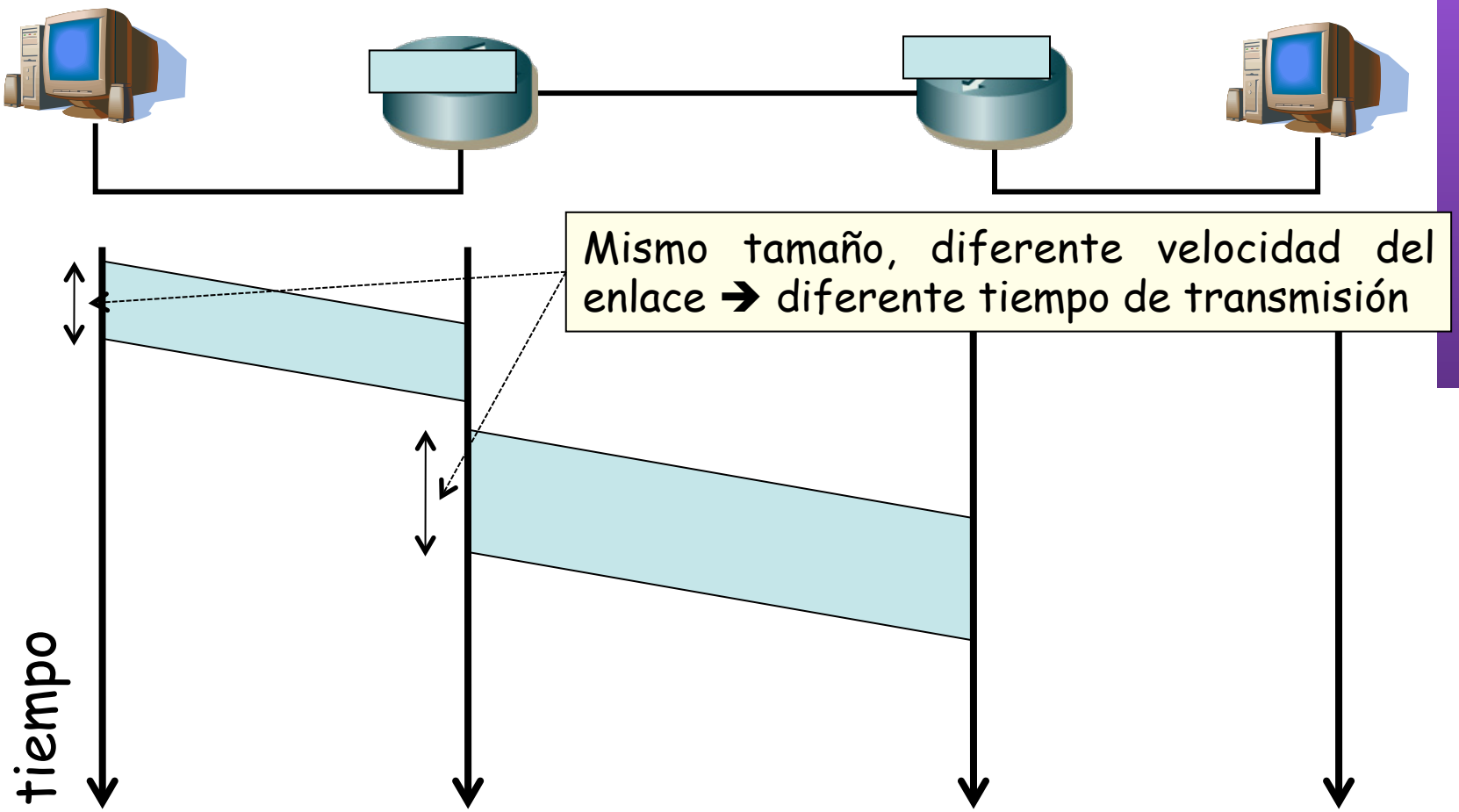
Retardos en redes

Transmisión y propagación



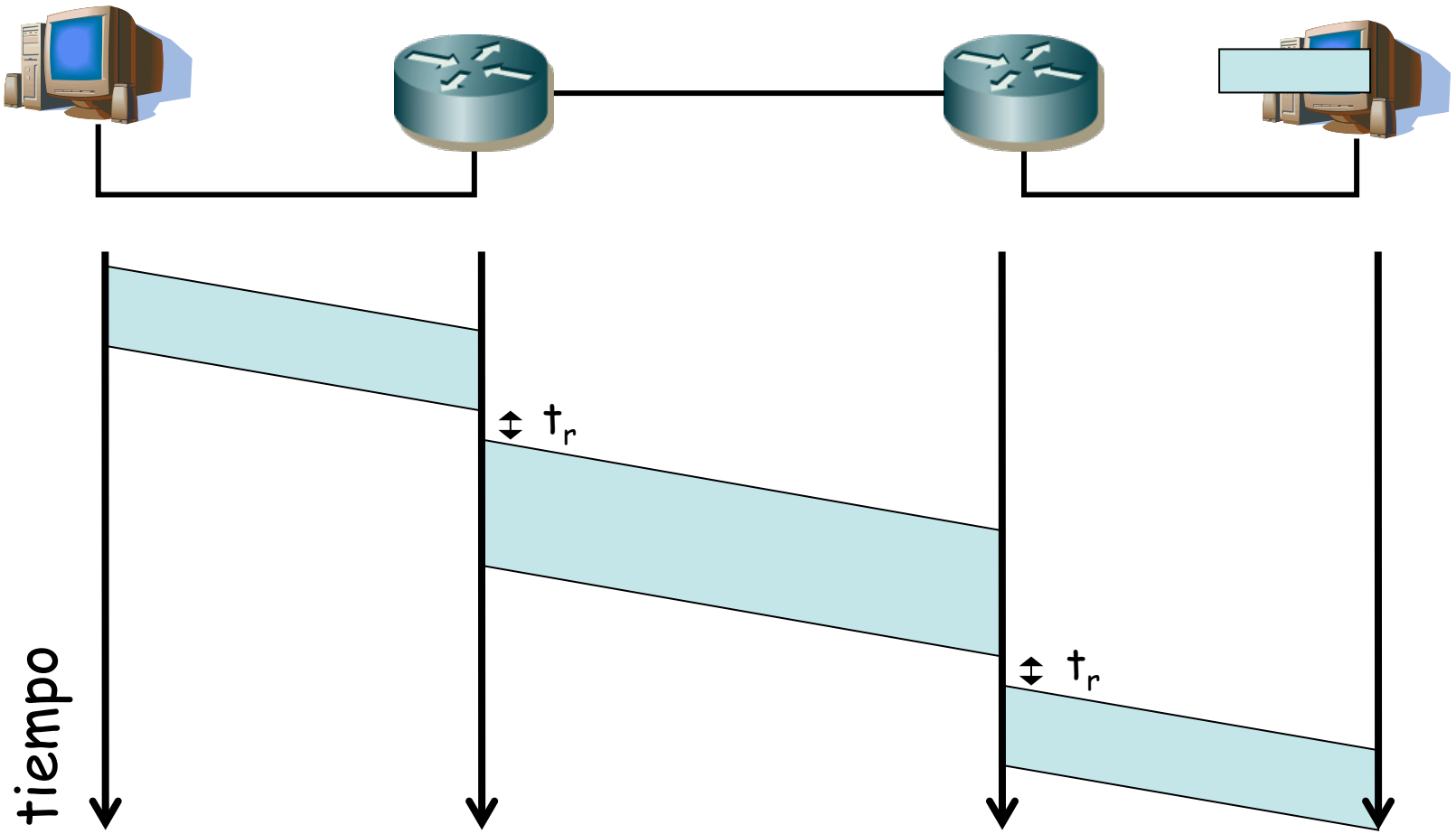
Store-and-forward

- El paquete completo debe llegar al conmutador de paquetes antes de que lo pueda retransmitir (. . .)



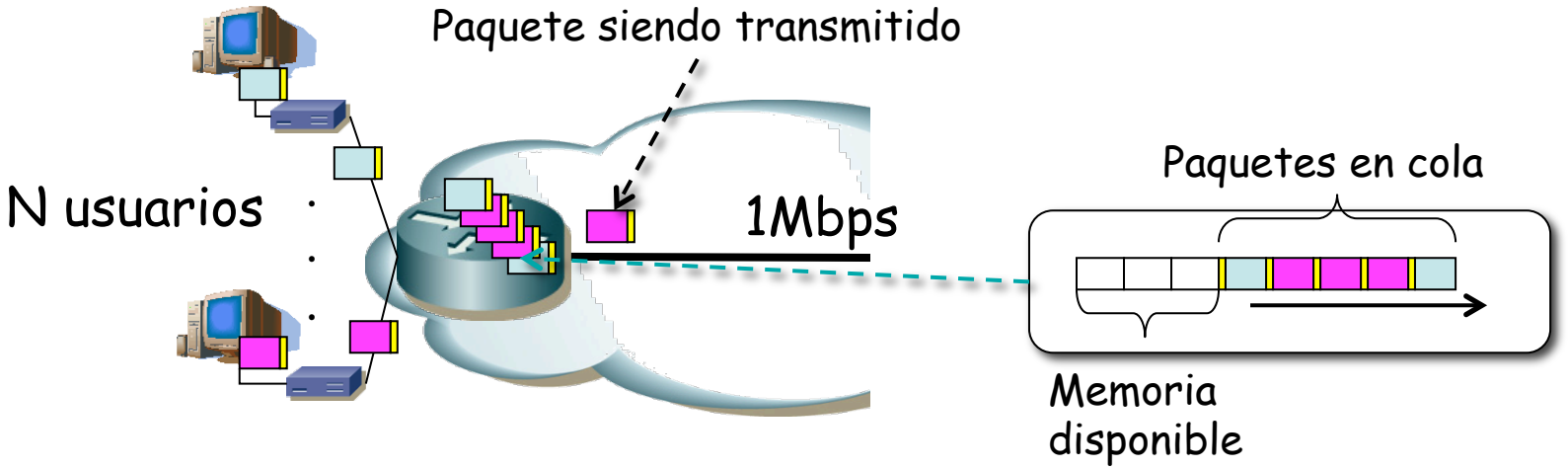
Tiempo de procesado

- El conmutador debe tomar una decisión para cada paquete, la cual lleva tiempo (t_r)



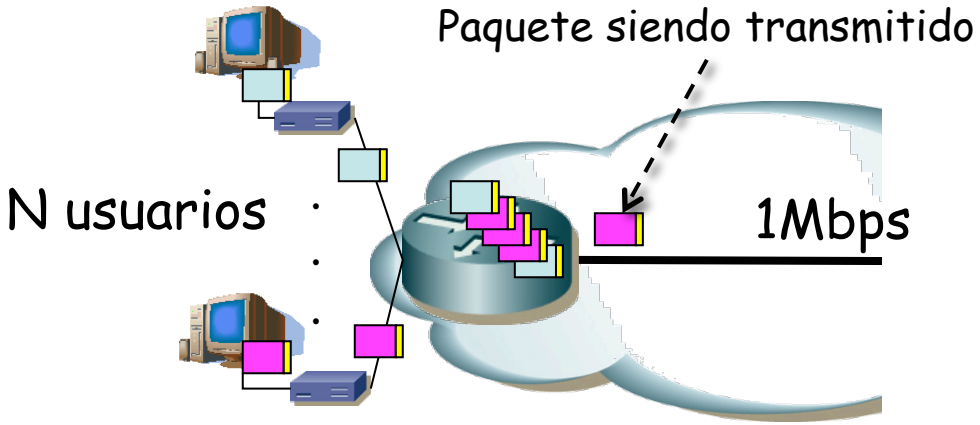
Retardo en cola

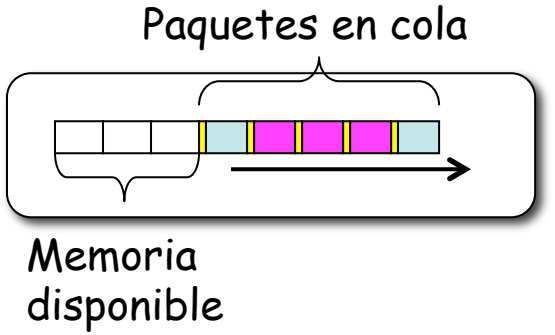
- Los paquetes pueden llegar al conmutador a una velocidad mayor que la capacidad del enlace de salida
- O pueden llegar varios simultáneamente por enlaces diferentes pero solo puede salir uno a la vez
- El comutador los almacena en memoria hasta poder enviarlos
- Esperan en una *cola* (normalmente en el interaz de salida)
- Si no queda espacio en memoria para almacenar un paquete, normalmente éste se pierde (*drop-tail policy*)



Planificación

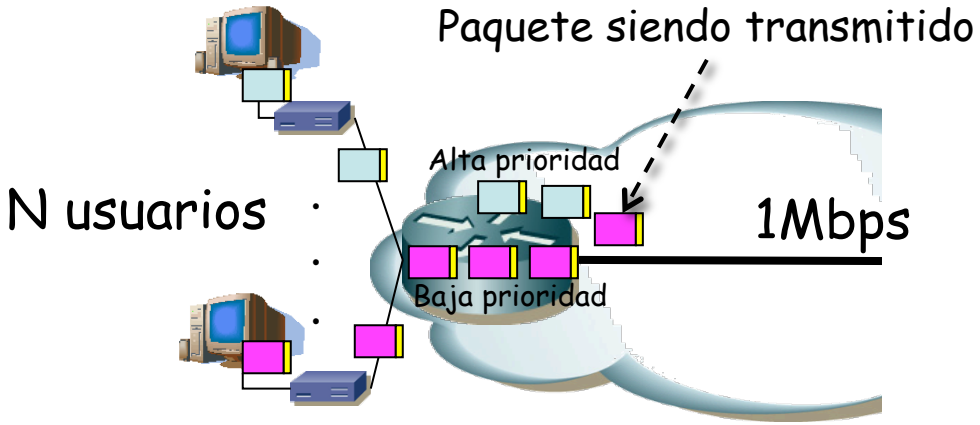
- ¿ En qué orden se atiende a los paquetes que hay en la cola ?
- FCFS: *First Come First Served*
  - También llamado FIFO (First In First Out)
  - Trato equitativo a diferentes flujos/usuarios/aplicaciones
  - Un paquete que requiera bajo retardo (voz) tiene que esperar a que se sirvan todos los anteriores en la cola
  - Asegurar límites en el retardo requiere caracterizar todas las fuentes de tráfico

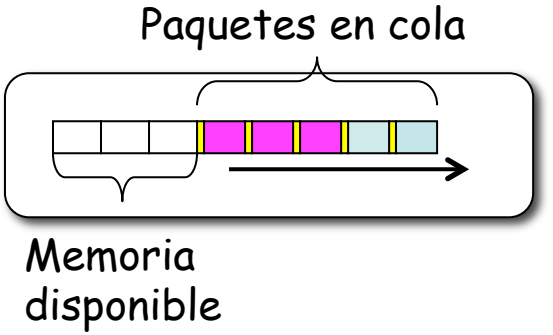




Planificación

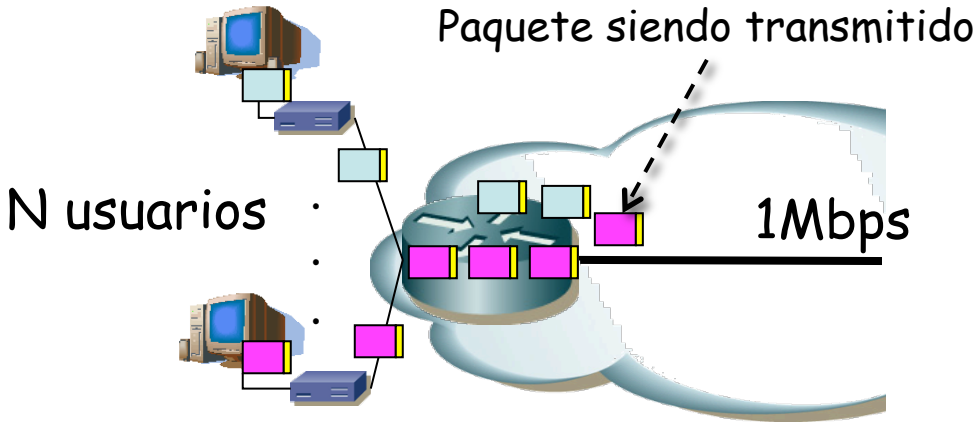
- ¿ En qué orden se atiende a los paquetes que hay en la cola ?
- **¿Otras alternativas?**
- Prioridades:
  - Clasificar los paquetes de entrada
  - Cada clase tiene una prioridad diferente
  - Solo se envían paquetes de una clase si las clases de prioridad superior no tienen paquetes en la memoria del router

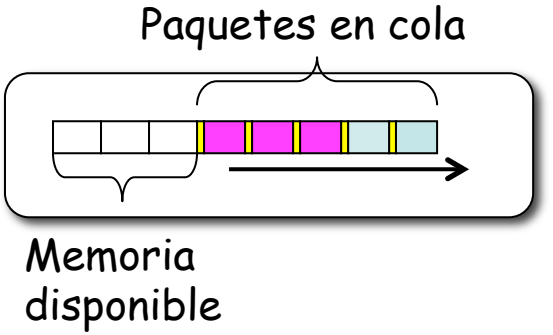




Planificación

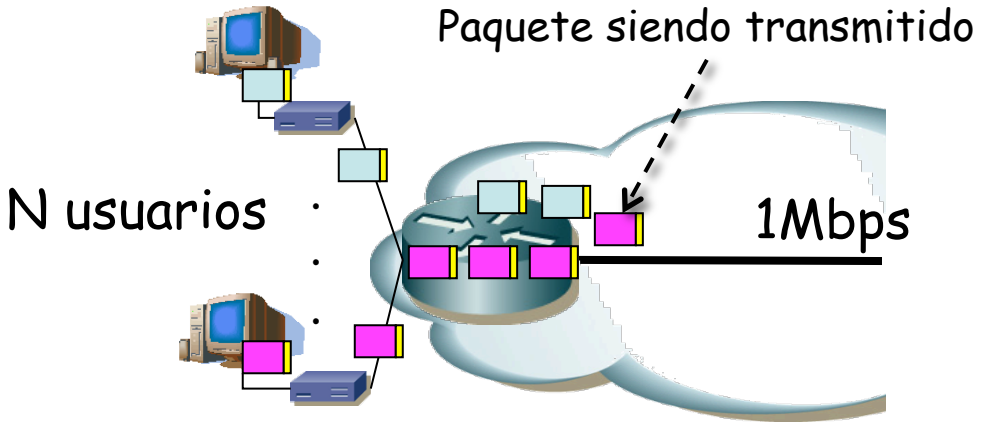
- ¿ En qué orden se atiende a los paquetes que hay en la cola ?
- **¿Otras alternativas?**
  - Round Robin
  - Weighted Round Robin
  - Deficit Round Robin
  - Generalized Processor Sharing
  - Weigthed Fair Queueing
  - ...

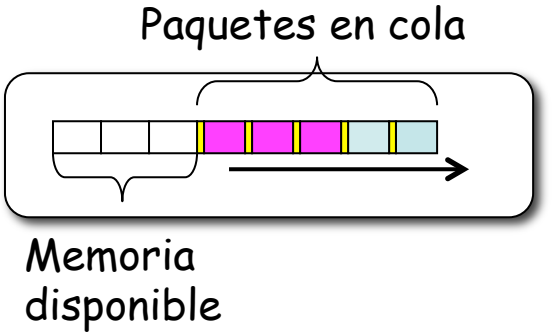




Planificación

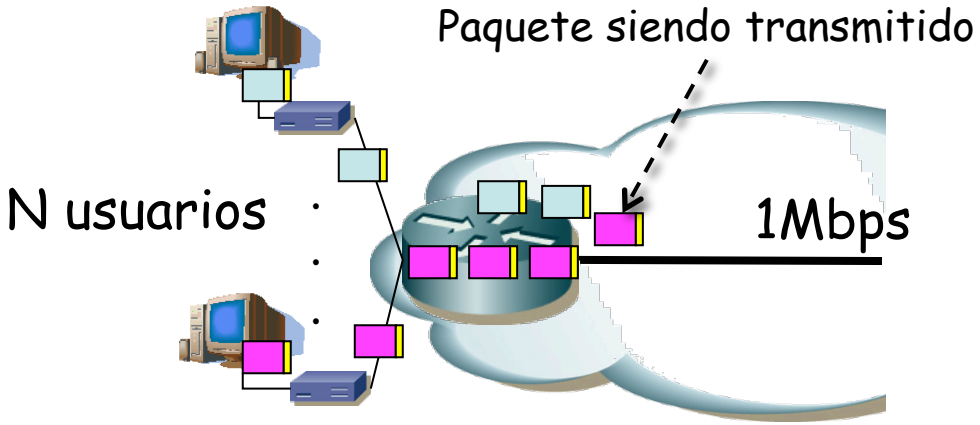
- ¿ En qué orden se atiende a los paquetes que hay en la cola ?
- ¿Otras alternativas?
- **Buscan:**
  - Hacer un reparto “justo” (*max-min fair*)
  - Protección: un flujo no pueda acaparar todos los recursos
  - Asegurar límites (al retardo, jitter, pérdidas...) predecibles
  - Simplicidad de implementación

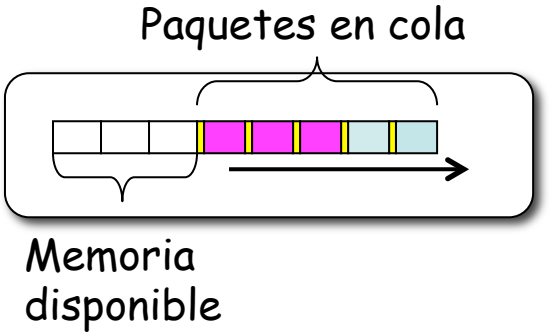




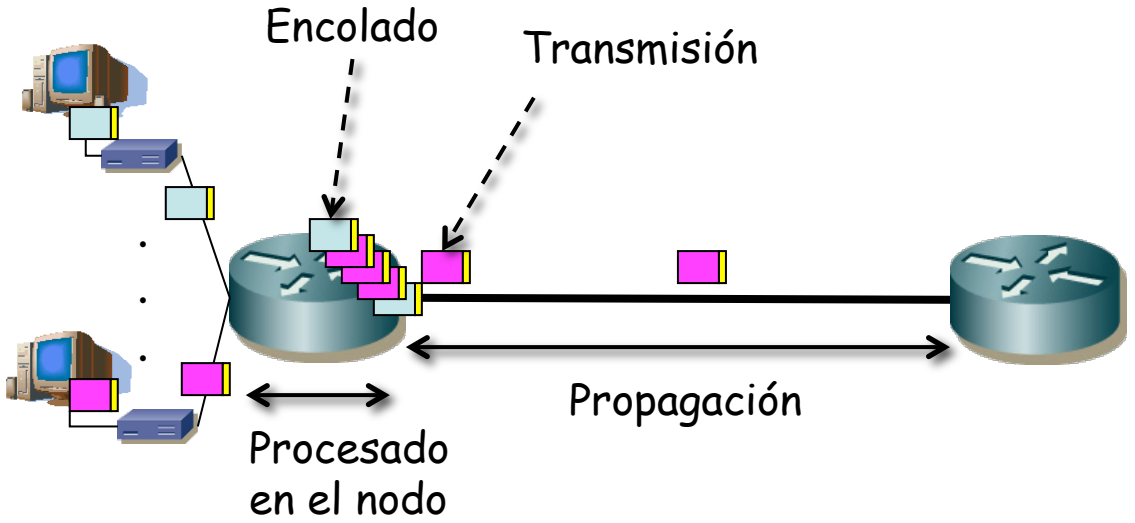
Planificación

- ¿ En qué orden se atiende a los paquetes que hay en la cola ?
- ¿Otras alternativas?
- **Necesario para:**
  - Ofrecer Calidad de Servicio (QoS, *Quality of Service*)





Retardos



$$d_{\text{nodo}} = d_{\text{proc}} + d_{\text{cola}} + d_{\text{trans}} + d_{\text{prop}}$$

d_{proc} = tiempo de procesado
- Unos µs

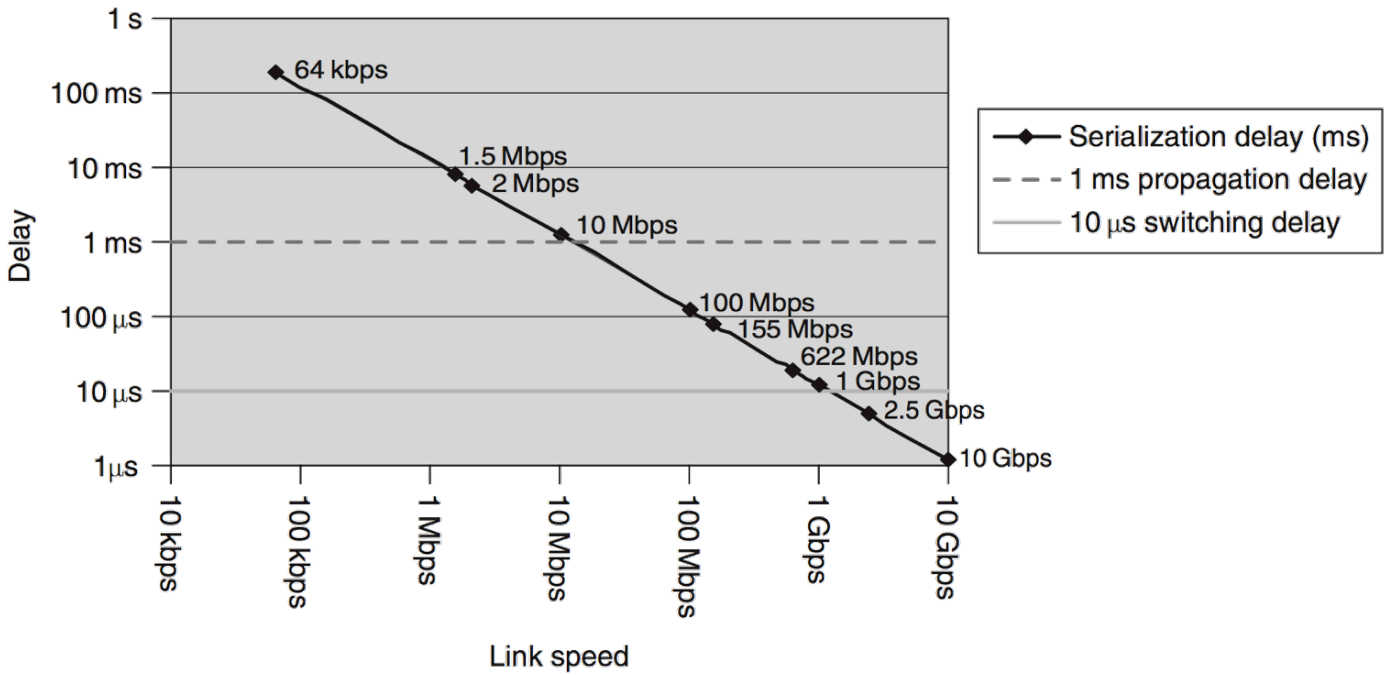
d_{cola} = retardo en cola
- Depende de la congestión

d_{trans} = retardo transmisión
- = L/R, significativo en enlaces de baja velocidad

d_{prop} = retardo propagación
- De unos µs a centenares de ms

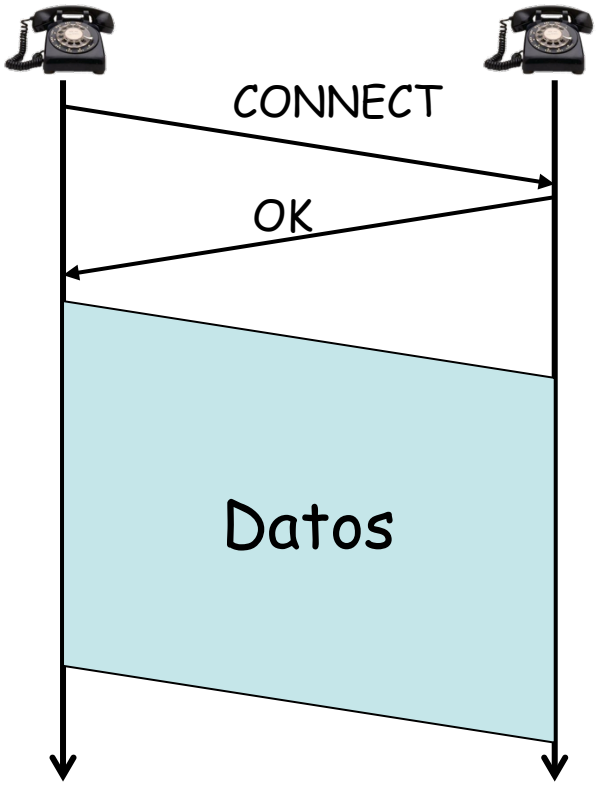
Retardos constantes

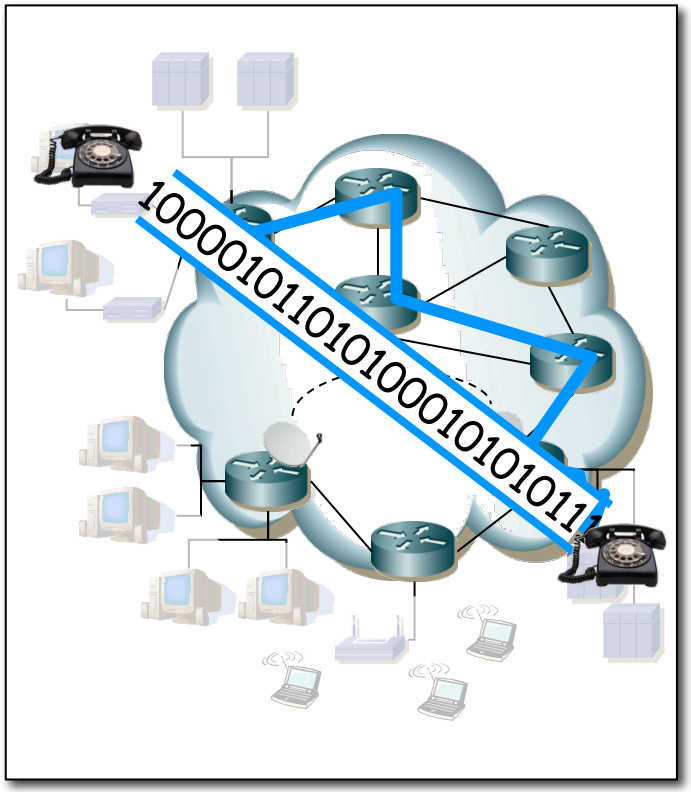
- Por cada salto intermedio habrá
  - Tiempo de transmisión (serialización)
  - Tiempo de propagación
  - Tiempo de procesado/conmutación
- Ejemplo comparativo
  - Paquete de 1500 bytes
  - Unos 200Km de fibra : 1ms de propagación



Comparar con...

- Conmutación de circuitos
- Retardo en conmutación analógica despreciable
- En digital del orden del tiempo de transmisión de un byte

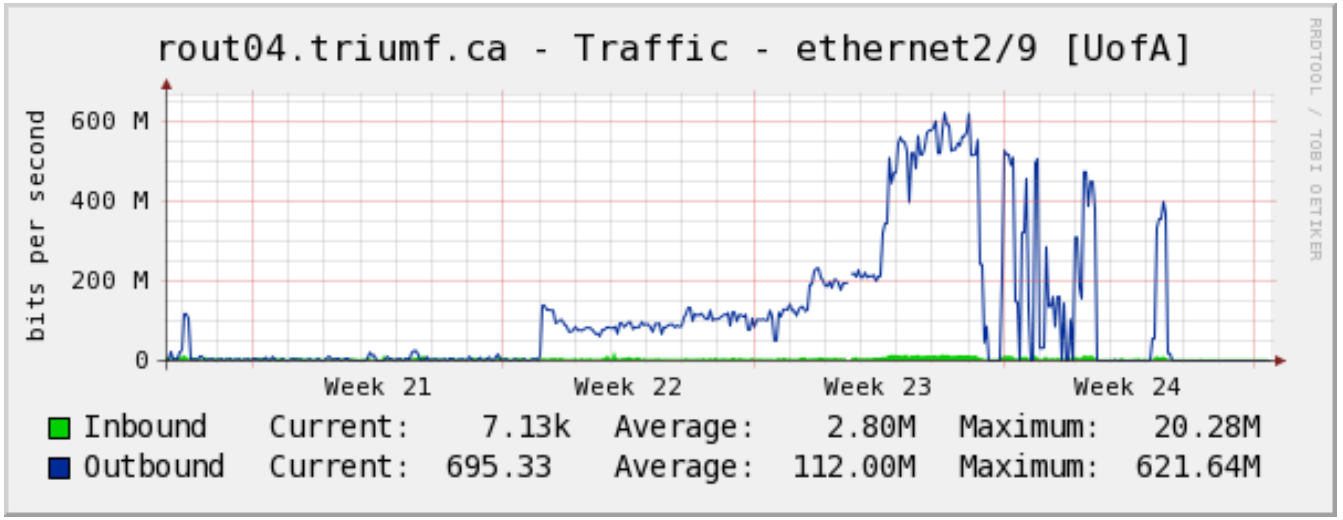




Throughput

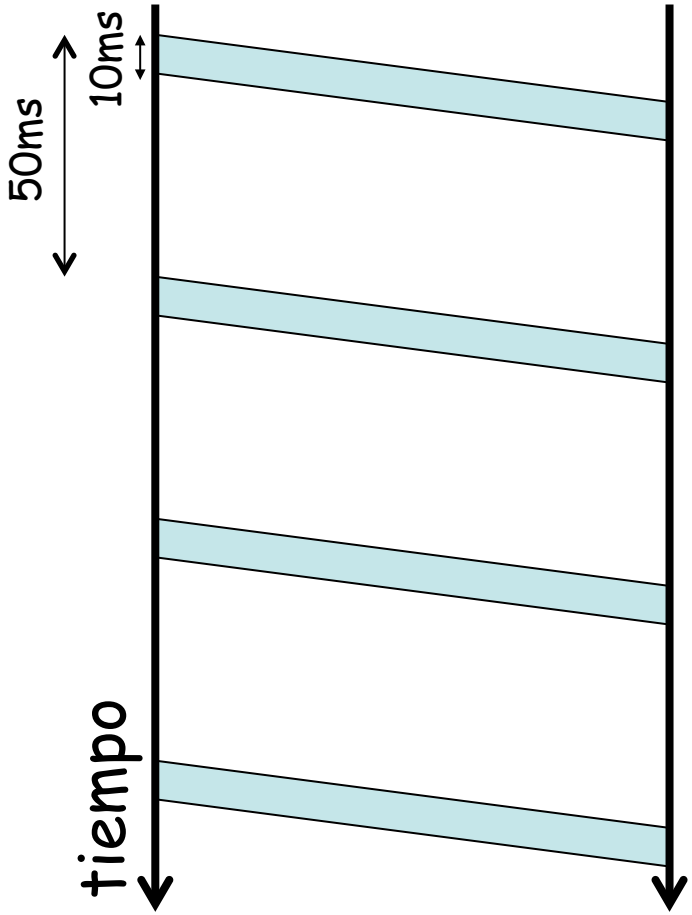
Bandwidth / Throughput

- **Throughput instantáneo**: tasa a la cual se transmiten o transfieren o reciben datos
- En el límite, si hay paquete es el bitrate del enlace y si no es 0
- **Throughput medio**: cantidad de datos transferidos en un intervalo de tiempo divididos por ese tiempo
- Ejemplo:
  - Transferencia de fichero de tamaño F bits en un tiempo T segundos ha sido en media a F/T bps



Throughput medio

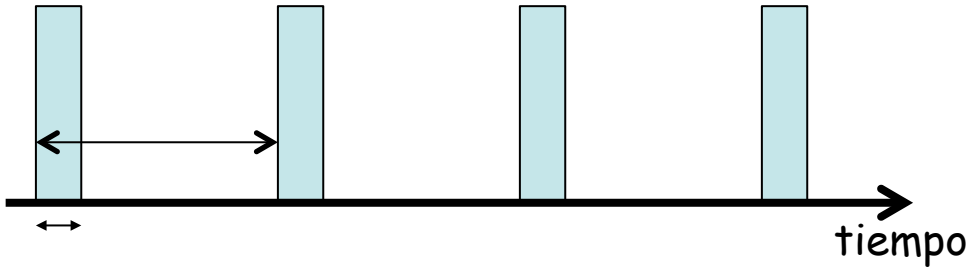
- Supongamos un paquete de 1250 bytes exactamente cada 50 ms en un enlace a 1 Mbps
- Tiempo de transmisión = 1250 x 8 / 1.000.000 = 10 ms



- Por ejemplo cada segundo se habrían transmitido:

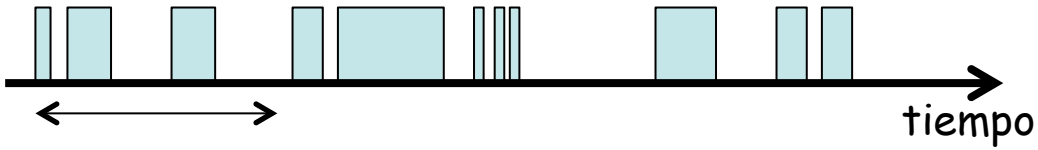
  1000 ms / 50 ms/pkt = 20 pkts
- 20 pkts/s
- 1250 bytes/pkt x 20 pkts/s =

  = 200.000 bps = 200 Kbps
- Es independiente del tiempo de propagación

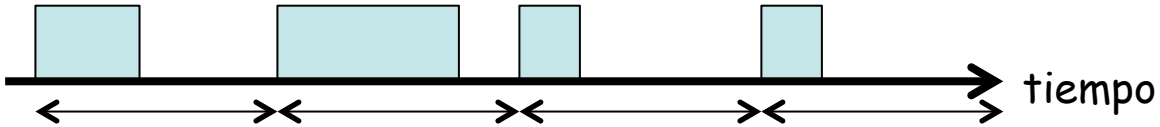


Throughput medio

- Supongamos unas llegadas más irregulares (y más habituales)
- Paquetes de diferentes tamaños
- Paquetes con separaciones variables

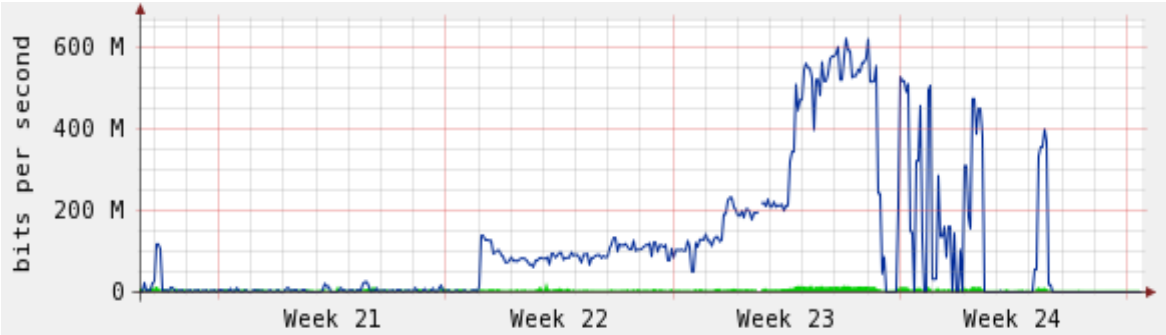


- ¿Cuál es el throughput medio?
- Podemos tomar un intervalo “grande” y agregar los bytes enviados o recibidos en ese intervalo
- En cada intervalo es la cantidad de bytes transmitidos entre la anchura del intervalo



Throughput medio

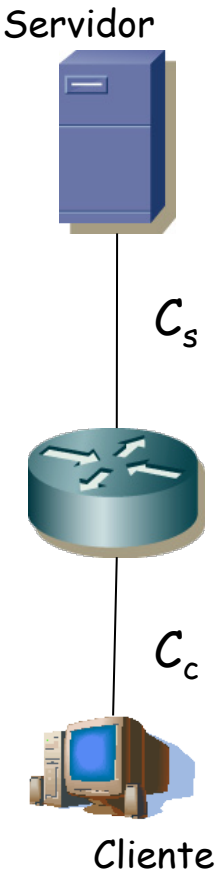
- Según anchura del intervalo
- Promedios más o menos “groseros”



Ejemplo

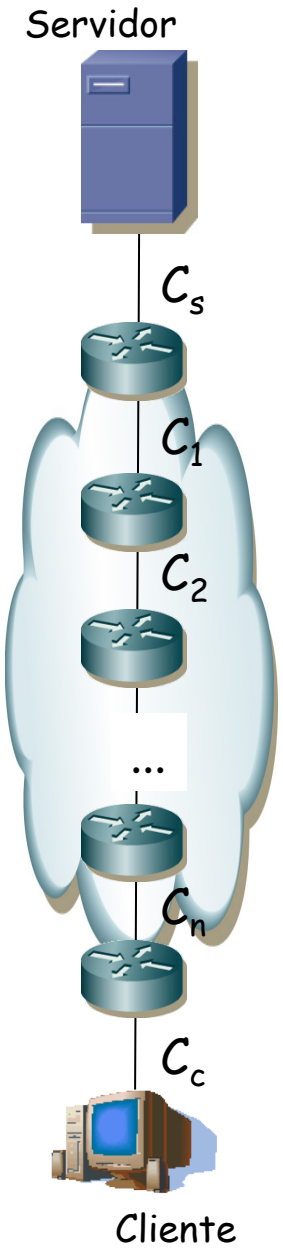
- Cliente, servidor y un conmutador de paquetes intermedio
- Capacidades de los enlaces C_S y C_C
- Cliente se descarga un fichero de tamaño F del servidor
- No hay más tráfico
- ¿Cuál es el mínimo tiempo que tardará el cliente en obtenerlo?

- Máximo throughput: mín{C_S,C_C}
- Mínimo tiempo: F/mín{C_S,C_C}
- "Cuello de botella" (*Bottleneck)*



Ejemplo 2

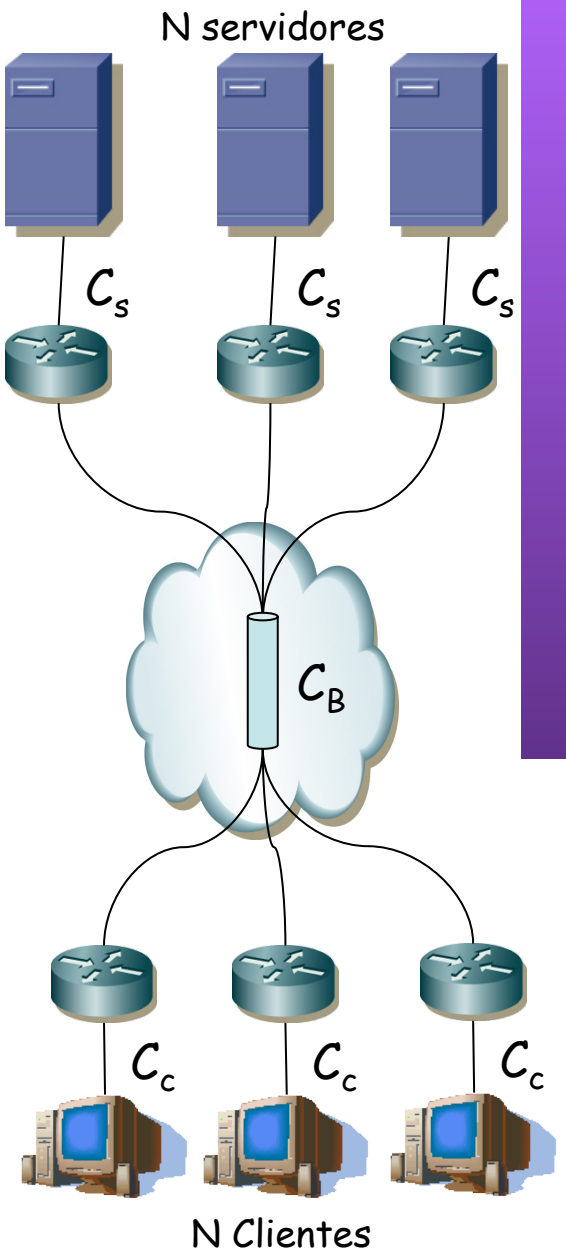
- Cliente, servidor y un conjunto de conmutadores de paquetes intermedios
- Capacidades de los enlaces C_S, C_C y C_i i=1...n
- Cliente se descarga un fichero de tamaño F del servidor
- No hay más tráfico
- Máximo throughput: mín$\{C_S,C_C,C_i\}_{i=1..n}$
- Hoy en día los enlaces en el núcleo de la red son de mucha mayor capacidad que los del acceso
- Con eso de nuevo máximo throughput: mín$\{C_S,C_C\}$



Ejemplo 3

- N Clientes y servidores
- Cada cliente descarga un fichero de un servidor
- Comparten un enlace en la red de capacidad C_B
- La capacidad en el resto de enlaces en el núcleo de la red es superior
- No hay más tráfico

- ¿Dónde está ahora el cuello de botella?
- Supongamos $C_C < C_S < C_B$
- Supongamos que la capacidad C_B se reparte equitativamente entre los N flujos
- Máximo throughput: mín{C_C,C_B/N}
- El cuello de botella puede estar en el acceso o en el núcleo



Pérdidas

Pérdidas

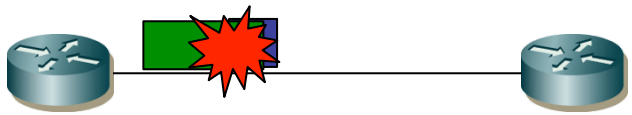
- Los paquetes podrían no llegar nunca a su destino

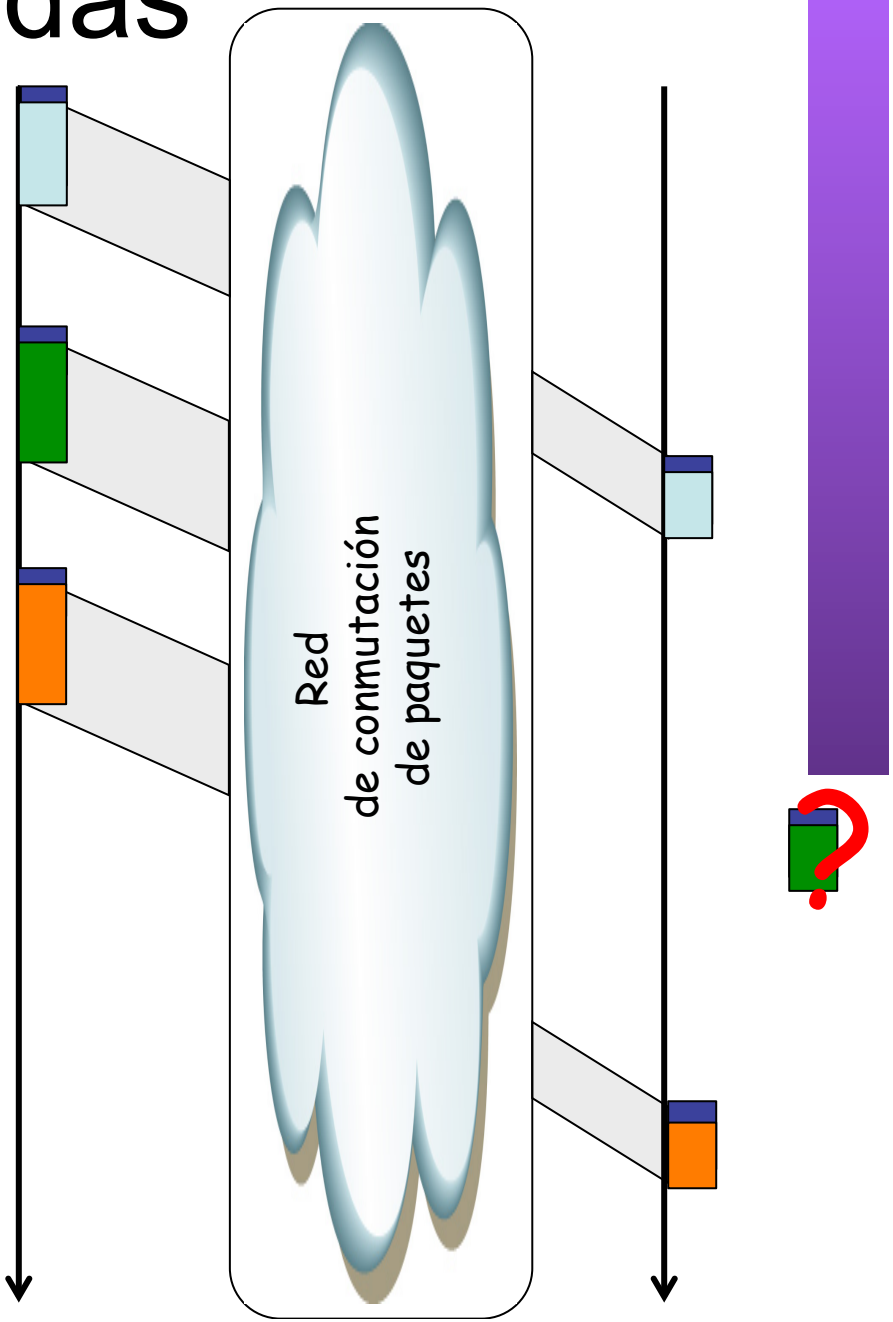
Posibles motivos

- Se corrompió y fue descartado en algún nodo de la red (CRCs)
- BER = Bit Error Rate
- Aproxima a la probabilidad de error de bit p_{err}
- Probabilidad de algún error en un paquete de N bits:

$$p_{epk} = 1 - (1 - p_{err})^N$$

- Asumiendo errores indep. (no ráfagas)
- Sin código "corrector" de errores
- Ejemplo: $p_{err}=10^{-6}$, N=12.000 ➔ ➔ $p_{epk}\approx 10^{-2}$
- (...)



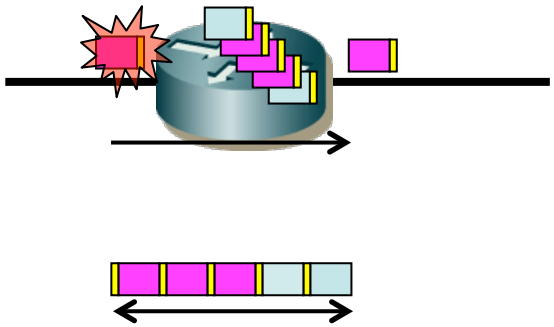


Pérdidas

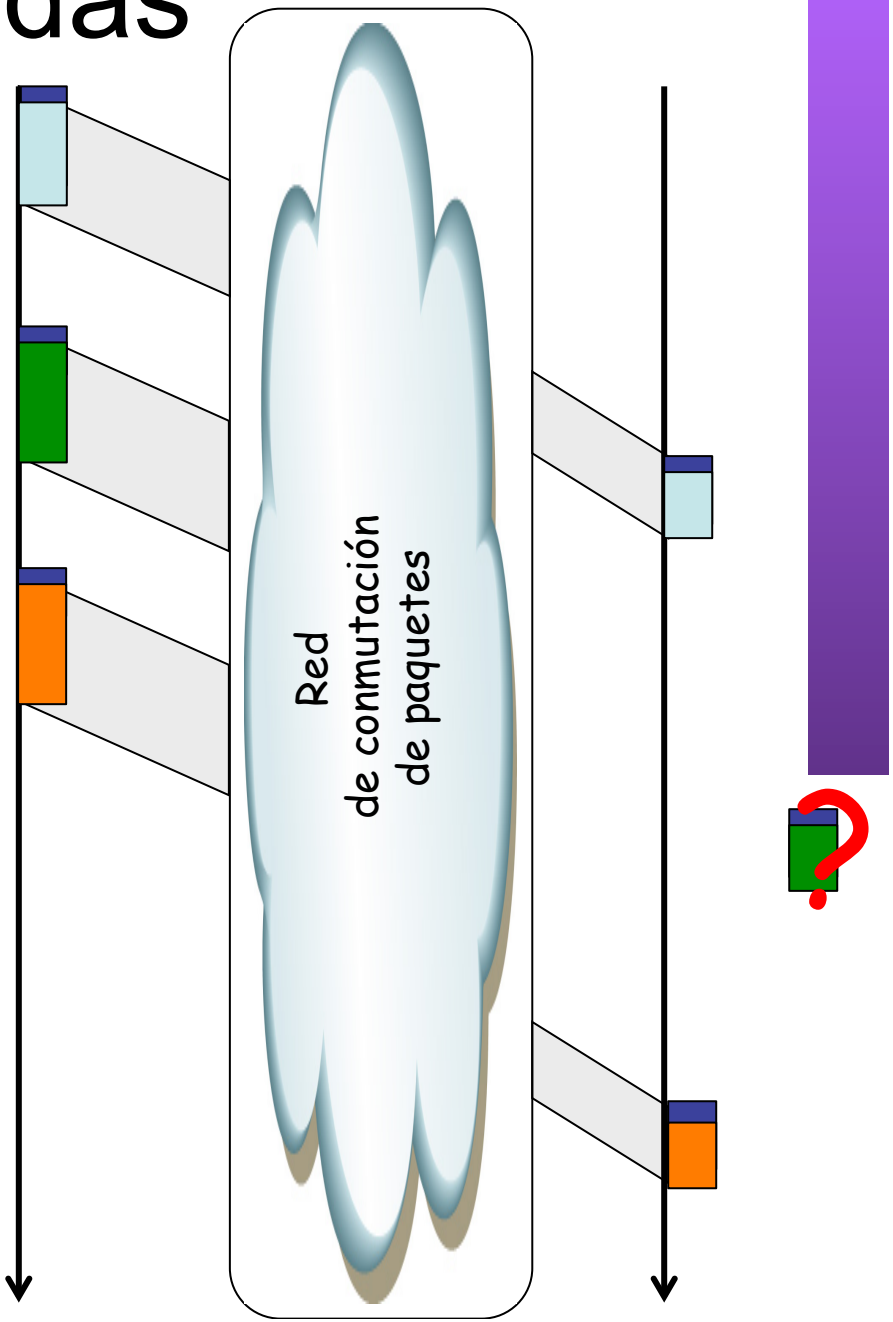
- Los paquetes podrían no llegar nunca a su destino

Posibles motivos

- Se descartó en un nodo de la red por desbordamiento de buffer
- (...)



Tamaño del buffer

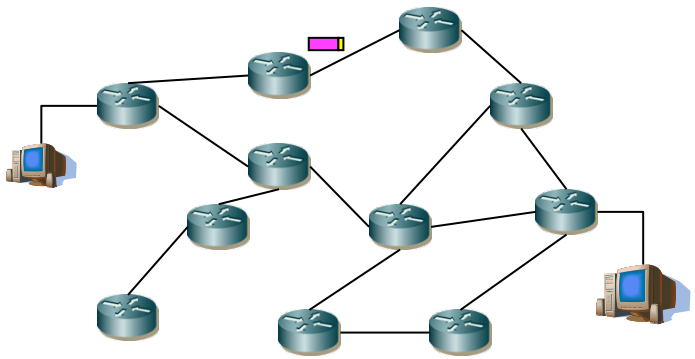


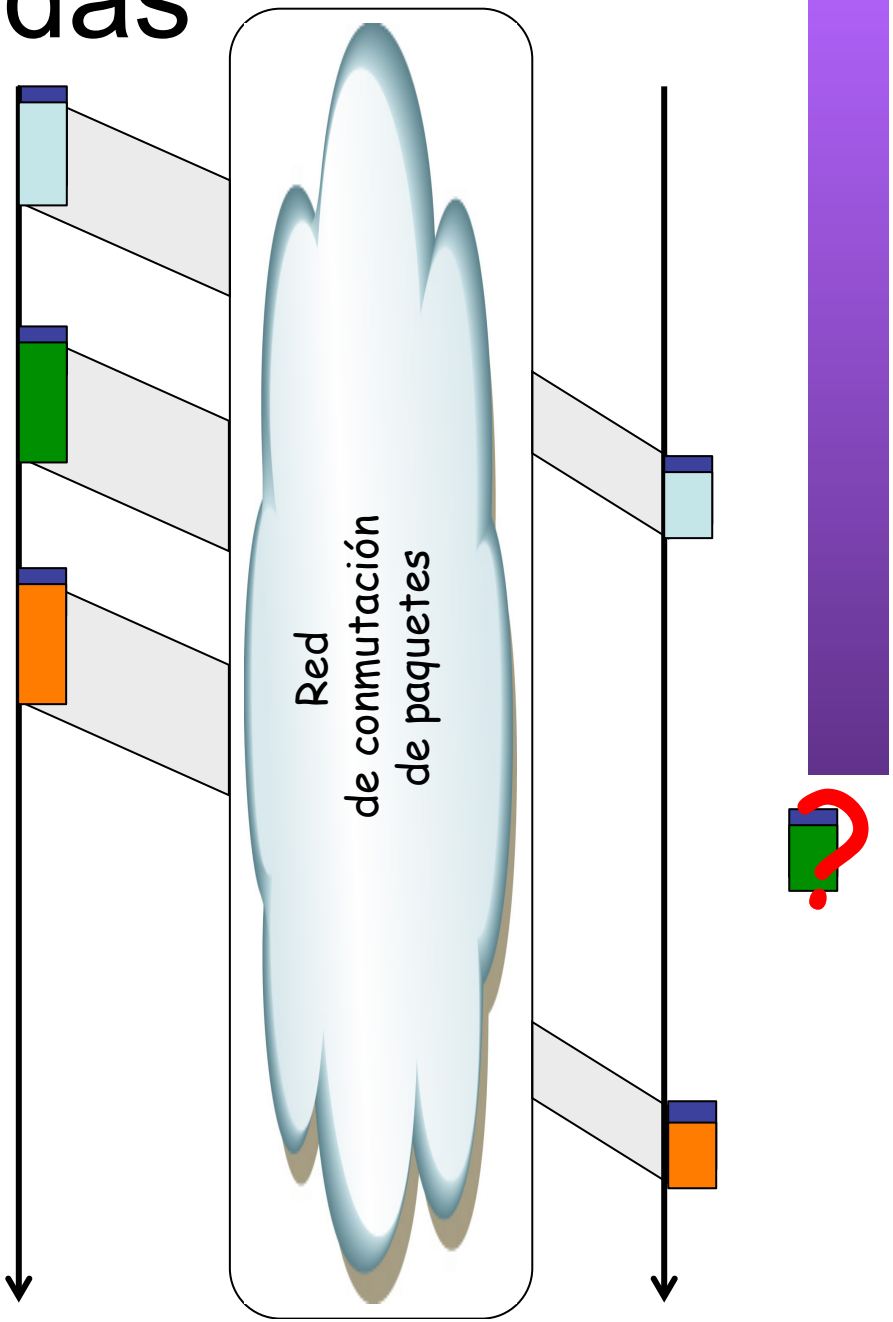
Pérdidas

- Los paquetes podrían no llegar nunca a su destino

Posibles motivos

- Se descartó por exceder el tiempo en la red
- (...)



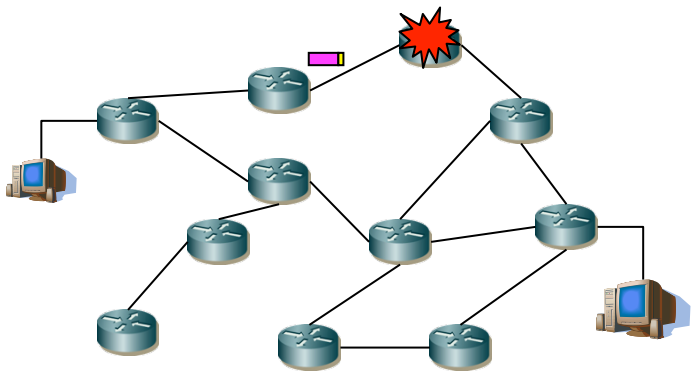


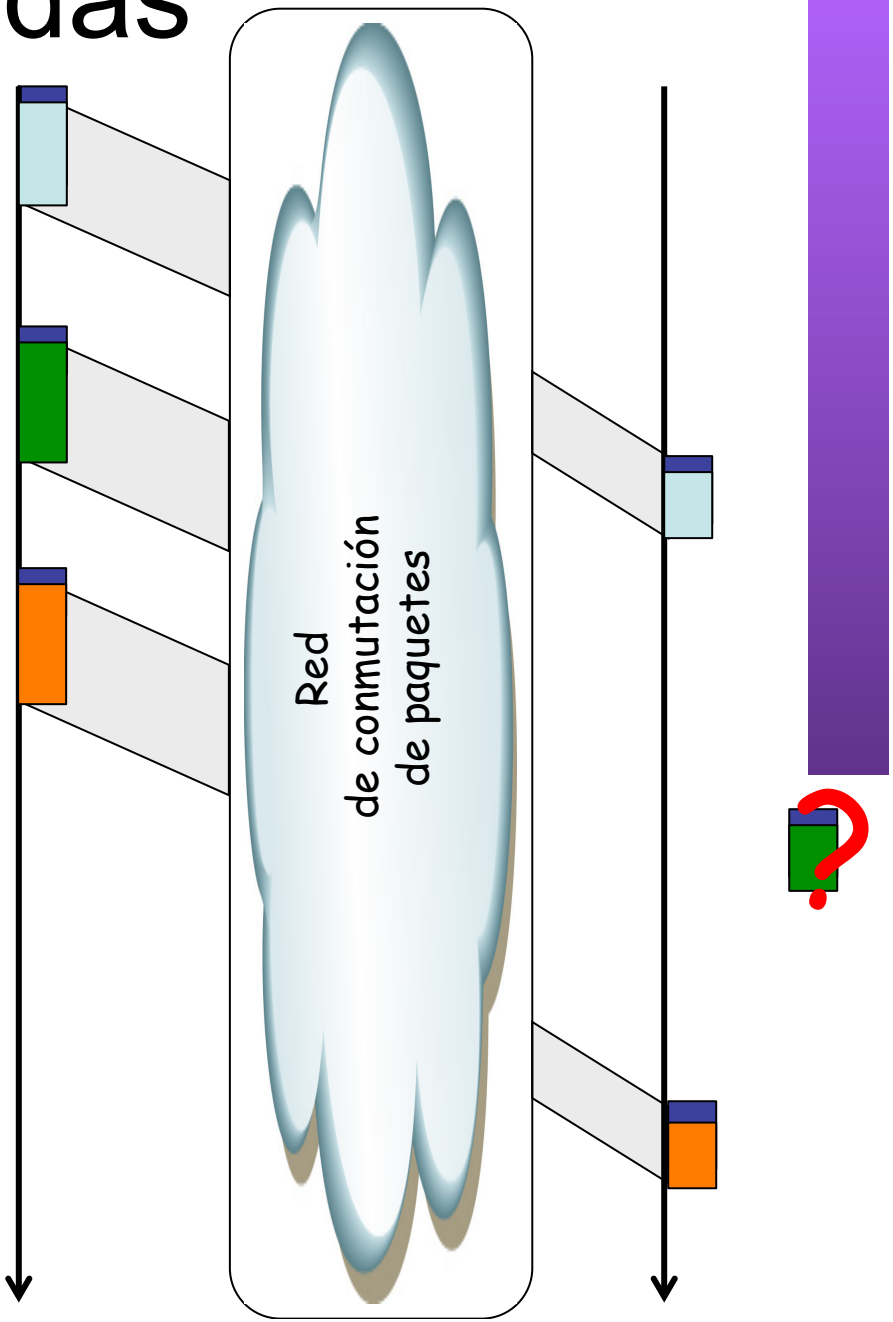
Pérdidas

- Los paquetes podrían no llegar nunca a su destino

Posibles motivos

- Fallo de un elemento de red
- Lleva un tiempo recalcular caminos
- (...)



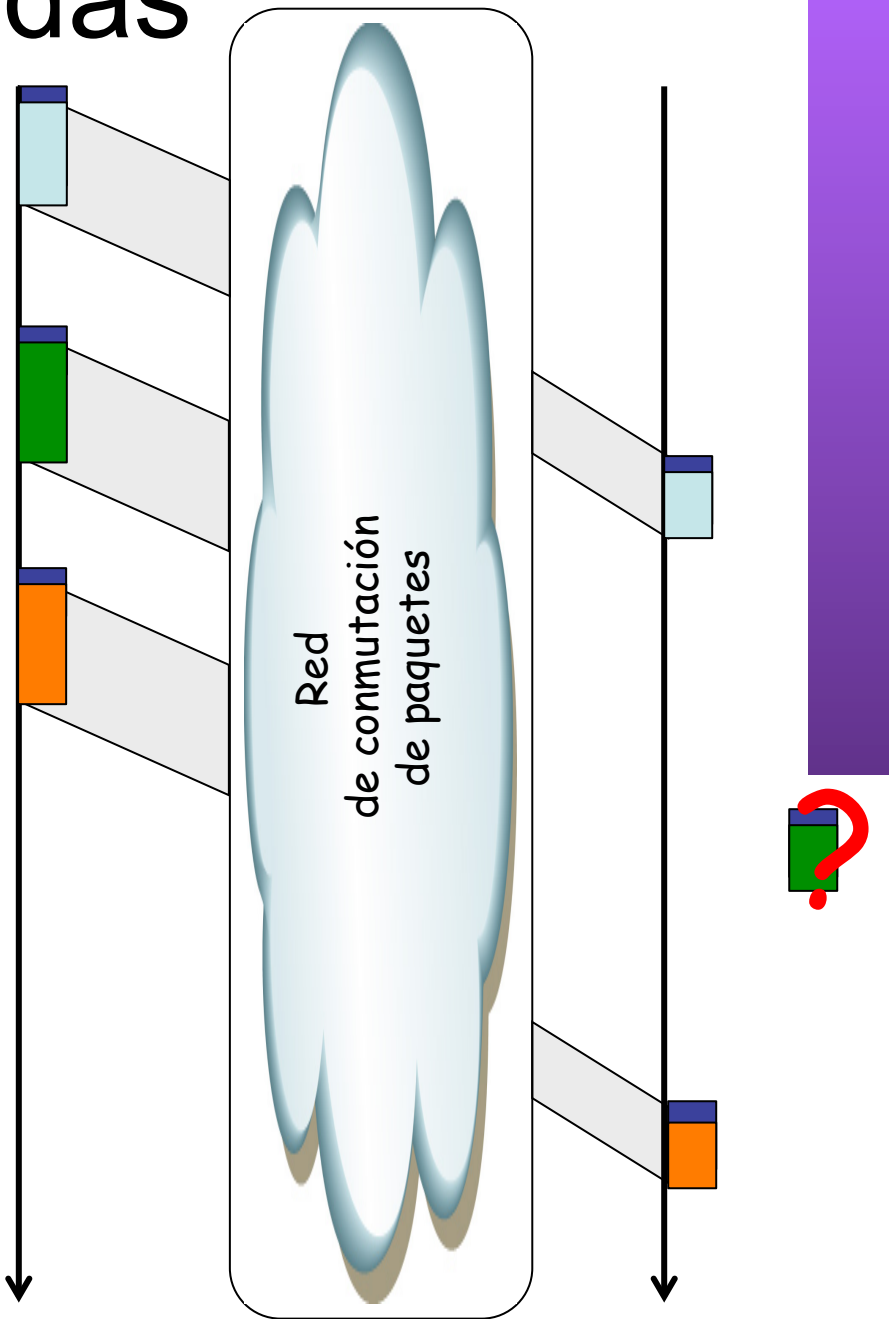


Pérdidas

- Los paquetes podrían no llegar nunca a su destino

Posibles motivos

- Descarte en nodo extremo por desbordamiento de buffer
- Puede ser culpa de la propia aplicación y del tiempo que le lleva procesar los datos recibidos



Resumen

- En redes de conmutación de paquetes:
  - Retardo, pérdidas y variación del retardo
  - El retardo múltiples componente
- En redes de conmutación de circuitos:
  - Encaminamiento y bloqueo
- En redes de conmutación de datagramas problemas adicionales de control de flujo, congestión, etc.