

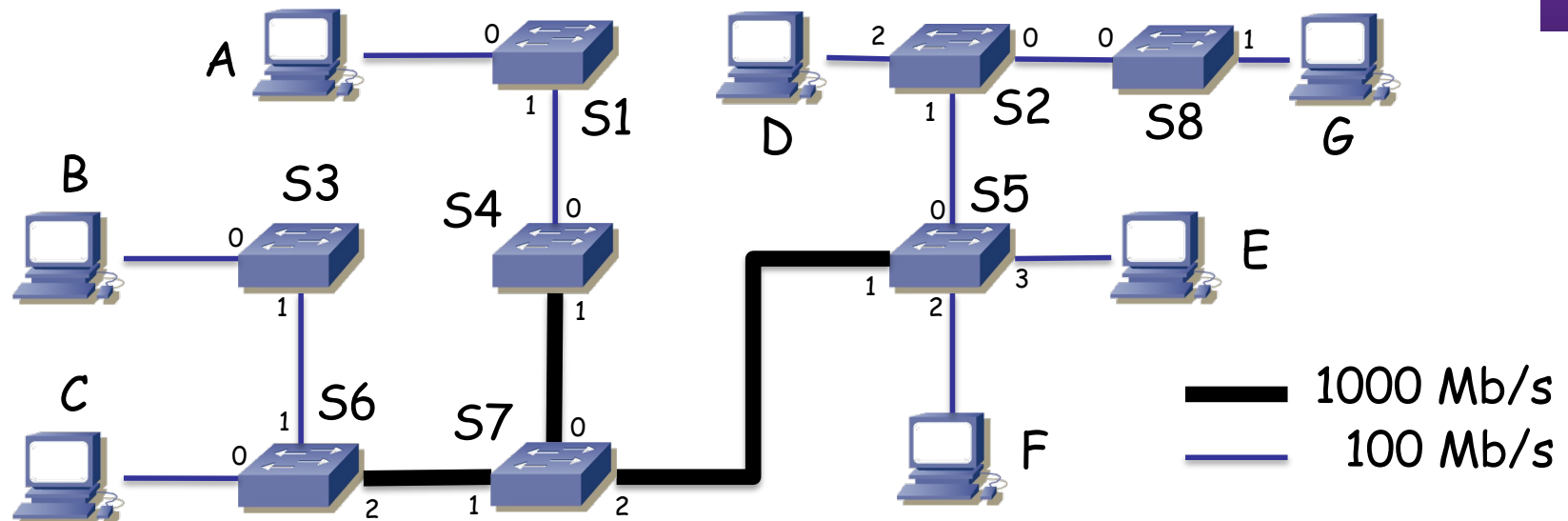
Retardo en redes

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

Arquitectura de Redes, Sistemas y Servicios

Conmutación

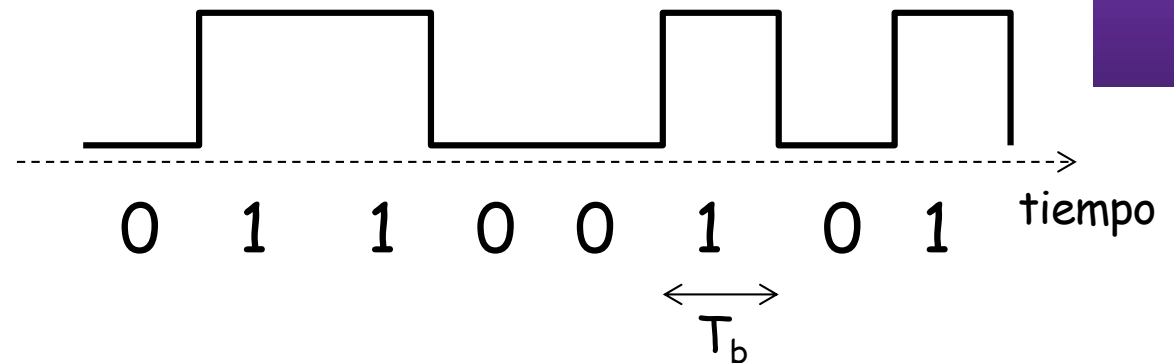
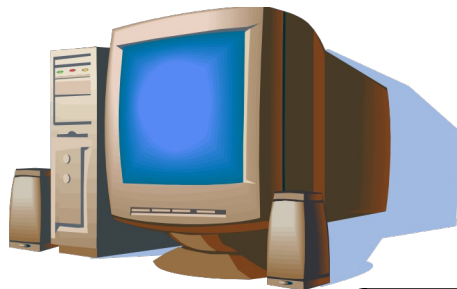
- Hemos visto conmutación de paquetes en dos tecnologías:
 - Ethernet (datagramas)
 - ATM (circuitos virtuales)
- Vamos a resolver ahora problemas generales que afectan a todas estas redes
- Por ejemplo: ¿Cuánto tiempo tarda un paquete en llegar de un host a otro?



Retardos en un enlace

Retardo de transmisión

- Tiempo que tarda el transmisor en colocar los bits en el canal
- También llamado retardo de serialización
- Bits por segundo (bps, b/s) (...)
- Ejemplo:
 - Longitud del paquete $L = 1500 \text{ Bytes} = 12000 \text{ bits}$
 - Tasa de transmisión $R = 57600 \text{ b/s}$
 - $T_b = 1 / 57600 = 0.00001736 = 17.36 \times 10^{-6} \text{ s} = 17.36 \mu\text{s}$
 - Tiempo de transmisión = $L/R = 12000 \text{ bits} / 57600 \text{ b/s} \approx 208 \text{ ms}$

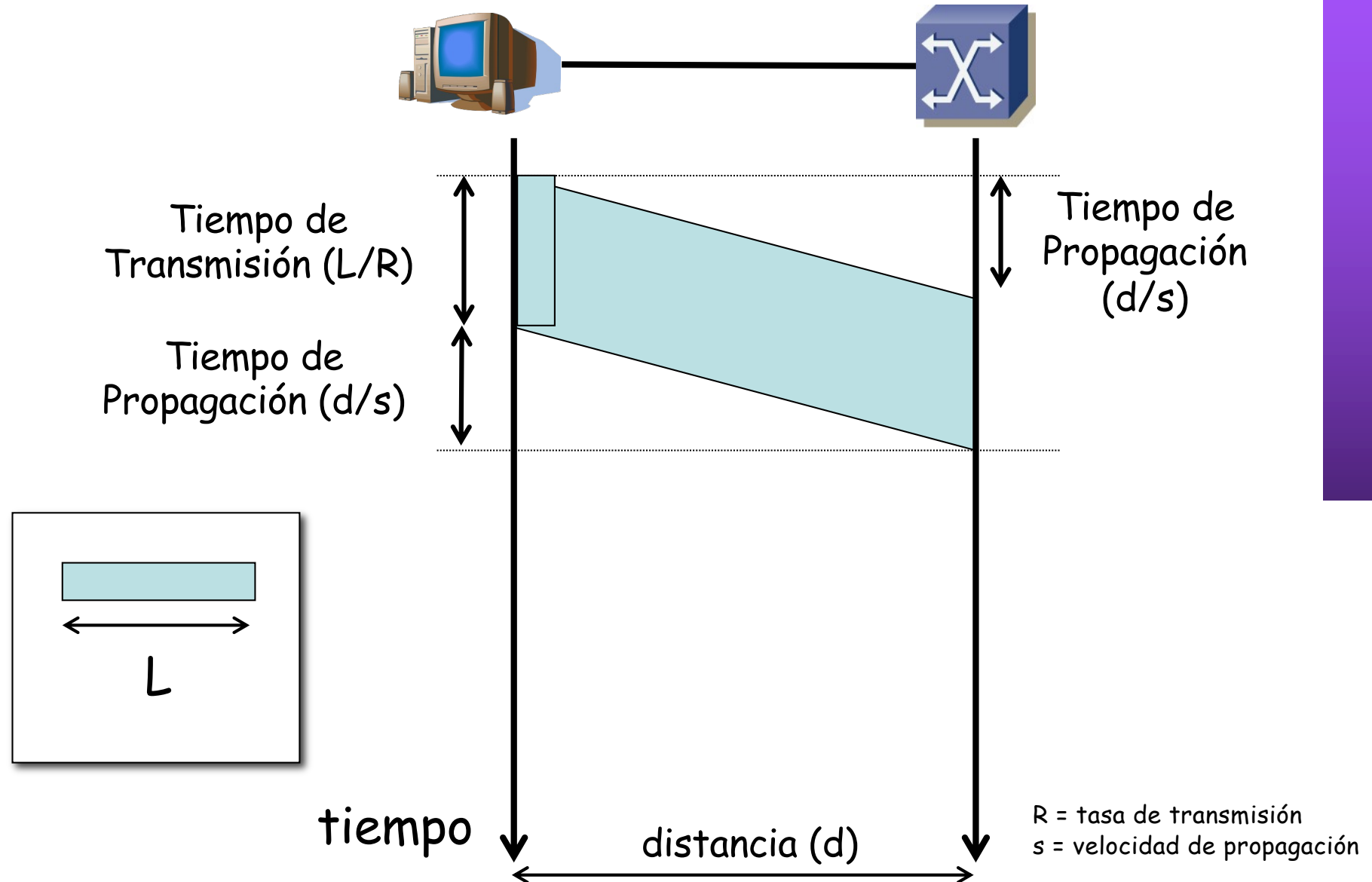


Retardo de propagación

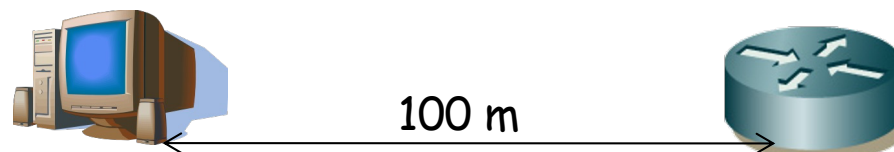
- Tiempo que tarda la señal en llegar al otro extremo del sistema de transmisión (...)
- Ejemplo:
 - Longitud del enlace físico $d = 2000 \text{ Km}$
 - Velocidad de propagación en el medio $s = 200000 \text{ Km/s}$
 - Retardo de propagación $= d/s = 2 \times 10^6 \text{ m} / (2 \times 10^8 \text{ m/seg}) = 10 \text{ ms}$
- La velocidad de transmisión y la velocidad de propagación son conceptos muy diferentes
- Velocidad de propagación en Km/s
 - En coaxial en torno a 250000 Km/s
 - En fibra óptica en torno a 200000 Km/s



Retardos de transmisión y propagación

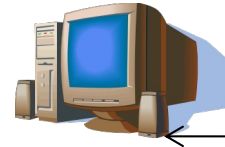


Ejemplo



Enunciado

- $L = 1500$ Bytes
- $R = 10$ Mbps
- $s = 200.000$ km/s
- $d = 100$ m
- ¿Cuándo empieza a recibirse?
- ¿Cuándo se ha terminado de recibir?



100 m

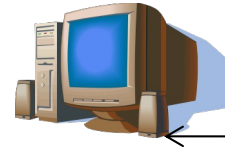


tiempo ↓

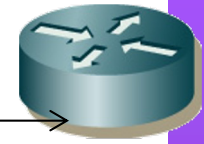


Enunciado

- $L = 1500$ Bytes
- $R = 100$ Mbps
- $s = 200.000$ km/s
- $d = 70$ km
- ¿Cuándo empieza a recibirse?
- ¿Cuándo se ha terminado de recibir?
- ¿Dónde está 0.17 ms tras empezar la transmisión?



70 km



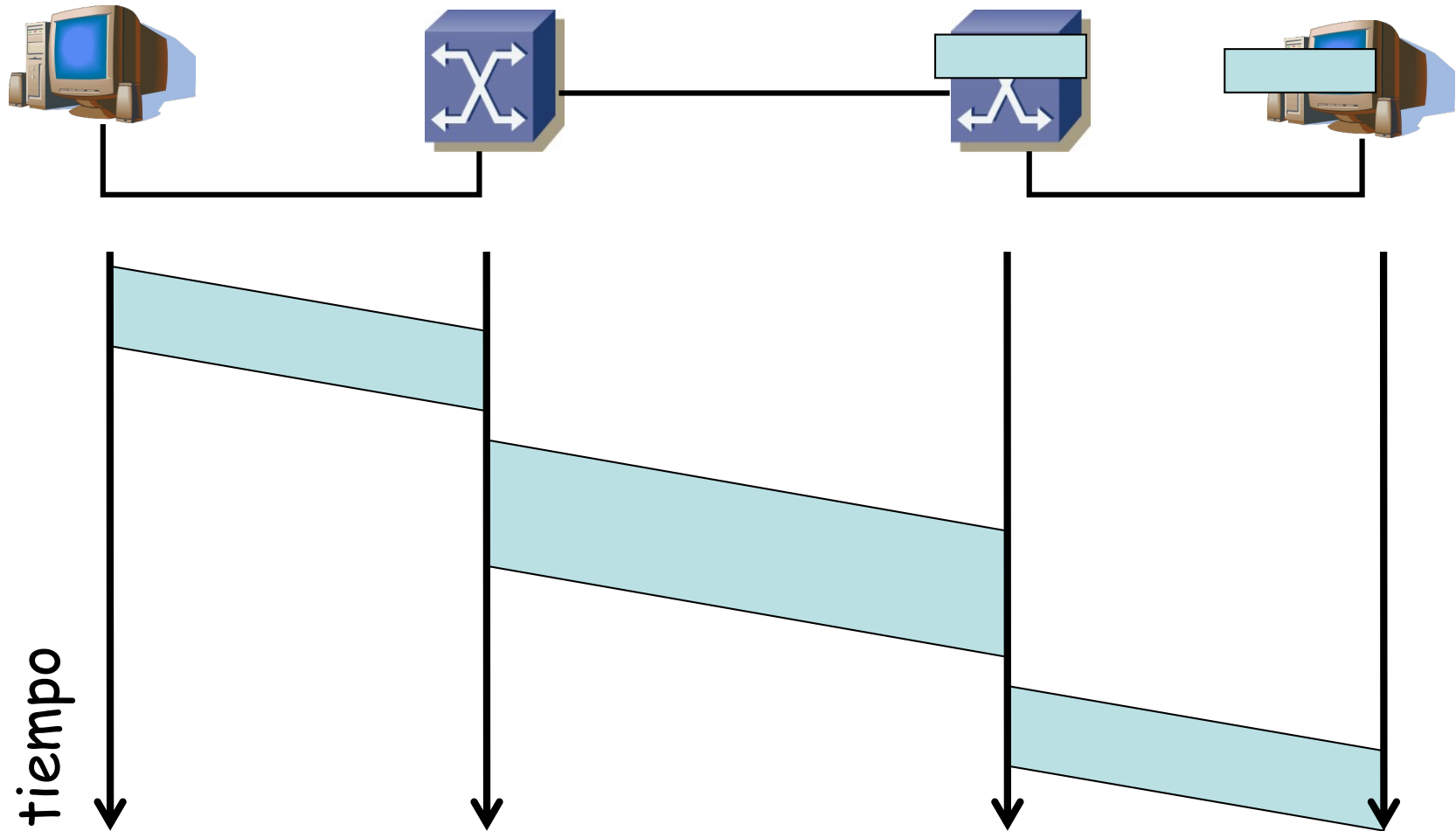
tiempo ↓



Retardo con varios “saltos”

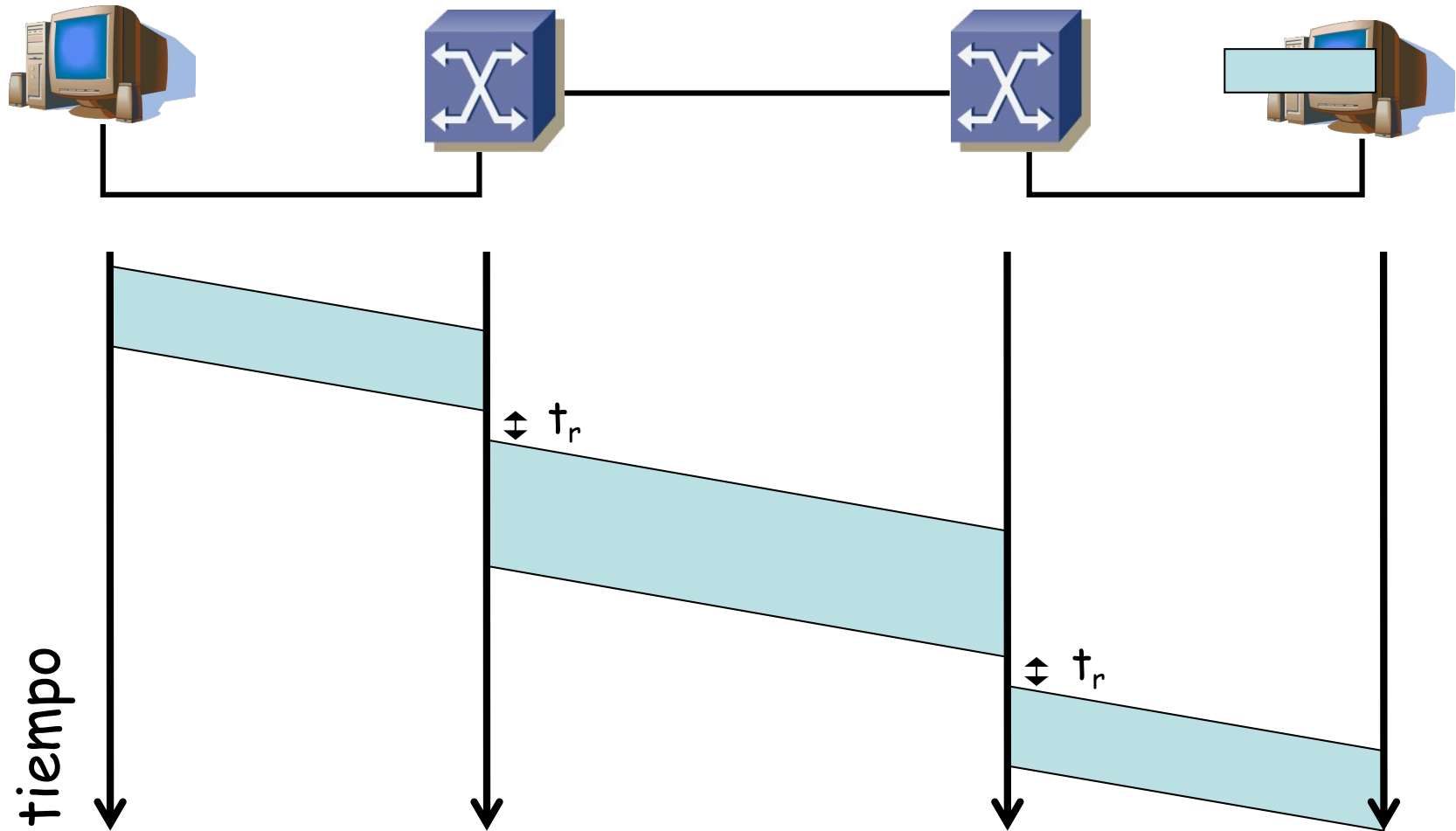
Store-and-forward

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



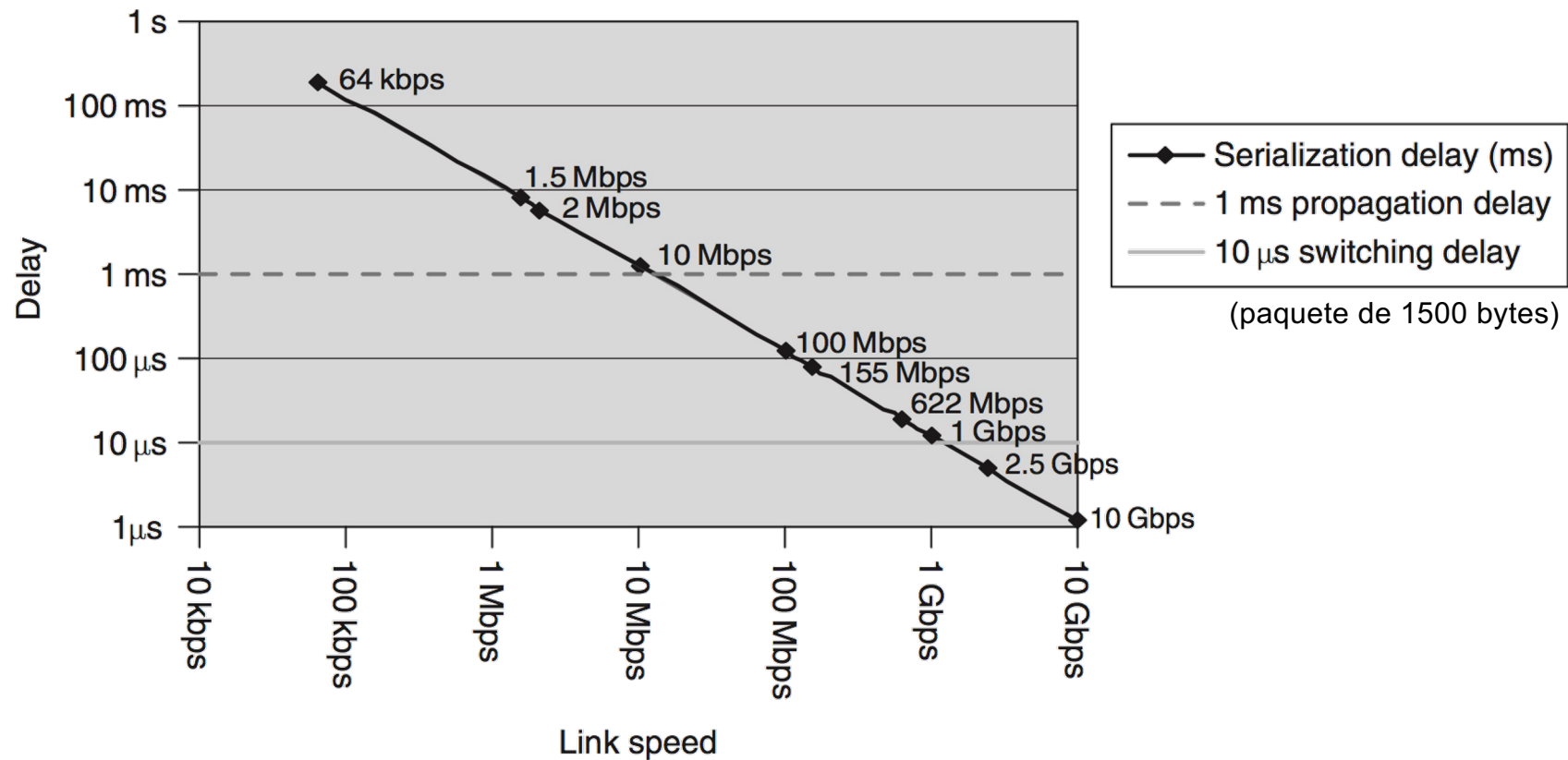
Tiempo de procesamiento

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



Comparativa

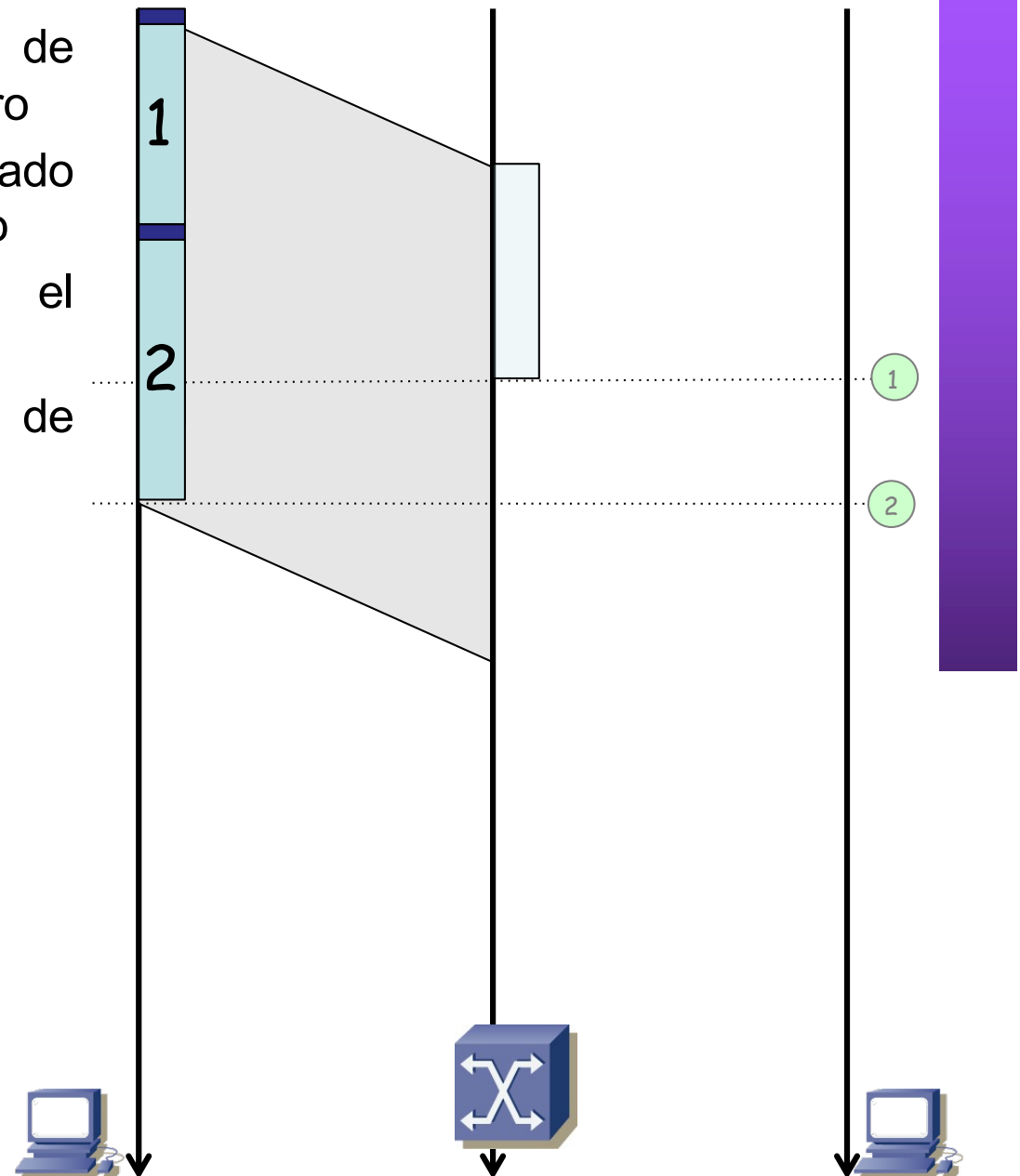
- El tiempo de transmisión (retardo de serialización) domina o no frente al de propagación según la velocidad de transmisión
- Y según la distancia del enlace ($1\text{ms} @ 300000\text{Km/s} = 300\text{Km}$)



Store & forward

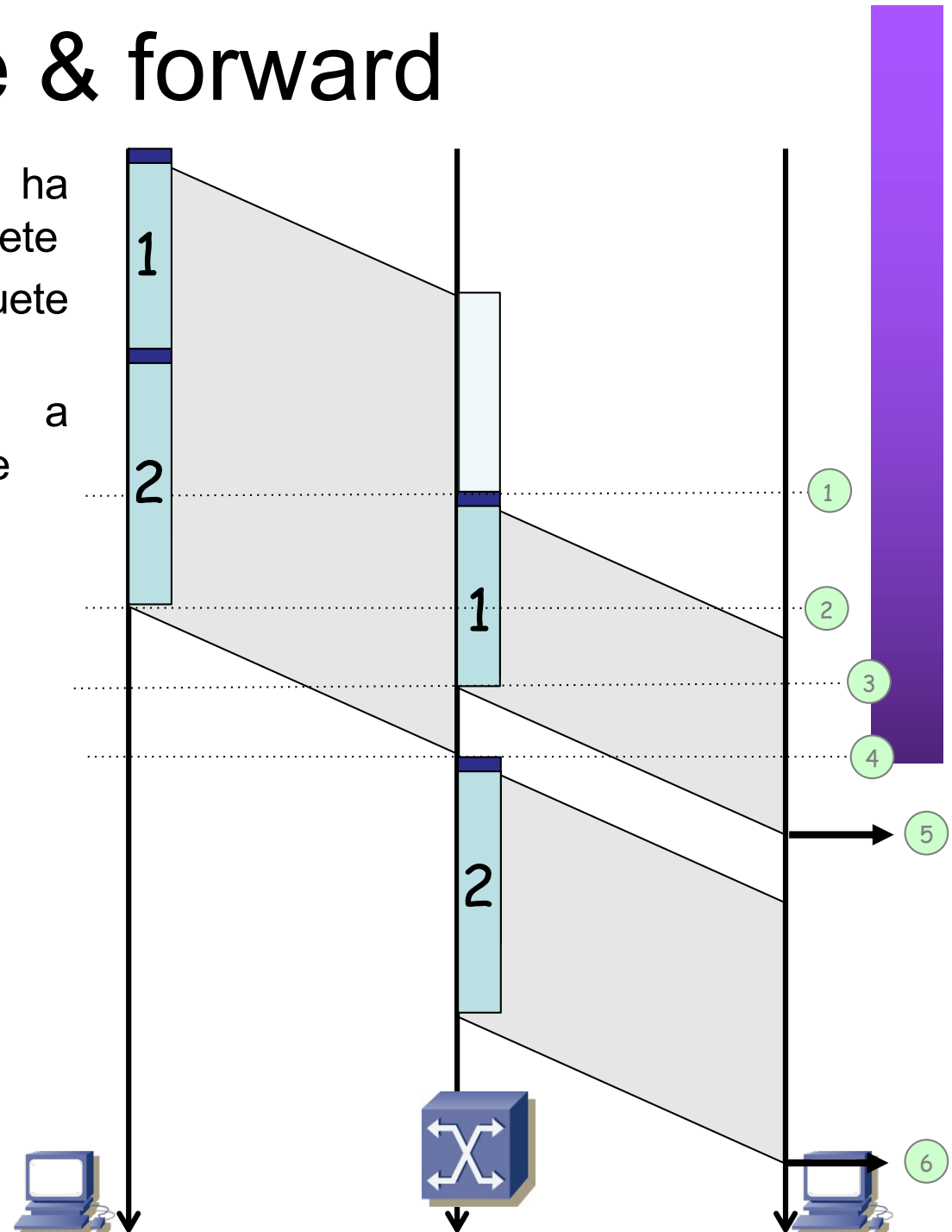
Store & forward

- Una vez que se termina de enviar un paquete puede otro
- En este caso ha terminado antes de recibirse el primero
- El tiempo (2) no lo ve el conmutador
- En (1) más el tiempo de procesado puede reenviar



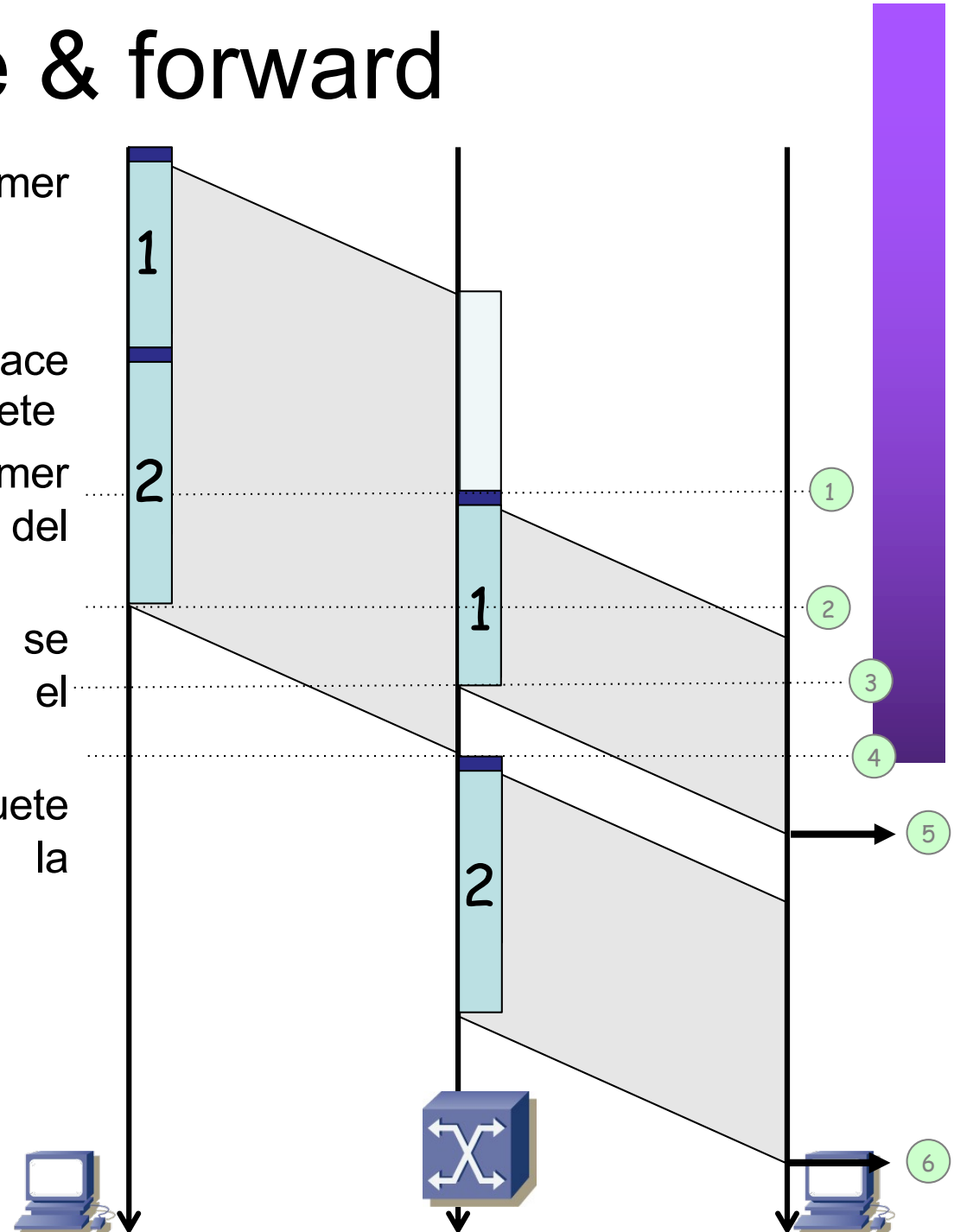
Store & forward

- En (3) el conmutador ha terminado de enviar el paquete
- No tiene el segundo paquete hasta (4)
- En (4) puede empezar a reenviar el segundo paquete



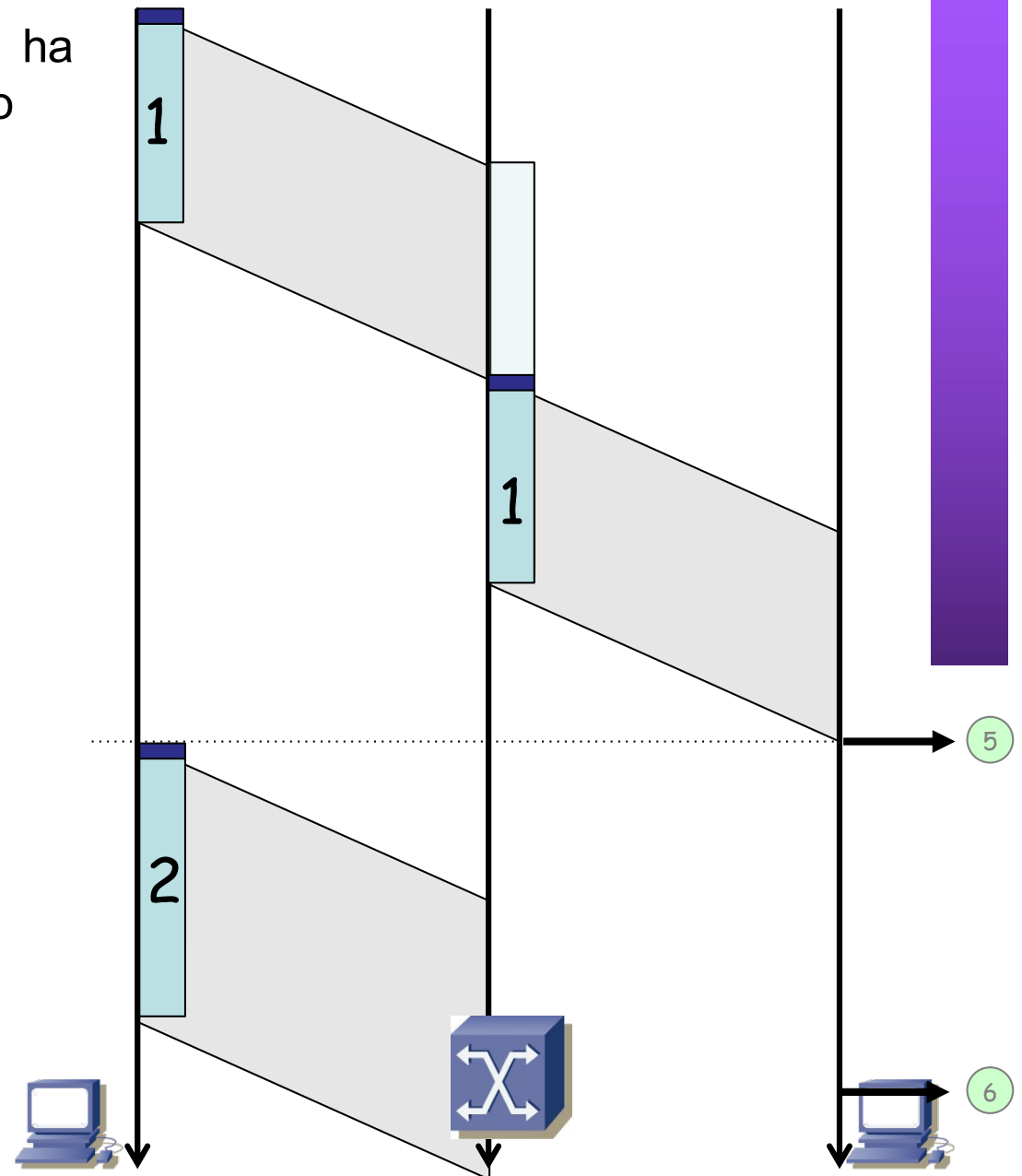
Store & forward

- En (5) está disponible el primer paquete
- En (6) el segundo
- Tenemos en el segundo enlace transmisión del primer paquete
- A la vez que en el primer enlace transmisión del segundo paquete
- Hay un tiempo en que no se está enviando nada en el segundo enlace
- Hasta que no tenga el paquete entero no puede tomar la decisión de reenvío



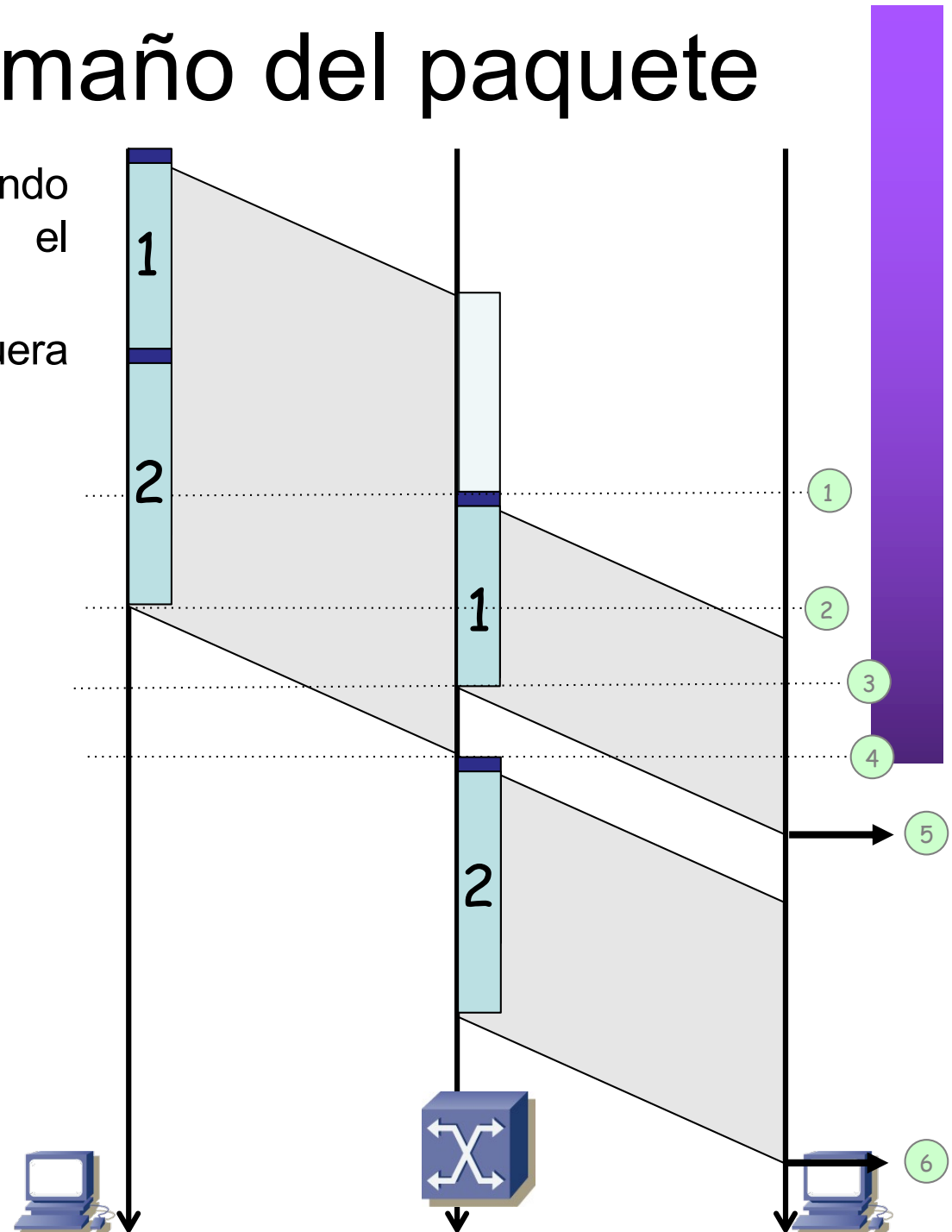
No ha sucedido esto

- El origen no sabe cuándo ha llegado el paquete al destino



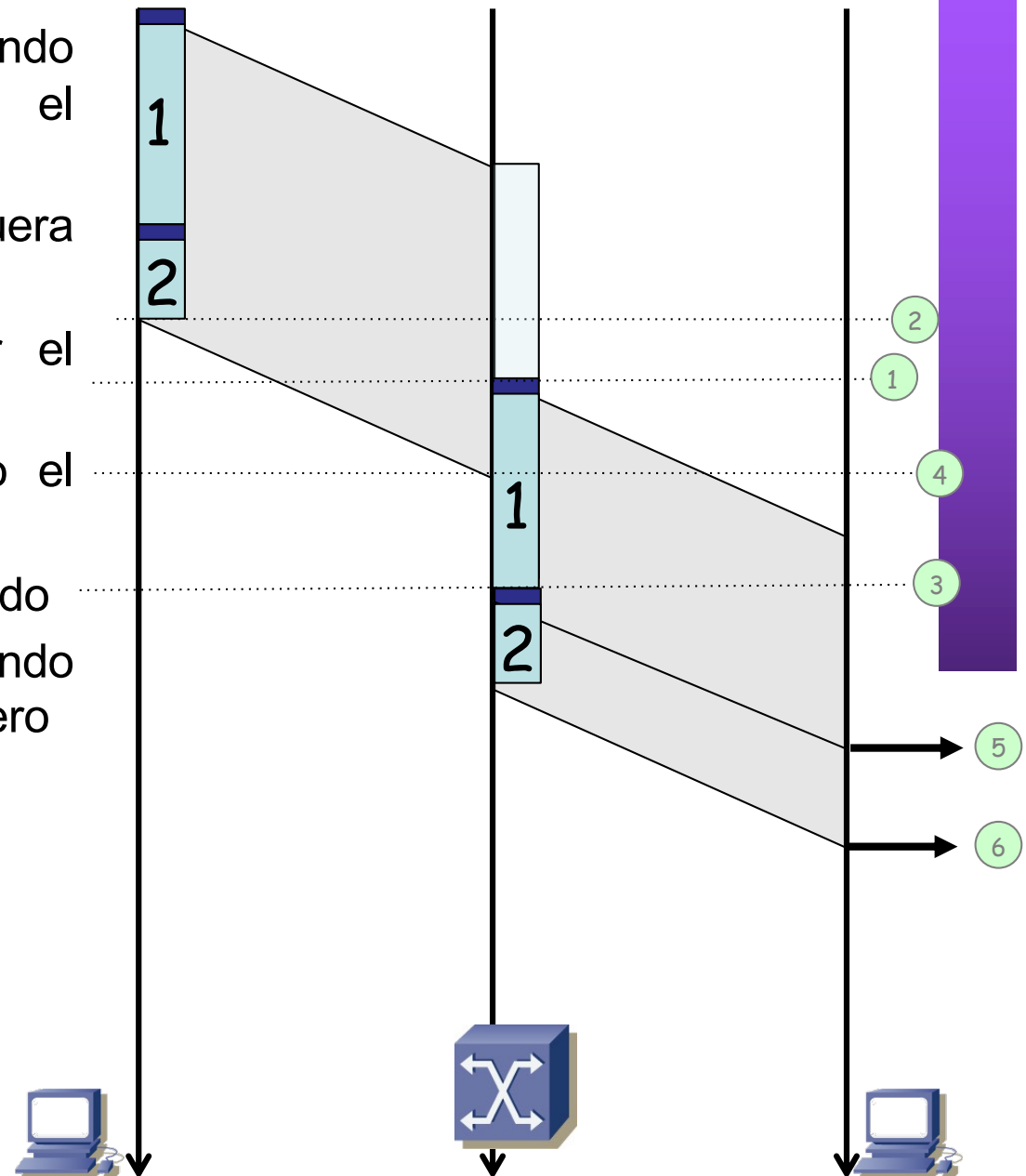
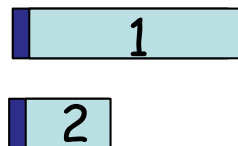
Efecto del tamaño del paquete

- En este ejemplo el segundo paquete era mayor que el primero
- Veamos qué sucedería si fuera más pequeño



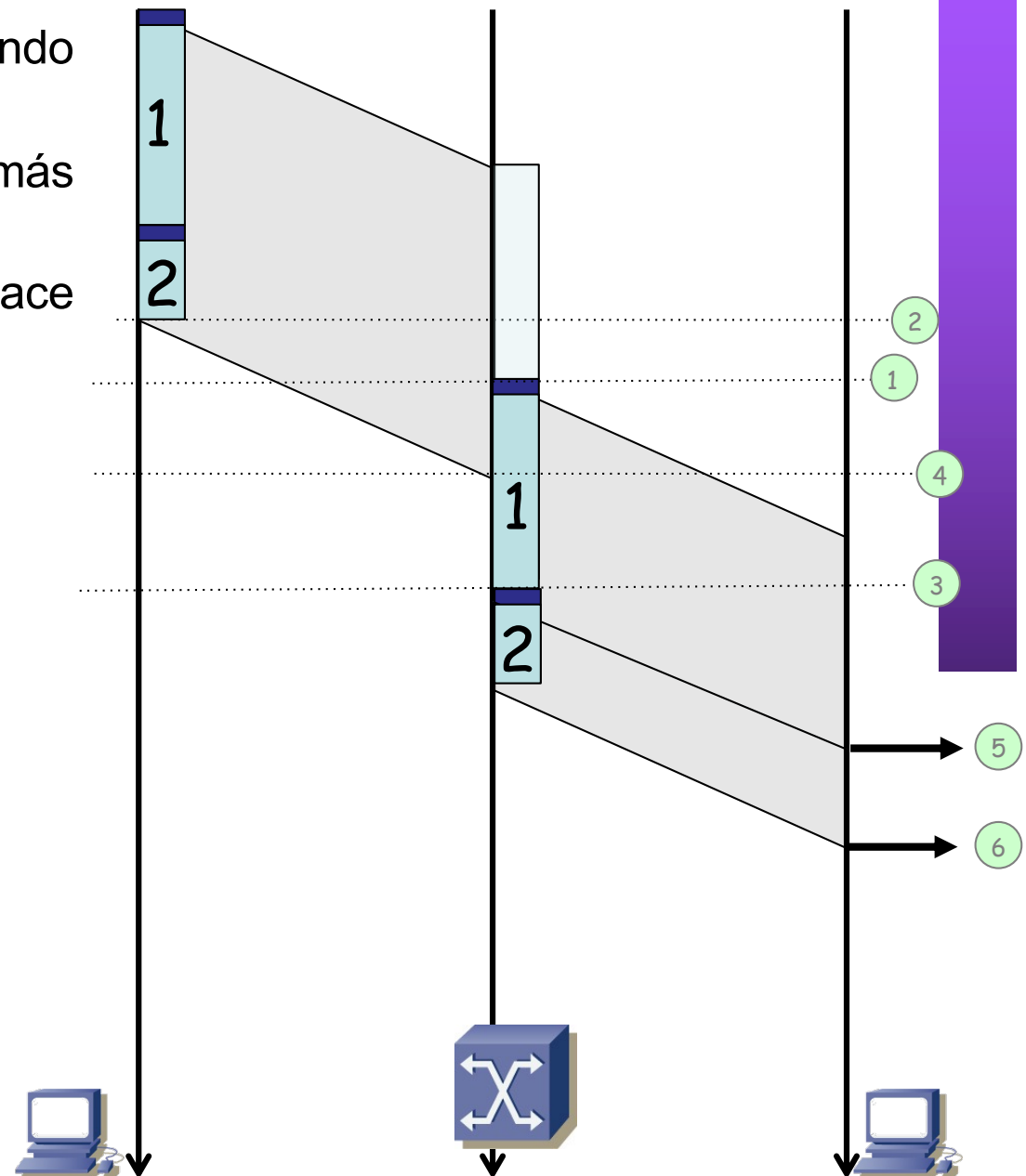
Efecto del tamaño del paquete

- En este ejemplo el segundo paquete era mayor que el primero
- Veamos qué sucedería si fuera más pequeño
- En (4) termina de llegar el segundo paquete al switch
- Pero aún está reenviando el primero
- Se queda en buffer esperando
- Podrá reenviarse cuando termine de enviarse el primero



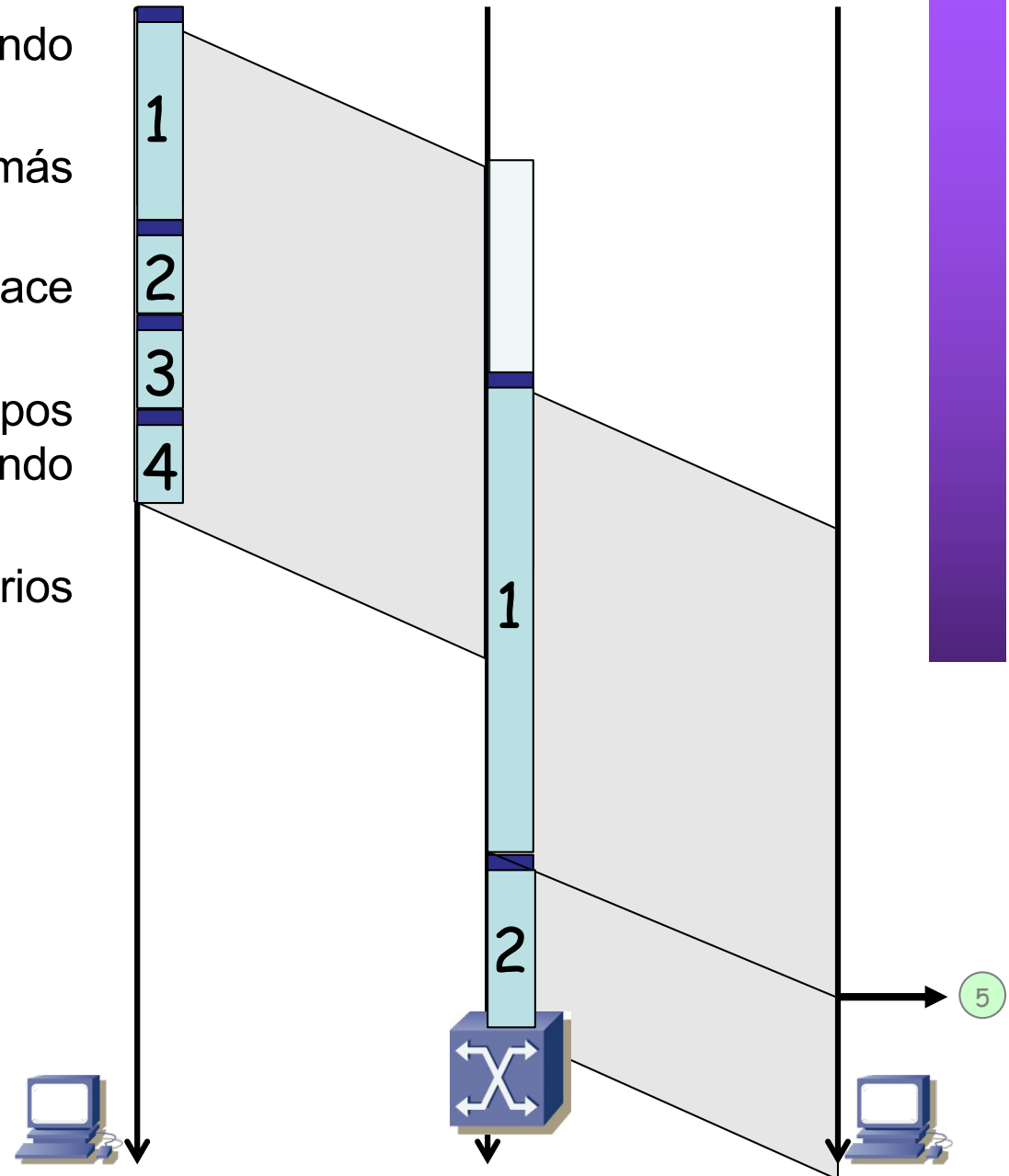
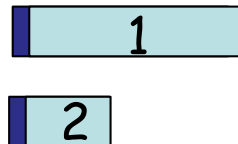
Efecto del tamaño del paquete

- Desde (4) hasta (3) el segundo paquete está en cola
- Podrían acumularse más paquetes
- Peor aún si el segundo enlace es de menor velocidad...



Efecto de la velocidad de tx

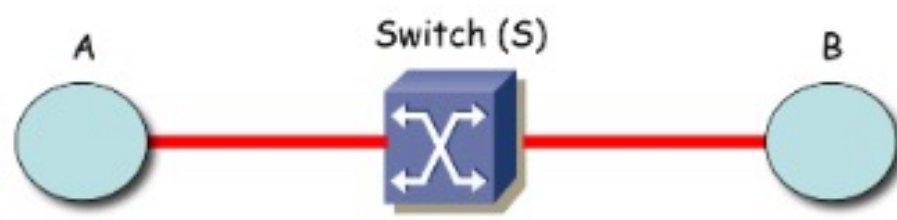
- Desde (4) hasta (3) el segundo paquete está en cola
- Podrían acumularse más paquetes
- Peor aún si el segundo enlace es de menor velocidad...
- Tendríamos mayores tiempos de transmisión en segundo enlace
- Podrían acumularse varios paquetes



Para saber más

Ejercicio 10 de la Hoja de Problemas 2

- La tecnología empleada se basa en conmutación de paquetes.
- Se va a enviar un bloque de datos de 1000 bytes. La PDU que se construye añade 10 bytes de cabecera.
- Una alternativa es enviar los 1000 bytes en un solo paquete que sería entonces de 1010 bytes.
- Otra posibilidad es partir ese bloque de datos en varios paquetes y enviarlos uno tras otro.
- Si en un caso cualquiera se divide en N paquetes aproximadamente iguales calcule el número de paquetes que se deberían emplear para minimizar el tiempo que transcurre entre que se empieza a enviar el primero en A y se termina de recibir el último en B.



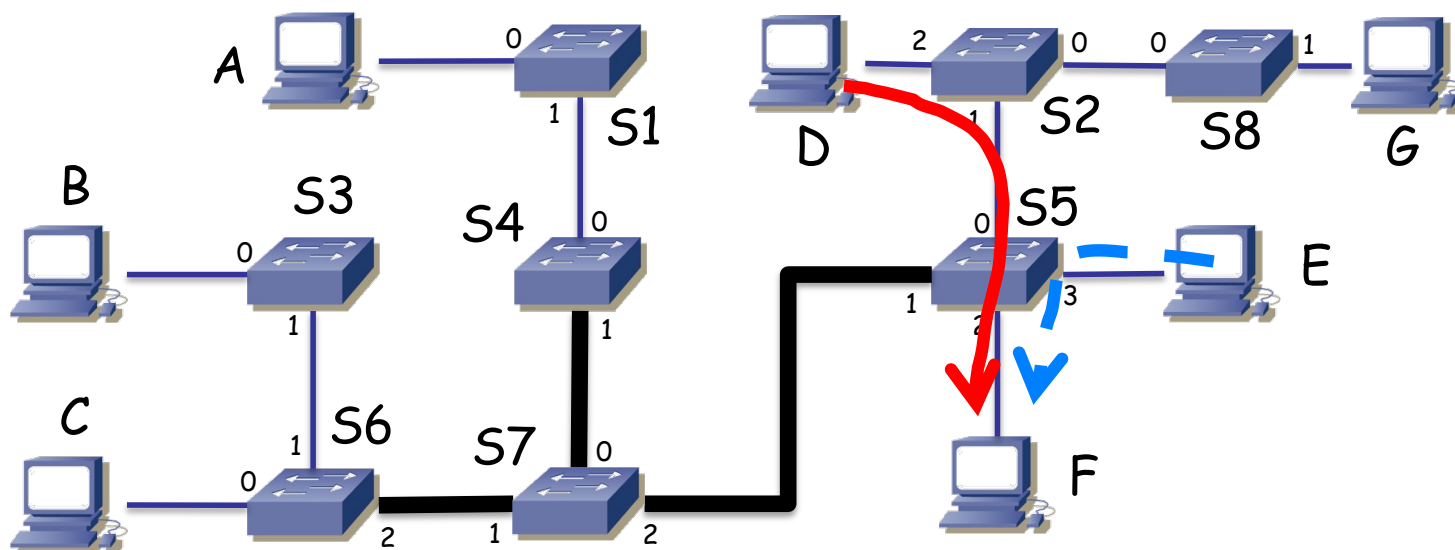
Retardo en cola

Retardo en cola: Causas

- Diferencias de tamaños y velocidades pueden causar que paquetes tengan que esperar en un conmutador
- También puede suceder por producirse llegadas por varias interfaces de forma casi simultánea

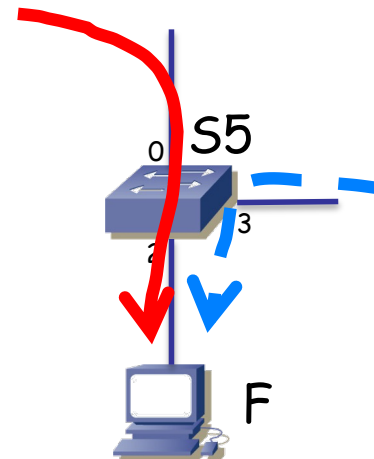
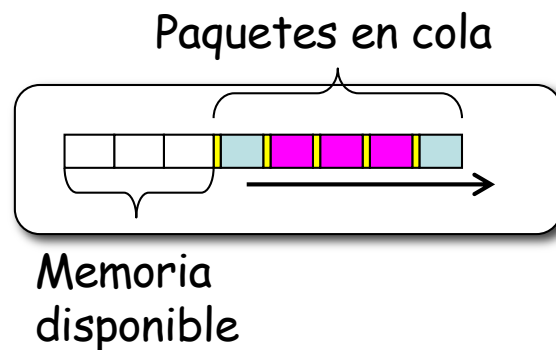
Múltiples entradas

- Será muy frecuente que lleguen paquetes al conmutador por varios enlaces, dirigidos al mismo puerto de salida



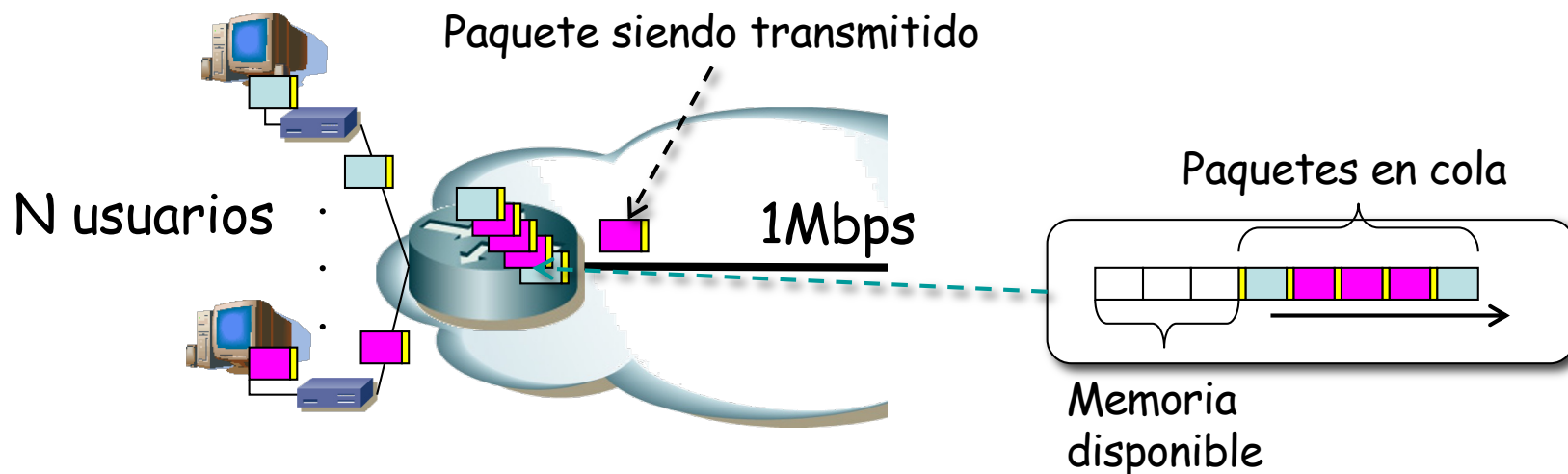
Múltiples entradas

- Será muy frecuente que lleguen paquetes al conmutador por varios enlaces, dirigidos al mismo puerto de salida
- Cada interfaz funciona de forma independiente, en paralelo
- Se puede estar recibiendo por varias interfaces a la vez
- Un paquete se podrá reenviar, el resto quedan en memoria
- Se crea una "cola de espera"



Retardo en cola

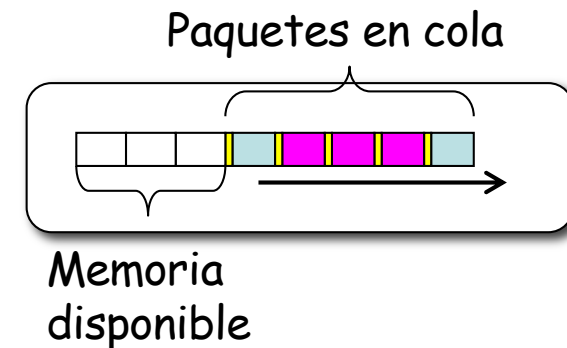
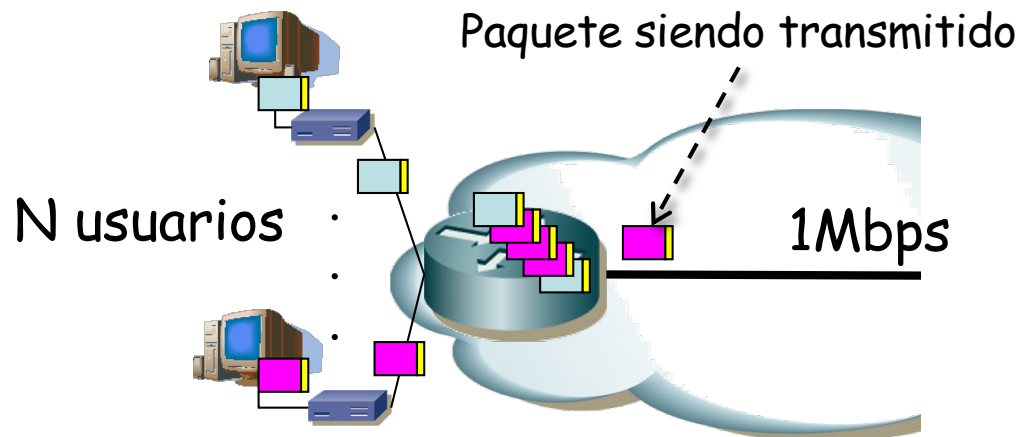
- El conmutador los almacena en memoria hasta poder enviarlos
- Esperan en una *cola* (normalmente asociada a la interfaz de salida)
- Si no queda espacio en memoria para almacenar un paquete, normalmente éste se descarta (*drop-tail policy*)
- Hablamos de situaciones de “congestión” (pérdidas y mayor retardo)
- Es independiente del rendimiento del nivel físico (calidad de la señal, probabilidad de error de bit, etc)



Planificación (*scheduling*)

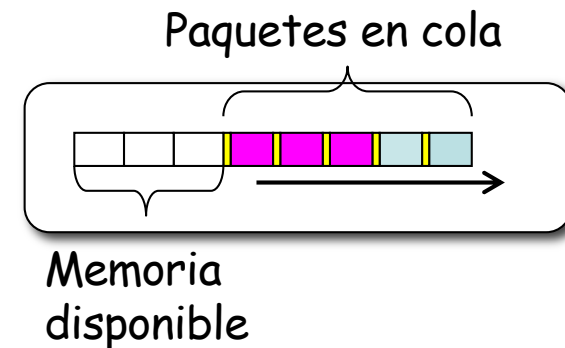
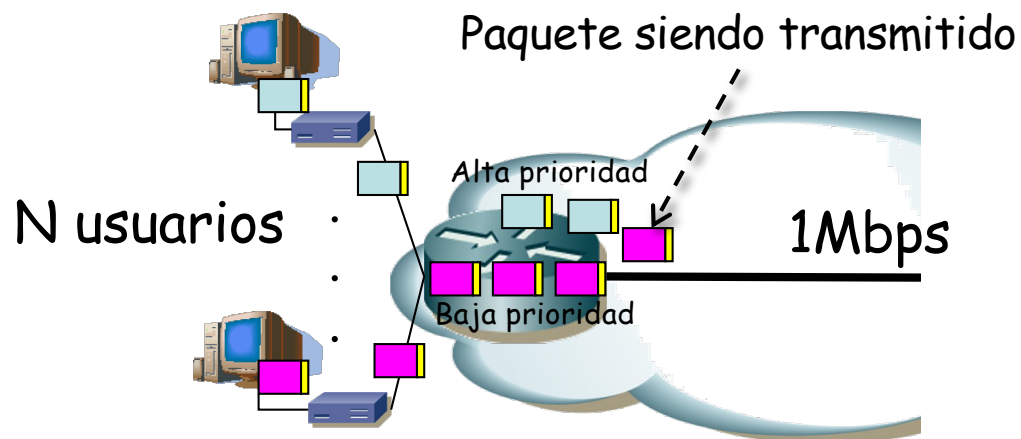
Planificación

- ¿ 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



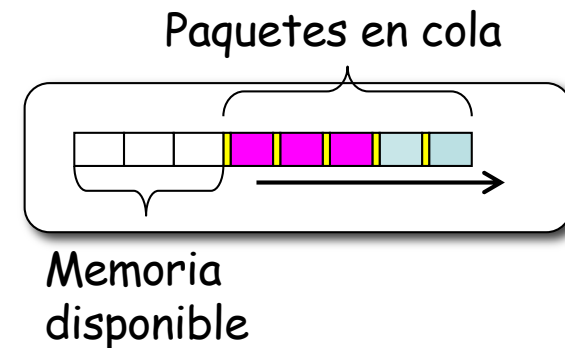
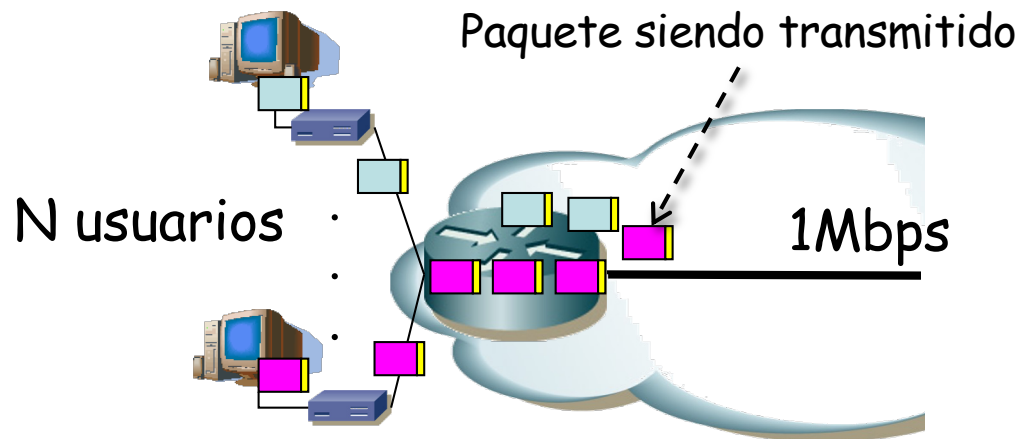
Planificación

- ¿ 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



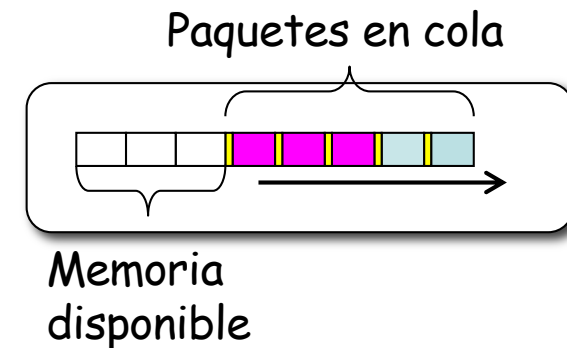
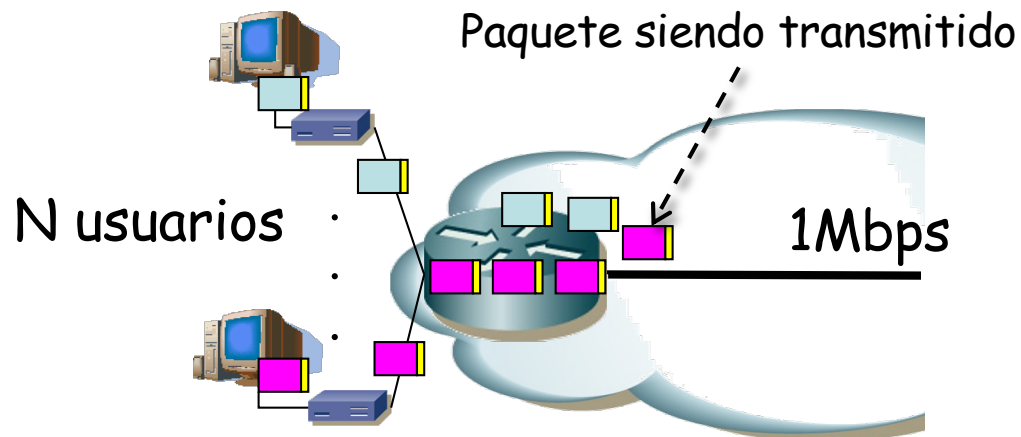
Planificación

- ¿ 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
 - ...



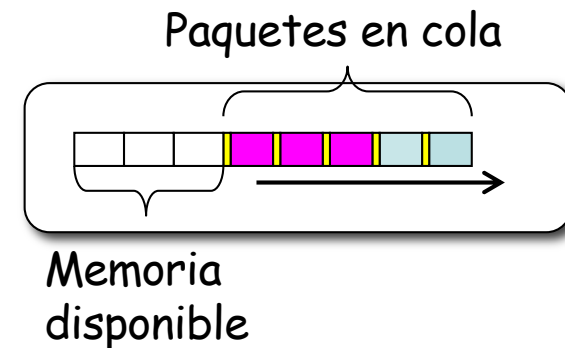
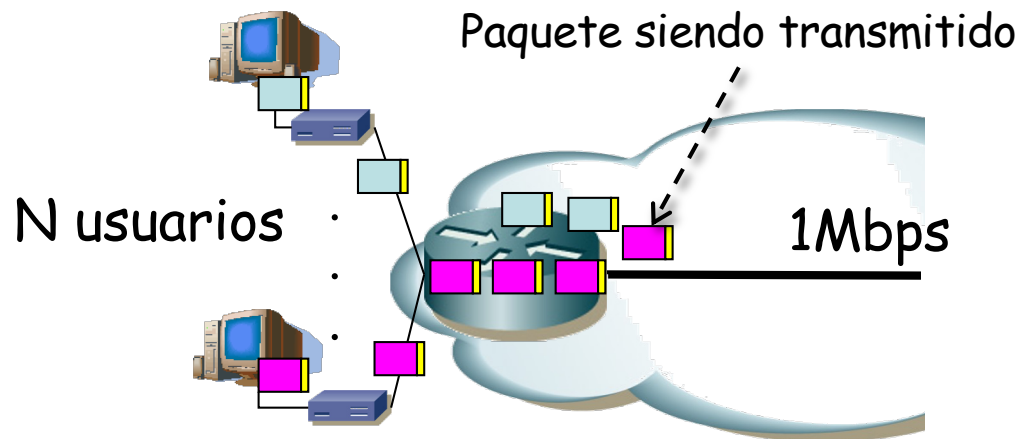
Planificación

- ¿ 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



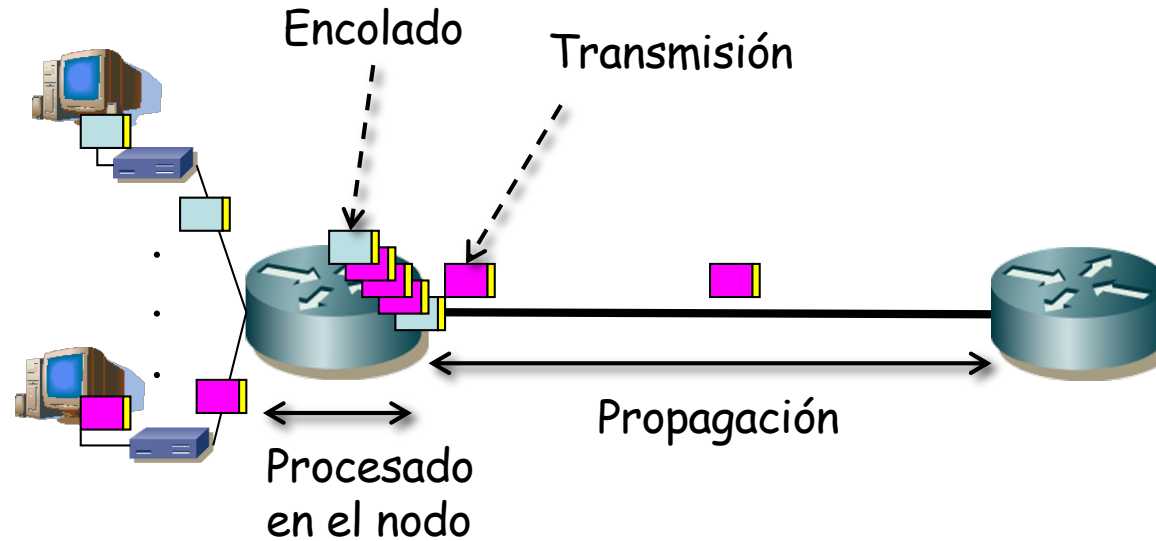
Planificación

- ¿ 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*)



Resumen de retardos

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