

# QoS: Parámetros de red

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

Grado en Ingeniería en Tecnologías de  
Telecomunicación, 3º

# ¿ Qué necesitan las apps ?

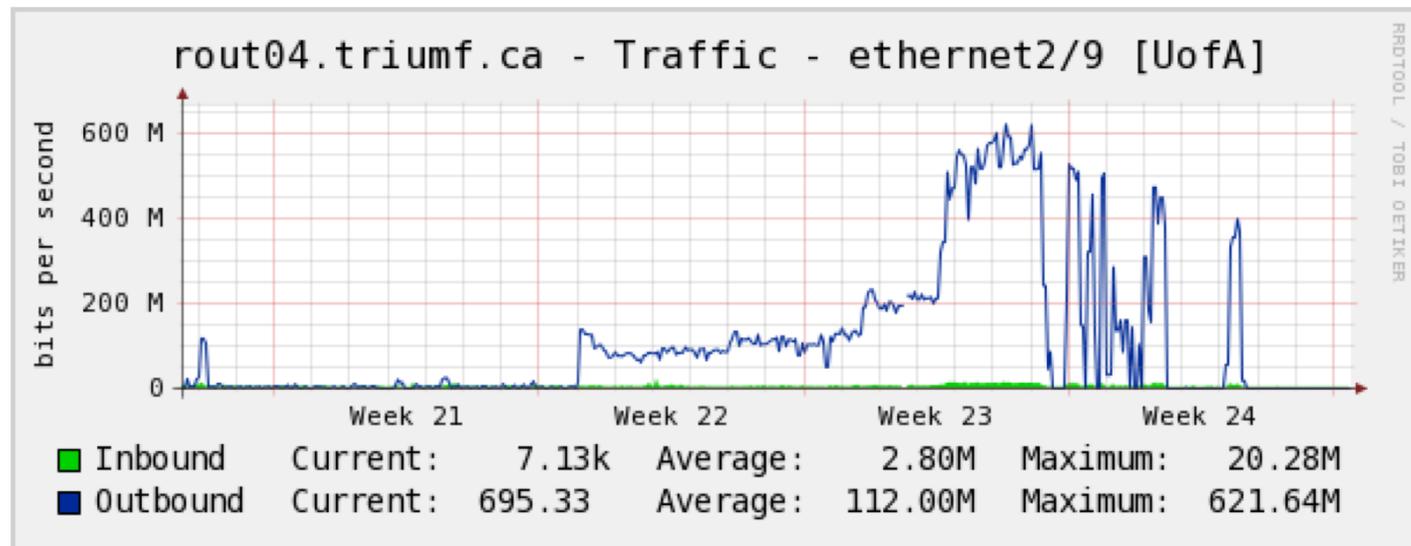
- Que sea predecible el comportamiento de la red
- Garantizar (depende de la aplicación):
  - Bandwidth (Throughput)
  - Delay
  - Variación en el retardo (jitter)
  - Pérdidas



# Throughput

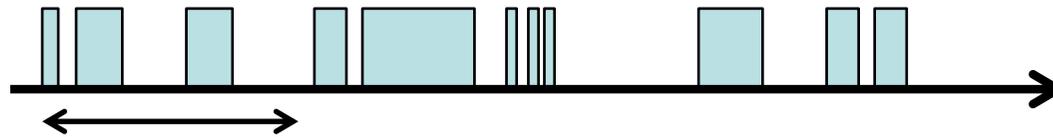
# Bandwidth / Throughput

- **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

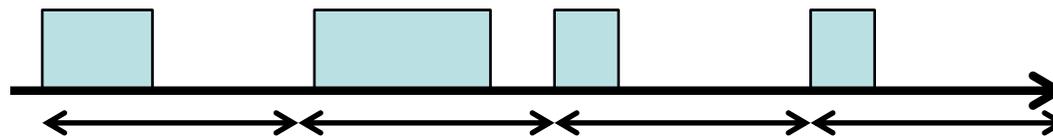


# Throughput medio

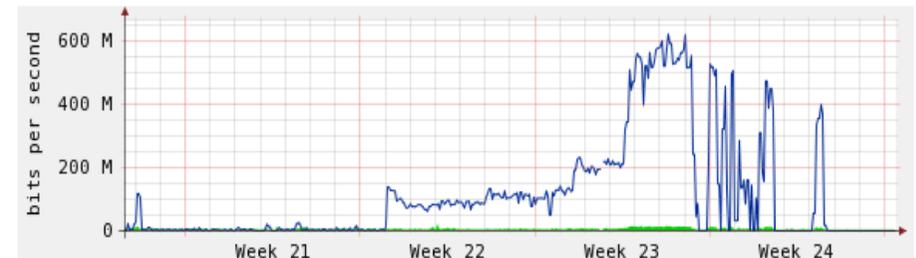
- Supongamos unas llegadas más irregulares (y más habituales)
- Paquetes de diferentes tamaños
- Paquetes con separaciones variables



- ¿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



- Según anchura del intervalo
- Promedios más “groseros”



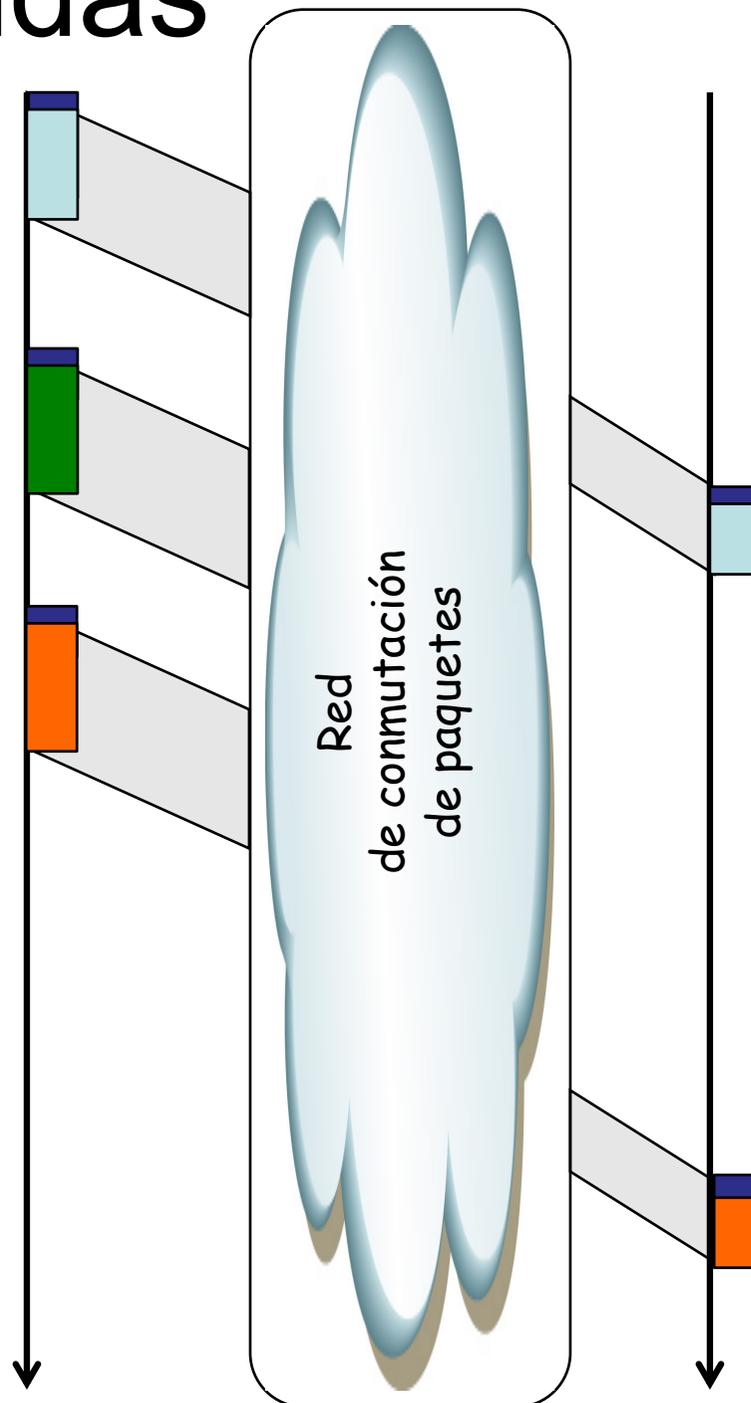
# Pérdidas

# Pérdidas

- Los paquetes podrían no llegar nunca a su destino

## Posibles motivos

- (...)



# Pérdidas

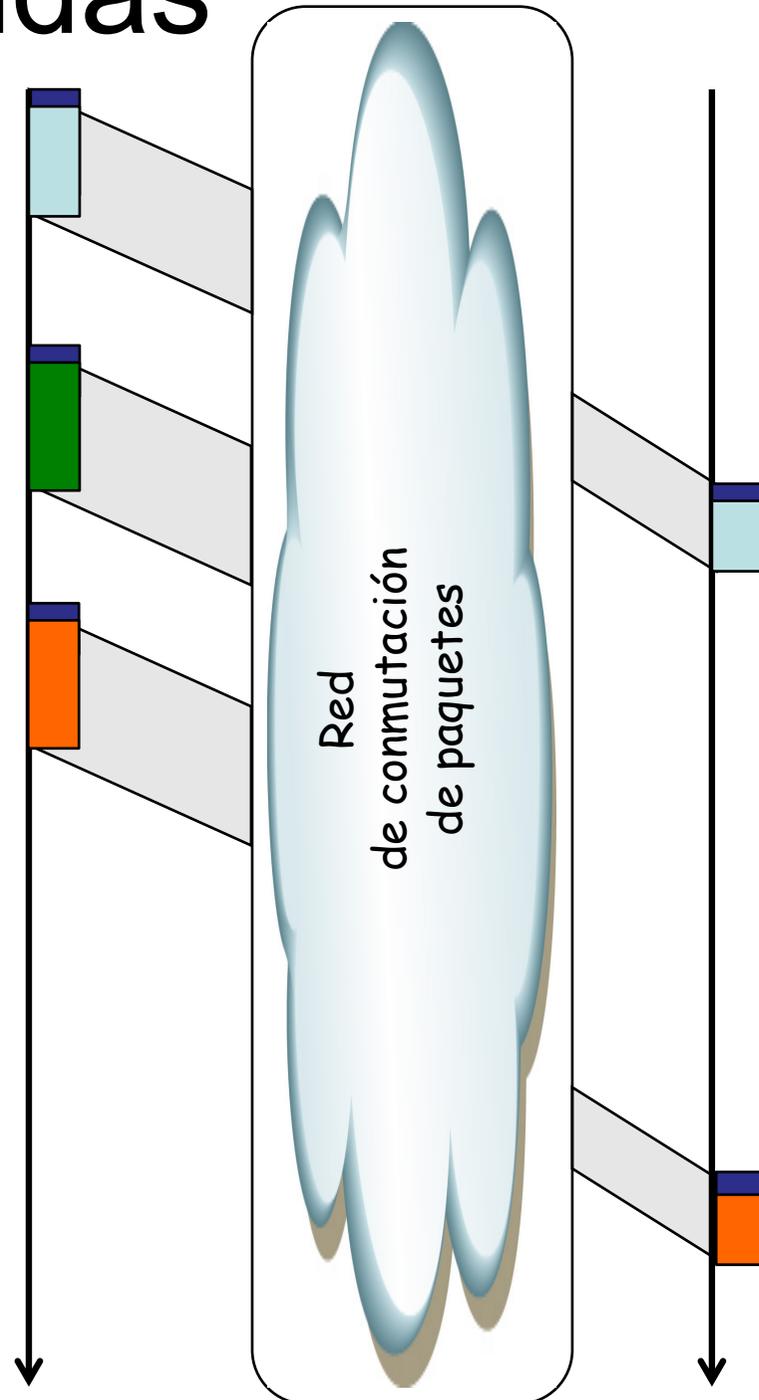
- Los paquetes podrían no llegar nunca a su destino

## 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:  $p_{err}=10^{-6}$ ,  $N=12.000 \rightarrow \rightarrow p_{epk} \approx 10^{-2}$
- (...)

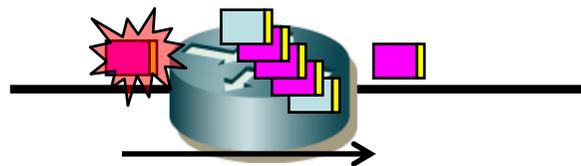


# Pérdidas

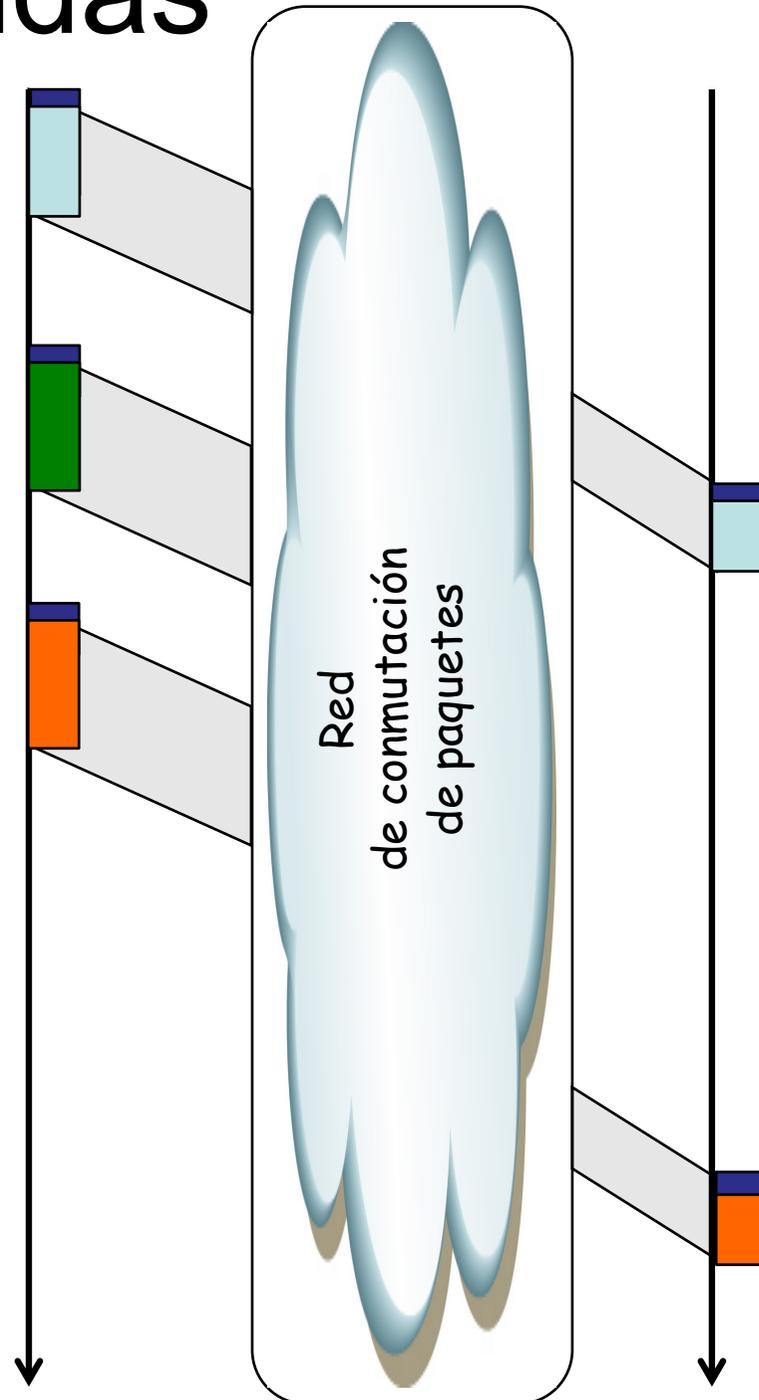
- Los paquetes podrían no llegar nunca a su destino

## Posibles motivos

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



Tamaño del buffer

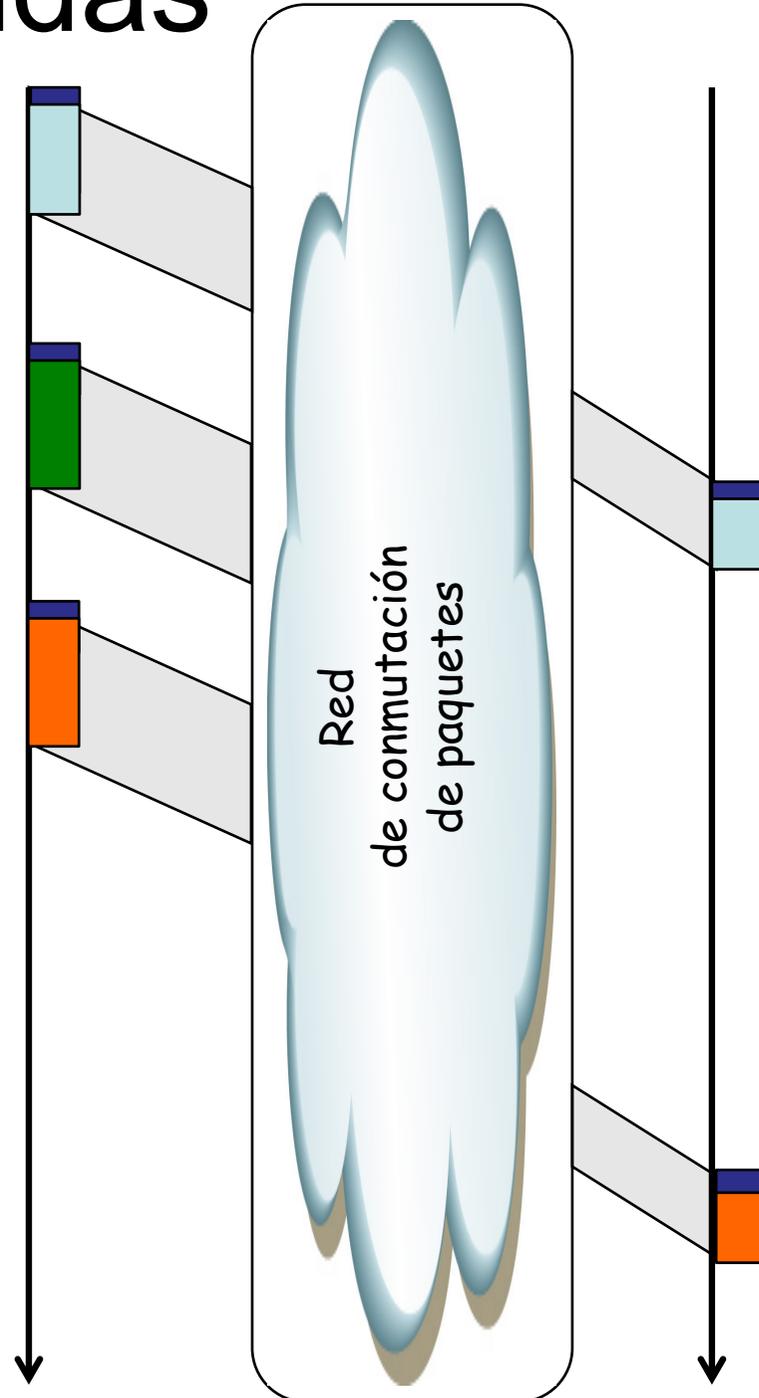
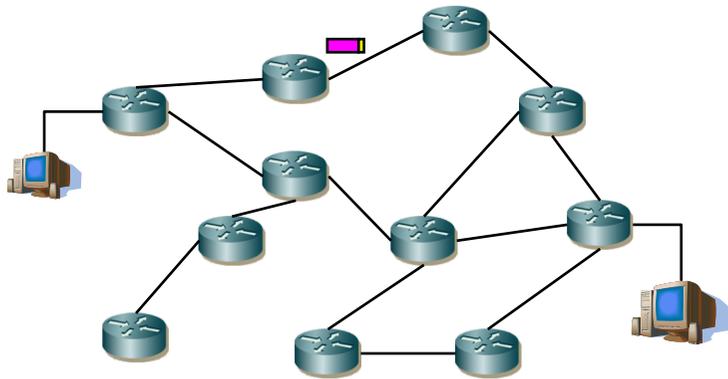


# Pérdidas

- Los paquetes podrían no llegar nunca a su destino

## Posibles motivos

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

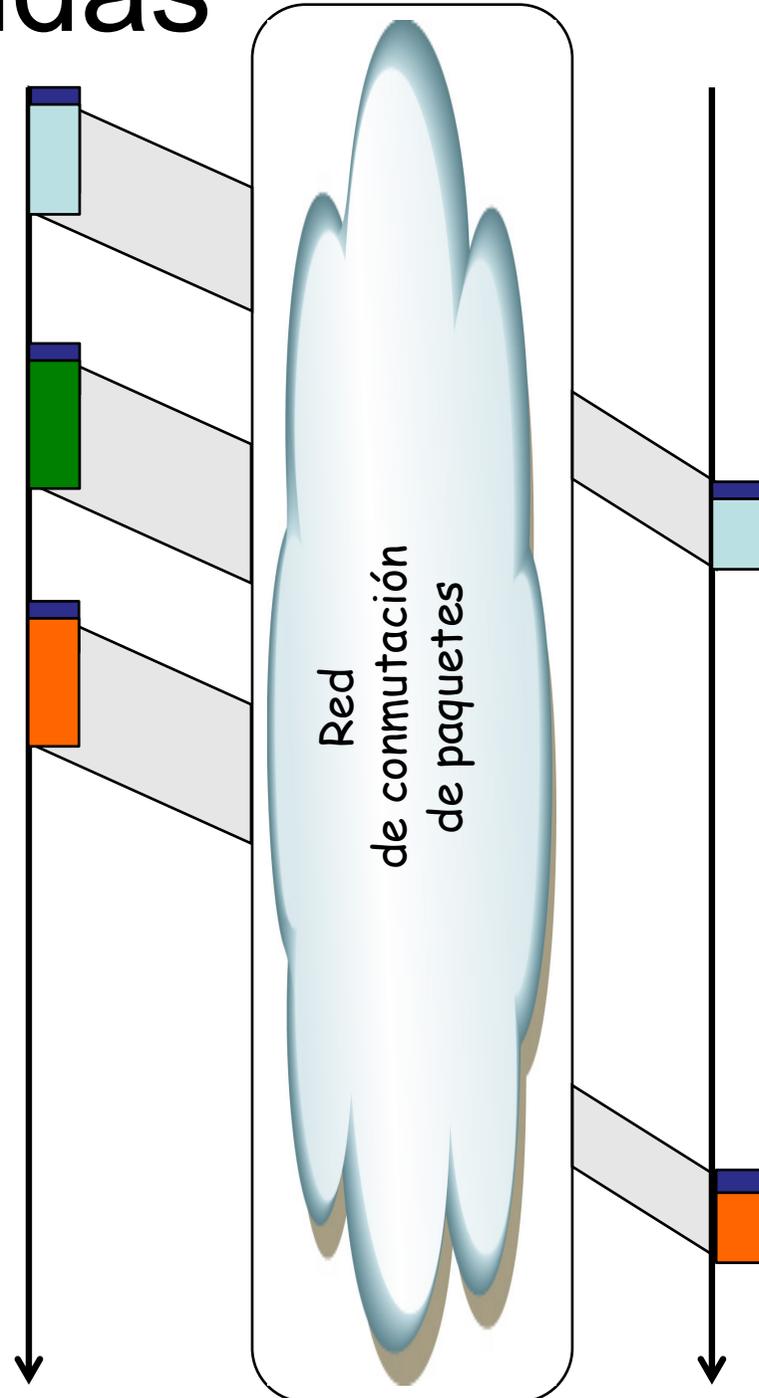
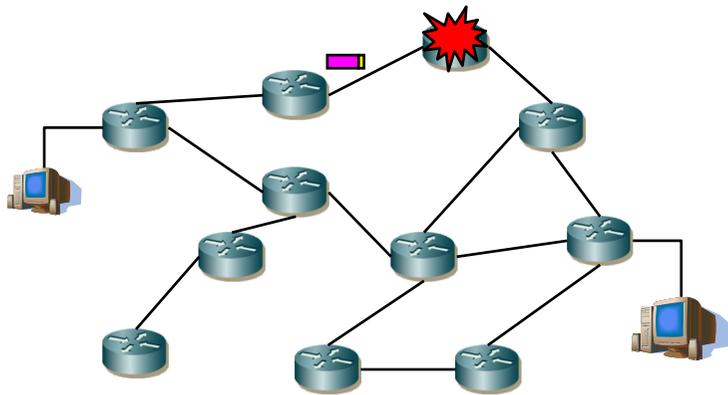


# Pérdidas

- Los paquetes podrían no llegar nunca a su destino

## Posibles motivos

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

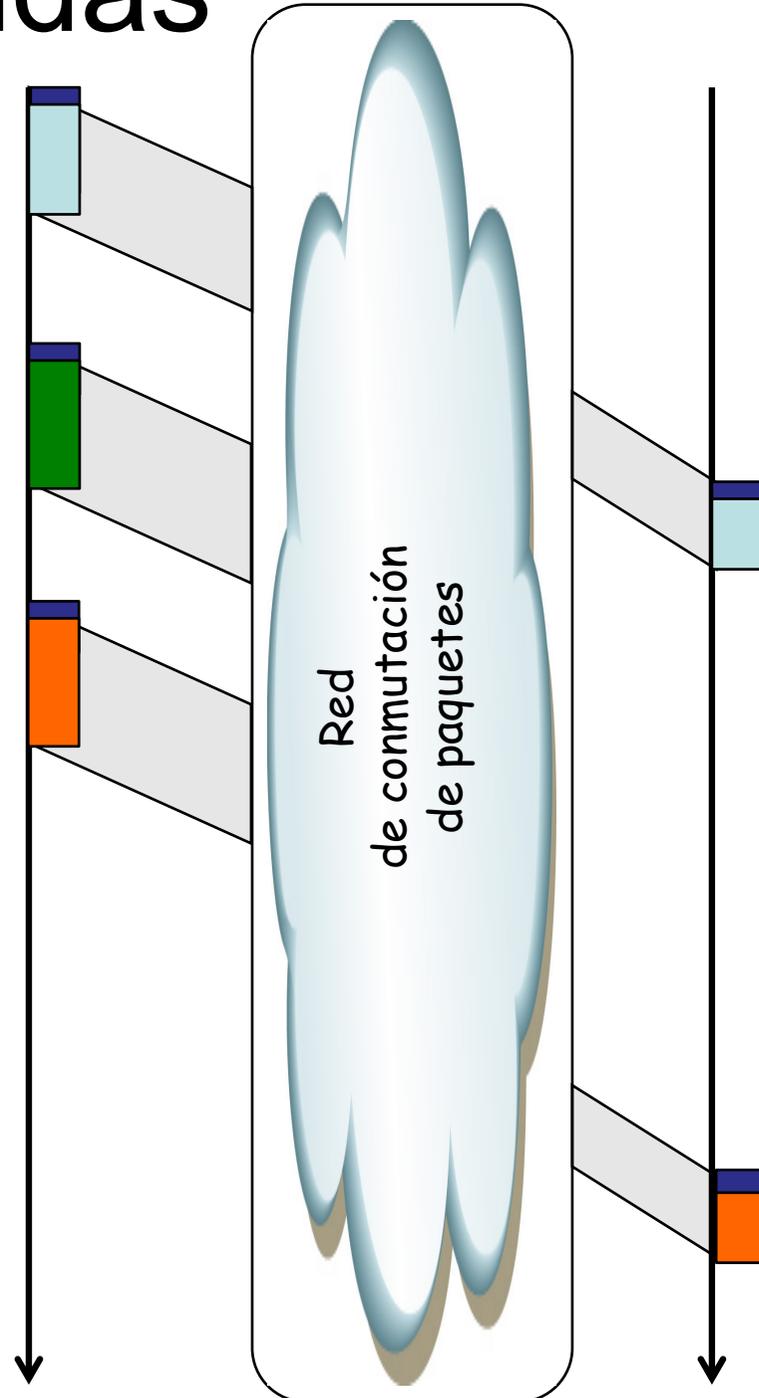


# Pérdidas

- 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



# Ejemplo

## Voz y pérdida de paquetes

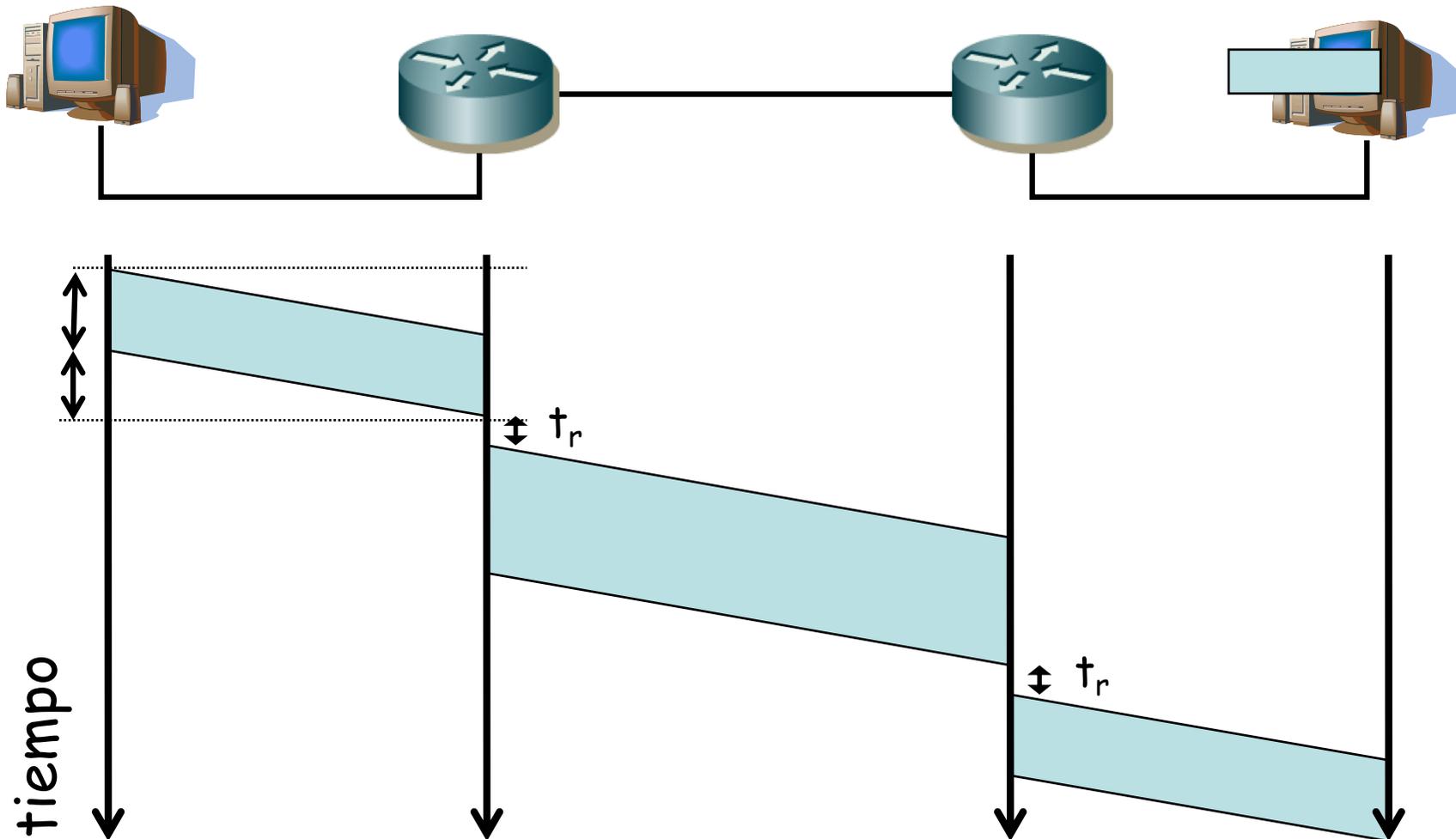
- Causan cortes y saltos
- Un paquete suele contener en torno a 20ms de muestras de voz
  - Si contiene menos: mayor ratio cabeceras/datos
  - Si contiene más: mayor retardo de formación
- Pérdida de 1 paquete se puede intentar recuperar (interpolación, etc)
- Pérdida de más de 1 paquete crea un corte que se nota claramente



# Retardo

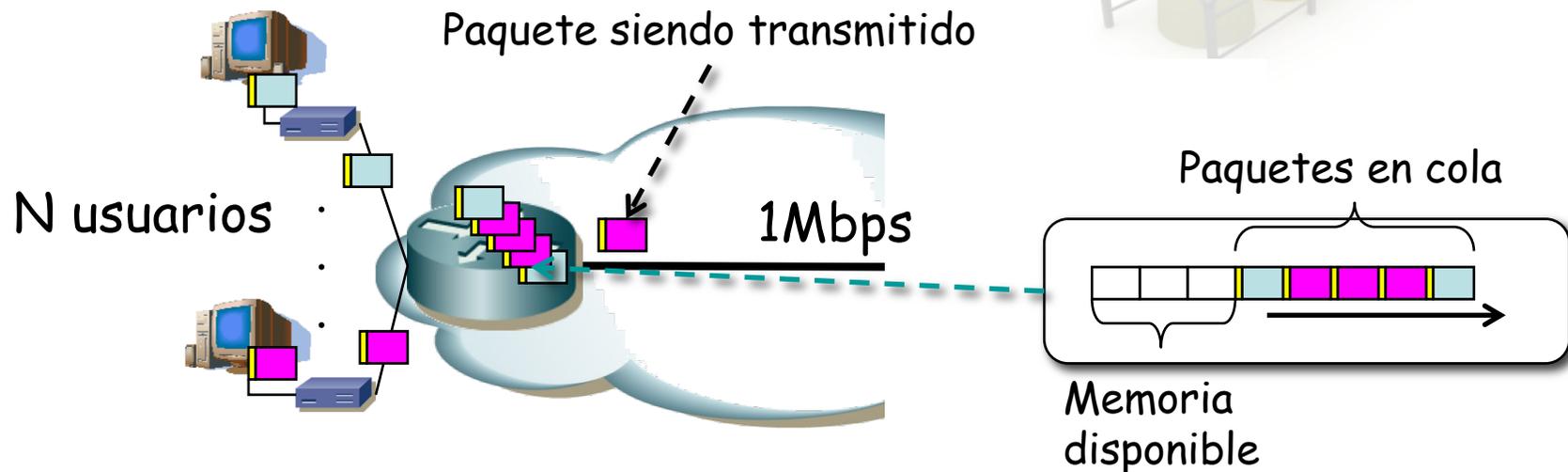
# Retardos

- Transmisión (serialización), propagación, procesado



# Retardo en cola

- Los paquetes pueden llegar al router 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 router 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*)

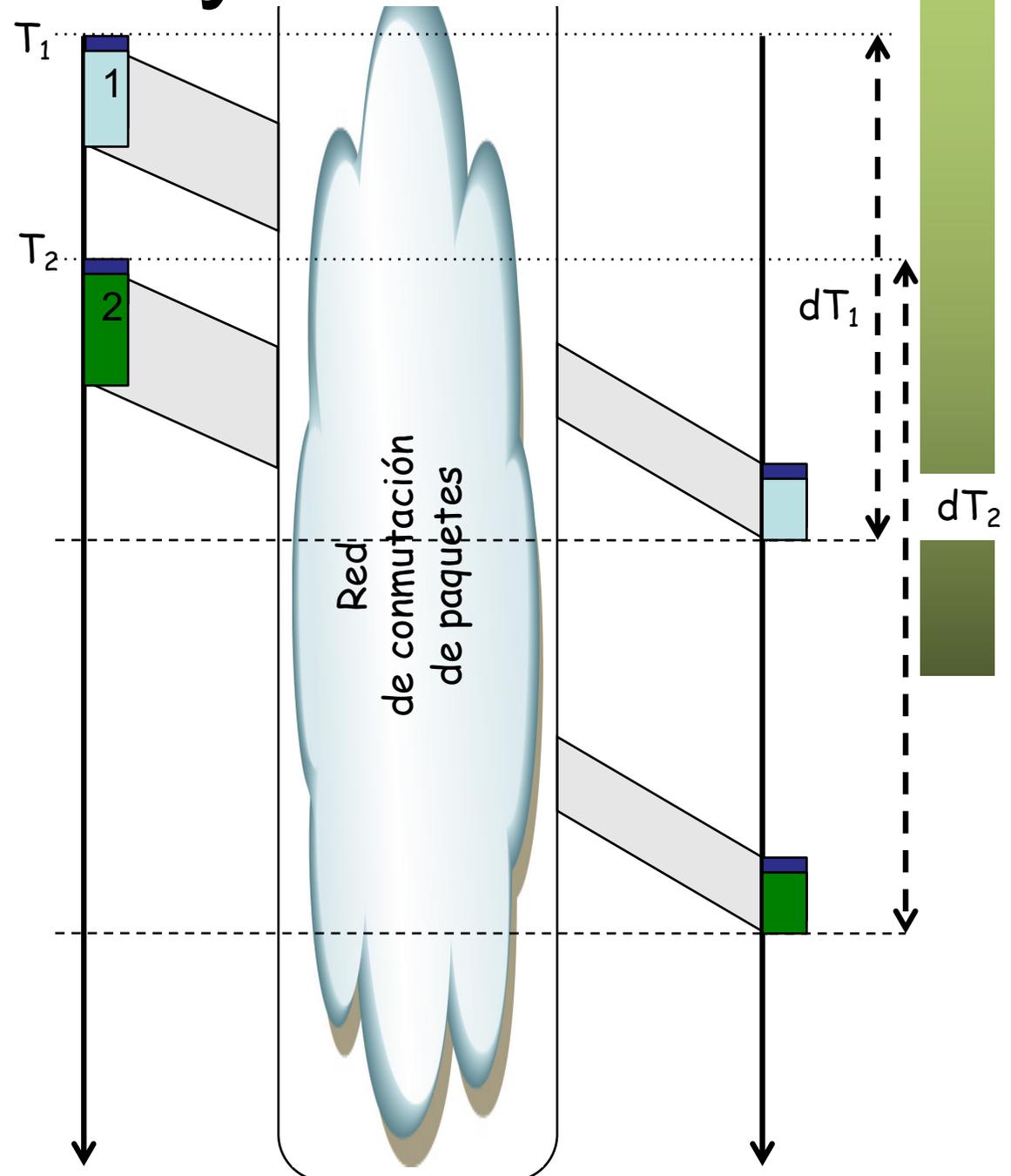


# Variación del retardo

# Packet Delay Variation

## Ejemplo

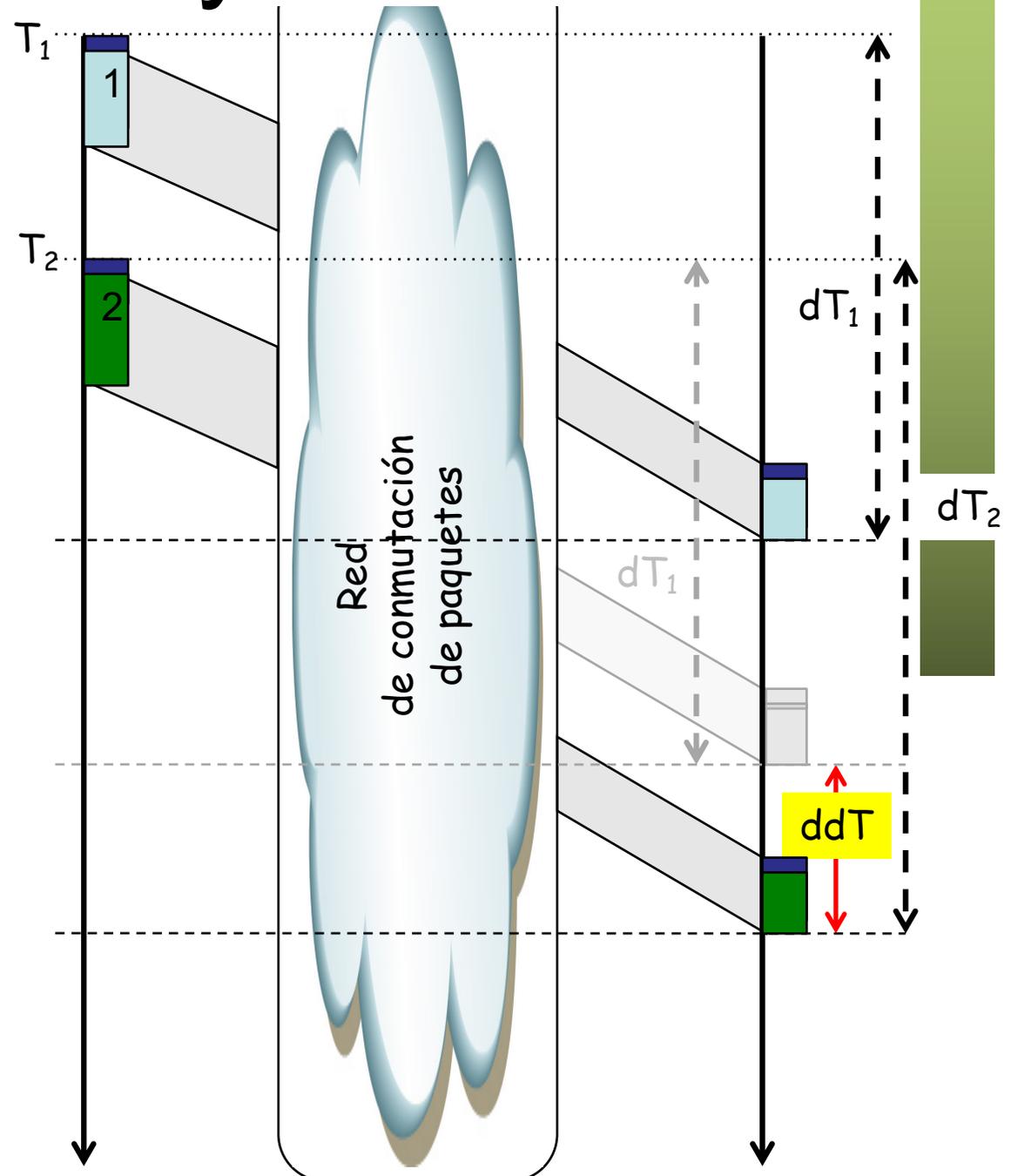
- Dos paquetes (1) y (2)
- Retardos  $dT_1$  y  $dT_2$
- $ddT = dT_2 - dT_1$
- Mide la diferencia entre cuándo ha llegado el segundo paquete y cuándo “debería” haber llegado
- El “debería” sería en el caso de mismo retardo ambos (...)



# Packet Delay Variation

## Ejemplo

- Dos paquetes (1) y (2)
- Retardos  $dT_1$  y  $dT_2$
- $ddT = dT_2 - dT_1$
- Mide la diferencia entre cuándo ha llegado el segundo paquete y cuándo “debería” haber llegado
- El “debería” sería en el caso de mismo retardo ambos (paquete en gris)
- Diferencia puede ser positiva o negativa (atrasarse o adelantarse)



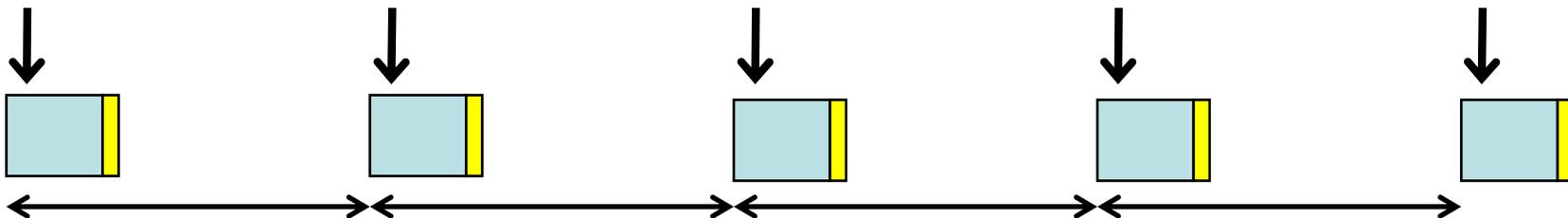
# Efectos del PDV

## Ejemplo

- Codec de voz que genera información digital a tasa constante
- Una vez paquetizada se convierte en paquetes equiespaciados
- (...)



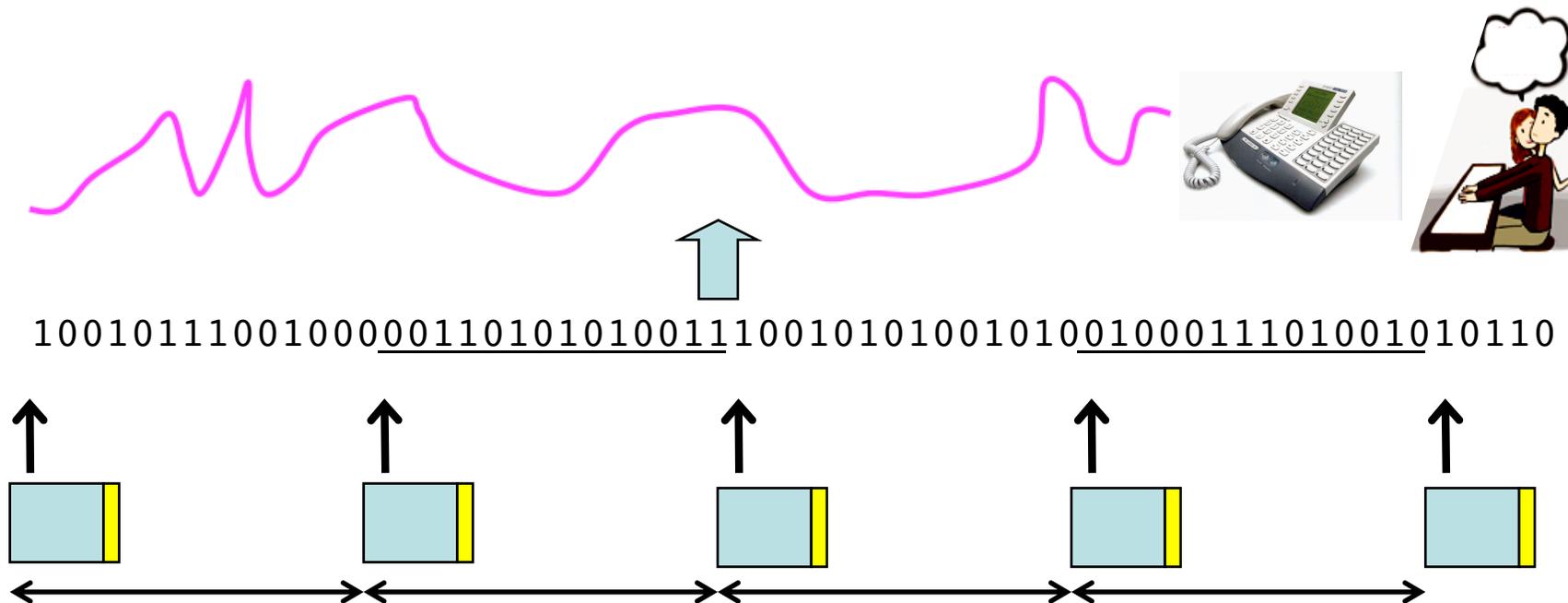
1001011100100000110101010011100101010010100100011101001010110



# Efectos del PDV

## Ejemplo

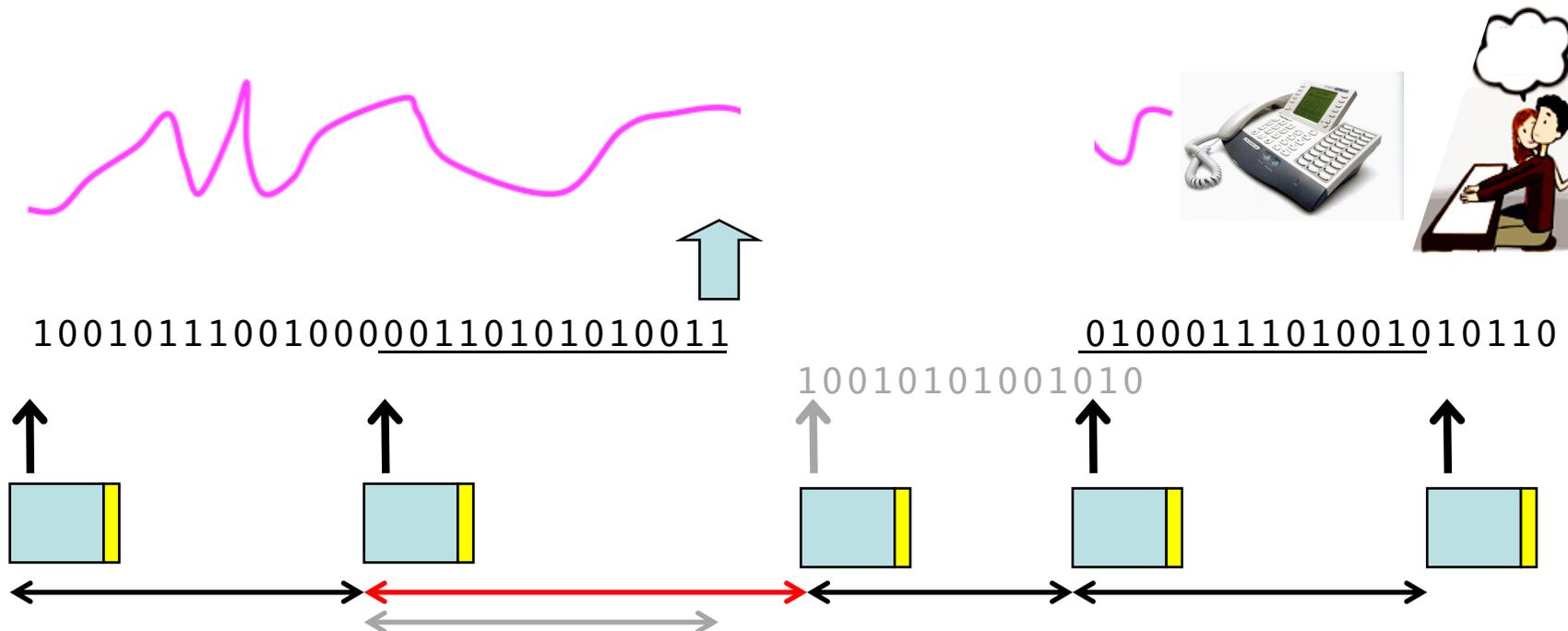
- Codec de voz que genera información digital a tasa constante
- Una vez paquetizada se convierte en paquetes equiespaciados
- En la decodificación se consumen a esa misma tasa
- (...)



# Efectos del PDV

## Ejemplo

- Codec de voz que genera información digital a tasa constante
- Una vez paquetizada se convierte en paquetes equiespaciados
- En la decodificación se consumen a esa misma tasa
- Un primer paquete sufre un retardo mayor que el anterior y puede que cuando llegue “ya sea tarde”
- Es decir, ya no sirve decodificarlo pues ya se ha producido el corte en la reproducción



# Efectos del PDV

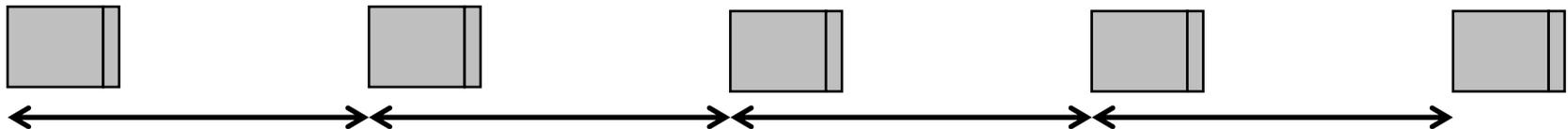
## Solución

- Retrasar comienzo de la reproducción mediante *buffering* en el cliente
- Supongamos que en  $t=0$  tiene el primer paquete y podría empezar a reproducir
- (...)

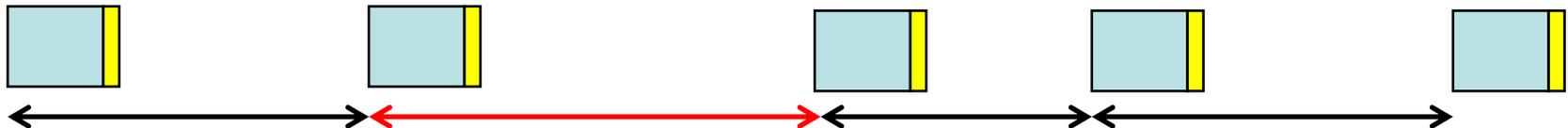
$t=0$   
↓

1001011100100000110101010011100101010010100100011101001010110

ideal



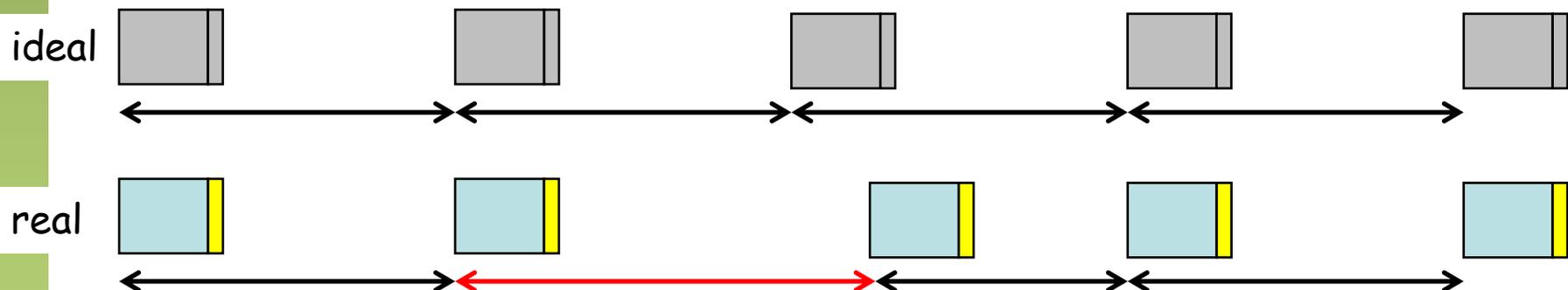
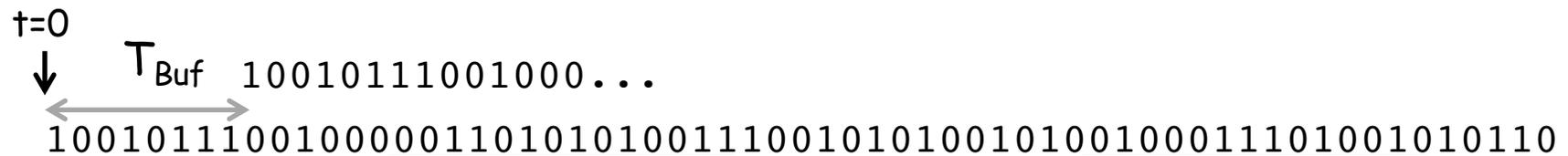
real



# Efectos del PDV

## Solución

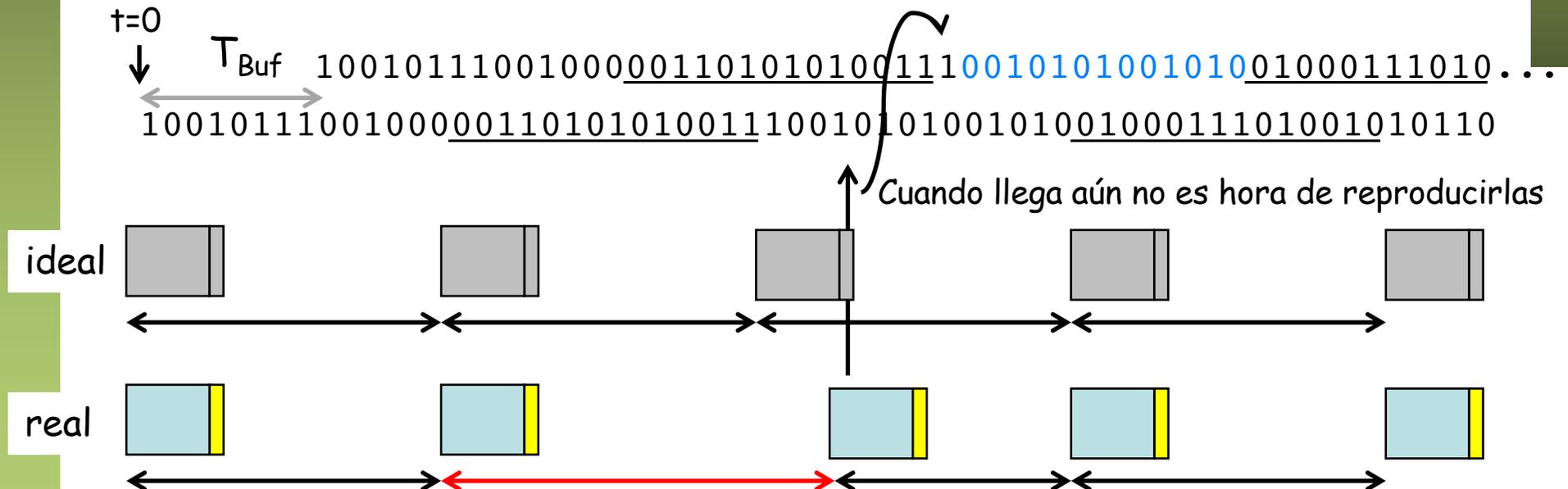
- Retrasar comienzo de la reproducción mediante *buffering* en el cliente
- Supongamos que en  $t=0$  tiene el primer paquete y podría empezar a reproducir
- Se introduce en memoria durante  $T_{Buf}$  (mientras tanto pueden llegar más paquetes, según el tiempo que se desee y lo grande que sea  $T_{Buf}$ )
- (...)



# Efectos del PDV

## Solución

- Retrasar comienzo de la reproducción mediante *buffering* en el cliente
- Supongamos que en  $t=0$  tiene el primer paquete y podría empezar a reproducir
- Se introduce en memoria durante  $T_{Buf}$  (mientras tanto pueden llegar más paquetes, según el tiempo que se desee y lo grande que sea  $T_{Buf}$ )
- El paquete muy retrasado entrará en el buffer y aún se estarán reproduciendo muestras de anteriores si su PDV es menor que  $T_{Buf}$



# Ejemplo

## Paquetes retrasados y jitter (en voz)

- Si un paquete llega demasiado tarde es equivalente a una pérdida
- Si el retardo general end-to-end es demasiado grande se pierde interactividad
- El jitter (variación en el retardo) se puede recuperar con buffer en el receptor
- Si el jitter es demasiado grande el buffer ha de ser tan grande que de nuevo implica demasiado retardo

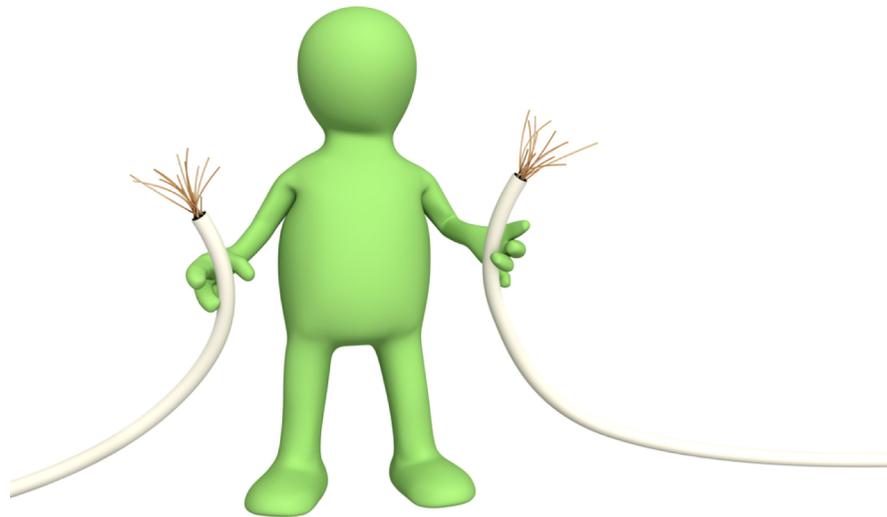


# Disponibilidad

# Availability

- **Network availability**

- Fracción de tiempo en que la conectividad está disponible
- Puede faltar la conectividad por cortes programador o por fallos
- Se tiene en cuenta la disponibilidad de cada elemento y se combinan, según estén en serie o en paralelo
- En serie deben estar todos disponibles (“disponible link 1 Y disponible link 2 Y disponible link 3 ...”)
- En paralelo debe estar alguno disponible (“disponible link 1 O disponible link 2 O disponible link 3 ...”)



# Availability

- **Service availability**

- Fracción de tiempo en que el servicio está disponible dentro de los parámetros de SLA del servicio
- Puede medirse independiente de la disponibilidad de red
- En ese caso no puede superarla o debe medirse condicionada a disponibilidad
- Puede implicar o no al comportamiento de servidores, según cómo se haya definido el servicio

