

# Transporte fiable

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

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

# Hasta ahora

- Protocolo mas simple
  - Stop and wait
  - Con números de secuencia para no entregar duplicados
  - Con ACK que indica cual es el dato que espero
  - Con timeout para el caso de que se pierdan paquetes
- **Garantiza fiabilidad sobre un canal con errores de bits o perdidas**
  - Si el protocolo funciona durante todo el tiempo que haga falta y no se pierden todos los paquetes al final la información llega
  - Lo que no garantiza es cuanto tarda en llegar
- Problemas
  - ¿Cuanto tarda? = ¿Cómo de rápido es el protocolo?

# Eficiencia de stop and wait Con errores de paquetes

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

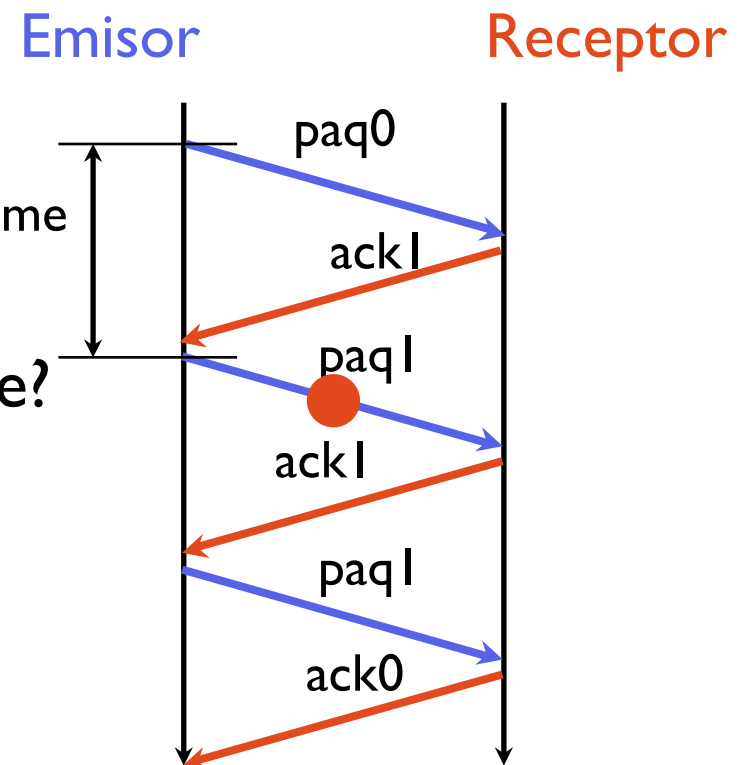
Arquitectura de Redes, Sistemas y Servicios  
Grado en Ingeniería en Tecnologías de Telecomunicación, 2º

# Eficiencia

- Cuanto se tardan en transferir  $s$  bytes con un protocolo de este tipo?
  - Dividimos en paquetes de tamaño  $c$

$s/c$  paquetes

Cuanto se tarda en enviar un paquete?  
 Si no hay errores 1 RTT  
 Si hay errores 2 RTTs?  
 Y si hay errores en la retransmision?



# Tiempo transmisión de un paquete

- Tiempo para transferir 1 paquete  
si la probabilidad de que un paquete se pierda es  $p$ 
    - 1 RTT con probabilidad  $(1-p)$
    - 2 RTT con probabilidad  $(1-p)*p$
    - 3 RTT con probabilidad  $(1-p)*p^2$
    - $n$  RTT con probabilidad  $(1-p)*p^{n-1}$   
(v.a. Distribucion geométrica)
- el numero medio de RTTs se deja como ejercicio  
pero es  $1/(1-p)$  RTTs

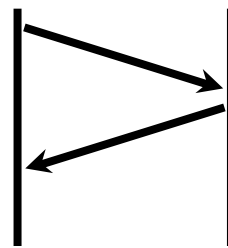
# Prestaciones

- Luego  $s$  bytes se transfieren en  
$$t = s/c * 1/(1-p) * RTT$$
  
y la velocidad de transferencia es  
$$s/t = c * (1-p) /RTT$$
- Y cuanto es eso en un caso real?
- Si elegimos tamaños de paquetes muy grandes hay que mandar menos, pero la probabilidad de perdida de un paquete es mayor
- Si elegimos paquetes pequeños hay que esperar un RTT al menos para mandar cada paquete

# Ejemplo

- Ejemplo:  
 Enlace de 1Gbps con un retardo de 15ms (4500Km), paquetes de 1000 bytes  
 A que velocidad puedo enviar?  
 Suponiendo sin errores

RoundTripTime  
 =30ms



$$v = \frac{tam}{RoundTripTime} = \frac{8000bits}{.03s} = 266Kbps$$

0.026% !! :-)

# Para pensar

- Si tengo un canal de 10Mbps, el retardo de propagación hasta el otro extremo es de unos 40ms. Si envío usando un protocolo stop and wait que usa paquetes de 1000bytes de datos, mas una cabecera pequeña de unos 40bytes
- ¿Cual es el throughput que consigo que se reciba en el otro lado?  
¿Cual es el throughput en el mejor caso? (si no hay ningún error)
- ¿Cual es en ese caso la utilización del canal? (que porcentaje de tiempo esta el canal enviando algo)
- ¿Y si hay una probabilidad de 5% de que un paquete se modifique? ¿y si es del 50%?
  
- ¿Y si es una red de tipo aloha con carga 40%?
- Entonces hay una probabilidad de perdida del paquete.
- ¿Que diferencia hay si el paquete se pierde o se corrompo?
  - En cualquier caso el receptor no recibe los datos del paquete
  - Pero lo demás funciona igual??



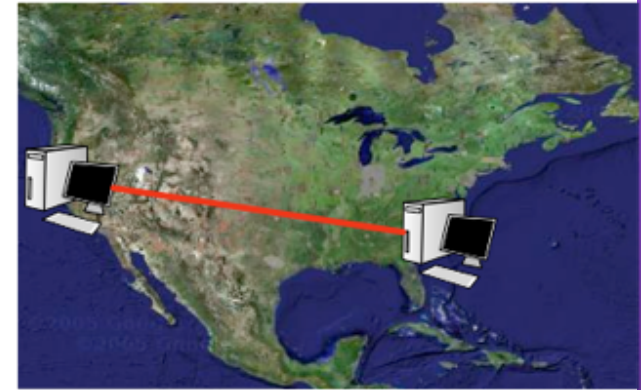
# Eficiencia de stop and wait Con pérdidas de paquetes

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

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

# Prestaciones

- Stop and wait es fiable con perdidas pero sigue siendo muy poco eficiente
- Ejemplo:  
 Enlace de 1Gbps con un retardo de 15ms (4500Km), paquetes de 1000 bytes  
 A que velocidad puedo enviar?
- Si los paquetes se pierden con probabilidad  $p$ 
  - RTT con probabilidad  $(1-p)$
  - RTT+TO con probabilidad  $(1-p)*p$
  - RTT+2TO con probabilidad  $(1-p)*p^2$
  - ...
  - RTT+n\*TO con probabilidad  $(1-p)*p^n$
- El timeout se procura elegir del orden del RTT
  - Mayor implica que reaccionamos despacio a los errores
  - Menor implica que se retransmiten paquetes que no hacia falta
- Las prestaciones son parecidas al anterior, pero con  $p$  probabilidad de perdida del paquete.  
 Normalmente en Internet
  - $P(\text{perdidadelpaquete}) \gg P(\text{corrupciondelpaquete})$
  - Y normalmente la modificación la detecta el nivel de enlace y lo descarta = pérdida



# Conclusiones

- Hay mecanismos y protocolos que permiten conseguir un transporte fiable sobre una red no fiable
  - Stop & wait : esperar hasta asegurarse de que cada paquete ha sido recibido

- Pero y las prestaciones?

*Si me bajo un fichero de 900MB por HTTP desde un servidor. El ping a ese servidor es de 60ms. Y mi acceso a Internet es de empresa a 100Mbps. Cuánto tardare como mínimo? Estoy limitado por el acceso?*

## Próximas clases:

- transporte fiable con mejores prestaciones
- problemas (pensar en las prestaciones de estos protocolos en casos reales)

# Sobre las unidades

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

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

# Nota sobre las unidades

- 1 byte son 8 bits (1B=8b)
- Aunque midiendo memoria se suelen usar prefijos k,M,G,T en potencias de 2 (por ejemplos k para  $2^{10}=1024$  M para  $2^{20}=1048576$  )

Hay un estandar para esto

KiB = 1024B    MiB =1048576

Pero para memoria, tamaños de ficheros, buffers y ventanas se suelen usar potencias de 2

- **En velocidades de transmisión de datos se usan los prefijos del S.I.**

1kB =  $10^3$ B    1MB =  $10^6$ B    1GB =  $10^9$ B    ...

- Las velocidades de transmisión se suelen dar en bits por segundo (kbps, Mbps...). Cuidado con la diferencia entre B y b

1MBps=1MB/s=8Mbps=8Mb/s

- Ejemplo en Ethernet la velocidad es 10Mbps. Un paquete de1000B se transmite en  $(1000B*8b/B)/10Mbps=0.0008s=0.8ms$