

Colección de problemas de STD

Tema 3: TCP

José María Barceló

Juan Carles Cruellas

Jordi Domingo Pascual

Jorge García Vidal

Josep Sunyol

Mayo 2003

Problema 1

Desde un ordenador personal, y a través de un módem, se establece una sesión ftp con el servidor de ftp de la UPC, al tiempo que se monitoriza su evolución haciendo uso de una versión para Windows 95 ® de tcpdump (windump). La tabla 1 muestra en la 1ª columna el número de línea de lo anotado en la 2ª. La 2ª columna muestra una captura de la pantalla de interacción del usuario con su cliente ftp y de éste con el servidor ftp. Lo tecleado por el usuario aparece en negrita y cursiva (el password –dirección de correo- ha sido alterado aunque se ha conservado el número de caracteres). Como puede observarse, una vez establecida la sesión ftp, el usuario se ha dirigido al directorio pub\doc\rfc y ha solicitado la transferencia del fichero *rfc-index.txt* que contiene una relación de todas las RFCs de Internet existentes. La 3ª columna de la tabla 1 muestra una parte de la captura de windump **EJECUTADO EN EL PC** (hasta el momento en que el cliente envía el password y el servidor de ftp lo acepta).

- A) A la vista del grafo de estados de TCP y de lo volcado en la 3ª columna de la tabla 1, especificar en que estado se encuentran los módulos TCP de ambos extremos DESPUES DE CADA UNO de los cuatro primeros segmentos que captura windump en el PC.
- B) Asociar los grupos de segmentos que viajan en un sentido y en otro a las fases por las que pasa la sesión FTP hasta que el password ha sido aceptado por el servidor (fases de FTP: aceptación de la sesión ftp por el servidor, petición de login, entrega de login, petición de password y entrega del mismo).

Cuando el usuario da la orden de transferir el fichero, se establece una nueva conexión TCP entre cliente y servidor. Así pues, en este momento hay dos conexiones TCP abiertas. La tabla 2 muestra partes importantes de esta conexión, **capturadas con windump EN EL SERVIDOR DE FTP (Daos cuenta de que en la tabla 1 la captura se realizaba en el PC y ahora la captura se realiza en el servidor de FTP).**

- C) ¿Cuál es el valor de la ventana de congestión del módulo TCP de la máquina defalla.upc.es en los puntos marcados del (1) al (7)?.
- D) ¿Cuál será aproximadamente el estado de ocupación de los buffers del módulo TCP del PC en los puntos (5) y (6)?.
- E) ¿Cuál creéis que debe ser el valor mínimo de la ventana de congestión del TCP de defalla.upc.es en el punto (8)? ¿Por qué se retransmite el segmento que acaba con el byte número 23361? ¿Cuál será el valor de la ventana de congestión en el punto (9)? ¿Cómo actuará el mecanismo de control de congestión e irá evolucionando la ventana de congestión a partir de aquí?.
- F) ¿Qué está sucediendo a partir del punto (10)? ¿Estaría acabada la sesión ftp para el usuario en el punto (11)? Razonad la respuesta.

TABLA 1: CAPTURA FTP (PARTE INTERACTIVA: CONEXIÓN DE CONTROL)

	CAPTURA DE PANTALLA DE USUARIO	CAPTURA POR WINDUMP (TCPDUMP)
1	C:\>ftp ftp.upc.es	203815.444968 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: S 10865331086533(O) win 8192 <mss1460 (DF)
2	Connected to defalla.upc.es	
3	220 0 0 0	
4	220 0 0 0 Server dFTP de la UPC	203833.553968 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: S 10865331086533(O) win 8192 <mss1460 (DF)
5	220 0 0 0	
6	220 U P C	203833.710154 defalla.upc.es.21 > 193153-26-166uc.nombres.tti.es.1064: S 2104269198(0) ack 1086534 win 8760 <mss1460 (DF)
7	220	
8	220	
9	220 defalla.upc.es FTP server (Thu Dec 23 17:20:7MET 1999) ready.	203833.726773 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: . ack 1 win 8760 (DF)
10	User (defalla.upc.es:(none)):anonymous	
11	331 Guest bgin ok, send your complete email address as password.	203844.147755 defalla.upc.es.21 > 193153-26-166uc.nombres.tti.es.1064: P 1:13(12) ack 1 win 8760 (DF) [tos 0x10]
12	Password: aaaaaaaa@aa.aa.aa	
13	230 Guest bgin ok, access restrictions apply.	203844.328946 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: . ack 13 win 8748 (DF)
14	ftp> cd pub	
15	250 CWD command successful.	203844.523575 defalla.upc.es.21 > 193153-26-166uc.nombres.tti.es.1064: P 13:164(151) ack 1 win 8760 (DF) [tos 0x10]
16	ftp> cd d0	
17	250 CWD command successful.	203844.644831 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: . ack 164 win 8597 (DF)
18	ftp> cd rfc	
19	250 Please read the file README	203847.157912 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: P 1:17(16) ack 164 win 8597 (DF)
20	250 it was last modified on Tue Aug 17 00:00:00 199 - 508 days ago	
21	250 CWD command successful.	203847.327138 defalla.upc.es.21 > 193153-26-166uc.nombres.tti.es.1064: . ack 17 win 8760 (DF) [tos 0x10]
22	ftp> get rfc-index.txt	
23	200 PORT command successful.	203847.342767 defalla.upc.es.21 > 193153-26-166uc.nombres.tti.es.1064: P 164:23269 ack 17 win 8760 (DF) [tos 0x10]
24	150 opening ASCII mode data connection for rfc-index.txt (452955 bytes).	
25	226 Transfer complete.	203847.448858 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: . ack 232 win 8529 (DF)
26	46340 bytes received in 6673 seconds (6.95K bytes/sec)	203855.059919 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: P 17:4229 ack 232 win 8529 (DF)
27	ftp> quit	
28	221 Goodbye.	203855.223013 defalla.upc.es.21 > 193153-26-166uc.nombres.tti.es.1064: P 232:28049 ack 42 win 8760 (DF) [tos 0x10]
29		
30		
31		
32		203855.327765 193153-26-166uc.nombres.tti.es.1064 > defalla.upc.es.21: . ack 280 win 8881 (DF)

TABLA 2 (INTERCAMBIO MASIVO DE DATOS: CONEXIÓN DE TRANSFERENCIA)

CAPTURA POR WINDUMP (TCPDUMP) SIN INDICACIÓN TEMPORAL -

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 S 211639454:211639454(0) win 8760 <mss1460 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 S 211639454:211639454(0) win 8760 <mss1460 (DF) [tos 0x8]
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 S 1124670:1124670(0) ack 211639454 2win 8760 <mss1460 (DF)
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . ack 1 win 8760 (DF) [tos 0x8]
```

(1)----->

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 1:146(1460) ack 1 win 8760 (DF) [tos 0x8]
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 1461 win 8760 (DF)
```

(2)----->

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 1461:2921(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 2921:4381(1460) ack 1 win 8760 (DF) [tos 0x8]
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 2921 win 8760 (DF)
```

(3)----->

```
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 4381 win 8760 (DF)
```

(4)----->

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 4381:5841(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 5841:7301(1460) ack 1 win 8760 (DF) [tos 0x8]
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 5841 win 8760 (DF)
```

(5)----->

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 7301:8761(1460) ack 1 win 8760 (DF) [tos 0x8]
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 7301 win 8760 (DF)
```

(6)----->

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 8761:10221(1460) ack 1 win 8760 (DF) [tos 0x8]
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 8761 win 8760 (DF)
```

(7)----->

Se han suprimido líneas

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 16061:17521(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 17521:18981(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 18981:20441(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 20441:21901(1460) ack 1 win 8760 (DF) [tos 0x8]
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 21901 win 8760 (DF)
```

(8)---->

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 21901:23361(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 23361:4821(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 24821:26281(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 26821:27741(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 27741:29201(1460) ack 1 win 8760 (DF) [tos 0x8]
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . 21901:23361(1460) ack 1 win 8760 (DF) [tos 0x8]
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 23361 win 8760 (DF)
```

(9)---->

Se han suprimido líneas

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 F 463133:463541(408) ack 1 win 8760 (DF) [tos 0x8]
```

(10)-->

```
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 . ack 46354 2win 8760 (DF)
193-15326166.uc.nombres.ttl.es.1075> defalla.upc.es.20 F 1:1(0) ack 46354 2win 8760 (DF)
```

```
defalla.upc.es.20> 193-15326166.uc.nombres.ttl.es.1075 . ack 2 win 8760 (DF) [tos 0x8]
```

(11)-->

Problema 2

Al host PC4 observem l'intercanvi de datagrames IP amb `tcpdump ip -i eth0 > trace`. El contingut del fitxer `trace` es el següent

```
12:55:45.92 PC4.local.9088 > Servidor2.local.www: S 3125459530:3125459530(0) win 255 <mss 536>
(1)——>
12:55:45.92 Servidor2.local.www > PC4.local.9088: S 2025570752:2025570752(0) ack 3125459531
win 32736 <mss 1460>
(2)——>
12:55:45.98 PC4.local.9088 > Servidor2.local.www: . ack 1 win 65534
(3)——>
Ara el client envia una comanda HTTP "GET" al servidor...
12:55:45.99 PC4.local.9088 > Servidor2.local.www: P 1:103(102) ack 1 win 65534
El servidor contesta...
12:55:46.00 Servidor2.local.www > PC4.local.9088: P 1:537(536) ack 103 win 32736 (DF)
12:55:46.03 PC4.local.9088 > Servidor2.local.www: . ack 537 win 64998
(4)——>
12:55:46.03 Servidor2.local.www > PC4.local.9088: . 537:1073(536) ack 103 win 32736 (DF)
12:55:46.03 Servidor2.local.www > PC4.local.9088: . 1073:1609(536) ack 103 win 32736
12:55:46.07 PC4.local.9088 > Servidor2.local.www: . ack 1073 win 64998
12:55:46.07 PC4.local.9088 > Servidor2.local.www: . ack 1609 win 64998
12:55:46.10 Servidor2.local.www > PC4.local.9088: . 1609:2145(536) ack 103 win 32736
12:55:46.10 Servidor2.local.www > PC4.local.9088: . 2145:2681(536) ack 103 win 32736
12:55:46.10 Servidor2.local.www > PC4.local.9088: . 2681:3217(536) ack 103 win 32736
12:55:46.11 Servidor2.local.www > PC4.local.9088: . 3217:3753(536) ack 103 win 32736
12:55:46.12 PC4.local.9088 > Servidor2.local.www: . ack 2145 win 64998
12:55:46.12 PC4.local.9088 > Servidor2.local.www: . ack 2681 win 64998

12:55:46.12 PC4.local.9088 > Servidor2.local.www: . ack 3217 win 64998
12:55:46.12 Servidor2.local.www > PC4.local.9088: . 3753:4289(536) ack 103 win 32736
12:55:46.13 PC4.local.9088 > Servidor2.local.www: . ack 3217 win 64998
12:55:46.13 Servidor2.local.www > PC4.local.9088: . 4289:4825(536) ack 103 win 32736
12:55:46.13 PC4.local.9088 > Servidor2.local.www: . ack 3217 win 64998
12:55:46.30 Servidor2.local.www > PC4.local.9088: . 4825:5361(536) ack 103 win 32736
12:55:46.32 Servidor2.local.www > PC4.local.9088: . 3217:3753(536) ack 103 win 32736
12:55:46.32 Servidor2.local.www > PC4.local.9088: . 3753:4289(536) ack 103 win 32736
12:55:46.33 PC4.local.9088 > Servidor2.local.www: . ack 5361 win 64998
(5)——>
Hem eliminat algunes línies...
12:55:57.32 Servidor2.local.www > PC4.local.9088: . 15217:15753(536) ack 103 win 32736
12:55:57.32 Servidor2.local.www > PC4.local.9088: . 15753:16289(536) ack 103 win 32736
12:55:57.63 PC4.local.9088 > Servidor2.local.www: . ack 15217 win 536
12:55:57.63 Servidor2.local.www > PC4.local.9088: . 16289:16825(536) ack 103 win 32736
12:55:57.63 Servidor2.local.www > PC4.local.9088: . 16825:17361(536) ack 103 win 32736
12:55:58.93 PC4.local.9088 > Servidor2.local.www: . ack 15753 win 0
12:55:58.93 PC4.local.9088 > Servidor2.local.www: . ack 17361 win 536
(6)——>
12:55:59.03 Servidor2.local.www > PC4.local.9088: F 17361:17361(0) ack 103 win 32736
(7)——>
12:55:59.03 PC4.local.9088 > Servidor2.local.www: . ack 17362 win 32736
(8)——>
12:55:59.03 PC4.local.9088 > Servidor2.local.www: F 103:103(0) ack 17362 win 32736
(9)——>
12:55:59.06 Servidor2.local.www > PC4.local.9088: ack 104 win 32736
```

1.A. Digues quin és l'estat del client i del servidor als punts (1), (2), (3), (4) i (5).

1.B. Quina serà la finestra de congestió del client en el punt 4.

1.C. Digues quina serà, probablement, l'ocupació dels buffers de transmissió i de recepció TCP del client i del servidor en els punts (5) i (6).

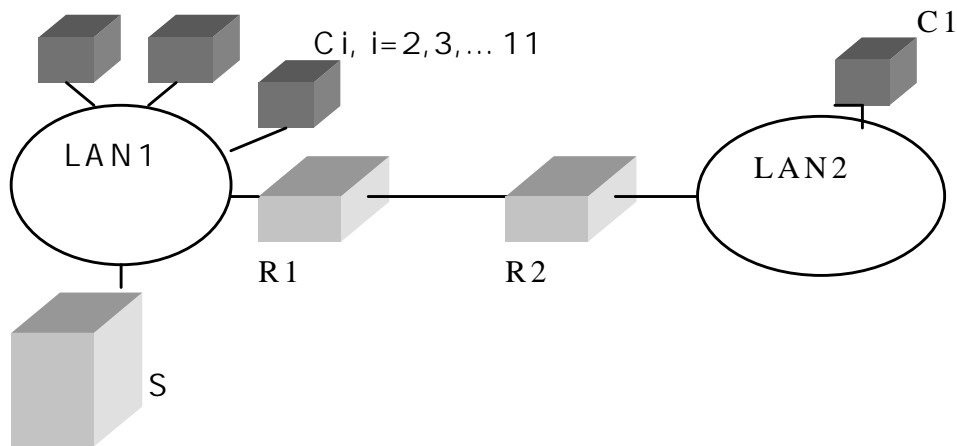
1.D. Per què tenim just abans del punt (5) tres acks enviats per el client al servidor, confirmant els mateixos bytes?.

1.E. Explica perquè el client envia un ack del byte 5361 després de la retransmissió del segments 3217:3753 (punt (5)).

Explica en quins estats es troben el client i el servidor en els punts (7), (8), i (9).

Problema 3

Tenemos la red que se muestra en la figura. S es un servidor de ficheros. Los clientes C_i ($i=1,2,\dots,11$) ejecutan una aplicación que lee un fichero de S y lo almacena en su disco local.



Suponemos que tanto los clientes como el servidor tienen una CPU con una velocidad de proceso muy alta (supondremos que es infinitamente rápida). Sin embargo el acceso a sus discos está limitado: el servidor puede leer de su disco como máximo a 8 MBps (es decir 64 Mbps) mientras que los clientes pueden escribir de forma sostenida en su disco como máximo a 2 MBps (16 Mbps).

Cuando se establece una conexión TCP tanto los clientes como el servidor reservan buffers de transmisión y de recepción de 16 KB.

Los routers R1 y R2 tienen buffers en cada interface de salida con capacidad limitada a 4.5 KB (3 paquetes de 1500 B).

Las dos LANs son de gran capacidad (100 Mbps) mientras que el enlace punto a punto es mucho más lento (100 Kbps). Los retardos de propagación son despreciables.

La MTU en toda la red es de 1500 B.

Las pérdidas por errores de transmisión en las dos LANs y en el enlace punto a punto son despreciables.

Apartado 1: Supongamos que el cliente C1 establece una conexión TCP con el servidor S.

- ¿Qué segmentos se intercambian durante el establecimiento de la conexión? ¿Qué valores tendrán en esos segmentos los siguientes flags: SYN, FIN, RST, ACK? ¿Qué MSS se anunciará?
- Explica los motivos por los que los números de secuencia iniciales en TCP se escogen de forma aleatoria.

Ejecutamos el comando `netstat -a` en el servidor S y obtenemos la siguiente salida:

Local Address	Remote Address	State
*.111	*.*	LISTEN
*.21	*.*	LISTEN
147.83.30.10.3045	147.83.30.33.60	ESTABLISHED
147.83.30.10.111	147.83.30.20.3200	SYN_RCVD
127.0.0.1.111	127.0.0.1.3200	ESTABLISHED
147.83.30.10.21	147.83.30.20.2000	FIN_WAIT1

- ¿Cuántos servidores tenemos activos en S? ¿Por qué en la columna "Local Address" encontramos dos direcciones IP distintas? ¿En qué conexión/conexiones S actúa como cliente?
- ¿Qué secuencia de intercambio de segmentos ha ocurrido probablemente en la conexión en estado SYN_RCVD? ¿Y en la conexión en estado FIN_WAIT1?. ¿Qué esperas que suceda en los próximos segundos en las dos anteriores conexiones?.

Apartado 2: Supongamos ahora que en la red solo hay activa la conexión entre S y C1.

- a) ¿Qué limita la velocidad de transferencia a largo plazo: la ventana de congestión o la ventana anunciada?(razona tu respuesta).
- b) Decir cuál será la ocupación (lleno/vacío) a largo plazo de los siguientes buffers:
 - Buffer de transmisión y buffer recepción de S.
 - Buffer de transmisión y buffer recepción de C1.
 - Buffer asociado al interface línea serie y buffer asociado al interface ethernet de R1.
 - Buffer asociado al interface línea serie y buffer asociado al interface ethernet de R2.
- c) ¿Qué ventanas anunciadas viajarían en los segmentos de S a C1? Idem para los segmentos que van de C1 a S.

Apartado 3: Cerramos la conexión anterior y abrimos una nueva conexión entre S y C2.

- a) ¿Qué limita, a largo plazo, la velocidad de transferencia: la ventana de congestión o la ventana anunciada?. Decir de forma aproximada cuál será, a largo la plazo, la velocidad de transferencia que obtendremos (en bps).
- b) ¿Cuál será la ocupación (lleno/vacío) a largo plazo de los siguientes buffers?:
 - Buffer de transmisión y buffer recepción de S.
 - Buffer de transmisión y buffer recepción de C2.
- c) ¿Qué ventanas anunciadas viajarían en los segmentos de S a C2? Idem para los segmentos que van de C2 a S.

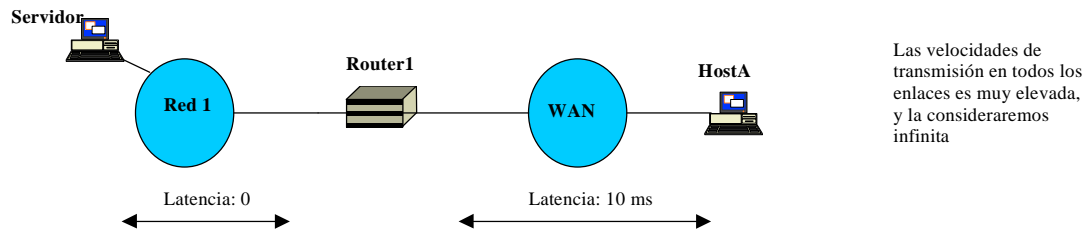
Apartado 4: Supongamos que ahora establecemos conexiones entre el servidor S y todos los clientes de la LAN1, es decir Ci, $i=2..11$.

- a) ¿Qué limita ahora, a largo plazo, la velocidad de transferencia? Decir de forma aproximada cuál será, a largo la plazo, la velocidad de transferencia que obtendremos *por cada conexión* (en bps).
- b) Decir cuál será la ocupación (lleno/vacío) a largo plazo de los siguientes buffers:
 - Buffers de transmisión y buffers recepción de S.
 - Buffer de transmisión y buffer recepción de Ci.
- c) ¿Qué ventanas anunciadas viajarían en los segmentos de S a Ci? Idem para los segmentos que van de Ci a S.

Apartado 5: Supongamos que desde el cliente C1 nos conectamos al puerto de carga del servidor S. Hacer una gráfica en la que se muestre la transferencia de los primeros 30 Paquetes de dicha conexión, indicando claramente en qué estado se halla la conexión TCP (Slow-Start o Congestion Avoidance). Suponemos que el cliente C1 envía un ACK por cada segmento de datos que recibe.

Pregunta 4

Tenemos la red mostrada en la figura:



El HostA se conecta al puerto de carga del servidor.

La latencia introducida en la red 1 es despreciable mientras que en la WAN tenemos una latencia de 10 ms.

En todos los enlaces la velocidad de transmisión es muy elevada y la consideraremos infinita.

Inicialmente supondremos que no se producen pérdidas y que el tamaño de los buffers del router es infinito.

Supondremos además que la velocidad de procesamiento del servidor y del Host es también infinitamente grande.

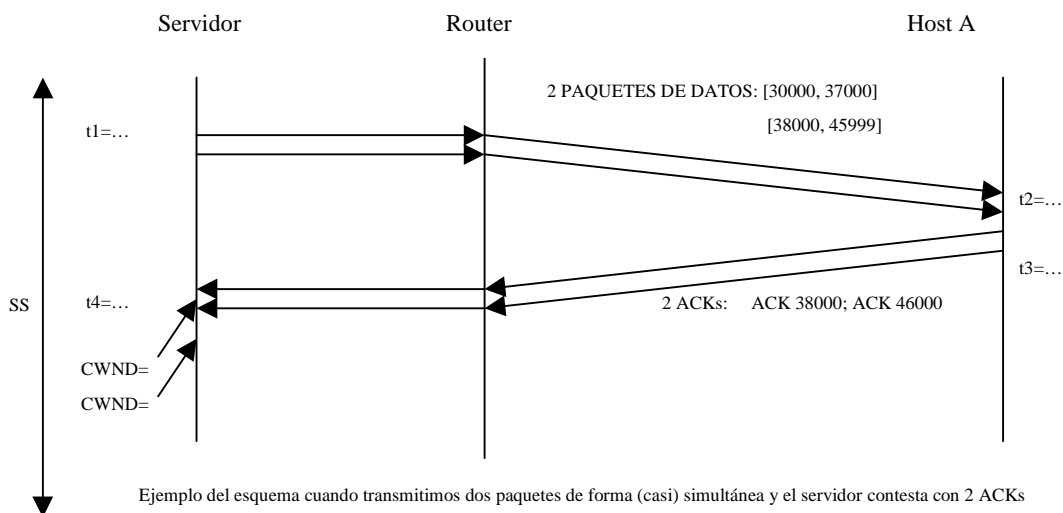
Cuando un segmento lleva datos tiene un tamaño de 8 KB y el valor máximo de la ventana de transmisión es de 64 KB.

Es decir:

$$\text{Ventana Transmisión} = \text{MIN}(64 \text{ KB}, \text{awnd}, \text{cwnd}),$$

donde *awnd* es la ventana anunciada y *cwnd* es la ventana de congestión.

a) Hacer un esquema como el mostrado en el que se indique CLARAMENTE los instantes de llegada y salida al Host A y al servidor de los segmentos TCP. Mostrar los primeros 140ms de tiempo. Si varios paquetes se transmiten o reciben en el mismo instante de tiempo, indicarlo tal como se muestra en el esquema ejemplo. Después de la recepción de cada segmento por parte del HostA se debe indicar el valor de la ventana de congestión. Supondremos que el Host A envía un ACK por cada segmento de datos que recibe. En cada segmento de datos indicar la numeración de los bytes que transporta. En los ACKs indicar hasta qué byte confirman. Suponemos que empezamos a numerar los bytes desde 0. (Intentar que el esquema quepa en una página).



b) ¿Cuál es el valor máximo de la ventana de transmisión que alcanzaremos? ¿Cuál es la velocidad media de transferencia de datos entre el servidor y el host A que tendremos en régimen estacionario (en Mbps)?

En el siguiente apartado supondremos que la red 1 introduce pérdidas en los segmentos. Más exactamente supondremos que si el servidor empieza a numerar los datos en el byte 0, tenemos una pérdida la primera vez que transmitimos el

segmento que lleva los bytes (32000,39999). Suponemos que los paquetes de tamaño 40 bytes (es decir, ACKs, SYNs, etc) que atraviesan la red *en cualquier sentido* nunca se pierden.

Hacemos las siguientes suposiciones:

- El TCP del servidor y del router sólo incorpora los mecanismos de Slow Start y de Congestion Avoidance pero no otros mecanismos como Fast retransmit, Fast recovery, etc.
- El valor del temporizador RTO de TCP se mantiene a un valor constante de 22 ms.

Tener en cuenta además que:

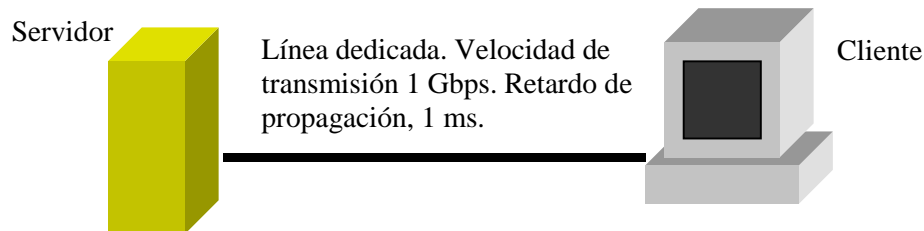
- Los ACKs duplicados no incrementan la ventana de congestión.
- Los ACKs acumulados son tomados como un único ACK a la hora de incrementar la ventana de congestión.
- La ventana de transmisión se refiere a los segmentos TCP que han sido enviados pero que no han sido confirmados.

c) Hacer una gráfica similar a la anterior, también hasta el instante 140 ms, en la que se muestre la evolución de la transmisión en este caso. Indicar la duración de las fases de Slow Start y las fases de Congestion Avoidance. Además del valor de *wnd*, especificar el valor de *ssthreshold*.

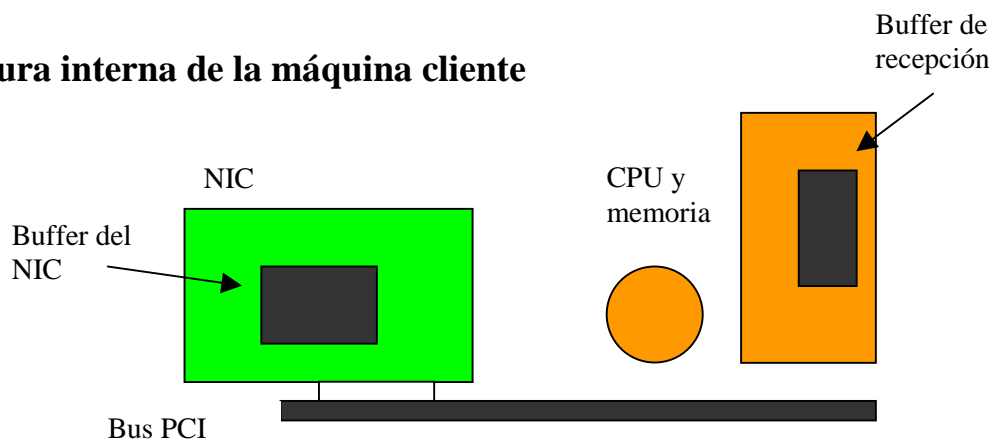
d) En el apartado anterior suponíamos que el temporizador RTO de TCP se mantiene constante. Sin embargo en realidad esto no sucederá así. Explicar brevemente qué mecanismo se utiliza para fijar el valor de dicho temporizador. NO es necesario que expliques el algoritmo concreto que se utiliza (algoritmo de Van Jacobson y Karels).

Pregunta 5

Supongamos que conectamos un servidor remoto de nuestra empresa a un cliente local a través de una línea dedicada a 1 Gbps. El retardo de propagación de la señal a lo largo de dicha línea es de 10 ms (ver figura).



Estructura interna de la máquina cliente



La máquina cliente es un PC equipado con una CPU que trabaja a una velocidad muy elevada (que consideraremos infinita) de forma que el proceso cliente consume los datos del buffer de recepción también a velocidad muy elevada (que consideraremos infinita). El bus de E/S de dicho sistema, al cual conectamos la tarjeta de red, es un bus PCI, con una velocidad máxima de transferencia de 1 Gbps. Sin embargo dicho bus es utilizado también por otros periféricos del sistema, de forma que la velocidad efectiva a la que la tarjeta de red puede enviar los datos a memoria es de sólo un 10% de la velocidad máxima de transferencia del bus.

El sistema operativo del cliente reserva 32 Kbytes para el buffer de recepción de datos TCP. El valor máximo de la ventana de congestión es de 64 Kbytes.

El servidor realiza hacia el cliente una transferencia continua de datos, enviados en tramas con MSS de 1000 Bytes. El cliente solo envía en sentido contrario ACKs TCP de 40 Bytes.

Supongamos que compramos para el cliente una tarjeta de red con un buffer de paquetes con una capacidad muy grande (ilimitada) y que la máquina cliente envía un ACK por cada segmento TCP correctamente recibido.

- Dibujar en forma de gráfica cuál será la evolución de la *ventana de transmisión* en el servidor, de la *ventana anunciada* por el cliente y de la velocidad de transferencia entre el servidor y el cliente. Considerar que el instante de tiempo $t=0$ corresponde a la finalización del proceso de establecimiento de la conexión. Indicar claramente los tiempos en el eje X. Indicar en dicha gráfica en que fase de control de congestión se haya el servidor.
- ¿Cuál será la máxima velocidad de transferencia que alcanzaremos?
- ¿Cuántos datos tendremos almacenados en el buffer de la tarjeta de red?
- ¿Cómo variarían las respuestas b) y c) si en vez de TCP usamos UDP como protocolo de transporte?

Supongamos ahora que la tarjeta de red que empleamos solo tiene capacidad de almacenamiento para 3 paquetes de datos.

- misma pregunta que en a).
- misma pregunta que en b)
- misma pregunta que d)