

## Solució del control

**Segon control de Xarxes de Computadors (XC), Grau en Enginyeria Informàtica**

**12/12/2011**

**Tardor 2011**

Duració: 1,5 hores. Respondre els problemes en fulls separats. Raonar les respostes

### Pregunta 1. (5 punts).

En una sessió del protocol de aplicació FTP capturamos una serie de intercambios de segmentos TCP. Se identifica cada intercambio con un número de línea y se incluye información varia como Origen y Destino, puertos TCP, flags activos (SYN, ACK, PSH, FIN), número de secuencia del primer octeto (Seq), número de acknowledge (Ack) y longitud de datos de usuario (Len).

NOTA: El protocolo FTP (File Transfer Protocol) es Cliente/Servidor y usa dos puertos en el Servidor, el 21 para control (solicitud de operaciones) y el 20 para transferencia de datos.

No.	Origen	Destino	Puertos	Flags	Núm. Sec.	Núm. Ack.	Longitud segmento
1	Cl	Serv	4484 > 21	[SYN]	Seq= 0	Ack= 0	Len=0
2	Serv	Cl	21 > 4484	[SYN, ACK]	Seq= 0	Ack= 1	Len=0
3	Cl	Serv	4484 > 21	[ACK]	Seq= 1	Ack= 1	Len=0
4	Serv	Cl	21 > 4484	[PSH, ACK]	Seq= 1	Ack= 1	Len=126
5	Cl	Serv	4484 > 21	[ACK]	Seq= 1	Ack= 127	Len=0
6	Cl	Serv	4484 > 21	[PSH, ACK]	Seq= 1	Ack= 127	Len=13
7	Serv	Cl	21 > 4484	[PSH, ACK]	Seq= 127	Ack= 14	Len=34
8	Cl	Serv	4484 > 21	[ACK]	Seq= 14	Ack= 161	Len=0
9	Cl	Serv	4484 > 21	[PSH, ACK]	Seq= 14	Ack= 161	Len=12
10	Serv	Cl	21 > 4484	[PSH, ACK]	Seq= 161	Ack= 26	Len=33
11	Cl	Serv	4484 > 21	[ACK]	Seq= 26	Ack= 194	Len=0
12	Cl	Serv	4484 > 21	[PSH, ACK]	Seq= 26	Ack= 194	Len= 27
13	Serv	Cl	21 > 4484	[PSH, ACK]	Seq= 194	Ack= 53	Len=27
14	Cl	Serv	4484 > 21	[PSH, ACK]	Seq= 53	Ack= 221	Len=15
15	Serv	Cl	20 > 4485	[SYN]	Seq= 0	Ack= 0	Len=0
16	Cl	Serv	4485 > 20	[SYN, ACK]	Seq= 0	Ack= 1	Len=0
17	Serv	Cl	20 > 4485	[ACK]	Seq= 1	Ack= 1	Len=0
18	Serv	Cl	21 > 4484	[PSH, ACK]	Seq= 221	Ack= 68	Len= 29
19	Cl	Serv	4485 > 20	[ACK]	Seq= 1	Ack= 1	Len=1460
20	Cl	Serv	4485 > 20	[ACK]	Seq= 1461	Ack= 1	Len=1460
21	Serv	Cl	20 > 4485	[ACK]	Seq= 1	Ack= 2921	Len=0
22	Cl	Serv	4485 > 20	[FIN, PSH, ACK]	Seq= 2921	Ack= 1	Len=635
23	Serv	Cl	20 > 4485	[ACK]	Seq= 1	Ack= 3557	Len=0
24	Serv	Cl	20 > 4485	[FIN, ACK]	Seq= 1	Ack= 3557	Len=0
25	Cl	Serv	4485 > 20	[ACK]	Seq= 3557	Ack= 2	Len= 0
26	Cl	Serv	4484 > 21	[ACK]	Seq= 68	Ack= 250	Len= 0
27	Serv	Cl	21 > 4484	[PSH, ACK]	Seq= 250	Ack= 68	Len=23
28	Cl	Serv	4484 > 21	[ACK]	Seq= 68	Ack= 273	Len=0
29	Cl	Serv	4484 > 21	[PSH, ACK]	Seq= 68	Ack= 273	Len=6
30	Serv	Cl	21 > 4484	[PSH, ACK]	Seq= 273	Ack= 74	Len=13
31	Serv	Cl	21 > 4484	[FIN, ACK]	Seq= 286	Ack= 74	Len=0
32	Cl	Serv	4484 > 21	[ACK]	Seq= 74	Ack=287	Len=0
33	Cl	Serv	4484 > 21	[FIN, ACK]	Seq= 74	Ack=287	Len=0
34	Serv	Cl	21 > 4484	[ACK]	Seq=	Ack=	Len=

- 1.A ¿Podemos saber en qué máquina (Cliente o Servidor) se ha realizado la captura? ¿Por qué?
- 1.B ¿Cuántos establecimientos y liberaciones de conexión TCP se observan en la captura? Indicar los números de línea.
- 1.C ¿Por qué en los envíos de datos de la conexión con el puerto 21 hay tantos segmentos con el bit de PUSH a 1? ¿Qué pueden significar?
- 1.D A la vista del volcado, ¿cuál es el valor de la ventana de congestión inicial que usa el cliente en su conexión con el servidor de puerto 20?
- 1.E Dar la secuencia de envío de datos si el fichero que el cliente sube al servidor ocupa 6 segmentos de 1460 octetos y la ventana de congestión inicial es uno (la habitual en el algoritmo de SS/CA). Indicar el tamaño de la ventana real cada vez que cambie. Suponer que los ACKs se envían en cuanto es posible.
- 1.F Para la secuencia anterior, ¿llegaríamos a entrar en la fase de CA si se pierde una sola vez el cuarto segmento?

### Solució:

#### 1.A

No, pues no disponemos de ninguna referencia (tiempos, duplicados o direcciones privadas).

#### 1.B

2 de cada:

	Servidor 21	Servidor 20
Establecimiento	Líneas 1-3	Líneas 15-17
Liberación	Líneas 31-34	Líneas 22-25

#### 1.C

Duració: 1,5 hores. Respondre els problemes en fulls separats. Raonar les respostes

La connexió con el puerto 21 es para control, por lo que los datos son pocos y hemos de enviarlos enseguida.

### 1.D

La ventana de congestión inicial es igual a 2 MSS, pues es después de enviar 2 MSS cuando espera un ACK; es decir, no ha esperado un ACK antes de enviar el segundo segmento.

### 1.E

cwnd= 1 MSS

Segmento 1	Cl > Serv	Seq=1	(en los envíos de datos siempre será Ack=1 y Len=1460).
	Serv > Cl	Ack=1461	(en los ACKs siempre será Seq=1 y Len=0).

cwnd = 2 MSS

Segmento 2	Cl > Serv	Seq=1461
Segmento 3	Cl > Serv	Seq=2921
	Serv > Cl	Ack=4381

cwnd = 4 MSS

Segmento 4	Cl > Serv	Seq=4381
Segmento 5	Cl > Serv	Seq=5841
Segmento 6	Cl > Serv	Seq=7301
	Serv > Cl	Ack=8761

cwnd = 7 MSS

### 1.F

Si se pierde el cuarto segmento, la secuencia anterior cambia:

...

cwnd = 4 MSS

Segmento 4	Cl > Serv	Seq=4381	(se pierde)
Segmento 5	Cl > Serv	Seq=5841	
Segmento 6	Cl > Serv	Seq=7301	

Salta el temporizador (RTO) y retransmite el segmento 4:

Segmento 4	Cl > Serv	Seq=4381
------------	-----------	----------

cwnd = 1 MSS (se reinicia la ventana de congestión)

Como ya había recibido los segmentos 5 y 6, envía un ACK reclamando el séptimo que no existe:

Serv > Cl	Ack=8761
-----------	----------

La ventana de congestión se incrementa en uno:

cwnd = 2 MSS

y en este momento debería entrar en la fase de CA, porque el umbral es la mitad de donde cayó, es decir 2 MSS, pero no entra porque ya ha acabado.

### Pregunta 2. (5 punts)

El client de la figura està connectat al seu ISP a través de la xarxa telefònica amb un mòdem intern a 56kbps. Suposa que el client es descarrega una pàgina web de 6000 bytes d'un servidor d'Internet. Suposa també: Els retards en Internet són negligibles en comparació als temps de transmissió en l'enllaç de 56 kbps. La MTU és de 1500 bytes; es fa servir l'opció *timestamp* (aquesta opció ocupa 12 bytes). Comenta les suposicions que facis i inventa't les dades que puguin faltar.



- 2.A Digueu quin serà el MSS, quants segments de dades enviarà el servidor i quants bytes de dades portarà cada un d'aquests segments. Quina mida (en bytes) tindran els datagrames que porten les confirmacions?
- 2.B Calcula quin és el temps de transmissió en l'enllaç de 56 kbps d'un segment de dades de mida màxima enviat per el servidor, i del corresponent ack enviat per el client. Dibuixa un diagrama de temps amb 3 eixos: client, ISP i servidor, que mostri la transmissió del segment i de l'ack.
- 2.C Suposa que l'únic retard rellevant és la transmissió dels segments de dades que envia el servidor en l'enllaç de 56 kbps. Dibuixa un diagrama de temps amb 3 eixos: client, ISP i servidor, on es vegin tots els segments que s'envien durant la connexió, i que reflecteixi la suposició anterior. Indica clarament els segments que s'envien: marca amb una G el segment amb el GET de HTTP que envia el client, amb S1, S2... els segments de dades que envia el servidor i A1, A2... les corresponents confirmacions. Indica en el diagrama quins segments portaran el flags de SYN i FIN activats. Indica sobre el diagrama el valor de la finestra de congestió (en segments), en cada instant en que canviï de valor.

Segon control de Xarxes de Computadors (XC), Grau en Enginyeria Informàtica	12/12/2011	Tardor 2011
---	------------	-------------

Duració: 1,5 hores. Respondre els problemes en fulls separats. Raonar les respostes

- 2.D** Suposa que l'únic retard rellevant és la transmissió dels segments de dades que envia el servidor en l'enllaç de 56 kbps (suposa que els temps de transmissió i retards de tots els altres segments és aproximadament igual a 0). Amb l'ajut del diagrama de l'apartat 2.C, estima el temps que durarà la descàrrega de la pàgina web i la velocitat efectiva assolida (punt de vista de l'aplicació).
- 2.E** Amb les aproximacions anteriors i l'ajut del diagrama de l'apartat 2.C, estima el RTT que mesurarà la connexió TCP en el socket del servidor quan arriba cadascuna de les confirmacions dels segments de dades que envia.

### Solució:

#### 1.A

El MSS sempre és la MTU - 40 (mida de les capçaleres IP i TCP sense opcions), per tant:

$MSS = 1500 - 40 = 1460$  bytes

Els segments de dades portaran  $1460 - 12 = 1448$  bytes, com a màxim (12 bytes del timestamp). Per tant, el Serv enviarà:  $\text{ceiling}(6000/1448) = \text{ceiling}(4,14) = 5$  segments. Els primers 4 segments portaran 1448 bytes, i l'últim  $6000 - 4 \cdot 1448 = 208$  bytes. Els 4 primers datagrames seran de 1500 bytes i l'últim de  $208 + 52 = 260$  bytes.

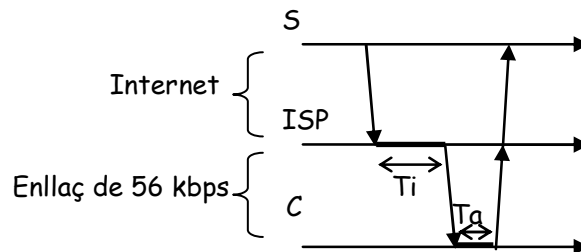
Els datagrams amb les confirmacions tindran  $40 + 12 = 52$  bytes (capçaleres IP+TCP, TCP amb 12 bytes de l'opció timestamp).

#### 1.B

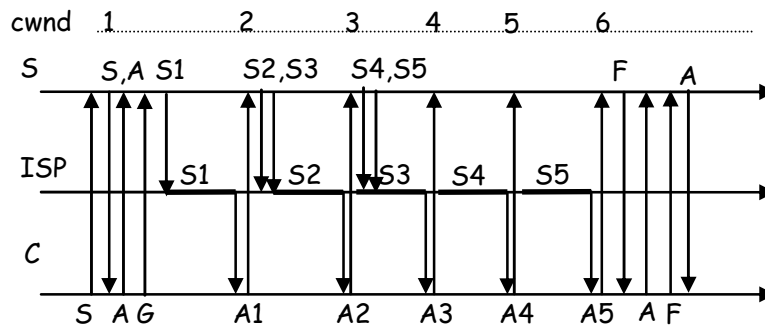
Temps de transmissió d'un segment de mida màxima:  $T_i = 1500 \cdot 8 \text{ bits} / 56 \text{ kbps} = 214,3 \text{ ms}$ .

Temps de transmissió d'una confirmació:  $T_a = 52 \cdot 8 \text{ bits} / 56 \text{ kbps} = 7,4 \text{ ms}$ .

No temin en compte les capçaleres que es puguin afegir a nivel d'enllaç.



#### 1.C



#### 1.D

Només considerem les transmissions dels segments de dades:

S'envien 5 segments amb 6000 bytes de dades i capçaleres de 52 bytes. És a dir, l'eficiència serà de  $6000 / (6000 + 5 \cdot 52) \approx 0,96$ .

Per tant, la velocitat efectiva serà de  $56 \cdot 0,96 = 53,76 \text{ kbps}$ .

El temps de transmissió serà de  $(6000 + 5 \cdot 52) \cdot 8 \text{ bits} / 56 \text{ kbps} \approx 0,89 \text{ segons}$

#### 1.E

Del diagrama deduïm:

Per S1:  $RTT \approx \text{temps de transmissió d'un segment de mida màxima} = 214,3 \text{ ms}$

Per S2:  $RTT \approx \text{temps de transmissió d'un segment de mida màxima} = 214,3 \text{ ms}$

Per S3:  $RTT \approx 2 \cdot \text{temps de transmissió d'un segment de mida màxima} = 428,6 \text{ ms}$

Per S4:  $RTT \approx 2 \cdot \text{temps de transmissió d'un segment de mida màxima} = 428,6 \text{ ms}$

Per S5:  $RTT \approx 2 \cdot \text{temps de transmissió d'un segment de mida màxima} + \text{temps de transmissió de l'últim segment (de 260 bytes)} = 428,6 \text{ ms} + 260 \cdot 8 \text{ bits} / 56 \text{ kbps} \approx 465 \text{ ms}$