

## Solució del control

Segon control de Xarxes de Computadors (XC), Grau en Enginyeria Informàtica		9/5/2016	Primavera 2016
NOM:	COGNOMS	GRUP	DNI

Duració: 1h15m. El test es recollirà en 25 minuts. Respondre el problemes en el mateix enunciat.

**Test.** (4 punts) Totes les preguntes són multiresposta: Valen la meitat si hi ha un error, 0 si més.

1. Marca les respostes que siguin correctes sobre el protocol TCP:

- ☐ El camp de checksum es calcula a partir només de dades de la pròpia capçalera TCP.
- ☐ El camp ACK Number és opcional.
- ☒ La validesa del camp ACK Number és opcional i depèn del Flag ACK
- ☐ La capçalera transporta el valor de la finestra de congestió

2. Per a calcular la finestra òptima d'una comunicació entre dos nodes units per una línia de transmissió on només hi ha una connexió TCP cal tenir en compte:

- ☐ Només la velocitat de transmissió de la línia
- ☐ Només la velocitat de transmissió de la línia i la mida del buffer de recepció
- ☒ Només la velocitat de transmissió de la línia, la mida del buffer de recepció i el RTT de la comunicació
- ☐ Només la velocitat de transmissió de la línia, la mida del buffer de recepció, el RTT de la comunicació i la probabilitat d'error de la línia

3. Quines afirmacions són correctes sobre el procés d'establiment de la connexió?

- ☐ Si el tercer dels segments de l'establiment es perd, un dels extrems de la connexió es queda en estat LISTEN
- ☒ El número de seqüència inicial (ISN) es reconeix explícitament en el procés d'establiment, fent incrementar en 1 els números de seqüència esperats per l'altre extrem
- ☒ Durant aquesta fase es negocia el MSS
- ☐ Durant aquesta fase es negocia la finestra de congestió inicial

4. Per a que el flux d'usuari d'un procés d'aplicació que envia dades TCP quedi bloquejat perquè el nivell TCP no pot acceptar més dades per a ser enviades, en un escenari en què la velocitat de tots els enllaços entre transmissor i receptor és la mateixa i no hi ha cap altra comunicació en la mateixa línia, cal que es donguin les següents condicions:

- ☐ L'aplicació ha d'utilitzar TCP en el mode bloquejant d'acord amb els flags usats durant l'inici de la connexió
- ☐ El valor de la finestra de congestió ha de ser major que el de la finestra advertida
- ☒ El buffer de transmissió de TCP en la banda transmissora ha d'estar ple
- ☒ El buffer de recepció de TCP en la banda receptora ha d'estar ple

5. Quines de les següents situacions són possibles a TCP?

- ☒ Després de d'enviar tota la finestra (cwnd) en una ràfega de segments, tots els segments són reconeguts i la finestra cwnd es duplica
- ☒ Després de d'enviar tota la finestra (cwnd) en una ràfega de segments, tots els segments són reconeguts i la finestra cwnd augmenta aproximadament en 1 segment
- ☒ Després de d'enviar tota la finestra (cwnd=4) en una ràfega de segments, tots quatre segments són reconeguts i la finestra cwnd augmenta en més de 2 segments, però menys que 4 (aproximadament 2.5)
- ☐ Després de d'enviar tota la finestra (cwnd) en una ràfega de segments, tots els segments són reconeguts i la finestra cwnd es quadruplica

6. Quines de les següents situacions són possibles a UDP?

- ☒ S'envia un datagrama UDP, i després de que es perdi no es torna a enviar més
- ☒ S'envia un datagrama UDP, i després de que es perdi l'aplicació el torna a enviar
- ☐ En el destí, els datagrames es reordenen a nivell UDP per a poder reconstruir la seqüència de datagrames original
- ☒ En el destí, es verifica el checksum dels datagrames i basat en això es descarten aquells que estan identificats com a corromputs

7. Quin RTT cal per a que la transmissió de segments des d'un equip a un altre no s'aturi per culpa de la finestra de TCP? Suposa dos nodes units per una línia de transmissió on només hi ha una connexió TCP, la cwnd té un valor prou gran com per a ser irrellevant, la velocitat de la línia és de 1000 bytes per segon i el valor de awnd és 400 bytes.

- ☒  $0 \leq RTT \leq 0.4s$
- ☐  $0.4s < RTT \leq 0.6s$
- ☐  $0.6s < RTT \leq 1s$
- ☐  $1s < RTT \leq 400s$

## Solució del control

Segon control de Xarxes de Computadors (XC), Grau en Enginyeria Informàtica		9/5/2016	Primavera 2016
NAME:	SURNAME	GROUP	DNI

Duration: 1h15m. The quiz will be collected in 25 minutes. Answer in the same questions sheet.

**Test.** (4 points) All questions are multiple choice: Count as half if there is one error, 0 if more.

1. Mark the correct answers about the TCP protocol:
  - ☐ The checksum field is calculated only from data coming from the TCP header itself.
  - ☐ The ACK field is optional
  - ☒ The validity of the ACK field is optional, and defined by the ACK flag
  - ☐ The TCP header carries the value of the congestion window
  
2. To calculate the optimal window for a communication between two nodes connected through a transmission line in which there is only one active TCP connection, the following parameters have to be taken into account:
  - ☐ Only the bitrate of the line
  - ☐ Only the bitrate of the line and the size of the reception buffer
  - ☒ Only the bitrate of the line, the size of the reception buffer and the RTT of the communication
  - ☐ Only the bitrate of the line, the size of the reception buffer, the RTT of the communication and the probability of error in the line
  
3. Which of the following statements referring to the connection establishment process are true?
  - ☐ If the third one of the connection establishment segments is lost, one of the ends of the connection will stay in state LISTEN
  - ☒ The Initial Sequence Number (ISN) is explicitly acknowledged in the establishment process, causing an increment of 1 in the sequence numbers expected by the other end of the communication.
  - ☒ During this phase, the MSS value is negotiated
  - ☐ During this phase, the initial value of the congestion window is negotiated.
  
4. To get one user application blocked in the process of writing data through a TCP communication because no more data can be accepted by the TCP layer, considering an scenario in which the bitrate of all links found between the transmitter and the receiver and in which no other active communications take place in the link, it is required that the following conditions hold:
  - ☐ The application uses TCP in blocking mode according to the flags used in the connection establishment
  - ☐ The value of the congestion window must be higher than the value of the advertised window
  - ☒ The TCP transmission buffer in the side of the data sender must be full
  - ☒ The TCP reception buffer in the side of the data receiver must be full
  
5. Which ones of the following situations are possible in TCP?
  - ☒ After sending a full window (cwnd) of segments, all the segments of the window are acknowledged and the size of the window doubles
  - ☒ After sending a full window (cwnd) of segments, all the segments of the window are acknowledged and the size of the window approximately increases in 1 segment
  - ☒ After sending a full window (cwnd=4) of segments, all 4 segments of the window are acknowledged and the size of the window increases in more than 2 segments but less than 4 (approximately 2.5)
  - ☐ After sending a full window (cwnd) of segments, all the segments of the window are acknowledged and the size of the window quadruples
  
6. Which ones of the following situations are possible in UDP?
  - ☒ One UDP datagram is sent, and after being lost it is not sent again
  - ☒ One UDP datagram is sent, and after being lost the application sends it again
  - ☐ At the destination, the datagrams are ordered at the UDP level to be able to rebuild the original sequence of datagrams
  - ☒ At the destination, the checksum of all datagrams is verified and based on this datagrams that are identified as corrupted are discarded
  
7. What value of RTT is needed in a transmission of segments between two systems to avoid any of them to get blocked over a transmission? You can assume that no other active connections use the same line that interconnects the two systems, that the value of cwnd is large enough to be negligible, that the bitrate of the line is 1000bytes per second, and that the value of awnd is 400 bytes.
  - ☒  $0 \leq \text{RTT} \leq 0.4\text{s}$
  - ☐  $0.4\text{s} < \text{RTT} \leq 0.6\text{s}$
  - ☐  $0.6\text{s} < \text{RTT} \leq 1\text{s}$
  - ☐  $1\text{s} < \text{RTT} \leq 400\text{s}$

## Solució del control

Segon control de Xarxes de Computadors (XC), Grau en Enginyeria Informàtica		9/5/2016	Primavera 2016
NOM:	COGNOMS	GRUP	DNI

Duració: 1h15m. El test es recollirà en 25 minuts. Respondre el problemes en el mateix enunciat.

### Pregunta 1. (6 puntos)

Las siguientes 29 líneas presentan información sobre parte de la captura de un intercambio de segmentos TCP entre una máquina Cliente (que llamaremos C) y una máquina Servidor (que llamaremos S).

Las columnas representan: 1) Número de línea del intercambio, 2) Identificador de máquina y port que envía, 3) Identificador de máquina y port que recibe, 4) Flags activos (S, P, F, .), 5) Indicación de si es un ack sin datos, 6) Número de secuencia : número de secuencia del siguiente segmento (tamaño de datos), 7) Número de ACK, 8) Tamaño de la ventana anunciada.

```

1.  10.1.0.3.1059 > 10.2.0.1.80: . ack 1 win 32120
2.  10.1.0.3.1059 > 10.2.0.1.80: P 1:93(92) ack 1 win 32120
3.  10.2.0.1.80 > 10.1.0.3.1059: P 1:213(212) ack 93 win 32120
4.  10.1.0.3.1059 > 10.2.0.1.80: . ack 213 win 23168
5.  10.2.0.1.80 > 10.1.0.3.1059: . 213:1661(1448) ack 93 win 32120
6.  10.1.0.3.1059 > 10.2.0.1.80: . ack 1661 win 32120
7.  .....
8.  10.1.0.3.1059 > 10.2.0.1.80: . ack 26277 win 23168
9.  10.2.0.1.80 > 10.1.0.3.1059: . 26277:27725(1448) ack 93 win 32120
10. 10.2.0.1.80 > 10.1.0.3.1059: . 27725:29173(1448) ack 93 win 32120
11. 10.2.0.1.80 > 10.1.0.3.1059: . 30621:32069(1448) ack 93 win 32120
12. 10.2.0.1.80 > 10.1.0.3.1059: . 32069:33517(1448) ack 93 win 32120
13. 10.1.0.3.1059 > 10.2.0.1.80: . ack 29173 win 23168
14. 10.2.0.1.80 > 10.1.0.3.1059: . 33517:34965(1448) ack 93 win 32120
15. 10.2.0.1.80 > 10.1.0.3.1059: . 34965:36413(1448) ack 93 win 32120
16. 10.2.0.1.80 > 10.1.0.3.1059: . 36413:37861(1448) ack 93 win 32120
17. 10.1.0.3.1059 > 10.2.0.1.80: . ack 29173 win 23168
18. 10.1.0.3.1059 > 10.2.0.1.80: . ack 29173 win 23168
19. 10.1.0.3.1059 > 10.2.0.1.80: . ack 29173 win 23168
20. 10.2.0.1.80 > 10.1.0.3.1059: . 29173:30621(1448) ack 93 win 32120
21. 10.1.0.3.1059 > 10.2.0.1.80: . ack 37861 win 23168
22. 10.2.0.1.80 > 10.1.0.3.1059: . 37861:39309(1448) ack 93 win 32120
23. .....
24. 10.2.0.1.80 > 10.1.0.3.1059: FP 499773:500213(440) ack 93 win 32120
25. 10.1.0.3.1059 > 10.2.0.1.80: . ack 493981 win 23168
26. 10.2.0.1.80 > 10.1.0.3.1059: . 493981:495429(1448) ack 93 win 32120
27. 10.1.0.3.1059 > 10.2.0.1.80: . ack 500214 win 23168
28. 10.1.0.3.1059 > 10.2.0.1.80: F 93:93(0) ack 500214 win 23168
29. 10.2.0.1.80 > 10.1.0.3.1059: . ack 94 win 32120

```

**1.A** (0.5 puntos) ¿Cuáles son las direcciones IP de C y S?

**10.1.0.3 para C (port 1059, mayor que 1024), y 10.2.0.1 para S (port 80, well-known).**

**1.B** (0.5 puntos) Justificar a partir de la captura quién inicia la conexión.

**C, porque en la línea 1 vemos el ACK final de la fase de establecimiento de conexión.**

**1.C** (0.5 puntos) ¿Por qué en la línea 4 no hay número de secuencia?

**Porque sólo es un ACK. El cliente no parece tener más datos para enviar.**

**1.D** (0.5 puntos) ¿Cuántos segmentos se intercambian en la desconexión? ¿Cuáles son las líneas correspondientes?

**4 segmentos: Líneas 24, 27, 28 y 29. Es un 3WH modificado, pues son 4 intercambios (no 3), ya que la respuesta al primer F se descompone en un ACK (línea 27) y el F en sí (línea 28).**

**1.E** (0.5 puntos) Si no ha habido pérdidas, ¿cuántos segmentos (con MSS bytes) parece que ha enviado S durante la línea 7?

**(26.277 - 1.661) / 1448 = 17.**

**1.F** (0,5 puntos) Antes del envío de la línea 9, ¿cuánto vale como mínimo la ventana de congestión?

$6 \text{ segmentos} * 1448 \text{ bytes/segmento} = 8.688 \text{ bytes}$ . Menos que la ventana anunciada = 23168 bytes = 16 segmentos de MSS bytes.

Sabemos que son 6 segmentos al menos, pues hasta la retransmisión de la línea 20, S ha enviado 8 segmentos, pero ha recibido ACKs por 2 segmentos (en la línea 13).

Hay otras maneras de calcularlo.

**1.G** (0,5 puntos) ¿En qué lado se ha hecho la captura? ¿Cómo lo sabemos?

En C, pues vemos los ACKs repetidos desde el cliente (ack 29173), pero no los segmentos de datos repetidos por el servidor.

**1.H** (1 punto) Dibujar (eje vertical para la ventana y eje horizontal para el tiempo) la evolución de la ventana de transmisión desde el momento del envío de la línea 9 hasta la línea 22, asumiendo que en la línea 9 la ventana de transmisión es igual a la ventana anunciada. **Indicar claramente en qué momento se cambia entre fases SS y CA, y el valor del umbral.**  
NOTA: No preocuparse por la escala del eje de tiempo, sino usar las líneas de la captura.

La ventana de transmisión vale siempre 16 MSS (ventana anunciada) hasta la retransmisión en la línea 20, que baja a una ventana de congestión de 1 MSS (en ese momento reiniciamos la fase SS). Antes podíamos estar en la SS inicial o en CA en función de si ha habido pérdidas o no en 7.

En la línea 21 sube a 2 MSS al llegar un ACK. El umbral pasa a valer  $16/2 = 8$  MSS. La ventana de transmisión no subirá más hasta llegar otro ACK.

**1.I** (0,75 puntos) Suponiendo que no hay pérdidas (aunque puedan aparecer en la captura) y que durante toda la conexión la ventana de TCP es igual a la ventana anunciada (es decir, el tiempo que se tarda en llegar a la ventana anunciada es despreciable), calcular aproximadamente la velocidad eficaz y cuánto durará la transmisión. Suponer RTT = 10 ms.

Hemos enviado 500.212 octetos, de los cuales 212 se han enviado inicialmente, por lo que parece que se ha enviado un fichero de 500.000 bytes. De ellos, se han enviado 440 en el último segmento con datos. Por tanto, se han enviado  $500.000 - 440$  bytes en segmentos de 1.448 octetos, lo que da 345 segmentos.

Sin embargo, con la aproximación indicada en el enunciado, podemos tener en cuenta simplemente que la ventana anunciada es de 16 MSS y eso es lo que enviaremos en cada RTT.

Por tanto, la velocidad aproximada será  $v = V_a / RTT = 23168 * 8 \text{ bits} / 0,01 \text{ s} = 18.534.400 \text{ bps} = 18,5 \text{ Mbps}$ .

La transmisión de los 500.000 bytes durará:  $t = 500.000 * 8 / 18.534.400 = 0,2158 \text{ s}$ .

**1.J** (0,75 puntos) Suponiendo que no hay pérdidas (aunque puedan aparecer en la captura) y que la ventana de congestión inicial es igual a 1 segmento con MSS bytes, ¿cuántos RTT transcurren hasta que la ventana de TCP alcanza el valor de la ventana anunciada? Volver a calcular la velocidad eficaz y cuánto durará la transmisión sin despreciar el tiempo en llegar a la ventana anunciada.

Para llegar a la ventana anunciada necesitamos 4 RTT (habiendo enviado  $1+2+4+8=15$  segmentos).

A la velocidad de la ventana anunciada, el número de segmentos de MSS bytes que nos queda por enviar es  $345 - 15 = 330$  más el último más pequeño. Estos 330 segmentos se enviarían en  $330/16=20,6 \rightarrow 21$  RTT. Los 4 RTT para que la ventana de congestión llegue a la ventana anunciada son muchos para ser despreciados respecto a los 21 RTT (el último de 440 bytes cabe en el último RTT) necesarios para acabar el envío. Con este cálculo más preciso vemos que tardamos  $21+4=25$  RTTs.

El tiempo total es por tanto:  $25 \text{ RTT} = 0,25 \text{ s}$  y la velocidad:  $(500.000 * 8) \text{ bits} / 25 * 0,01 \text{ s} = 16 \text{ Mbps}$ .