



Universidad rey Juan Carlos

Escuela Técnica Superior de Ingeniería de Telecomunicación

Sistemas Telemáticos

Practica 2

TCP

Iván Moreno Martín

2º GITT

1. Estados de una conexión TCP

1.1. Establecimiento de la conexión

a) Indica en qué estado se encuentra el servidor antes de arrancar el cliente.

Antes de arrancar el cliente el servidor se encuentra en modo escucha aparece como LISTEN

b) Indica qué estados atraviesa el servidor hasta que el cliente se encuentra conectado.

Una vez conectamos pc1 como cliente, los estados de pc2 son:

- 1.Listen
- 2.Syn_Recv
3. Established

c) Apunta el valor de la ventana anunciada por el servidor.

La ventana anunciada enviada por el servidor aparece en la captura de la siguiente forma:

12.0.0.2.7777 > 11.0.0.2.54590 [...] ack 1625173416 win 36 [...]

Es decir, la ventana anunciada por el servidor es de 36.

1.2. Intercambio de datos

a) Explica el valor de Recv-Q teniendo en cuenta que has pulsado más caracteres para enviar en el cliente.

El valor que aparece en Recv-Q es de 36 (tamaño de ventana anunciada). Esto está acumulado en el buffer del servidor.

b) Explica los segmentos enviados por el cliente que muestra la captura de tráfico de r1.

Como el servidor está suspendido cuando le llegan mensajes de pc1 la ventana anunciada es 0, con ello dice a pc1 que no puede enviar más datos. Cuando Arranquemos pc2 el tamaño de ventana volver a ser de 36, es decir restablece su valor.

1.3. Finalización de la conexión

a) ¿Por qué se mantiene tanto tiempo en este estado?

Se mantiene ese tanto tiempo en ese estado porque está a la espera de los ACK's correspondientes al fin de conexión para asegurarse que no se han perdido paquetes.

b) Explica por qué ya no aparecen conexiones TCP en pc2.

Ya no aparecen porque ha recibido los paquetes con fin de ack, con lo cual el estado de pc2 vuelve a estar a LISTEN, a la espera de una nueva conexión.

2. Slow Start en el inicio de la conexión

a) Indica el valor del flightsize en los siguientes instantes: 1'5s, 2'5s, 3'5s, 4'5s, 5s, 6s.

Para saber el valor de flightsize miramos la gráfica TCP trace en cada uno de los instantes indicados, por tanto :

flightsize(1,5s) = 3
cwnd(1,5s)=3

flightsize(2,5s) = 6
cwnd(2,5s)=6

flightsize(3,5s)=7
cwnd= (3,5s)=12

flightsize(4,5s)=11
cwnd(4,5s)= 19

flightsize(5s)= 11
cwnd(5s)= 19

flightsize(6s)=17
cwnd(6s)= 30

nota: para obtener cwnd lo podemos calcular como por ejemplo en el instante 4,5s como cwnd anterior (=12)+ nº de ACK'S (=7) = 19

b) ¿En qué instantes de tiempo de esta captura es menor la ventana de control de flujo que la ventana de control de congestión?

c) ¿Por qué se envían inicialmente 3 segmentos con datos y no 1?

Porque tenemos como límite 3 MSS, es decir con esto podemos enviar 1 a 3 MSS.

d) ¿Cuántos segmentos con nuevos datos se podrían enviar en el instante 3'5s?

En el instante 3,5 vemos que tenemos un tamaño de ventana $cwnd = 12$, pero sólo mandamos 7, por tanto podríamos mandar $12 - 7 = 5$

e) ¿Cuántos segmentos con nuevos datos se podrían enviar en el instante 4'5s?

En el caso del instante 4,5s tenemos $cwnd = 19$ y enviamos 11 por tanto aun podríamos mandar 8 más.

3. Control de Congestión tras Timeout

a) ¿Por qué la primera vez se envían 2 segmentos con datos alrededor del instante 1s?

Porque en este caso estamos en Slow Start y como máximo estaremos en 3 MSS.

b) Entre el instante 3s y 3'5s se envían 6 segmentos. ¿Podrían haberse enviado más segmentos en ese instante? Razona la respuesta.

En este caso el tamaño de $cwnd = 10$ por tanto $10 - 6 = 4$, por lo que se podrían haber mandado hasta 4 segmentos más.

2. Localiza la retransmisión que se produce alrededor del instante 7'5s. Estudia el comportamiento de TCP tras el timeout. Analiza tanto la gráfica como cada uno de los segmentos. Responde a las siguientes preguntas:

a) ¿Qué número de secuencia tiene el segmento retransmitido?

Como podemos ver en la siguiente captura de pantalla el número de secuencia retransmitido es el 8689

Time	Source	Destination	Protocol	Length
26 7.929996	11.0.0.2	12.0.0.2	TCP	151
27 9.464425	12.0.0.2	11.0.0.2	TCP	60
28 9.464505	11.0.0.2	12.0.0.2	TCP	151
29 9.464539	11.0.0.2	12.0.0.2	TCP	151
30 10.485977	12.0.0.2	11.0.0.2	TCP	60
33 10.486000	12.0.0.2	11.0.0.2	TCP	60
31 10.486077	11.0.0.2	12.0.0.2	TCP	151

Destination Port: 7000
[Stream index: 0]
[TCP Segment Len: 1448]
Sequence number: 8689 (relative sequence number)
[Next sequence number: 10137 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Header Length: 32 bytes

b) ¿Qué valor tiene el congestion-threshold en el instante 8s?

El valor de threshold al tener timeout lo calculamos como el maximo entre $\text{flagsize} / 2$ y 2 MSS , es decir $\text{threshold} = \max(\text{flagsize}/2, 2\text{MSS})$

Calculamos $\text{flagsize}/2$ y como resultado obtenemos $6/2 = 3 > 2\text{MSS}$ por tanto $\text{threshold} = 3$

c) ¿Qué tamaño tiene la ventana de congestión en el instante 8s?

Como en el instante anterior hemos activado timeout y estamos en Slow Start entonces el tamaño de $\text{cwnd} = 1$

d) ¿Por qué se envían 2 segmentos alrededor del instante 9s?

En el instante anterior (8s) el tamaño de $\text{cwnd} = 1$, entonces, en la siguiente transmisión de segmentos y por estar en timeout, le sumamos uno a cwnd con lo que ahora tenemos $\text{cwnd} = 2$ con lo que sólo mandamos dos segmentos.

e) ¿Podría enviarse algún segmento más en el instante 9s? ¿Por qué?

En este caso en concreto no porque estamos ocupando el tamaño máximo de la ventana de control de congestión

f) ¿En qué modo de control de congestión se haya la conexión en el instante 5s? ¿Por qué?

El modo en el que nos encontramos en el instante 5s es Slow Start, como vemos el tamaño de ventana de control de congestión

g) ¿En qué modo de control de congestión se haya la conexión en el instante 10'5s? ¿Por qué?

El modo en el que nos encontramos en el instante 10,5s es de Additive Increase, como vemos se está transmitiendo cerca de la capacidad de la red, es decir cuando $\text{cwnd} = \text{congestionThreshold}$.

4. Fast Retransmit / Fast Recovery

a) ¿Cuántas retransmisiones debidas a timeout se observan en la traza? Identifica en qué instante se producen.

Observando las trazas capturadas en fichero vemos que se produce un timeout en el instante 10,089s

b) ¿Cuántas retransmisiones debidas a Fast Retransmit se observan en la traza? Identifica en qué instante se producen.

Se producen 3 retransmisiones debidas a Fast Retransmit y se producen en los instantes 5,04s; 16,1451s; 21,1949s.

c) Observa la primera ocasión en la que se produce Fast Retransmit. Indica cuántos paquetes se envían después de haber recibido el primer ACK nuevo. Explica la razón por la que se puede enviar ese número de paquetes. Fast Retransmit.

Se envían 6 paquetes esto se debe a que lo permite tanto cwnd como la ventana anunciada.

d) Poco después de producirse la primera retransmisión debida a Fast Retransmit se envían nuevos segmentos de datos antes de recibir ningún ACK. ¿Cuántos nuevos segmentos de datos se envían? ¿Por qué se pueden enviar en ese instante? ¿Se podría enviar alguno más?

5. Control de Congestión y sondas de ventana

a) Localiza en la traza los anuncios de ventana de tamaño 0 enviados por el servidor al cliente. ¿Qué segmentos son los que transportan estos anuncios?

Los segmentos son el 80 y 82 en el que el servidor envía que el tamaño de ventana es 0

b) Observa las sondas de ventana que envía el cliente en ese periodo. ¿Qué segmentos son los que transportan las sondas de ventana?

Los segmentos 81 y 83 en los que se mandan Keep-Alive

c) Estudia el comportamiento de TCP entre los segmentos 43 y 87. Responde a las siguientes preguntas:

1) Indica cuál es el número de byte más alto que puede enviarse tras recibirse los siguientes segmentos: 41, 43, 51, 79.x

Ultimo byte que puede enviarse = ultimo byte asentido + tamaño de ventana anunciada

Segmento 41 → $UBPE = (222177 - 1) + 44888 = 67064$

Segmento 43 → $UBPE = (30865 - 1) + 36200 = 67064$

Segmento 51 → $UBPE = (32313 - 1) + 34752 = 67064$

Segmento 79.x → $UBPE = (67065 - 1) + 4344 = 71408$

2) ¿Cuántos bytes nuevos pueden enviarse tras recibirse el segmento 80?

Se puede ver en la captura que la ventana está llena. Con lo que nos salta un valor de ventana anunciada es 0. Por lo que no se pueden enviar más bytes de datos.

3) ¿Cuántos bytes de datos transportan los segmentos con sondas de ventana?

Los únicos bytes que aparecen son los de cabecera, que en este caso son 32, 20 fijos y 12 de opciones.

4) Tras haberse cerrado la ventana, ¿en qué segmento vuelve a abrirse? ¿Cuál es el tamaño de la nueva ventana anunciada?

Tras haberse cerrado la ventana, ésta se vuelve a abrir en el segmento 84 con un tamaño de ventana anunciada de 2896.

5) Tras recibirse el segmento 84, ¿qué limita al emisor, la ventana de control de flujo o la de congestión?

La ventana que limita al emisor es la ventana de control de flujo.

d) Indica en qué modo de control de congestión se encuentra la conexión TCP en los siguientes puntos:

1) Antes del segmento 43.

Antes del segmento 43 estamos en Slow Start

2) Entre el segmento 43 y 87

Entre estos dos segmentos estamos en Slow Start

3) Después del segmento 87

Después del segmento 87 estamos en Slow Start

e) Indica en qué momentos de la conexión es la ventana de control de flujo la que limita al emisor, y en cuáles es la ventana de control de congestión.

En el caso que la ventana de control de flujo limita la conexión esta entre el segmento 74 al segmento 85. Al no tener ningún timeout y al estar en Slow Start no limita al emisor.

6. Fast Retransmission y SACKs

1. Dado que SACK es una opción que pueden usar ciertas implementaciones de TCP, indica cómo se ponen de acuerdo los extremos de una conexión TCP en que ambos tienen implementada dicha opción y pueden usar SACK. ¿En qué segmentos de la conexión ocurre ese acuerdo?

Estos SACK son una opción de TCP y deberían aparecer al principio de la conexión, en nuestro caso podemos ver que en el segmento 1 aparece dentro de Transmission Control Protocol → Options y dentro de estas opciones aparece :

TCP SACK Permitted option : True

Que quiere decir que SACK está activado. Esto aparece tanto en el segmento 1 como en el 2 de la conexión.

2. Indica cuál es el primer segmento de la captura donde se muestran asentimientos selectivos. ¿Qué números de secuencia se están asintiendo? ¿Qué números de segmento son los que se están asintiendo?

El primer segmento de la captura donde se muestran los asentimientos es el 79, en este caso está asintiendo [52049, 65081], número de secuencia 1 = 52049, número de secuencia 2 = 65081 - 1 = 65080.

Los números de segmentos son para seq1 = 52049 es el 62, y para el seq2 = 65080 es el 73.

3. Explica si el segmento 78 es una retransmisión por timeout o es una retransmisión rápida.

Si miramos la gráfica podemos ver que es una retransmisión por timeout porque tiene un periodo de inactividad de aproximadamente 1S, es decir, hay un largo periodo de inactividad.

4. Explica en qué modo de control de congestión está la máquina 11.0.0.10 justo después de haber enviado el segmento 78. Indica el valor de threshold en ese instante.

Como vemos en la siguiente imagen nos encontramos en modo Fast Retransmit, el valor de threshold = $\text{flagsize} / 2 = 48 / 2 = 24$.

5. Justo antes de enviar el segmento 83, ¿qué segmentos puede suponer la máquina 11.0.0.10 que le faltan a 13.0.0.10?

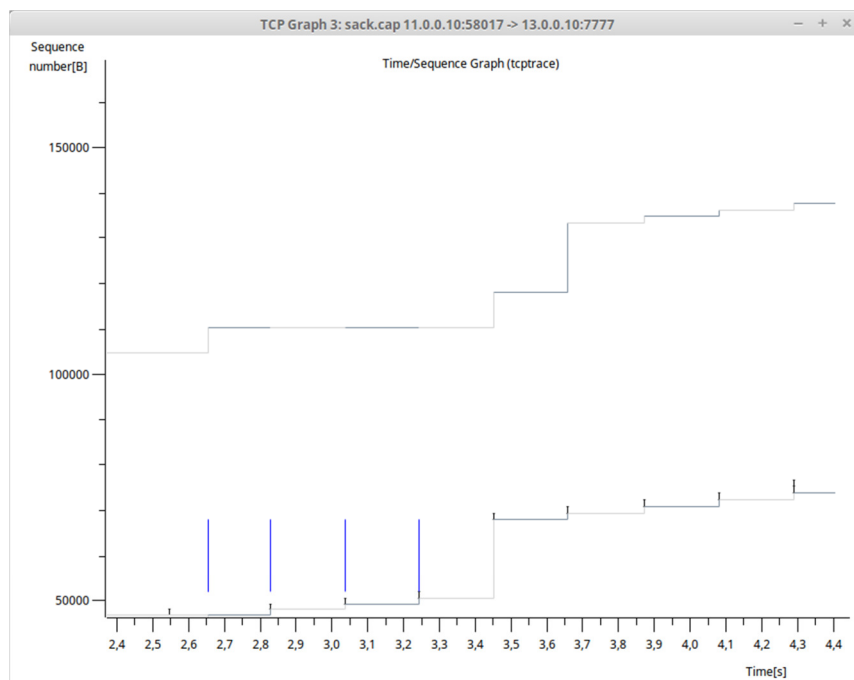
Justo antes de enviar el segmento 83, miramos el segmento 82, donde le indica que le falta el segmento 48241, es decir, ACK= 48241. Por tanto los segmentos anteriores están pendientes de conformación. Pero más específicamente puede suponer que le faltan los segmentos 58, 59 y 60, es decir los números de secuencia que indica el SACK del 52049 hasta el 67976.

6. Si 11.0.0.10 sabe que a 13.0.0.10 le faltan varios segmentos, ¿por qué justo después de enviar el segmento 83 no retransmite todos los que sabe que le faltan?

Porque al producirse timeout en el segundo 2,5s el tamaño de la ventana de congestión vuelve a ser 1 con lo cual sólo se puede mandar un segmento.

7. Explica si el segmento 262 es una retransmisión por timeout o es una retransmisión rápida.

Como podemos ver en la gráfica parece que nos encontramos en Fast Retransmit.



8. Explica en qué modo de control de congestión está la máquina 11.0.0.10 justo después de haber enviado el segmento 262. Indica el valor de threshold en ese instante.

Se puede ver que en la captura estamos en modo Fast Retransmit, el valor de threshold es $\text{flagsize} / 2 = 4$

9. Al recibir el segmento 257, ¿qué segmento/s puede suponer la máquina 11.0.0.10 que le falta/n a 13.0.0.10?

Todos aquellos posteriores al segmento número 245.

10. A recibir el segmento 259, ¿qué segmento/s puede suponer la máquina 11.0.0.10 que le falta/n a 13.0.0.10?

Todos aquellos que sean posteriores al segmento 246.

11. Fíjate en las diferencias que hay entre el segmento 257 y 259. ¿Qué crees que ha provocado que la máquina 13.0.0.10 haya enviado ese asentimiento?

Como vemos esos segmentos aparecen casi al mismo tiempo esto puede deberse a un timeout o que en el modo en el que están (congestion avoidance).

12. A recibir el segmento 277, ¿qué segmento/s puede suponer la máquina 11.0.0.10 que le falta/n a 13.0.0.10?

Alguno posterior a las secuencias [241737, 257665][259113, 260561].

13. Al recibir el segmento 669, ¿qué segmento/s puede suponer la máquina 11.0.0.10 que le falta/n a 13.0.0.10?

La máquina puede suponer que le pueden faltar los segmentos del 617 al 659

14. Explica si el segmento 747 es una retransmisión por timeout o es una retransmisión rápida.

Por lo que vemos en la captura el segmento 747 es una retransmisión por timeout

15. ¿Por qué hay 2 retransmisiones juntas? Segmentos 747 y 749.

Estas retransmisiones se producen porque se han recibido 10ACKs duplicados del segmento 729, por tanto la máquina 11.0.0.10 realiza un Fast Retransmit de ese segmento. En el último ACK vemos que le faltan las secuencias 612121 y 613569 y por eso se retransmiten de manera consecutiva.