

HOJA DE PROBLEMAS 3:

UDP/TCP/Java

- 1) Suponer una comunicación UDP entre dos máquinas de Internet A y B. El camino entre ambas atraviesa variedad de elementos de interconexión (concentradores, conmutadores y routers). Un paquete UDP generado por A con destino B tiene un error de un bit corrupto en su cabecera UDP a la mitad de camino ¿Qué elemento o elementos de red se darían cuenta de tal fallo? ¿Devolvería algún mensaje de notificación al origen?
- 2) Se desea implementar un servicio novedoso de intercambio de ficheros para la próxima Navarparty en el que el servidor que dispone del fichero lo descarga sobre todos los clientes interesados de manera simultánea, es decir, comienza a la vez en todos y la descarga es para todos a la misma tasa. Sabemos que todas las máquinas se encuentran en la misma red de área local ¿Qué esquema de protocolos utilizaría para implementar por encima esa aplicación si el objetivo principal es reducir la carga sobre la red? Determine las funcionalidades básicas (direccionamiento, detección de errores y/o fiabilidad) que tendría que implementar su aplicación por no cubrirse por protocolos inferiores. En general, ¿Qué problemas de nivel de enlace se podrían encontrar en esa red compuesta por más de 500 máquinas sobre la misma LAN Ethernet conmutada a 10/100?
- 3) Se ha realizado una captura de una aplicación que no funciona, resultando los 2 siguientes paquetes capturados, mostrando el timestamp y el contenido de cada paquete en hexadecimal desde el primer byte de nivel de enlace que se sabe que es Ethernet

```
09:24:34.821328
  0x0000: 0015 c65e 6ec2 0018 5107 30f2 0800 4500
  0x0010: 003f 73a6 4000 4011 78ff 82ce 9e8d 82ce
  0x0020: a9de 909d 0063 002b a410 616c 6775 6965
  0x0030: 6e20 6573 6375 6368 616e 646f 2065 6e20
  0x0040: 6573 7465 2070 7565 7274 6f3f 0a
09:24:34.821816
  0x0000: 0018 5107 30f2 0015 c65e 6ec2 0800 4500
  0x0010: 005b 41c8 0000 fc01 2ed1 82ce a9de 82ce
  0x0020: 9e8d 0303 4b42 0000 0000 4500 003f 73a6
  0x0030: 4000 3d11 7bff 82ce 9e8d 82ce a9de 909d
  0x0040: 0063 002b a410 616c 6775 6965 6e20 6573
  0x0050: 6375 6368 616e 646f 2065 6e20 6573 7465
```

- a) Identifique el protocolo de transporte utilizado por la aplicación y la dirección IP y puerto del servidor.
- b) Extraiga el mensaje enviado por la aplicación en el primer paquete (supuesto que sea texto ASCII).
- c) En base al segundo paquete, identifique la posible causa de que no funcione correctamente la aplicación.
- d) Determine si el servidor está localizado en la misma o diferente LAN; y en el caso de que sea diferente la distancia geográfica aproximada entre ambas LAN.

- 4) Suponga que se quiere transmitir un fichero de 1Mbyte a través de una conexión TCP sobre una red con MTU 1500. Determinar el número de paquetes de datos y el número de paquetes de control (SYN,FIN,ACK, etc) que se enviarán en esa transmisión.
- 5) Suponer una máquina A que manda un segmento TCP de SYN ISN_A a otra B. Entonces A recibe sólo un SYN ISN_B de B.
- ¿En qué estado del diagrama de transición de estados está la conexión para A?
 - ¿Cual es el siguiente segmento que tendrá que enviar A?
 - ¿Qué tipo de acción se está realizando?
 - ¿A qué nuevo estado se pasa?

- 6) La siguiente traza ha sido capturada en la red de la universidad

```
1 0.963491 IP 130.206.168.45.60905 > 193.252.23.108.110: S 2357731200:2357731200(0) win 65535
2 0.964072 IP 193.252.23.108.110 > 130.206.168.45.60905: S 1061601894:1061601894(0) ack 2357731201 win 5792
3 0.964129 IP 130.206.168.45.60905 > 193.252.23.108.110: . ack 1061601895 win 65535
4 1.111168 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061601895:1061601927(32) ack 2357731201 win 5792
5 1.111753 IP 130.206.168.45.60905 > 193.252.23.108.110: P 2357731201:2357731218(17) ack 1061601927 win 65535
6 1.112349 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731218 win 5792
7 1.200422 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061601927:1061601956(29) ack 2357731218 win 5792
8 1.200834 IP 130.206.168.45.60905 > 193.252.23.108.110: P 2357731218:2357731232(14) ack 1061601956 win 65535
9 1.201287 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731232 win 5792
10 1.711614 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061601956:1061601991(35) ack 2357731232 win 5792
11 1.712040 IP 130.206.168.45.60905 > 193.252.23.108.110: P 2357731232:2357731238(6) ack 1061601991 win 65535
12 1.712630 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731238 win 5792
13 1.861177 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061601991:1061602000(9) ack 2357731238 win 5792
14 1.861596 IP 130.206.168.45.60905 > 193.252.23.108.110: P 2357731238:2357731244(6) ack 1061602000 win 65535
15 1.862059 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731244 win 5792
16 2.064350 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061602000:1061602005(5) ack 2357731244 win 5792
17 2.065276 IP 130.206.168.45.60905 > 193.252.23.108.110: F 2357731244:2357731244(0) ack 1061602005 win 65535
18 2.065890 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731245 win 5792
19 2.065894 IP 193.252.23.108.110 > 130.206.168.45.60905: F 1061602005:1061602005(0) ack 2357731245 win 5792
20 2.065957 IP 130.206.168.45.60905 > 193.252.23.108.110: . ack 1061602006 win 65535
```

- Indique cómo se podría saber a qué aplicación pertenece.
- Indique cual es el cliente y cual el servidor en esta acción (con sus direcciones IP)
- Determine el tamaño de datos que envía la aplicación en cada sentido.
- ¿Puede estimar donde se ha capturado la traza (cerca del cliente o del servidor, o en un punto intermedio)?
- Haga una tabla para el cliente y otra para el servidor, indicando en qué estado de conexión estaba TCP al principio y en qué estado ha quedado después de enviar o recibir cada paquete mostrado.

- 7) Suponer una aplicación servidora sobre TCP que para cada cliente que se le conecta le devuelve cada 50ms un mensaje con la cotización de determinado valor en bolsa. Cada mensaje consiste en 100 bytes de datos TCP y se envían 10 mensajes, tras lo cual el servidor desconecta al cliente. Suponer activados algoritmo de Nagle y Delayed ACK.
- Determinar la traza de paquetes en formato `tcpdump` correspondiente a una conexión de un cliente con dirección `ipC`, puerto `pC`, con un servidor de dirección `ipS`, puerto `pS`, corriendo este servicio. Suponer la traza capturada en el lado del cliente, RTT 100ms, MSS 1460bytes, y buffer recepción 12MSS.
 - Determinar la evolución de la ventana de congestión con el tiempo para cada equipo.
 - ¿Cómo se podría configurar TCP para que el cliente reciba cada mensaje del servidor lo antes posible?
- 8) Entre dos máquinas B y A, se establece una conexión TCP en la que A es la máquina servidora. El servidor de la máquina A transfiere un fichero de 8000 bytes a toda máquina cliente que se le conecte y le cierra la conexión correctamente. Considerar que:
- A y B poseen direcciones `ipA` e `ipB` respectivamente.
 - Existe 1 RTT entre las máquinas A y B, ya que se encuentran separadas en 2 puntos de Internet.
 - El puerto de servidor A es el 99 y el de cliente B el 4444.
 - A y B están conectadas a redes de área local Ethernet.
 - El buffer de recepción de A y B es de 1 MSS.
 - Número de secuencia inicial de A y B es 0.
- Determinar la traza de paquetes en formato `tcpdump` que se podría capturar en el extremo de la máquina A con al menos los siguientes campos: nº paquete, timestamp, tamaño completo de la trama a nivel Ethernet, ip/puerto origen, ip/puerto destino, flags, nº secuencia, nº confirmación, tamaño datos TCP y ventana anunciada por el receptor de A tras cada paquete. Suponer que está activa la opción de anuncio del MSS de TCP
 - Si en el camino entre A y B existiera un segmento de red con MTU 1060, determinar cómo llegaría a B el primer paquete con datos del fichero enviado por A. Detallar para cada paquete su tamaño a nivel Ethernet y los campos de protocolos que considere significativos en esta ocasión.
 - ¿Cuántos paquetes de ACK se retornarían a A como respuesta al paquete de datos del apartado b)?
 - Se cambia el buffer de recepción a 2 MSS. Determine la duración de la conexión

- e) Determinar si sería posible las siguientes conexiones, consecutivamente en este orden sin que finalice ninguna de ellas, y por qué
- [1] Una máquina C desde su puerto 80 se conecta a la máquina A con un servicio en el puerto 99.
 - [2] La máquina A con un servidor en el puerto 99 se conecta a otra máquina D con un servidor en el puerto 80.
 - [3] Una máquina C desde su puerto 80 se conecta con otra máquina D con un servicio en el puerto 67.
 - [4] Una máquina C desde su puerto 9981 se conecta con una máquina D con un servicio también en el puerto 9981.
- f) Repetir el ejercicio suponiendo que:
- El umbral de slow start es 8 MSS para A y B
 - La ventana de congestión inicial es 1 MSS para A y B.
 - El buffer de recepción de A y B es de 20 MSS.
- 9) Suponga que se ha recogido la siguiente traza de una conexión TCP colocando el tcpdump en la máquina 130.206.169.222. Se sabe que las dos máquinas que aparecen están conectadas a 20 Kbytes/s.
- a) El paquete 1 parece que se ha perdido. ¿Cómo se ha detectado esa pérdida? ¿Puede ser que no se haya perdido?
 - b) ¿Ha habido alguna otra pérdida? ¿Cómo se ha detectado?
 - c) ¿Cuál es la ventana de congestión de 169.229.131.109 tras los paquetes 8 y 10?
 - d) ¿Cuál es el tamaño mínimo de buffer de recepción de 130.206.169.222 que hace que la velocidad de la conexión no esté limitada por este tamaño?
 - e) ¿Cuántas conexiones simultáneas se podrán establecer entre máquinas de Internet y el puerto 1244 de la máquina 130.206.169.222 mientras la conexión de la traza sigue activa? ¿Y con el puerto 80 de la máquina 169.229.131.109?

Nº	paq.	Traza
1	17.077761	130.206.169.222.1244 > 169.229.131.109.http: S 3155219005:3155219005(0) win 5840 <mss 1460 >
2	20.077783	130.206.169.222.1244 > 169.229.131.109.http: S 3155219005:3155219005(0) win 5840 <mss 1460 >
3	20.300796	169.229.131.109.http > 130.206.169.222.1244: S 715589519:715589519(0) ack 3155219006 win 32850 <mss 1326>
4	20.300902	130.206.169.222.1244 > 169.229.131.109.http: . ack 1 win 5840
5	20.301301	130.206.169.222.1244 > 169.229.131.109.http: P 1:114(113) ack 1 win 5840
6	20.523803	169.229.131.109.http > 130.206.169.222.1244: . ack 114 win 32850
7	20.529210	169.229.131.109.http > 130.206.169.222.1244: . 1:1315(1314) ack 114 win 32850
8	20.529284	130.206.169.222.1244 > 169.229.131.109.http: . ack 1315 win 7884
9	20.530337	169.229.131.109.http > 130.206.169.222.1244: P 2629:3943(1314) ack 114 win 32850
10	20.530437	130.206.169.222.1244 > 169.229.131.109.http: . ack 1315 win 7884
11	20.753019	169.229.131.109.http > 130.206.169.222.1244: . 3943:4307(364) ack 114 win 32850
12	20.753118	130.206.169.222.1244 > 169.229.131.109.http: . ack 1315 win 7884
13	20.754520	169.229.131.109.http > 130.206.169.222.1244: . 4307:5621(1314) ack 114 win 32850
14	20.754595	130.206.169.222.1244 > 169.229.131.109.http: . ack 1315 win 7884
15	20.978778	169.229.131.109.http > 130.206.169.222.1244: . 1315:2629(1314) ack 114 win 32850
16	20.978885	130.206.169.222.1244 > 169.229.131.109.http: . ack 5621 win 10512
17	20.979898	169.229.131.109.http > 130.206.169.222.1244: . 5621:6935(1314) ack 114 win 32850
18	20.979985	130.206.169.222.1244 > 169.229.131.109.http: . ack 6935 win 14454

```

19 | 21.204180 169.229.131.109.http > 130.206.169.222.1244: . 8249:8403(154) ack 114 win 32850
20 | 21.204265 130.206.169.222.1244 > 169.229.131.109.http: . ack 6935 win 14454
21 | 21.205418 169.229.131.109.http > 130.206.169.222.1244: . 6935:8249(1314) ack 114 win 32850
22 | 21.205521 130.206.169.222.1244 > 169.229.131.109.http: . ack 8403 win 17082
    
```

- 10) Suponga que se ha recogido la siguiente traza de una conexión TCP colocando un sniffer en un punto intermedio del camino que une ambas máquinas de la conexión.
- ¿Se puede decir a qué tipo de servicio pertenece esta conexión?
 - ¿Cuál es el número de secuencia real presente en la cabecera TCP del paquete 10?
 - ¿Podemos determinar la MTU del camino? ¿Por qué?
 - ¿Existe situación de congestión en la conexión? ¿Cómo se detecta?
 - ¿Cuál podría ser el paquete 17 que continuase la traza que envía el 10.0.0.2 a 192.168.0.100 suponiendo que no hay pérdidas y que 10.0.0.2 no envía nuevos datos?

Nº paq	
1	192.168.0.100.4480 > 10.0.0.2.5001: S 684446768:684446768(0) win 64240 <mss 1460> (DF)
2	10.0.0.2.5001 > 192.168.0.100.4480: S 3097576443:3097576443(0) ack 684446769 win 17520 <mss 1460> (DF)
3	192.168.0.100.4480 > 10.0.0.2.5001: . ack 1 win 64240 (DF)
4	192.168.0.100.4480 > 10.0.0.2.5001: P 1:501(500) ack 1 win 64240 (DF)
5	10.0.0.2.5001 > 192.168.0.100.4480: . ack 501 win 16520 (DF)
6	192.168.0.100.4480 > 10.0.0.2.5001: P 501:1001(500) ack 1 win 64240 (DF)
7	10.0.0.2.5001 > 192.168.0.100.4480: . ack 1001 win 16520 (DF)
8	192.168.0.100.4480 > 10.0.0.2.5001: P 1001:2461(1460) ack 1 win 64240 (DF)
9	10.0.0.2.5001 > 192.168.0.100.4480: . ack 1001 win 16520 (DF)
10	192.168.0.100.4480 > 10.0.0.2.5001: P 2461:3921(1460) ack 1 win 64240 (DF)
11	10.0.0.2.5001 > 192.168.0.100.4480: . ack 1001 win 16520 (DF)
12	192.168.0.100.4480 > 10.0.0.2.5001: P 3921:5381(1460) ack 1 win 64240 (DF)
13	10.0.0.2.5001 > 192.168.0.100.4480: . ack 1001 win 16520 (DF)
14	192.168.0.100.4480 > 10.0.0.2.5001: P 1001: 2461 (1460) ack 1 win 64240 (DF)
15	192.168.0.100.4480 > 10.0.0.2.5001: P 5381:6841(1460) ack 1 win 64240 (DF)
16	10.0.0.2.5001 > 192.168.0.100.4480: . ack 2461 win 17520 (DF)
17	

Nota: (DF)=activado campo de la cabecera IP Don't fragment

- 11) Suponga que se ha recogido la siguiente traza de una conexión web colocando un sniffer en el propio extremo 130.206.169.222.
- ¿Cuál es la ventana de congestión inicial de esta conexión para el servidor?
 - ¿Por qué los datos que envía el servidor son de 1448 bytes y no del tamaño del MSS?
 - ¿Cuál es el RTT aproximado entre cliente y servidor?
 - El paquete 24 ¿Es una retransmisión? ¿Por qué?
 - ¿De qué tamaño es el fichero enviado por el servidor?
 - ¿Entra en funcionamiento el control de flujo? ¿Por qué?

- g) Determinar el valor de la ventana de congestión (en MSSs) y el umbral de slow start (en MSSs) para las máquinas cliente y servidor después del envío de cada paquete de la traza. Suponer un umbral inicial de 8 MSS

Nº paq	
1	08:28:29.166721 IP 130.206.169.222.4391 > 130.206.158.134.http: S 60936803:60936803(0) win 5840 <mss 1460>
2	08:28:29.270150 IP 130.206.158.134.http > 130.206.169.222.4391: S 3932451824:3932451824(0) ack 60936804 win 5792 <mss 1460>
3	08:28:29.270509 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 1 win 5840
4	08:28:29.270650 IP 130.206.169.222.4391 > 130.206.158.134.http: P 1:121(120) ack 1 win 5840
5	08:28:29.374155 IP 130.206.158.134.http > 130.206.169.222.4391: . 1:1449(1448) ack 121 win 362
6	08:28:29.374193 IP 130.206.158.134.http > 130.206.169.222.4391: . 1449:2897(1448) ack 121 win 362
7	08:28:29.374979 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 1449 win 8688
8	08:28:29.375476 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 2897 win 11584
9	08:28:29.478151 IP 130.206.158.134.http > 130.206.169.222.4391: P 2897:4345(1448) ack 121 win 362
10	08:28:29.478185 IP 130.206.158.134.http > 130.206.169.222.4391: . 4345:5793(1448) ack 121 win 362
11	08:28:29.478198 IP 130.206.158.134.http > 130.206.169.222.4391: . 5793:7241(1448) ack 121 win 362
12	08:28:29.478997 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 4345 win 14480
13	08:28:29.479477 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 5793 win 17376
14	08:28:29.479505 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 7241 win 20272
15	08:28:29.582161 IP 130.206.158.134.http > 130.206.169.222.4391: . 8689:10137(1448) ack 121 win 362
16	08:28:29.582191 IP 130.206.158.134.http > 130.206.169.222.4391: P 10137:11585(1448) ack 121 win 362
17	08:28:29.582205 IP 130.206.158.134.http > 130.206.169.222.4391: P 11585:13033(1448) ack 121 win 362
18	08:28:29.582216 IP 130.206.158.134.http > 130.206.169.222.4391: F 14310:14310(0) ack 121 win 362
19	08:28:29.583459 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 7241 win 20272
20	08:28:29.583464 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 7241 win 20272
21	08:28:29.583468 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 7241 win 20272
22	08:28:29.583474 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 7241 win 20272
23	08:28:29.686162 IP 130.206.158.134.http > 130.206.169.222.4391: . 7241:8689(1448) ack 121 win 362
24	08:28:29.686193 IP 130.206.158.134.http > 130.206.169.222.4391: P 13033:14310(1277) ack 121 win 362
25	08:28:29.687457 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 13033 win 23168
26	08:28:29.687463 IP 130.206.169.222.4391 > 130.206.158.134.http: . ack 14311 win 26064
27	08:28:29.690822 IP 130.206.169.222.4391 > 130.206.158.134.http: R 121:121(0) ack 14311 win 26064

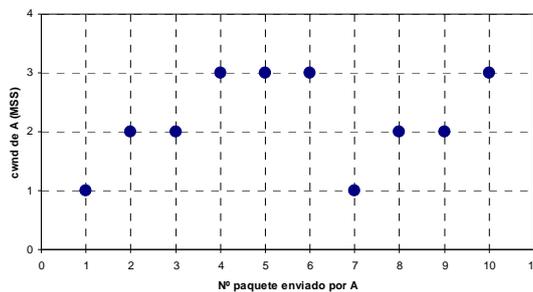
- 12) Suponga que se ha recogido la siguiente traza de una conexión web colocando un sniffer en el punto intermedio del camino entre ambas máquinas. Suponer un umbral de slow start de 8MSS.

- Estime la ventana de congestión inicial que como mínimo han utilizado el cliente y servidor vista la traza.
- ¿Se detecta la pérdida de algún paquete en la transmisión? ¿Cuántos? ¿Por qué?
- ¿Por qué se están recibiendo los ACKs duplicados 19 y 20 cuando ha habido retransmisión en 18?
- ¿Cómo es posible recibir el paquete de datos 26 cuando ya se había mandado un FIN en ese mismo sentido de la conexión?
- Determine los paquetes que se encuentran en zona de slow start y en zona de congestion avoidance desde el punto de vista del servidor.

Nº paq	
1	42.852793 IP 130.206.169.222.4319 > 130.206.158.134.80: S 1991692015:1991692015(0) win 5840 <mss 1460>
2	42.852859 IP 130.206.158.134.80 > 130.206.169.222.4319: S 1344552705:1344552705(0) ack 1991692016 win 5792 <mss 1460>
3	42.853229 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 1 win 5840
4	42.853371 IP 130.206.169.222.4319 > 130.206.158.134.80: P 1:121(120) ack 1 win 5840
5	42.853451 IP 130.206.158.134.80 > 130.206.169.222.4319: . ack 121 win 362
6	42.856103 IP 130.206.158.134.80 > 130.206.169.222.4319: . 1:1449(1448) ack 121 win 362
7	42.856161 IP 130.206.158.134.80 > 130.206.169.222.4319: . 1449:2897(1448) ack 121 win 362
8	42.857213 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 1449 win 8688
9	42.857320 IP 130.206.158.134.80 > 130.206.169.222.4319: . 4345:5793(1448) ack 121 win 362

10	42.857219 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 2897 win 11584
11	42.857380 IP 130.206.158.134.80 > 130.206.169.222.4319: . 5793:7241(1448) ack 121 win 362
12	42.857406 IP 130.206.158.134.80 > 130.206.169.222.4319: . 7241:8689(1448) ack 121 win 362
13	42.858215 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 2897 win 11584
14	42.858341 IP 130.206.158.134.80 > 130.206.169.222.4319: . 8689:10137(1448) ack 121 win 362
15	42.858222 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 2897 win 11584
16	42.858416 IP 130.206.158.134.80 > 130.206.169.222.4319: P 10137:11585(1448) ack 121 win 362
17	42.858703 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 2897 win 11584
18	42.858785 IP 130.206.158.134.80 > 130.206.169.222.4319: P 2897:4345(1448) ack 121 win 362
19	42.859202 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 2897 win 11584
20	42.859706 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 2897 win 11584
21	42.859712 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 11585 win 14480
22	42.860017 IP 130.206.158.134.80 > 130.206.169.222.4319: P 13033:14310(1277) ack 121 win 362
23	42.860359 IP 130.206.158.134.80 > 130.206.169.222.4319: F 14310:14310(0) ack 121 win 362
24	42.861202 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 11585 win 14480
25	42.861208 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 11585 win 14480
26	43.063480 IP 130.206.158.134.80 > 130.206.169.222.4319: P 11585:13033(1448) ack 121 win 362
27	43.064698 IP 130.206.169.222.4319 > 130.206.158.134.80: . ack 14311 win 17376
28	43.068611 IP 130.206.169.222.4319 > 130.206.158.134.80: R 121:121(0) ack 14311 win 17376

13) Una aplicación TCP en una máquina A con dirección ipA y puerto cliente pA se conecta a una máquina B con dirección ipB y puerto pB, de forma que en cuanto se produce la conexión, A le manda a B 12.000 bytes de datos y al terminar A finaliza la conexión. Suponer que A siempre tiene datos para mandar. Para ambas máquinas considerar MSS 1460 bytes, buffer recepción 12MSS, ssthresh 8MSS, RTT 100ms y temporizador Delayed Ack 100ms. Además considerar que el perfil de evolución de la ventana de congestión de A es la siguiente donde el eje horizontal son los paquetes enviados por A sin considerar las fases de establecimiento y cierre de la conexión:



- Determinar la traza de paquetes en formato tcpdump que correspondería a la conexión representada en ese perfil de la ventana de congestión, identificando los que corresponden a cada nº de paquete del perfil, e incluyendo los paquetes de establecimiento y cierre de la conexión. Suponer la traza capturada en la máquina A.
 - ¿Implementa esta conexión Delayed Ack o el algoritmo de Nagle?
 - Determine los paquetes de A que se encuentran en cuanto a la ventana de congestión en zona de slow start y congestion avoidance.
- 14) La siguiente traza en formato tcpdump se ha capturado en el extremo cliente. Suponiendo activados todos los mecanismos opcionales de TCP y un RTT que separa ambas máquinas:
- Determinar el número de bytes de datos TCP que transmite cliente y servidor.

- b) Determinar el número de bytes de datos TCP que podemos determinar que el servidor ha recibido con seguridad.
- c) ¿Por qué el tamaño máximo de datos TCP de los paquetes no coincide con el MSS anunciado al principio de la conexión?
- d) En la traza hay paquetes que se han perdido ¿Qué paquetes son (indicar el nº de paquete y por qué)? ¿Cómo detecta TCP que se han perdido todos ellos?
- e) ¿Cuál es por lo menos el valor de la ventana de congestión inicial del cliente?
- f) ¿Cuál es el valor de la ventana de congestión tras los paquetes 9, 10, 13 y 15?
- g) Determine el timestamp de cada paquete suponiendo que el primero tiene lugar en el segundo 0 (puede rellenarlo en la misma tabla).

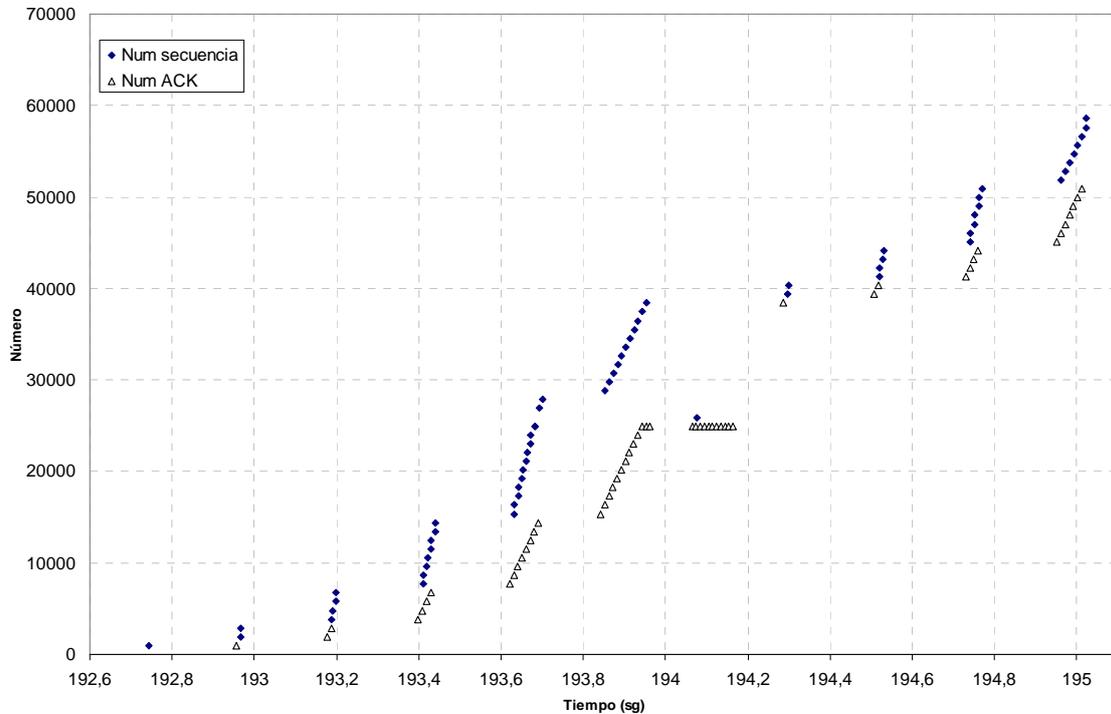
Nº	Timestamp	Descripción
1	0	10.1.1.1.34621 > 10.1.2.1.9999: S 3720297321:3720297321(0) win 5840 <mss 1460,sackOK,timestamp 23181 0,nop,wscale 1>
2		10.1.2.1.9999 > 10.1.1.1.34621: S 3719205918:3719205918(0) ack 3720297322 win 5792 <mss 1460,sackOK,timestamp 22977 23181,nop,wscale 1>
3		10.1.1.1.34621 > 10.1.2.1.9999: . ack 1 win 2920
4		10.1.1.1.34621 > 10.1.2.1.9999: . 1:1449(1448) ack 1 win 2920
5		10.1.1.1.34621 > 10.1.2.1.9999: . 1449:2897(1448) ack 1 win 2920
6		10.1.1.1.34621 > 10.1.2.1.9999: P 2897:4345(1448) ack 1 win 2920
7		10.1.2.1.9999 > 10.1.1.1.34621: . ack 1 win 2896
8		10.1.1.1.34621 > 10.1.2.1.9999: . 4345:5793(1448) ack 1 win 2920
9		10.1.1.1.34621 > 10.1.2.1.9999: . 1:1449(1448) ack 1 win 2920
10		10.1.2.1.9999 > 10.1.1.1.34621: . ack 1449 win 4344
11		10.1.1.1.34621 > 10.1.2.1.9999: . 5793:7241(1448) ack 1 win 2920
12		10.1.1.1.34621 > 10.1.2.1.9999: . 7241:8689(1448) ack 1 win 2920
13		10.1.1.1.34621 > 10.1.2.1.9999: . 1449:2897(1448) ack 1 win 2920
14		10.1.1.1.34621 > 10.1.2.1.9999: . 1449:2897(1448) ack 1 win 2920
15		10.1.2.1.9999 > 10.1.1.1.34621: . ack 4345 win 5792
16		10.1.1.1.34621 > 10.1.2.1.9999: . 8689:10137(1448) ack 1 win 2920
17		10.1.1.1.34621 > 10.1.2.1.9999: P 10137:11585(1448) ack 1 win 2920

15) Suponga una conexión TCP entre dos máquinas cliente A y servidor B. La aplicación servidora de la máquina B transfiere un fichero de 100 Mbytes a toda aplicación cliente que se le conecte y le cierra la conexión correctamente. Considerar las siguientes características de la conexión:

- El RTT entre A y B es 50ms.
- A y B poseen direcciones ipA e ipB respectivamente.
- El puerto del cliente A es el pA y el de servidor B el pB.
- El ancho de banda del cuello de botella es 3 Mbps, y no existe ningún otro tráfico simultáneo sobre los mismos enlaces.
- Ambas máquinas están conectadas a sendas redes Ethernet.
- El umbral de slow start es 8 MSS para ambas máquinas
- La ventana de congestión inicial es 1 MSS para ambas máquinas.
- El buffer de recepción de A y B es de 7.300 bytes.
- Suponer activados: Algoritmo de Nagle, Delayed ACK

- a) Determinar la velocidad máxima de transferencia que se consigue en el sentido de B a A. ¿Se producirá pérdida de algún paquete en esta conexión completa?. Justifique la respuesta.
 - b) Determinar los 20 primeros paquetes de la traza de paquetes en formato *tcpdump* que se podría capturar de esta conexión en el extremo de la máquina B con al menos los siguientes campos: n° paquete, timestamp, tamaño completo de la trama a nivel Ethernet, ip/puerto origen, ip/puerto destino, flags, n° secuencia, n° confirmación, tamaño datos TCP
 - c) Suponer que se pierde el tercer paquete que lleva datos de servidor a cliente ¿Cuáles serían los siguientes 6 paquetes que vienen a continuación en una traza como la del apartado anterior?
- 16) Suponer una conexión TCP entre un navegador y un servidor web. Una vez establecida la conexión el navegador solicita una URL (mensaje de datos TCP 100 bytes) a la que contesta el servidor con el fichero de la web deseada. En la figura se muestra la evolución de los números de secuencia de los paquetes de datos que manda el servidor y los números de confirmación TCP que devuelve el cliente. Para facilitar su interpretación, el número de secuencia representa la suma del número de secuencia TCP y los datos TCP del paquete, de forma que en verdad representa el número de secuencia del último byte de datos de cada paquete. La captura se ha realizado en un punto en el camino entre navegador y servidor web. Considere un MSS de 960 bytes negociado por ambos extremos, un temporizador de retransmisión de 2sg para ambos extremos, que el servidor siempre tiene datos para enviar y el servidor es el encargado de finalizar la conexión al terminar la transferencia.
- a) Determine aproximadamente el RTT entre los dos extremos de la conexión TCP. Determine la localización aproximada del punto de captura con respecto a los dos extremos de la conexión TCP medido en RTT desde el punto de captura a los dos extremos de la conexión.
 - b) Determine si TCP detecta congestión en la red ¿Cómo se ha detectado?
 - c) ¿Qué mecanismos opcionales de TCP implementa o no, y por qué?
 - d) Dibuje el valor aproximado de la ventana de congestión del servidor que corresponde en función del tiempo (eje x) y medido en MSSs (eje y).
 - e) Determine las zonas de slow start y congestion avoidance en la figura.
 - f) Estime el valor de umbral ssthresh que tenía la conexión al inicio
 - g) Supuesto que en la figura aparecen todos los paquetes de datos para finalizar la descarga de la página web. Determine todos los paquetes restantes necesarios de (datos o control) para finalizar la conexión. Para ello utilice un formato de traza *tcpdump* con cliente con dirección ipC, puerto pC, con servidor de dirección ipS, puerto pS, y ventana anunciada por el receptor de 30MSS

- h) Determine a qué lado del punto de captura (en el camino del navegador al punto de captura o del punto de captura al servidor) se encuentra el cuello de botella en cuanto a velocidad para la conexión. ¿Podría estimar el ancho de banda del cuello de botella?



17) Suponga una conexión TCP entre dos máquinas cliente A y servidor B. La aplicación cliente de la máquina A transfiere un fichero de 50 Mbytes al servidor tan pronto como se conecta con él, a la máxima velocidad posible (siempre tiene datos para enviar), y es el cliente el que también cierra la conexión correctamente tras finalizar el envío del fichero. La máquina B está saturada atendiendo a muchos clientes simultáneamente y por tanto la aplicación servidora sólo es capaz de leer del buffer de recepción TCP un segmento pasados 250ms desde que llegó al buffer. Considerar además las siguientes características de la conexión:

- El RTT entre A y B es 100ms.
- A y B poseen direcciones ipA e ipB respectivamente.
- El puerto del cliente A es el pA y el de servidor B el pB.
- El ancho de banda del cuello de botella es 2 Mbps, y no existe ningún otro tráfico simultáneo sobre los mismos enlaces.
- Ambas máquinas están conectadas a sendas redes Ethernet 100BaseT (100Mbps).
- El umbral de slow start es 9 MSS para ambas máquinas
- La ventana de congestión inicial es 1 MSS para ambas máquinas.
- El buffer de recepción de A y B es de 8 MSS.
- Suponer activado Algoritmo de Nagle y desactivado Delayed ACK

- a) Determinar la velocidad media de transferencia del fichero que se consigue. Justifique la respuesta.

- b) Determinar si se produce pérdida de algún paquete a lo largo de esta conexión. Justifique la respuesta.
- c) Determinar los 25 primeros paquetes de la traza de paquetes en formato *tcpdump* que se podría capturar en un punto intermedio entre la máquina A y B con al menos los siguientes campos: n° paquete, timestamp, ip/puerto origen, ip/puerto destino, flags, n° secuencia, n° confirmación, tamaño datos TCP, tamaño ventana anunciada por el receptor
- d) Determinar el valor de la ventana de congestión de A y de B tras cada paquete de la traza anterior.
- e) Determinar el número secuencia real del paquete de FIN mandado por la máquina A suponiendo que el número de secuencia inicial de A es 47.
- f) En el caso de que usted fuese el administrador del servidor B y pudiese dimensionarlo y configurarlo correctamente para atender a los clientes a la máxima velocidad. ¿Qué cambiaría y en tal caso qué velocidad de transferencia se conseguiría con el cliente A? Justifique la respuesta.

18) Suponga que se ha recogido la siguiente traza de una conexión TCP de un servicio en el que el servidor siempre tiene datos para enviar cuando se le conecta un cliente hasta que termina la transferencia. Suponer un umbral de slow start inicial de 8MSS.

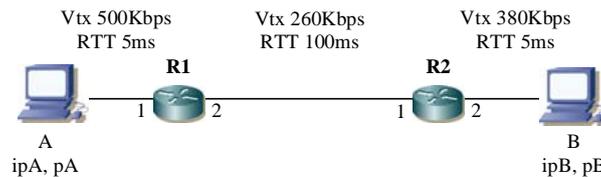
- a) Determine o estime, cuando sea posible lo siguiente. Justifique la respuesta:
 - el punto donde se ha realizado la captura en el camino de cliente a servidor
 - el tamaño del buffer de recepción del cliente
 - el tamaño de datos recibidos por el cliente
 - si está activado Delayed ACK
 - el número de secuencia real del paquete 8 en la red
- b) Determine si TCP ha detectado congestión y cómo la ha detectado, identificando los números de paquetes implicados. De producirse, localice en qué punto del camino entre cliente y servidor se ha producido la congestión. Justifique la respuesta
- c) ¿Se puede afirmar si el tráfico de esta conexión comparte la red con conexiones de otros usuarios? ¿Cuál es la velocidad máxima alcanzada por la conexión? Justifique la respuesta
- d) Determine, cuando sea posible, los valores de la ventana de congestión, umbral de slow start y la ventana disponible de envío del servidor medidos en MSSs tras recibir cada paquete (en los que esté como destino) y antes de enviar cada paquete (en los que esté como origen).
- e) Suponga un segundo cliente que se encuentra en la misma red que el cliente original y se conecta simultáneamente al mismo servidor (mismo instante de establecimiento de la conexión). Presente dos gráficas con eje horizontal el tiempo en RTTs y eje vertical el valor de la ventana de congestión del servidor

para con cada cliente para ambas transferencias simultáneas con ese servidor pero sólo los primeros 4 RTTs

Nº paq	Timestamp	Paquete
1	0,000882	66.6.6.6:4324> 88.8.8.8:21: S 84532523: 84532523 (0) win 14600 <mss 1460>
2	0,050049	88.8.8.8:21 > 66.6.6.6:4324: S 9431:9431(0) ack 1 win 14600 <mss 1460>
3	0,051013	66.6.6.6:4324> 88.8.8.8:21: . ack 1 win 14600
4	0,100730	88.8.8.8:21 > 66.6.6.6:4324: P 1:1461(1460) ack 1 win 14600
5	0,101380	66.6.6.6:4324> 88.8.8.8:21: . ack 1461 win 13140
6	0,150382	88.8.8.8:21 > 66.6.6.6:4324: P 1461:2921(1460) ack 1 win 14600
7	0,153686	66.6.6.6:4324> 88.8.8.8:21: . ack 2921 win 14600
8	0,200321	88.8.8.8:21 > 66.6.6.6:4324: P 5841:7301(1460) ack 1 win 14600
9	0,202645	88.8.8.8:21 > 66.6.6.6:4324: P 4381:5841(1460) ack 1 win 14600
10	0,204381	66.6.6.6:4324> 88.8.8.8:21: . ack 2921 win 14600
11	0,206997	66.6.6.6:4324> 88.8.8.8:21: . ack 2921 win 14600
12	2,257823	88.8.8.8:21 > 66.6.6.6:4324: P 2921:4381(1460) ack 1 win 14600
13	2,250244	66.6.6.6:4324> 88.8.8.8:21: . ack 7301 win 14600
14	2,302571	88.8.8.8:21 > 66.6.6.6:4324: P 7301:8761 (1460) ack 1 win 14600
15	2,304349	88.8.8.8:21 > 66.6.6.6:4324: P 8761:10221 (1460) ack 1 win 14600
16	2,300352	66.6.6.6:4324> 88.8.8.8:21: . ack 7301 win 14600
17	4,351232	88.8.8.8:21 > 66.6.6.6:4324: P 7301:8761 (1460) ack 1 win 14600
18	4,350647	66.6.6.6:4324> 88.8.8.8:21: . ack 10221 win 14600
19	4,400685	88.8.8.8:21 > 66.6.6.6:4324: P 10221:11681(1460) ack 1 win 14600
20	4,400731	88.8.8.8:21 > 66.6.6.6:4324: F 11681: 11681 (0) ack 1 win 14600
21	4,401059	66.6.6.6:4324> 88.8.8.8:21: . ack 11682 win 14600
22	4,402755	66.6.6.6:4324> 88.8.8.8:21: F 1:1(0) ack 11682 win 14600
23	4,450425	88.8.8.8:21 > 66.6.6.6:4324: . ack 2 win 14600

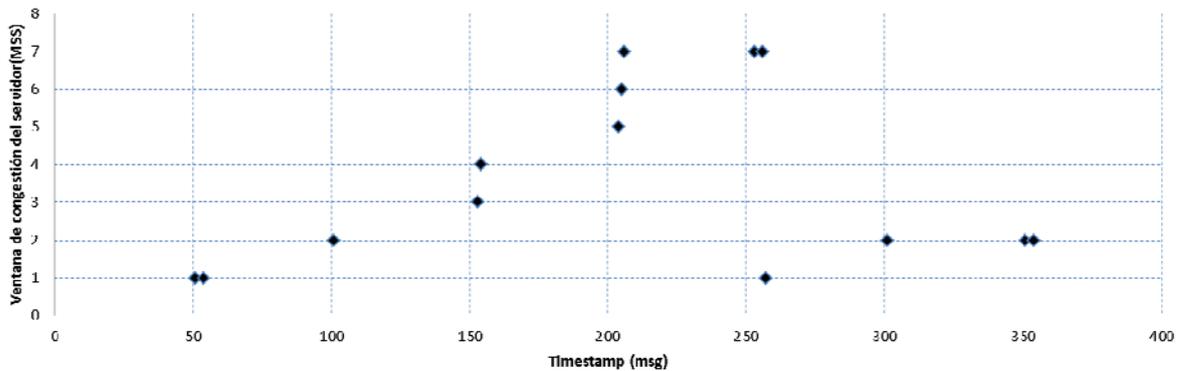
19) Suponga el escenario de la figura en el que una aplicación corriendo en la máquina A establece una conexión TCP con una aplicación en la máquina B para mandarle un fichero de 20 Kbytes en cuanto se establece la conexión, a la máxima velocidad posible (siempre tiene datos para enviar). El segmento de red A-R1 tiene una velocidad de transmisión de 500Kbps, el segmento R1-R2 de 260Kbps y el segmento R2-B de 380Kbps. El RTT entre A y R1 es 5ms, entre R1 y R2 100ms, y entre R2 y B 5ms. Considerar además las siguientes características de la conexión:

- A y B poseen direcciones ipA e ipB respectivamente.
- El puerto del cliente A es el pA y el de servidor B el pB.
- Las direcciones IP de los routers son IPRX,Y donde X es el número de router e Y el número de interfaz.
- No existe otro tráfico simultáneo en las redes.
- El MSS es 1000 bytes en las 2 redes extremo y con ese MSS no hay fragmentación en la red intermedia.
- El umbral de slow start es 8 MSS para ambas máquinas.
- La ventana de congestión inicial es 1 MSS para ambas máquinas.
- El buffer de recepción es de 5 MSS para ambas máquinas.
- Suponer activado Algoritmo de Nagle, desactivado Delayed ACK y activada la opción de anuncio de MSS.



- a) Determinar los 25 primeros paquetes de la traza de paquetes en formato tcpdump que se podría capturar en el router R2 con al menos los siguientes campos: n° paquete, timestamp, ip/puerto origen, ip/puerto destino, flags, n° secuencia, n° confirmación, tamaño datos TCP, tamaño ventana anunciada por el receptor. Determinar también el valor de la ventana de congestión y umbral de slow start de A tras cada paquete enviado o recibido por A. Justifique las consideraciones que considere más significativa.
 - b) Determinar las zonas de slow start y congestion avoidance sobre los números de paquete de la traza. Justifique la respuesta.
 - c) En el caso de producirse una error de transmisión en el segmento entre R1 y R2 que corrompe un bit de los datos TCP de la conexión anterior. ¿Qué elemento y capa de protocolo se da cuenta del error y descarta el paquete? ¿Qué elemento y capa de protocolo se encargará de retransmitir los datos perdidos?
- 20) Suponer que un cliente corriendo en una máquina C se descarga un fichero de un servidor 130.206.160.215: para ello le manda unos datos TCP eligiendo el fichero y a continuación el servidor se lo manda. El perfil de evolución de la ventana de congestión del servidor durante la conexión se muestra en la figura y cada punto indica el valor de la ventana de congestión justo después de haber recibido un paquete en el servidor desde el cliente con el flag de ACK activado. Cada ACK se grafica con el timestamp del instante en el que se ve pasar por el punto de captura. Considere:
- MSS de 1460 bytes negociado por ambos extremos
 - Ventana de congestión inicial 1MSS para ambos extremos
 - Que el servidor tiene siempre datos para enviar
 - Buffer de recepción de 20MSS para ambos extremos
 - Umbral de slow start de 8 MSS para ambos extremos
 - El punto de captura de la traza es en el cliente
 - La dirección IP del cliente es ipC. Suponer ipS del servidor para simplificar.
- a) Determinar los mecanismos opcionales de TCP que implementa o no la conexión. Justifique la respuesta
 - b) Estime la velocidad máxima que alcanza esta conexión TCP. ¿Esta velocidad está limitada por la red? Justifique la respuesta
 - c) Determine la traza en formato tcpdump (incluyendo el número de paquete y timestamp) que se capturaría en el punto de captura
 - d) Determine el valor del umbral de slow start del servidor y de la ventana disponible en el servidor después de recibir o mandar los paquetes de la traza anterior según corresponda.

- e) La ventana de congestión inicial en la implementación TCP de los sistemas operativos actuales suele variar entre 1 y 4 MSSs. Si se aumentase la ventana de congestión inicial a 10MSS, determine los beneficios y perjuicios que supondría.

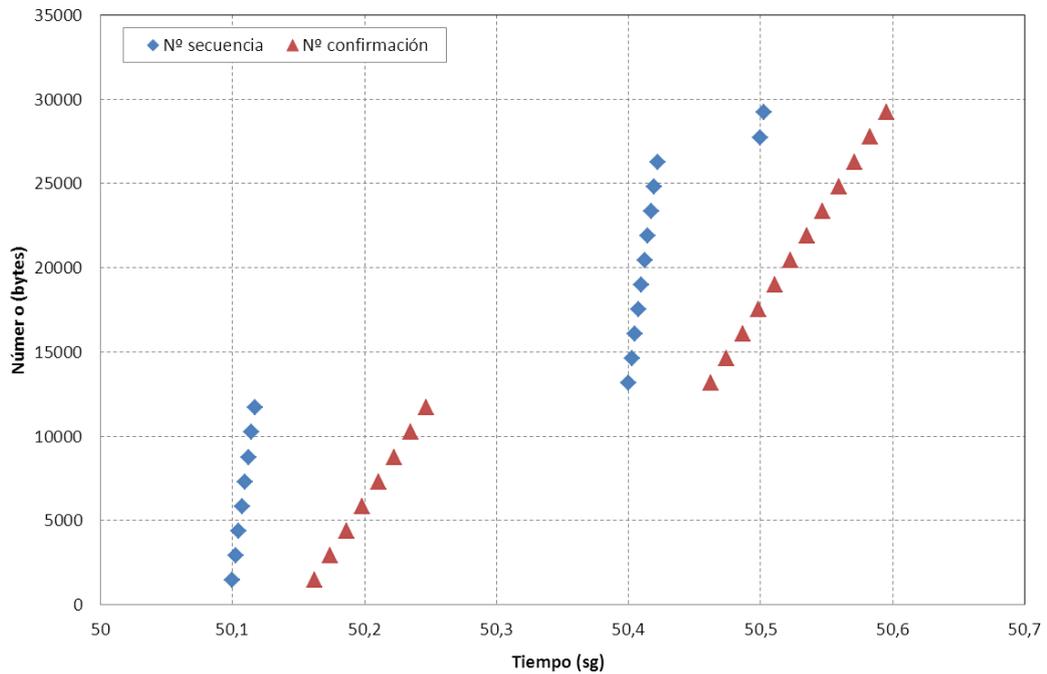


- 21) Suponga que se ha recogido la siguiente traza de una conexión TCP de un servicio desconocido en determinado punto de la red. Suponer un umbral de slow start inicial de 8MSS y un temporizador de retransmisión inicial de 2 sg.

Nº paq	Tiempo (sg)	Paquete
1	0,000000	2.2.2.2.2222 > 3.3.3.3.3333: S 34244:34244 (0) win 14600 <mss 1460>
2	0,050000	3.3.3.3.3333 > 2.2.2.2.2222: S 882:882 (0) ack 1 win 14600 <mss 1460>
3	0,050604	2.2.2.2.2222 > 3.3.3.3.3333: . ack 1 win 14600
4	0,051208	2.2.2.2.2222 > 3.3.3.3.3333: P 1:1461(1460) ack 1 win 14600
5	0,065493	2.2.2.2.2222 > 3.3.3.3.3333: P 1461:2921(1460) ack 1 win 14600
6	0,100604	3.3.3.3.3333 > 2.2.2.2.2222: P 1:1461(1460) ack 1 win 14600
7	0,101208	2.2.2.2.2222 > 3.3.3.3.3333: . ack 1461 win 13140
8	0,101210	3.3.3.3.3333 > 2.2.2.2.2222: . ack 1461 win 14600
9	0,101812	2.2.2.2.2222 > 3.3.3.3.3333: P 2921:4300 (1379) ack 1 win 13140
10	0,114890	3.3.3.3.3333 > 2.2.2.2.2222: P 1461:2921(1460) ack 1461 win 14600
11	0,115493	2.2.2.2.2222 > 3.3.3.3.3333: ack 2921 win 11680
12	0,115494	3.3.3.3.3333 > 2.2.2.2.2222: . ack 2921 win 14600
13	0,129175	3.3.3.3.3333 > 2.2.2.2.2222: P 2921:4381(1460) ack 2921 win 14600
14	0,129779	2.2.2.2.2222 > 3.3.3.3.3333: . ack 4381 win 10220
15	0,151112	3.3.3.3.3333 > 2.2.2.2.2222: . ack 4300 win 14600
16	0,151208	3.3.3.3.3333 > 2.2.2.2.2222: P 4381:5841(1460) ack 4300 win 14600
17	0,151812	2.2.2.2.2222 > 3.3.3.3.3333: . ack 5841 win 13140
18	0,165493	3.3.3.3.3333 > 2.2.2.2.2222: P 5841:7301(1460) ack 4300 win 14600
19	0,166097	2.2.2.2.2222 > 3.3.3.3.3333: . ack 7301 win 11680
20	0,179779	3.3.3.3.3333 > 2.2.2.2.2222: P 7301:8761 (1460) ack 4300 win 14600
21	0,180383	2.2.2.2.2222 > 3.3.3.3.3333: . ack 8761 win 10220
22	0,201812	3.3.3.3.3333 > 2.2.2.2.2222: P 13141:14601(1460) ack 4300 win 14600
23	0,202416	2.2.2.2.2222 > 3.3.3.3.3333: . ack 8761 win 13140
24	0,216097	3.3.3.3.3333 > 2.2.2.2.2222: P 14601:16061(1460) ack 4300 win 14600
25	0,216701	2.2.2.2.2222 > 3.3.3.3.3333: . ack 8761 win 11680
26	0,230383	3.3.3.3.3333 > 2.2.2.2.2222: P 16061:17521(1460) ack 4300 win 14600
27	0,230987	14.2.2.2.2222 .3.3.3.3333: . ack 8761 win 10220

- a) Determine o estime, cuando sea posible, lo siguiente. Justifique la respuesta:
- el RTT que ve el cliente y el RTT que ve el servidor

- el tamaño de datos TCP enviados por el cliente
 - qué mecanismos opcionales de TCP están activados
 - el número de secuencia real del paquete 18 en la red
- b) Determine, cuando sea posible, los valores de la ventana de congestión, umbral de slow start y la ventana disponible de cliente y todos ellos también de servidor medidos en MSSs justo después de recibir en el servidor el paquete 27.
- c) Determine las líneas de traza correspondiente a los paquetes 28, 29 y 30 que continúan la traza suponiendo que el servidor tiene más datos para enviar.
- d) Determine la razón que limita la velocidad de esta conexión TCP. ¿A qué velocidad o velocidades de transmisión sobre la red está limitada? Suponer enlaces full-duplex en todo el camino. Justifique la respuesta.
- e) Una de las máquinas de la conexión es muy lenta ya que está muy cargada atendiendo otras tareas. Identifique de qué máquina se trata y en qué condiciones podría suponer un problema para la conexión. Justifique la respuesta.
- f) Suponer que se pierde el paquete 5 ¿Cuándo detectaría TCP esa pérdida y retransmitiría ese paquete? Justifique la respuesta.
- g) En un cliente TCP cualquiera escrito en Java ¿El método bind() u otro método que lo incluya es obligatorio? ¿Y en un cliente UDP? Justifique la respuesta.
- 22) Suponer una conexión TCP para la que se muestra en la figura la evolución de los números de secuencia de los paquetes de datos que manda el servidor y los números de confirmación TCP que le devuelve el cliente, cuando la conexión lleva establecida 50 segundos. Para facilitar su interpretación, el número de secuencia representa la suma del número de secuencia TCP y los datos TCP del paquete, de forma que en verdad representa el número de secuencia del último byte de datos de cada paquete. La captura se ha realizado en la red del servidor. Considere un temporizador de retransmisión de 2sg para ambos extremos.
- a) Determine el tipo de conexión TCP, interactiva o masiva u otra combinación ¿El servidor tiene siempre datos para enviar? Justifique la respuesta.
- b) Determine, si es posible, el RTT de la conexión TCP. Justifique al respuesta.
- c) Determine si TCP detecta congestión en la red. Justifique la respuesta.
- d) ¿Puede determinar si entra en funcionamiento el control de flujo o el control de congestión?
- e) Estime, si es posible, la velocidad de la red en la red del servidor, en la red del cliente y la del cuello de botella entre ambos.
- f) En la figura suponga que se pierde el 8º paquete en número de secuencia. Rehaga la figura en las nuevas condiciones.



23) A continuación tiene un programa cliente TCP en Java modelo. Determine los cambios necesarios (use los números de línea como referencia) para convertirlo en un cliente TCP que corre en la máquina 7.7.7.7, usando como puerto cliente el 9999, y se conecta al servidor 130.206.160.1 en el puerto 81. Una vez establecida la conexión el cliente tiene que mandar la cadena de texto “DOWNLOAD”, a lo cual el servidor le devuelve por la misma conexión el contenido de un fichero de texto. El servidor cierra la conexión al terminar de enviar el contenido del fichero. El cliente tiene que mostrar el contenido del fichero de texto recibido en pantalla.

```

1. import java.io.*;
2. import java.net.*;
3. class TCPClient {
4.     public static void main(String argv[]) throws Exception
5.     {
6.         String sentence;
7.         String modifiedSentence;
8.         BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
9.         Socket clientSocket = new Socket("server.com", 6789);
10.        DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
11.        BufferedReader inFromServer = new BufferedReader(new
12.        InputStreamReader(clientSocket.getInputStream()));
13.        sentence = inFromUser.readLine();
14.        outToServer.writeBytes(sentence + '\n');
15.        modifiedSentence = inFromServer.readLine();
16.        System.out.println("FROM SERVER: " + modifiedSentence);
17.        clientSocket.close();
18.    }

```