



Universidad
Carlos III de Madrid

TECNOLOGÍAS *DEL* SECTOR FINANCIERO

Práctica 2: Mensajería de baja latencia

Andoni Alcelay

MÁSTER UNIVERSITARIO EN TECNOLOGÍAS DEL SECTOR FINANCIERO: FINTECH

Introducción

En este documento se explicarán los resultados obtenidos en la práctica 2 realizada con shockets y wireshark. El documento está dividido en 3 partes donde 2 de ellas son resultados e implementación de diferentes tipos de shockets para enviar y recibir mensajes ejecutando una aplicación servidor y otra cliente. El último punto será analizar todas estas implementaciones con whireshark y analizar sus resultados.

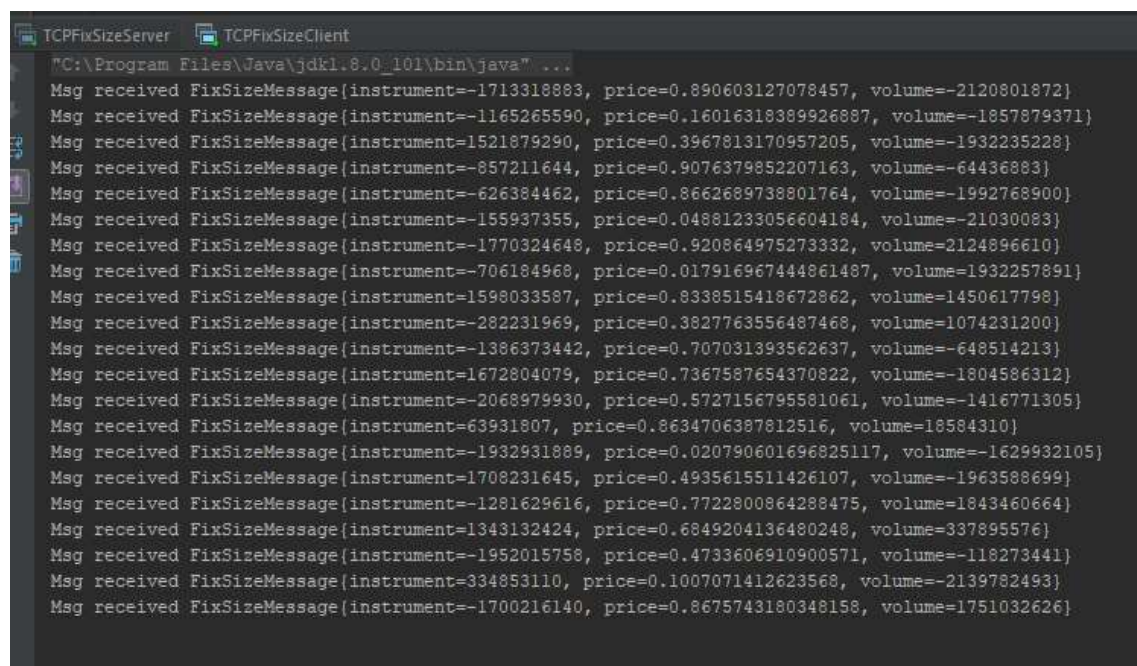
Práctica 1: Creación de comunicaciones TCP

En esta práctica crearemos un canal de comunicación simple TCP en Java, probando el envío y recepción y la serialización de mensajes. Cada mensaje representa un precio de mercado, con su instrumento, cantidad y precio. Esta práctica se divide en dos partes, la mensajería de tamaño fijo y la de tipo variable.

Conexión de tamaño fijo

Tras seguir los pasos de la práctica, se procede a ejecutar la aplicación en el orden correcto, es decir, empezando por lanzar el servidor y después el cliente. El resultado ha sido el mostrado en las ilustraciones 1 y 2.

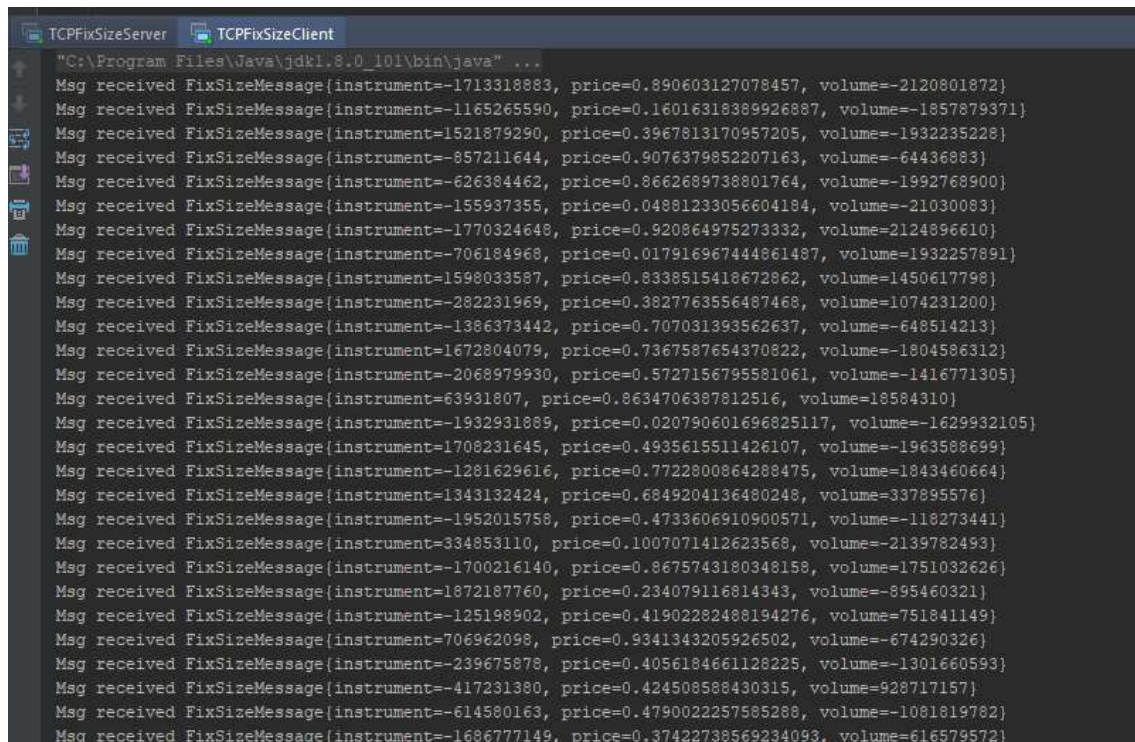
Servidor:



```
TCPSizeServer TCPFixedSizeClient
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
Msg received FixSizeMessage[instrument=-1713318883, price=0.890603127078457, volume=-2120801872]
Msg received FixSizeMessage[instrument=-1165265590, price=0.16016318389926887, volume=-1857879371]
Msg received FixSizeMessage[instrument=1521879290, price=0.3967813170957205, volume=-1932235228]
Msg received FixSizeMessage[instrument=-857211644, price=0.9076379852207163, volume=-64436883]
Msg received FixSizeMessage[instrument=-626384462, price=0.8662689738801764, volume=-1992768900]
Msg received FixSizeMessage[instrument=-155937355, price=0.04881233056604184, volume=-21030083]
Msg received FixSizeMessage[instrument=-1770324648, price=0.920864975273332, volume=2124896610]
Msg received FixSizeMessage[instrument=-706184968, price=0.017916967444861487, volume=1932257891]
Msg received FixSizeMessage[instrument=1598033587, price=0.8338515418672862, volume=1450617798]
Msg received FixSizeMessage[instrument=-282231969, price=0.3827763556487468, volume=1074231200]
Msg received FixSizeMessage[instrument=-1386373442, price=0.707031393562637, volume=-648514213]
Msg received FixSizeMessage[instrument=1672804079, price=0.7367587654370822, volume=-1804586312]
Msg received FixSizeMessage[instrument=-2068979930, price=0.5727156795581061, volume=-1416771305]
Msg received FixSizeMessage[instrument=63931807, price=0.8634706387812516, volume=18584310]
Msg received FixSizeMessage[instrument=-1932931889, price=0.020790601696825117, volume=-1629932105]
Msg received FixSizeMessage[instrument=1708231645, price=0.4935615511426107, volume=-1963588699]
Msg received FixSizeMessage[instrument=-1281629616, price=0.7722800864288475, volume=1843460664]
Msg received FixSizeMessage[instrument=1343132424, price=0.6849204136480248, volume=337895576]
Msg received FixSizeMessage[instrument=-1952015758, price=0.4733606910900571, volume=-118273441]
Msg received FixSizeMessage[instrument=334853110, price=0.1007071412623568, volume=-2139782493]
Msg received FixSizeMessage[instrument=-1700216140, price=0.8675743180348158, volume=1751032626]
```

Ilustración 1: Ejecución de sockets tamaño fijo, servidor

Cliente:



```

TCPFixSizeServer TCPFixSizeClient
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
Msg received FixSizeMessage{instrument=-1713318883, price=0.890603127078457, volume=-2120801872}
Msg received FixSizeMessage{instrument=-1165265590, price=0.16016318389926887, volume=-1857879371}
Msg received FixSizeMessage{instrument=1521879290, price=0.3967813170957205, volume=-1932235228}
Msg received FixSizeMessage{instrument=-857211644, price=0.9076379852207163, volume=-64436883}
Msg received FixSizeMessage{instrument=-626384462, price=0.8662689738801764, volume=-1992768900}
Msg received FixSizeMessage{instrument=-155937355, price=0.04881233056604184, volume=-21030083}
Msg received FixSizeMessage{instrument=-1770324648, price=0.920864975273332, volume=2124896610}
Msg received FixSizeMessage{instrument=-706184968, price=0.017916967444861487, volume=1932257891}
Msg received FixSizeMessage{instrument=1598033587, price=0.8338515418672862, volume=1450617798}
Msg received FixSizeMessage{instrument=-282231969, price=0.3827763556487468, volume=1074231200}
Msg received FixSizeMessage{instrument=-1386373442, price=0.707031393562637, volume=-648514213}
Msg received FixSizeMessage{instrument=1672804079, price=0.7367587654370822, volume=-1804586312}
Msg received FixSizeMessage{instrument=-2068979930, price=0.5727156795581061, volume=-1416771305}
Msg received FixSizeMessage{instrument=63931807, price=0.8634706387812516, volume=18584310}
Msg received FixSizeMessage{instrument=-1932931889, price=0.020790601696825117, volume=-1629932105}
Msg received FixSizeMessage{instrument=1708231645, price=0.4935615511426107, volume=-1963588699}
Msg received FixSizeMessage{instrument=-1281629616, price=0.7722800864288475, volume=1843460664}
Msg received FixSizeMessage{instrument=1343132424, price=0.6849204136480248, volume=337895576}
Msg received FixSizeMessage{instrument=-1952015758, price=0.4733606910900571, volume=-118273441}
Msg received FixSizeMessage{instrument=334853110, price=0.1007071412623568, volume=-2139782493}
Msg received FixSizeMessage{instrument=-1700216140, price=0.8675743180348158, volume=1751032626}
Msg received FixSizeMessage{instrument=1872187760, price=0.234079116814343, volume=-895460321}
Msg received FixSizeMessage{instrument=-125198902, price=0.41902282488194276, volume=751841149}
Msg received FixSizeMessage{instrument=706962098, price=0.9341343205926502, volume=-674290326}
Msg received FixSizeMessage{instrument=-239675878, price=0.4056184661128225, volume=-1301660593}
Msg received FixSizeMessage{instrument=-417231380, price=0.424508588430315, volume=928717157}
Msg received FixSizeMessage{instrument=-614580163, price=0.4790022257585288, volume=-1081819782}
Msg received FixSizeMessage{instrument=-1686777149, price=0.37422738569234093, volume=616579572}

```

Ilustración 2: Ejecución de sockets tamaño fijo, cliente

Conexión de tamaño variable

En este caso el tamaño de los mensajes que se envía es variable, para este primer intento se utilizar el valor 8 para el tamaño del mensaje aleatorio que se genera, valor que se le pasa al método `generateRandomMsg()`;

Los resultados son los que se muestran en las ilustraciones 3 y 4 respectivamente:

Servidor

```
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.9740674294447723, volume=-1424182709, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.7203936591237928, volume=-669993917, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.7969009862928788, volume=-1357627176, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.9707286556282084, volume=-509171165, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.7757193021165529, volume=-816071453, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.714702432798532, volume=-760199809, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.3967567070565944, volume=407071429, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.38678929492948677, volume=586488827, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.7457384095184781, volume=-1397625433, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.355381763970933, volume=1492835860, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.7021341407387195, volume=1654465678, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
About to send msg of size 04
```

Ilustración 3: Mensajería de tamaño variable (8), servidor

Cliente

```
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
Read MsgSize 40
Msg received VariableSizeMessage{price=0.9740674294447723, volume=-1424182709, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
Read MsgSize 40
Msg received VariableSizeMessage{price=0.7203936591237928, volume=-669993917, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
Read MsgSize 40
Msg received VariableSizeMessage{price=0.7969009862928788, volume=-1357627176, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
Read MsgSize 40
Msg received VariableSizeMessage{price=0.9707286556282084, volume=-509171165, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
Read MsgSize 40
Msg received VariableSizeMessage{price=0.7757193021165529, volume=-816071453, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
Read MsgSize 40
Msg received VariableSizeMessage{price=0.714702432798532, volume=-760199809, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
Read MsgSize 40
Msg received VariableSizeMessage{price=0.3967567070565944, volume=407071429, instrument='INST{8}{8}{8}{8}{8}{8}{8}' }
Read MsgSize 40
```

Ilustración 4: Mensajería de tamaño variable (8), cliente

Para la segunda parte de la práctica, hay que intentar cambiar este tamaño de los mensajes enviados, que actualmente es 8. Para este segundo intento se le subirá ese valor a 80000. Los resultados son mostrados en las ilustraciones 5 y 6.

Servidor:

```
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
About to send msg of size 04
Message sent: VariableSizeMessage{price=0.7688412083608532, volume=723625207, instrument='INST{80000}{80000}{80000}{80000}{80000}{80000}{80000}' }
About to send msg of size 04
|
```

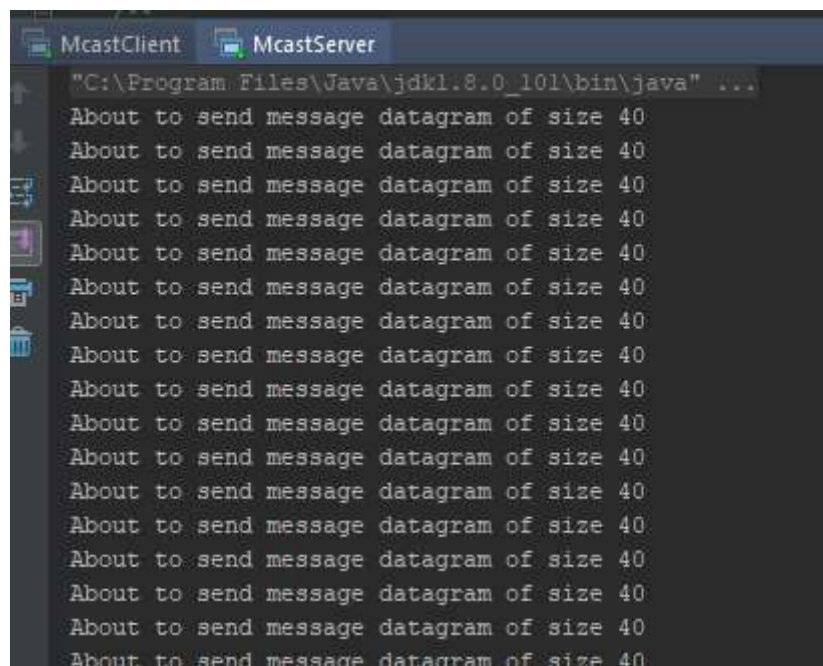
Ilustración 5: Mensajería de tamaño variable (80000), servidor

Práctica 2: Creación de comunicaciones Multicast

En esta práctica crearemos un canal de comunicación simple Multicast en Java, probando el envío y recepción y la serialización de mensajes. Cada mensaje representa un precio de mercado, con su instrumento, cantidad y precio. A diferencia de TCP, en Multicast las comunicaciones son UDP, es decir no hay sesión, ni control de flujo, ni de orden, etc.

Tras ejecutar el cliente y el servidor el resultado ha sido el mostrado en las ilustraciones 9 y 10.

Servidor:



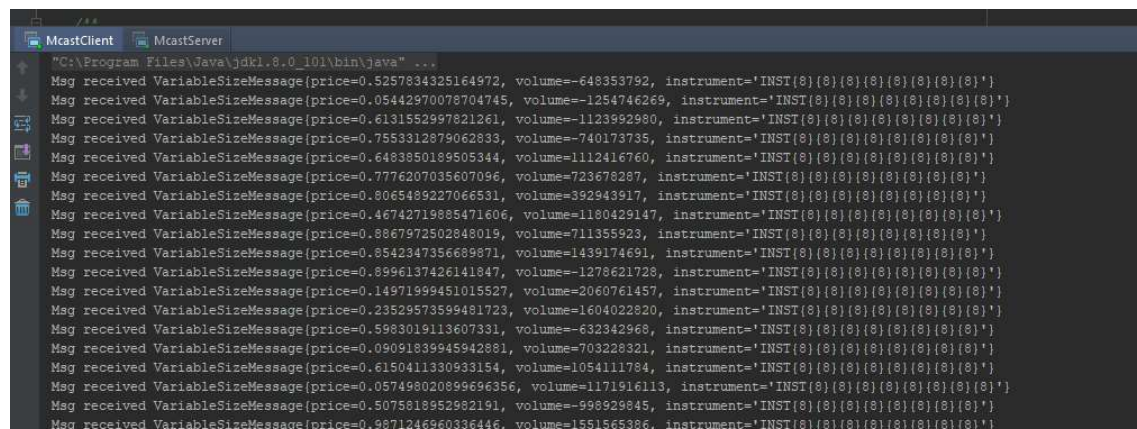
```

McastClient McastServer
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40
About to send message datagram of size 40

```

Ilustración 9: Mensajería multicast, servidor

Cliente:



```

McastClient McastServer
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
Msg received VariableSizeMessage(price=0.5257834325164972, volume=-648353792, instrument='INST(8){8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.05442970078704745, volume=-1254746269, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.6131552997821261, volume=-1123992980, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.7553312879062833, volume=-740173735, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.6483950189505344, volume=1112416760, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.7776207035607096, volume=723678287, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.8065489227066531, volume=392943917, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.46742719885471606, volume=1180429147, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.8867972502848019, volume=711355923, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.8542347356689871, volume=1439174691, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.8996137426141847, volume=1278621728, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.14971999451015527, volume=2060761457, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.23529573599481723, volume=1604022820, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.5983019113607331, volume=-632342968, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.09091839945942881, volume=703228321, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.6150411330933154, volume=1054111784, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.057498020899696356, volume=1171916113, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.5075818952982191, volume=-998929845, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')
Msg received VariableSizeMessage(price=0.9871246960336446, volume=1551565386, instrument='INST(8){8}{8}{8}{8}{8}{8}{8}')

```

Ilustración 10: Mensajería multicast, cliente

Como en la práctica anterior se cambiará el tamaño del mensaje a enviar a 80000, y podrá verse que no es capaz de enviar porque en este caso se usan Datagramas, y estos tienen un tamaño fijo que no se puede sobrepasar.

```

McastClient  McastServer
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
About to send message datagram of size 560016
Exception in thread "main" java.net.SocketException: The message is larger than the maximum supported by the underlying transport: Datagram send failed
    at java.net.DualStackPlainDatagramSocketImpl.socketSend(Native Method)
    at java.net.DualStackPlainDatagramSocketImpl.send(DualStackPlainDatagramSocketImpl.java:136)
    at java.net.DatagramSocket.send(DatagramSocket.java:693)
    at com.cnebrera.uc3.tech.lesson2.mcast.McastServer.sendMessage(McastServer.java:55)
    at com.cnebrera.uc3.tech.lesson2.mcast.McastServer.main(McastServer.java:25)

Process finished with exit code 1

```

Ilustración 11: Error por sobrepasar el tamaño máximo de Datagram

Wireshark

Ciente sin servidor

Como se ve en la ilustración 12, se ha hecho un envío de paquetes con el servidor apagado, esto se ve porque hay envíos por el puerto 6789 que no se reciben nunca, pueden verse con color rojo

No.	Time	Source	Destination	Protocol	Length	Info
33	0.578654	127.0.0.1	127.0.0.1	TCP	52	63981 → 6789 [SYN] Seq=0 Win=64240 Len=0 MSS=65495 WS=256 SACK_PERM=1
34	0.578654	127.0.0.1	127.0.0.1	TCP	40	6789 → 63981 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
40	1.079745	127.0.0.1	127.0.0.1	TCP	52	[TCP Retransmission] 63981 → 6789 [SYN] Seq=0 Win=64240 Len=0 MSS=65495 WS=256 SACK_PERM=1
41	1.079745	127.0.0.1	127.0.0.1	TCP	40	6789 → 63981 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
49	1.580407	127.0.0.1	127.0.0.1	TCP	52	[TCP Retransmission] 63981 → 6789 [SYN] Seq=0 Win=64240 Len=0 MSS=65495 WS=256 SACK_PERM=1
50	1.580407	127.0.0.1	127.0.0.1	TCP	40	6789 → 63981 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Offset	Hex	ASCII
0000	45 00 00 34 53 0c 40 00 80 06 00 00 7f 00 00 01	E..4S@.....
0010	7f 00 00 01 f9 ed 1a 85 fa 5b 5c 90 00 00 00 00[.....
0020	80 02 fa f6 10 9a 00 00 02 04 ff d7 01 03 03 08	...[.....
0030	01 01 04 02

Ilustración 12: Envío de paquetes sin respuesta

En cambio en la ilustración 13 la cosa cambia, se envían paquetes por el puerto 6789 que sí que tienen respuesta, y esta se puede ver en la siguiente traza al envío.

Cliente con servidor

No.	Time	Source	Destination	Protocol	Length	Info
124	11.464794	127.0.0.1	127.0.0.1	TCP	56	6789 → 64069 [PSH, ACK] Seq=113 Ack=1 Win=525568 Len=0
125	11.464794	127.0.0.1	127.0.0.1	TCP	40	64069 → 6789 [ACK] Seq=1 Ack=129 Win=525312 Len=0
147	12.466886	127.0.0.1	127.0.0.1	TCP	56	6789 → 64069 [PSH, ACK] Seq=129 Ack=1 Win=525568 Len=0
148	12.466886	127.0.0.1	127.0.0.1	TCP	40	64069 → 6789 [ACK] Seq=1 Ack=145 Win=525312 Len=0
149	13.465443	127.0.0.1	127.0.0.1	TCP	56	6789 → 64069 [PSH, ACK] Seq=145 Ack=1 Win=525568 Len=0
150	13.465443	127.0.0.1	127.0.0.1	TCP	40	64069 → 6789 [ACK] Seq=1 Ack=161 Win=525312 Len=0
151	14.465770	127.0.0.1	127.0.0.1	TCP	56	6789 → 64069 [PSH, ACK] Seq=161 Ack=1 Win=525568 Len=0
152	14.465770	127.0.0.1	127.0.0.1	TCP	40	64069 → 6789 [ACK] Seq=1 Ack=177 Win=525312 Len=0
153	15.466108	127.0.0.1	127.0.0.1	TCP	56	6789 → 64069 [PSH, ACK] Seq=177 Ack=1 Win=525568 Len=0
154	15.466108	127.0.0.1	127.0.0.1	TCP	40	64069 → 6789 [ACK] Seq=1 Ack=193 Win=525312 Len=0
155	16.466419	127.0.0.1	127.0.0.1	TCP	56	6789 → 64069 [PSH, ACK] Seq=193 Ack=1 Win=525568 Len=0
156	16.466419	127.0.0.1	127.0.0.1	TCP	40	64069 → 6789 [ACK] Seq=1 Ack=209 Win=525312 Len=0
157	17.466752	127.0.0.1	127.0.0.1	TCP	56	6789 → 64069 [PSH, ACK] Seq=209 Ack=1 Win=525568 Len=0
158	17.466752	127.0.0.1	127.0.0.1	TCP	40	64069 → 6789 [ACK] Seq=1 Ack=225 Win=525312 Len=0
159	18.467038	127.0.0.1	127.0.0.1	TCP	56	6789 → 64069 [PSH, ACK] Seq=225 Ack=1 Win=525568 Len=0
160	18.467038	127.0.0.1	127.0.0.1	TCP	40	64069 → 6789 [ACK] Seq=1 Ack=241 Win=525312 Len=0

```

> Frame 159: 56 bytes on wire (448 bits), 56 bytes captured (448 bits)
  Raw packet data
    Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
    Transmission Control Protocol, Src Port: 6789, Dst Port: 64069, Seq: 225, Ack: 1, Len: 16
    Data (16 bytes)

```

0000	45 00 00 38 54 7b 42 00	80 06 00 00 7f 00 00 01	E::8T:@-----
0010	7f 00 00 01 1a 85 fa 45	1b 97 07 2f dd b6 c2 50	-----E---P
0020	50 18 08 05 f4 6c 00 00	1c cd a9 7e 3f e1 95 74	P-----t
0030	23 31 53 f7 b8 62 12 83		#15-b-

Ilustración 13: Envío de paquetes con respuesta