Botnet Screenshots

Tölvusamskipti

13. október, 2019

Dr. Jacky Mallett

Egill Aron Þórisson, Nökkvi Karlsson

Below are screenshots that prove that all our server commands and client commands work as they are described in the README.

CLIENT COMMAND: CONNECT <IP> <PORT>:
The picture below shows a client telling a server(101) to connect to another server(102)
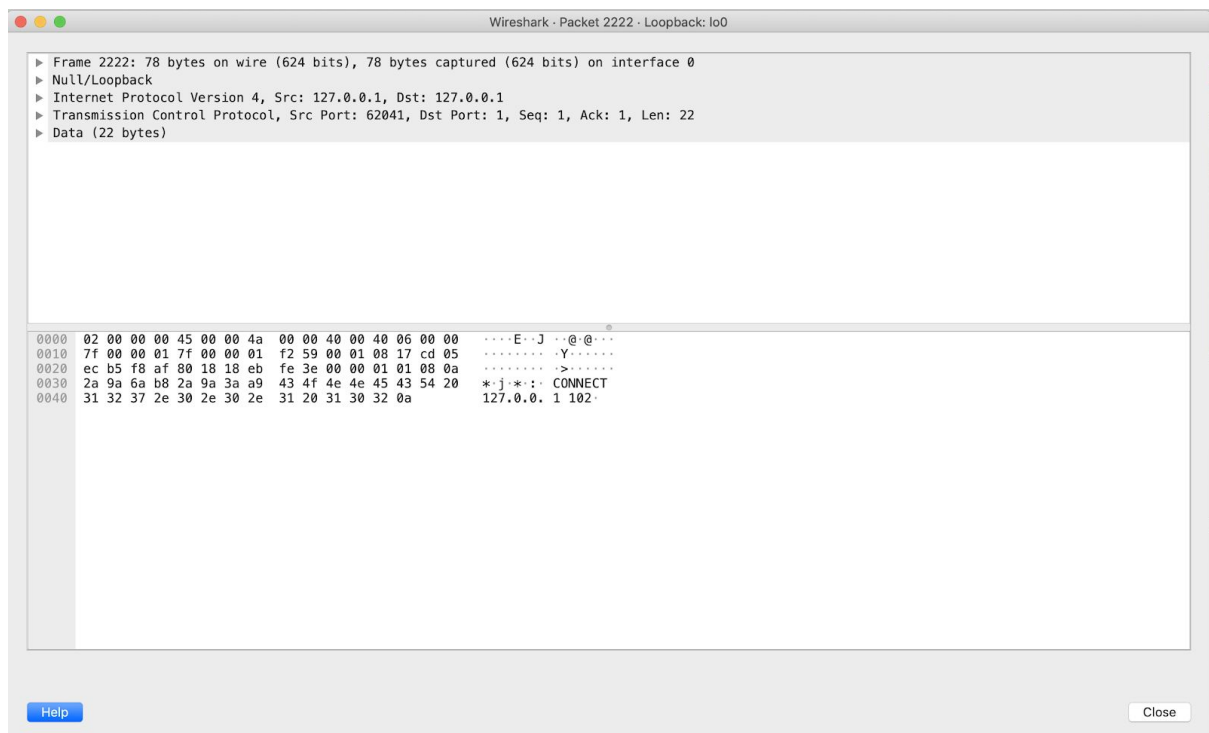

This is the wireshark trace of the client sending the CONNECT command to the server.

SERVER COMMAND: LISTSERVERS,<MY_GROUP_ID>

The picture below shows the two servers connected after sending a
LISTSERVERS,<MY_GROUP_ID> to each other.

Below is a picture that shows the wireshark trace of one of the servers sending
LISTSERVERS,<MY_GROUP_ID> to the other.
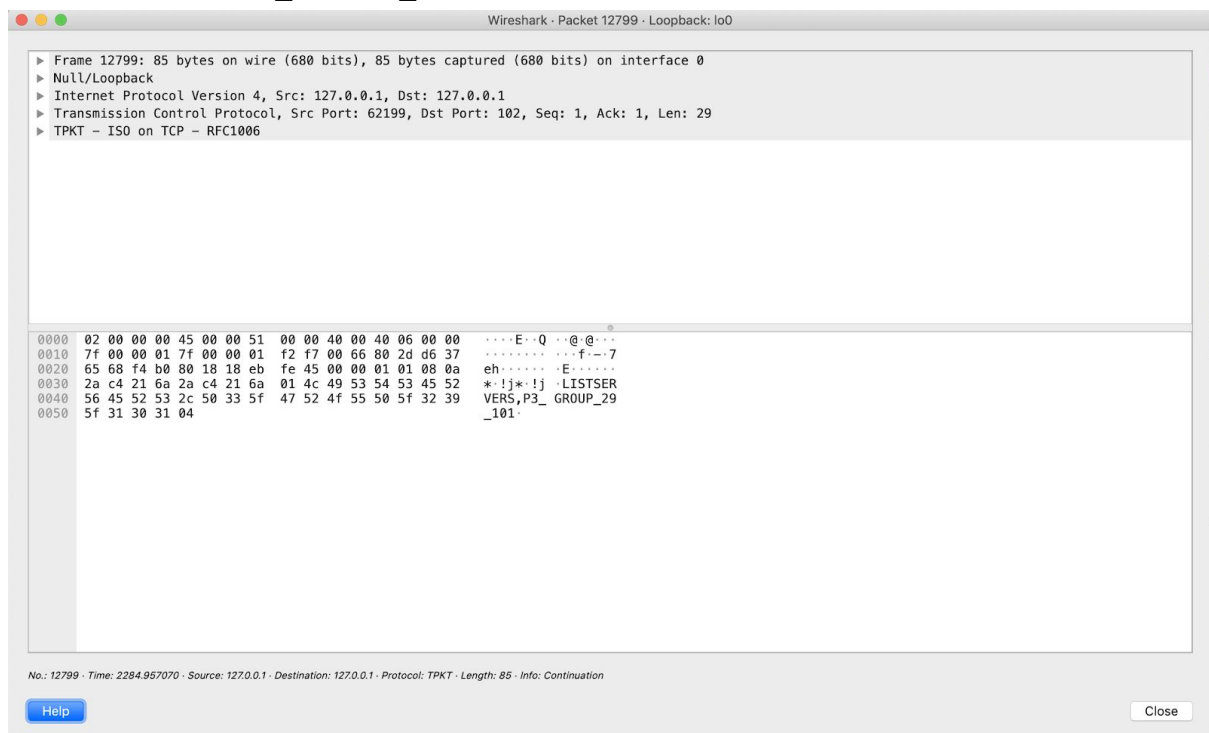
Below is a picture that shows the wireshark trace communication of one of the servers sending the response to LISTSERVERS,<MY_GROUP_ID> which is SERVERS,…….
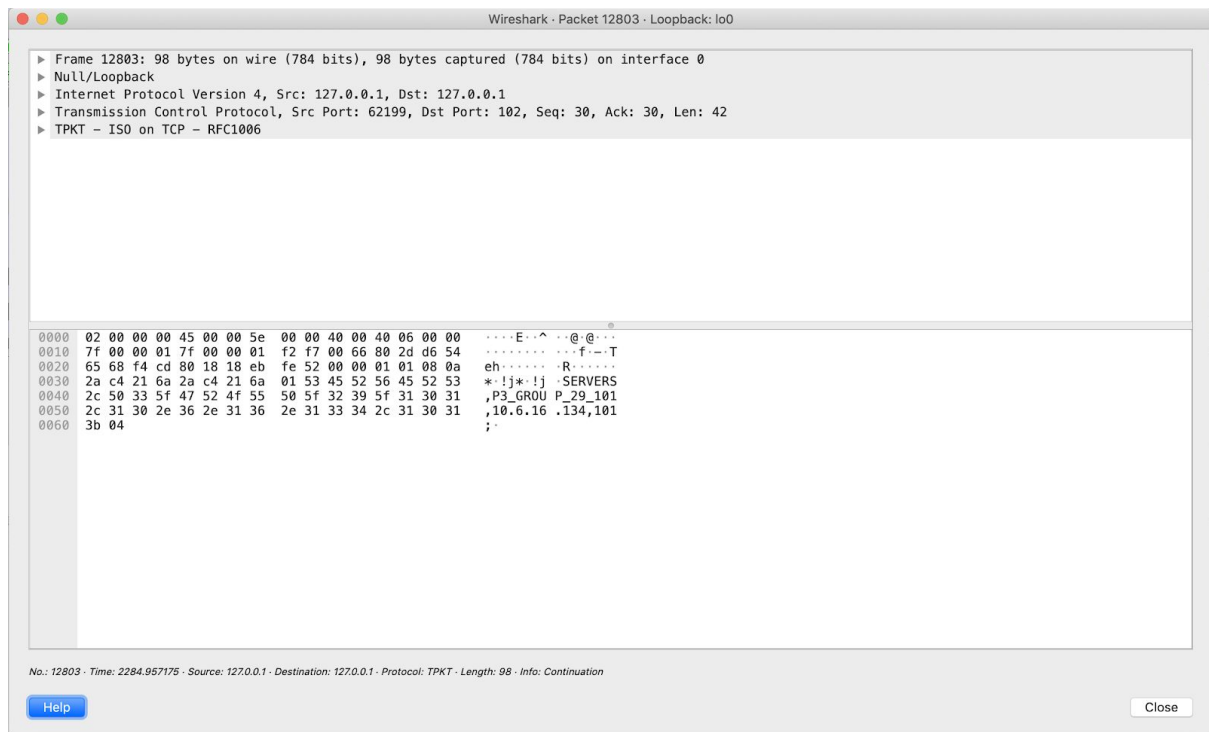
CLIENT COMMAND: LISTSERVERS

Below is a picture that shows a client sending a LISTSERVERS command to a server he is connected to and the server responds with a list of servers he is directly connected to.
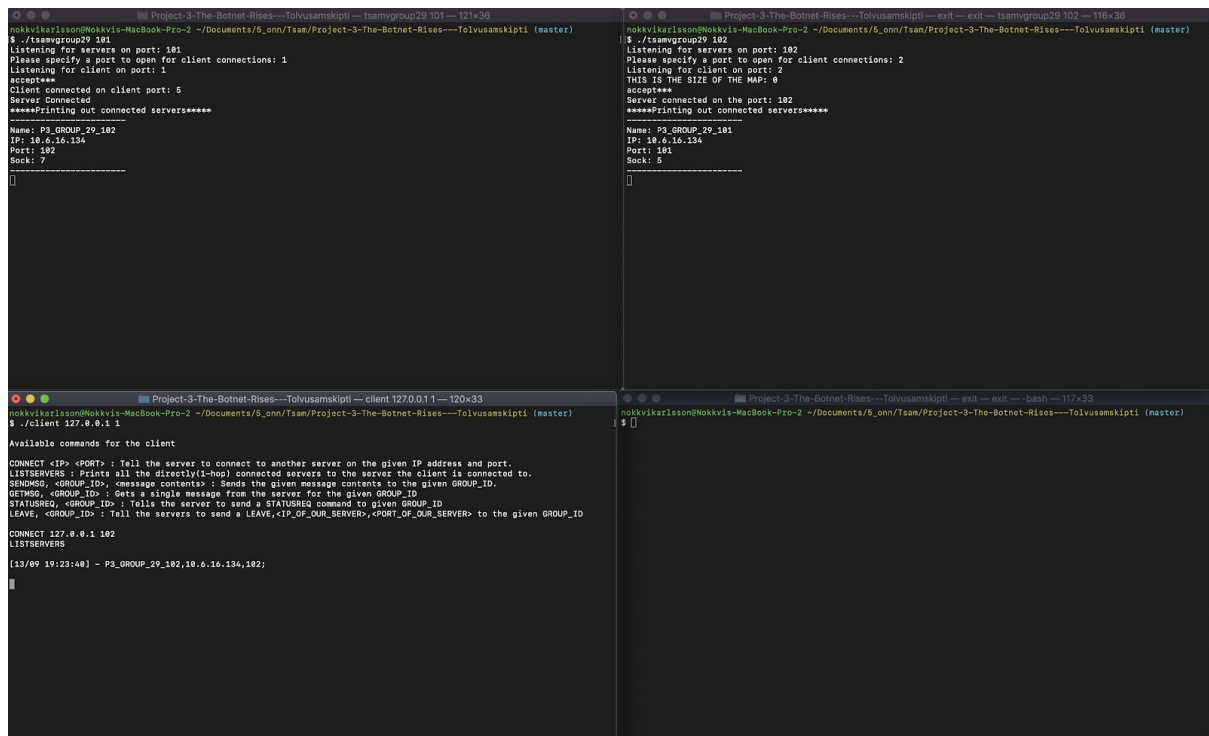
Below we can see the client sending a LISTSERVERS command to the server.
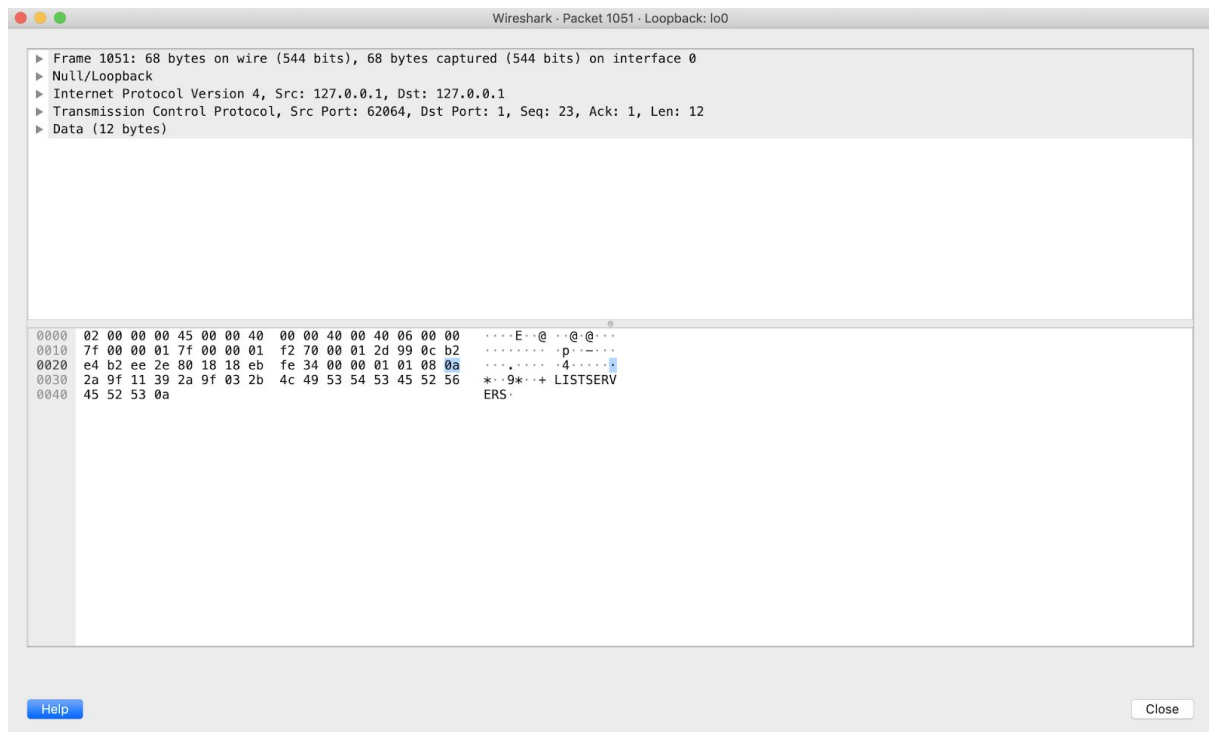
Below we can see the servers responding to the LISTSERVER client command and sending back a list of all the servers that are directly connected to him.
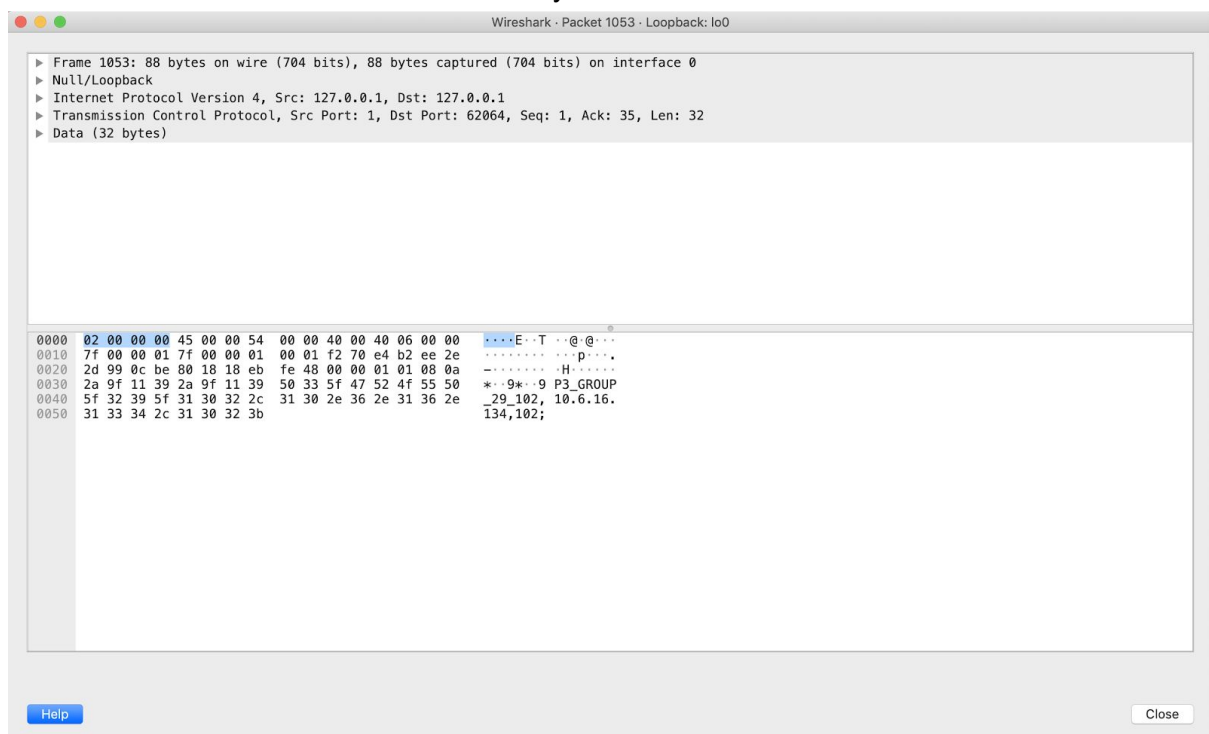
SERVER COMMANDS: KEEPALIVE,<no. messages> & GET_MSG,<GROUP_ID> & SEND_MSG,<FROM_GROUP_ID>,<TO_GROUP_ID>,<No. messages>.

The picture below shows server number 101 sending a KEEPALIVE,<1> to server number 102 and then 102 sends a GET_MSG,<server_number_102> back to server 101 to retrieve the message 101 is storing. When server 101 receives the GET_MSG command he sends a SEND_MSG,<FROM_GROUP_ID>,<TO_GROUP_ID>,<message> back.

This picture shows the KEEPALIVE,<no. Messages> being sent.
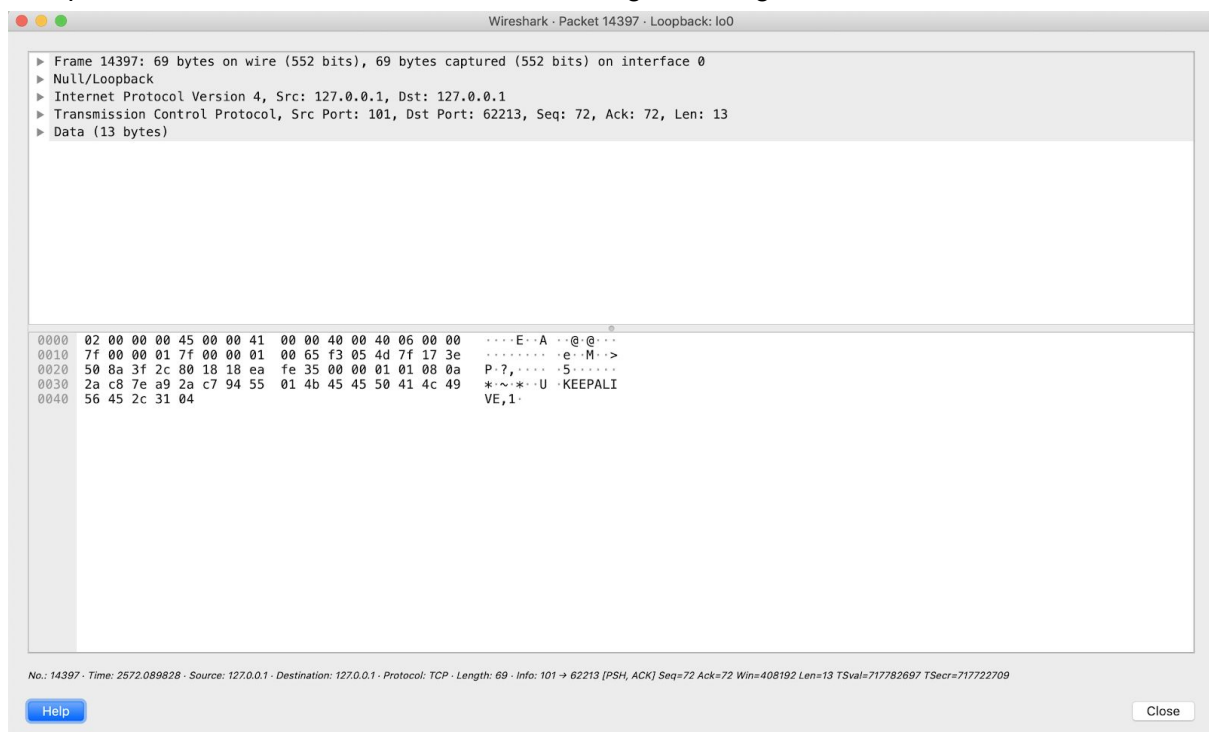
The picture below shows the GET_MSG,<GROUP_ID> command being sent.
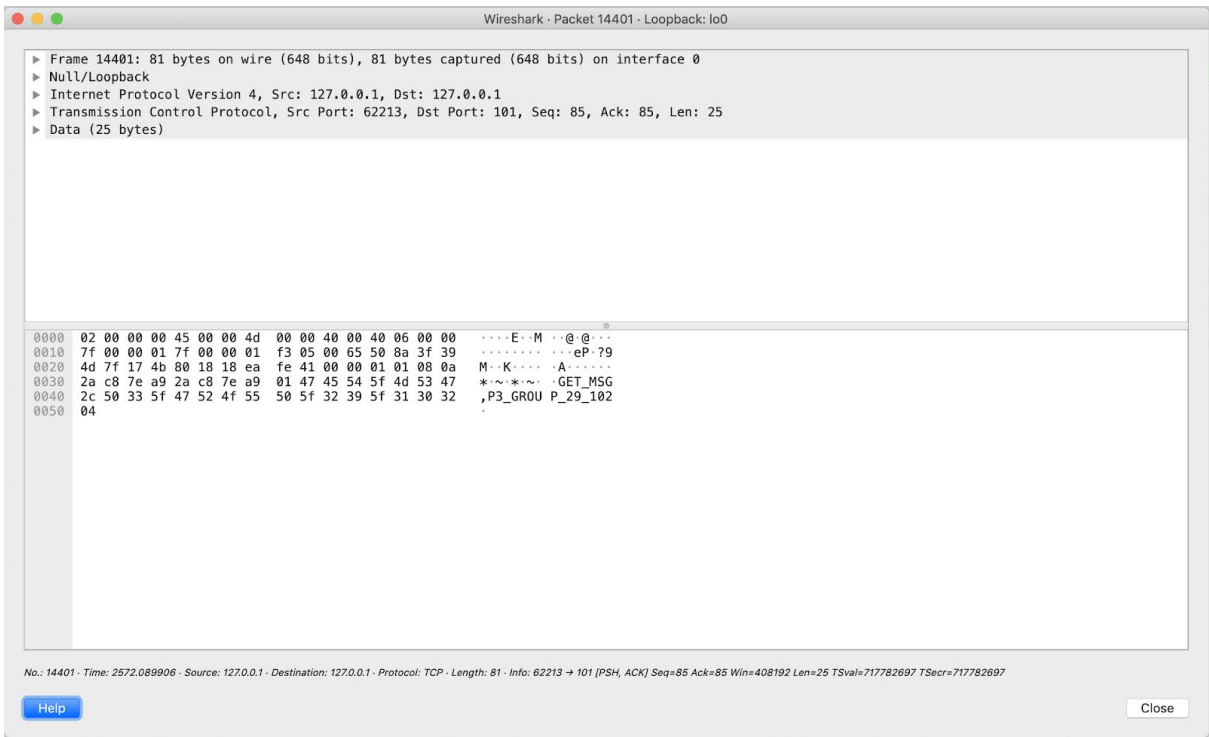
The picture below shows the
SEND_MSG,<FROM_GROUP_ID>,<TO_GROUP_ID>,<message> command being sent.
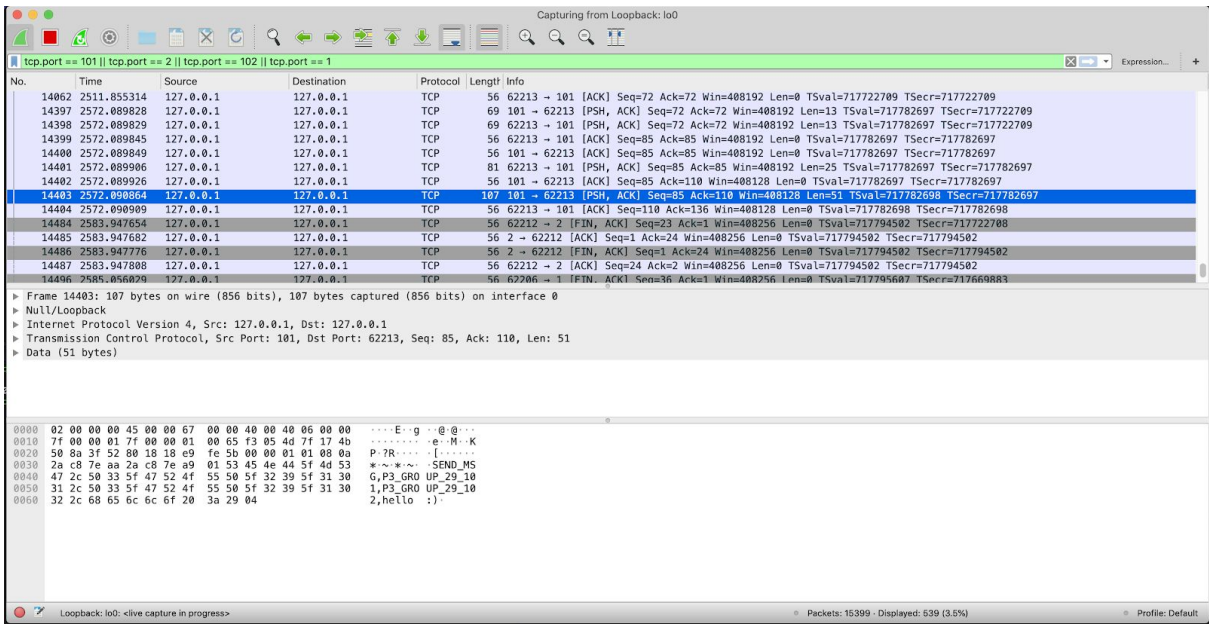
CLIENT COMMAND: SENDMSG, <TO_GROUP_ID>, <message>

The picture below shows a client sending SENDMSG, <GROUP_ID_OF_SERVER_102> to server 101 which server 101 sends to server 102.

This picture shows the SENDMSG, <GROUP_ID_OF_SERVER_102> command being sent from the client to server 101.
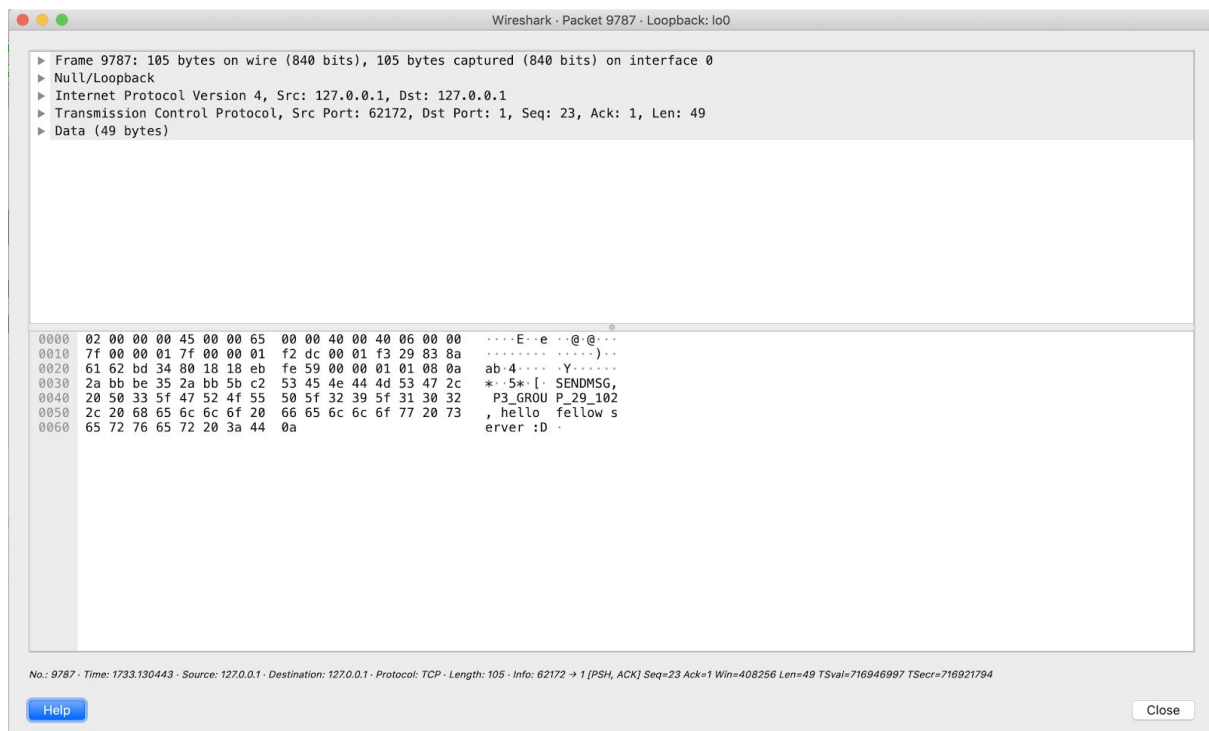
The picture below shows server 101 sending the message to 102.

Wireshark · Packet 9789 · Loopback: lo0

▶ Frame 9789: 121 bytes on wire (968 bits), 121 bytes captured (968 bits) on interface 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 62173, Dst Port: 102, Seq: 72, Ack: 72, Len: 65
▶ TPKT — ISO on TCP — RFC1006

```
0000  02 00 00 00 45 00 00 75  00 00 40 00 40 06 00 00   ····E··u ··@·@···
0010  7f 00 00 01 7f 00 00 01  f2 dd 00 66 95 c9 38 6f   ········ ···f··8o
0020  33 93 0b 00 80 18 18 ea  fe 69 00 00 01 01 08 0a   3······· ·i······
0030  2a bb be 35 2a bb 5b ca  01 53 45 4e 44 5f 4d 53   *··5*·[· ·SEND_MS
0040  47 2c 50 33 5f 47 52 4f  55 50 5f 32 39 5f 31 30   G,P3_GRO UP_29_10
0050  31 2c 50 33 5f 47 52 4f  55 50 5f 32 39 5f 31 30   1,P3_GRO UP_29_10
0060  32 2c 68 65 6c 6c 6f 20  66 65 6c 6c 6f 77 20 73   2,hello  fellow s
0070  65 72 76 65 72 20 3a 44  04                        erver :D ·
```

Help   Close

CLIENT COMMAND: GETMSG, <GROUP_ID>

The picture belows shows a client send a GETMSG, <GROUP_ID> to a server and the server sends back a single message back to the client for the given GROUP_ID.

Here we can see the GETMSG, <GROUP_ID> being sent.
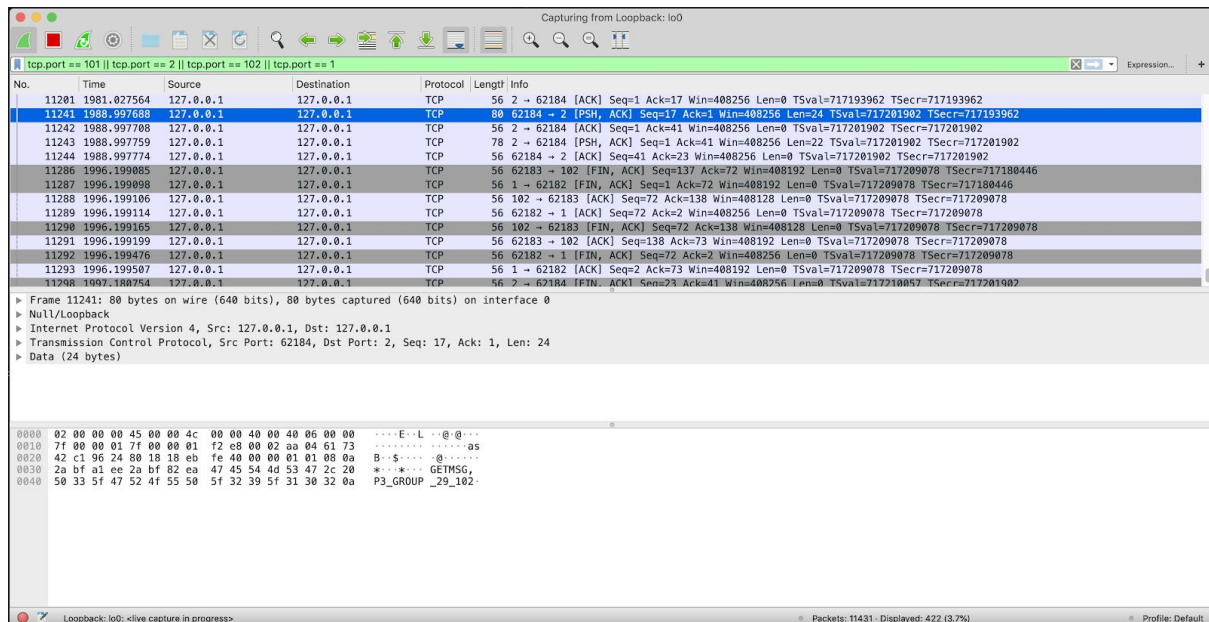
Here we can see the server sending the client a single message for the given GROUP_ID
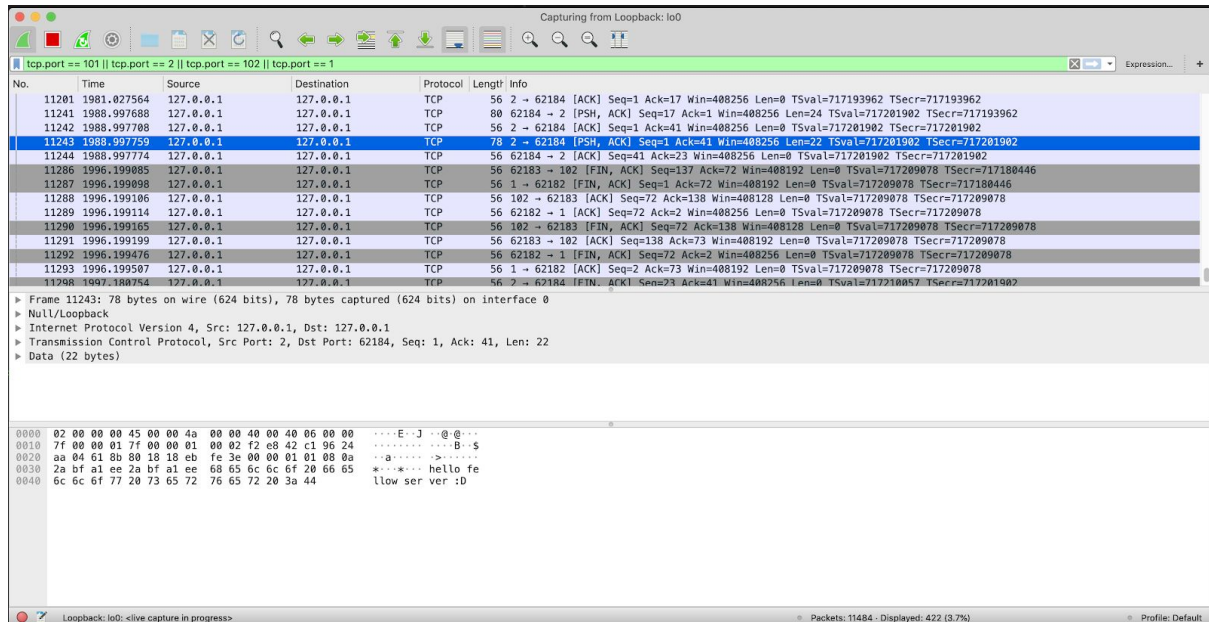
CLIENT COMMAND: LEAVESERVER,<GROUP_ID>
& SERVER COMMAND: LEAVESERVER,<IP>,<PORT>

The picture belows shows a client sending a LEAVESERVER,<GROUP_ID> to server 101
telling the server to drop connection to 102 and send him a
LEAVESERVER,<SERVER_101_IP>,<SERVER_101_PORT> command to him.
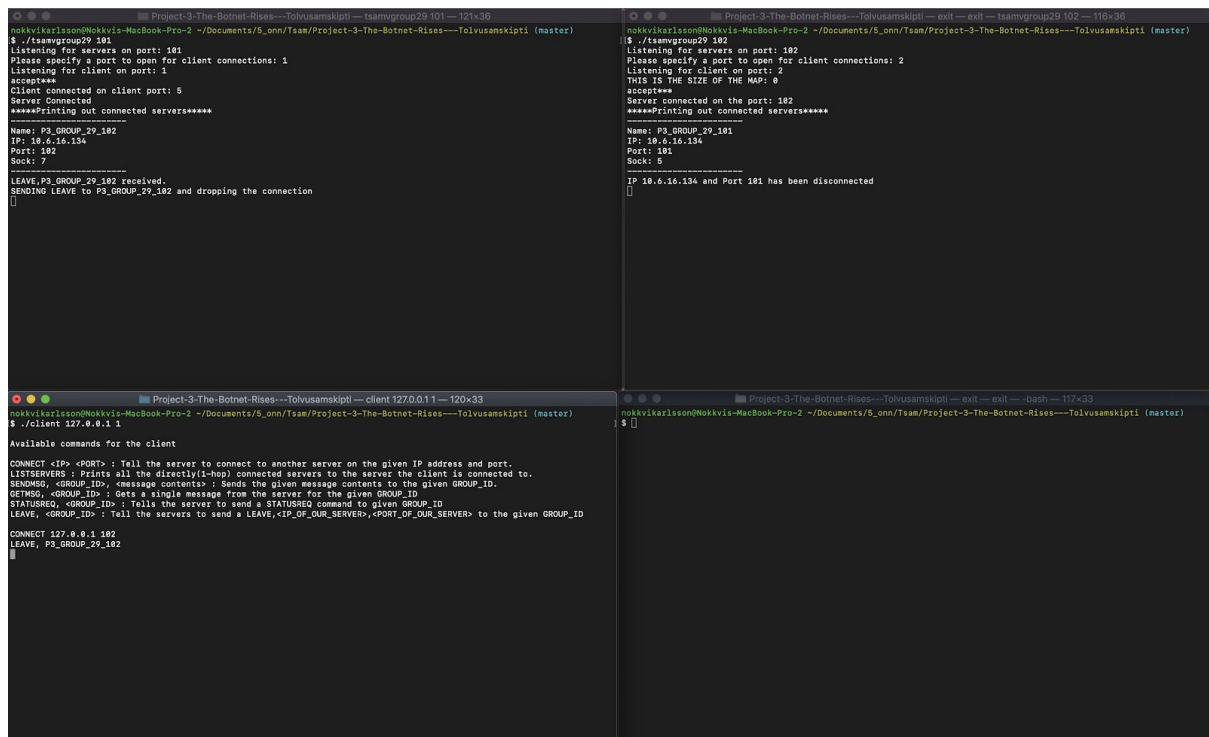
Here we can see the client sending a LEAVESERVER,<GROUP_ID>.
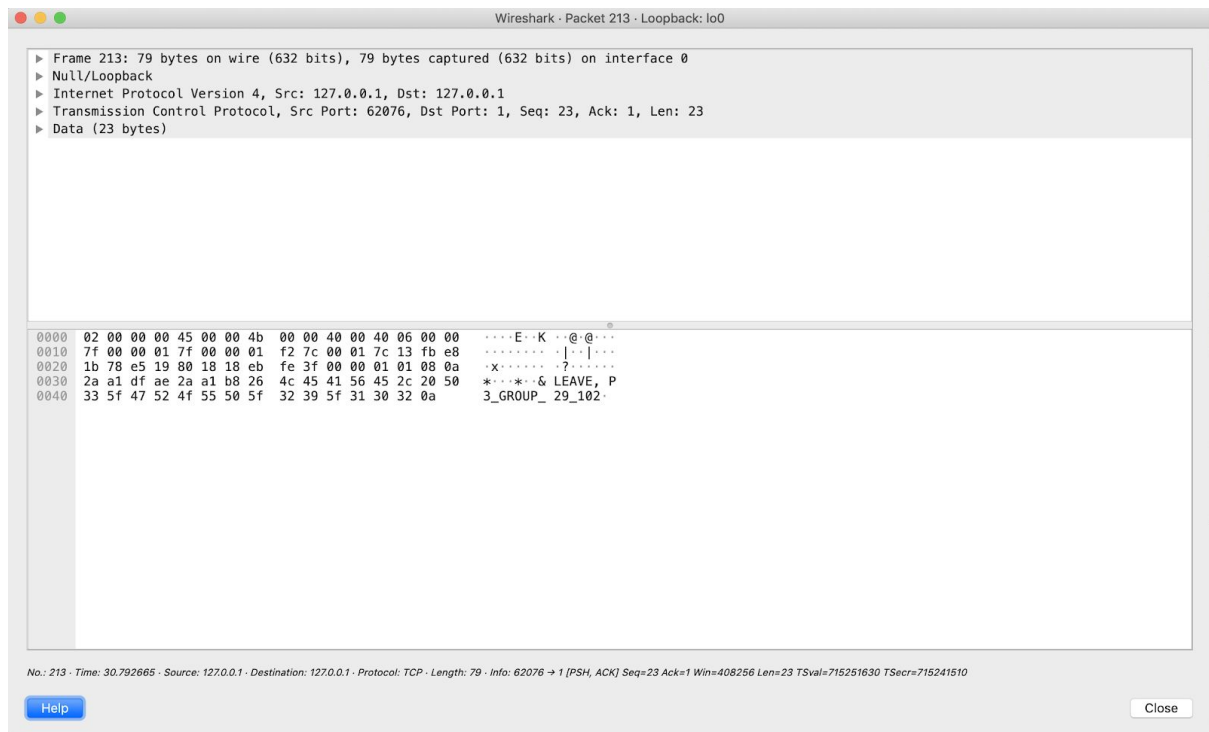
Here we can see the server sending a LEAVESERVER,<IP>,<PORT> to the other server to notify him that he has dropped the connection.
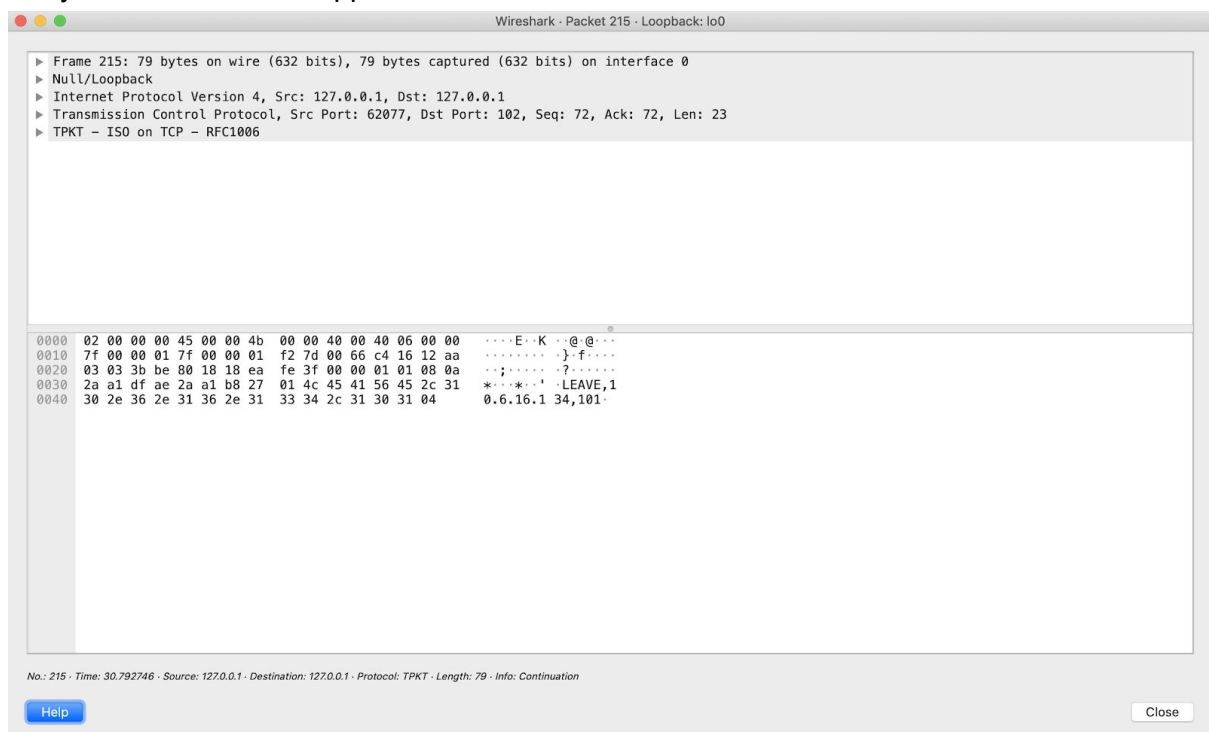
CLIENT COMMAND: STATUSREQ, <GROUP_ID>
& SERVER COMMAND: STATUSREQ,<GROUP_ID>

Here below we can see the client send a STATUSREQ, <GROUP_ID> command to server number 101 then he sends STATUSREQ,<GROUP_ID> message to the other server which is number 102. Server number 102 then responds to server number 101 with a STATUSRESP,....

Below we can see the client sending a STATUSREQ, <GROUP_ID> command to server 101.


```
●●●                        Wireshark · Packet 6612 · Loopback: lo0
▶ Frame 6612: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 62144, Dst Port: 1, Seq: 23, Ack: 1, Len: 27
▶ Data (27 bytes)




0000  02 00 00 00 45 00 00 4f  00 00 40 00 40 06 00 00   ····E··O ··@·@···
0010  7f 00 00 01 7f 00 00 01  f2 c0 00 01 63 0b 0d ef   ········ ····c···
0020  d2 e1 cb f7 80 18 18 eb  fe 43 00 00 01 01 08 0a   ········ ·C······
0030  2a b2 cc cb 2a b2 9e 22  53 54 41 54 55 53 52 45   *···*··" STATUSRE
0040  51 2c 20 50 33 5f 47 52  4f 55 50 5f 32 39 5f 31   Q, P3_GR OUP_29_1
0050  30 32 0a                                           02·

No.: 6612 · Time: 1144.541822 · Source: 127.0.0.1 · Destination: 127.0.0.1 · Protocol: TCP · Length: 83 · Info: 62144 → 1 [PSH, ACK] Seq=23 Ack=1 Win=408256 Len=27 TSval=716360907 TSecr=716348962
        Help                                                                              Close
```

Here we see server number 101 sending a STATUSREQ,<GROUP_ID> command to server number 102.


```
●●●                        Wireshark · Packet 6614 · Loopback: lo0
▶ Frame 6614: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 62145, Dst Port: 102, Seq: 72, Ack: 72, Len: 27
▶ TPKT — ISO on TCP — RFC1006




0000  02 00 00 00 45 00 00 4f  00 00 40 00 40 06 00 00   ····E··O ··@·@···
0010  7f 00 00 01 7f 00 00 01  f2 c1 00 66 ba 6f 52 7e   ········ ···f·oR~
0020  9a ed 0e 8b 80 18 18 ea  fe 43 00 00 01 01 08 0a   ········ ·C······
0030  2a b2 cc cb 2a b2 9e 23  01 53 54 41 54 55 53 52   *···*··# ·STATUSR
0040  45 51 2c 50 33 5f 47 52  4f 55 50 5f 32 39 5f 31   EQ,P3_GR OUP_29_1
0050  30 31 04                                           01·

No.: 6614 · Time: 1144.541917 · Source: 127.0.0.1 · Destination: 127.0.0.1 · Protocol: TPKT · Length: 83 · Info: Continuation
        Help                                                                              Close
```
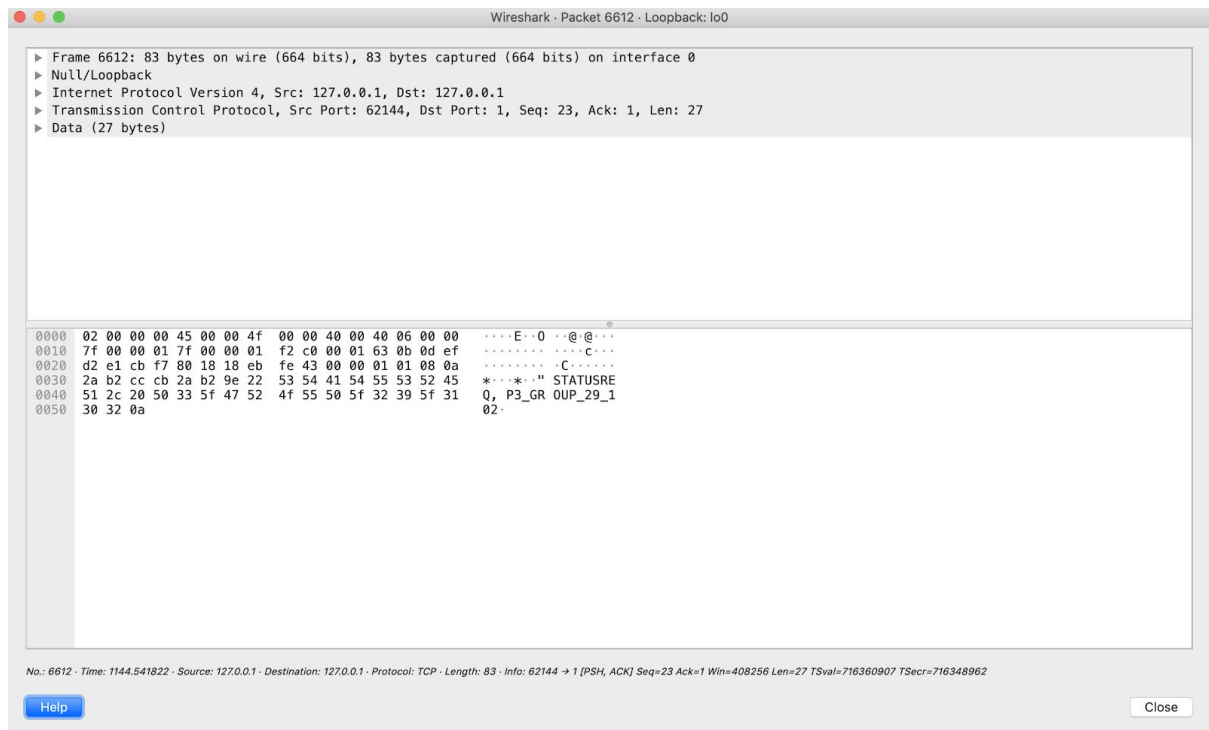
Here we see server number 102 send a STATUSRESP,... back to server number 101.

Wireshark · Packet 6616 · Loopback: lo0

▶ Frame 6616: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 102, Dst Port: 62145, Seq: 72, Ack: 99, Len: 44
▶ TPKT – ISO on TCP – RFC1006

```
0000  02 00 00 00 45 00 00 60  00 00 40 00 40 06 00 00   ····E··`  ··@·@···
0010  7f 00 00 01 7f 00 00 01  00 66 f2 c1 9a ed 0e 8b   ········  ·f······
0020  ba 6f 52 99 80 18 18 ea  fe 54 00 00 01 01 08 0a   ·oR·····  ·T······
0030  2a b2 cc cb 2a b2 cc cb  01 53 54 41 54 55 53 52   *···*···  ·STATUSR
0040  45 53 50 2c 50 33 5f 47  52 4f 55 50 5f 32 39 5f   ESP,P3_G  ROUP_29_
0050  31 30 32 2c 50 33 5f 47  52 4f 55 50 5f 32 39 5f   102,P3_G  ROUP_29_
0060  31 30 31 04                                        101·
```

No.: 6616 · Time: 1144.541992 · Source: 127.0.0.1 · Destination: 127.0.0.1 · Protocol: TPKT · Length: 100 · Info: Continuation

Help   Close