UDP HANDSHAKE

TESTING DOCUMENT

Abstract

A document which tests a simple localhost UDP handshake. Screenshots of the application and wireshark captures have been included to demonstrate the traffic of the application.

Justin Cervantes

Cervantes.jfa@gmail.com

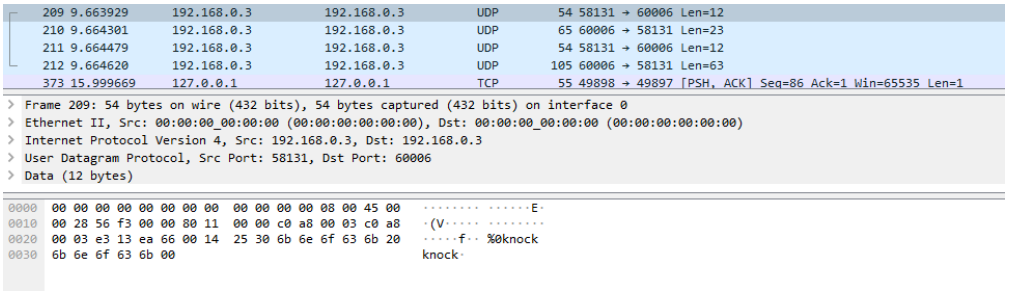
Contents

Test Cases

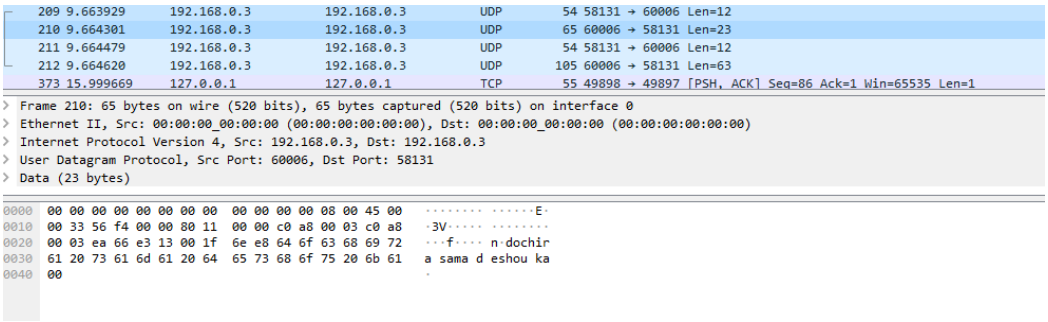
Main Scenario

Description: The user will first launch the UDP Server, then launch the UDP client. The server starts by waiting for the message “knock knock”, once received, it sends the response “dochira sama deshou ka”. The client, once it receives its reply, sends back “KanfyooSHas”, which the server will reply to with “o nkgx gtj o luxmkz o ykk gtj o xkskshkx o ju gtj o atjkxyzgtj”.
Success criteria: The only output displayed other than the introduction in the console of what each app is, should be the UDP client printing out the final part of the handshake.
Status: Passed

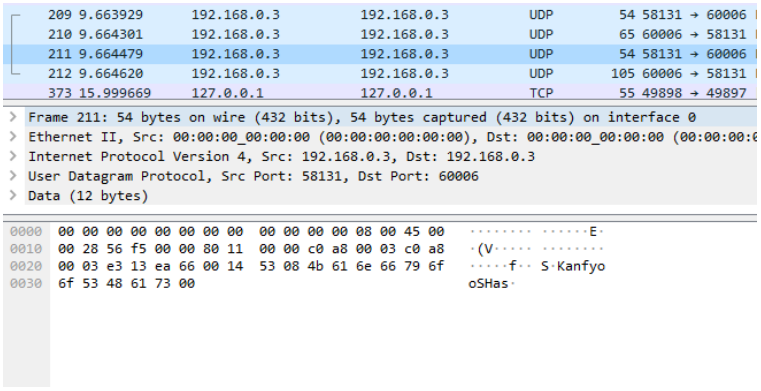
-knock knock-



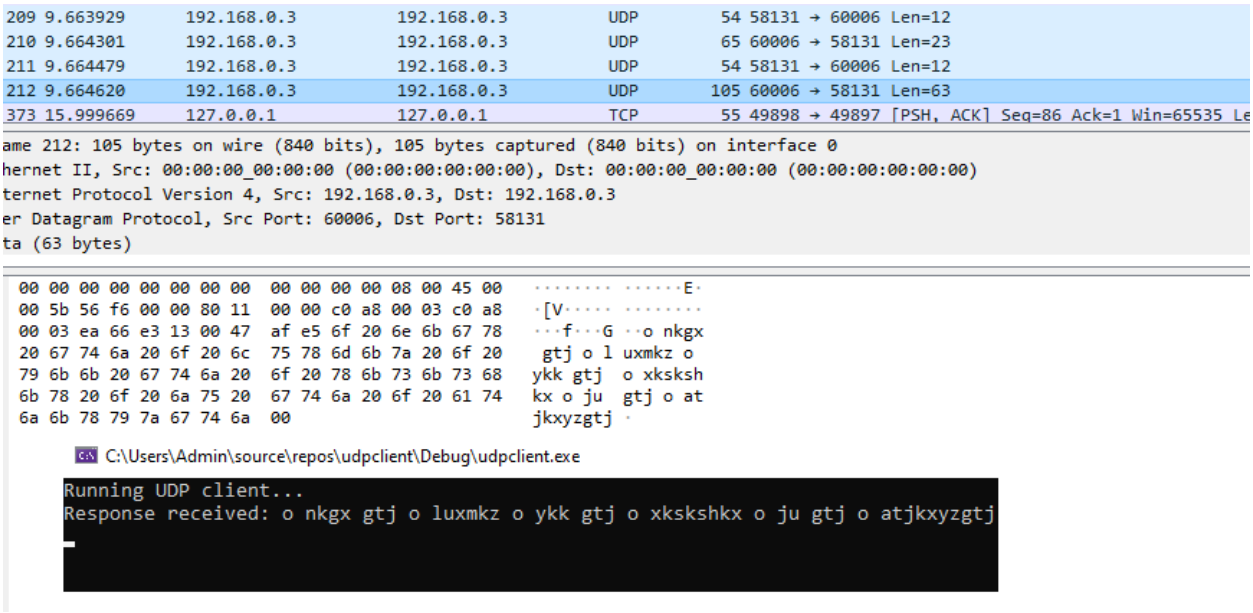
- dochira sama deshou ka-



- KanfyooSHas –



- o nkgx gtj o luxmkz o ykk gtj o xkskshkx o ju gtj o atjkxyzgtj -

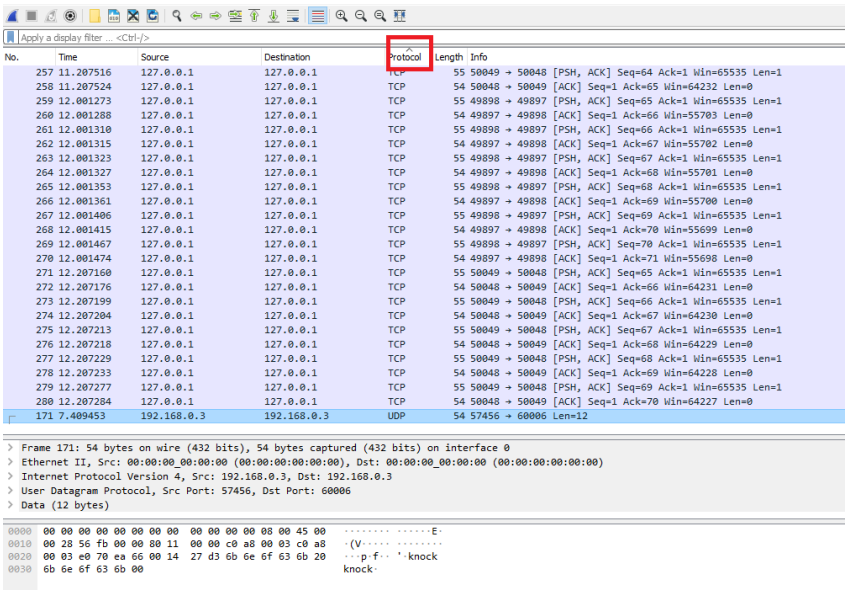


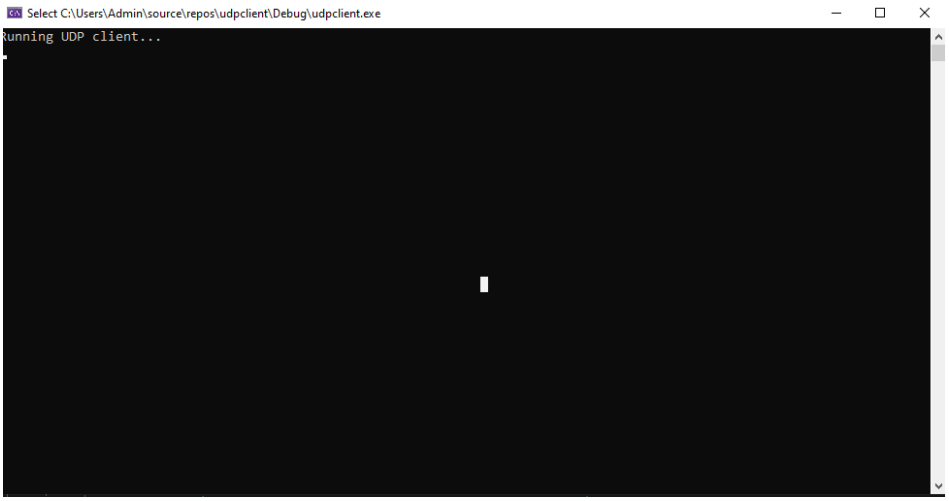
Alternate Scenario: Server not started

Description: The user will first launch the UDP Client without an active session of the UDP Server. The client will fail to start the handshake, but will send the first datagram out with “knock knock”

Success criteria: Only the header message for the UDP Client should appear.

Status: Passed



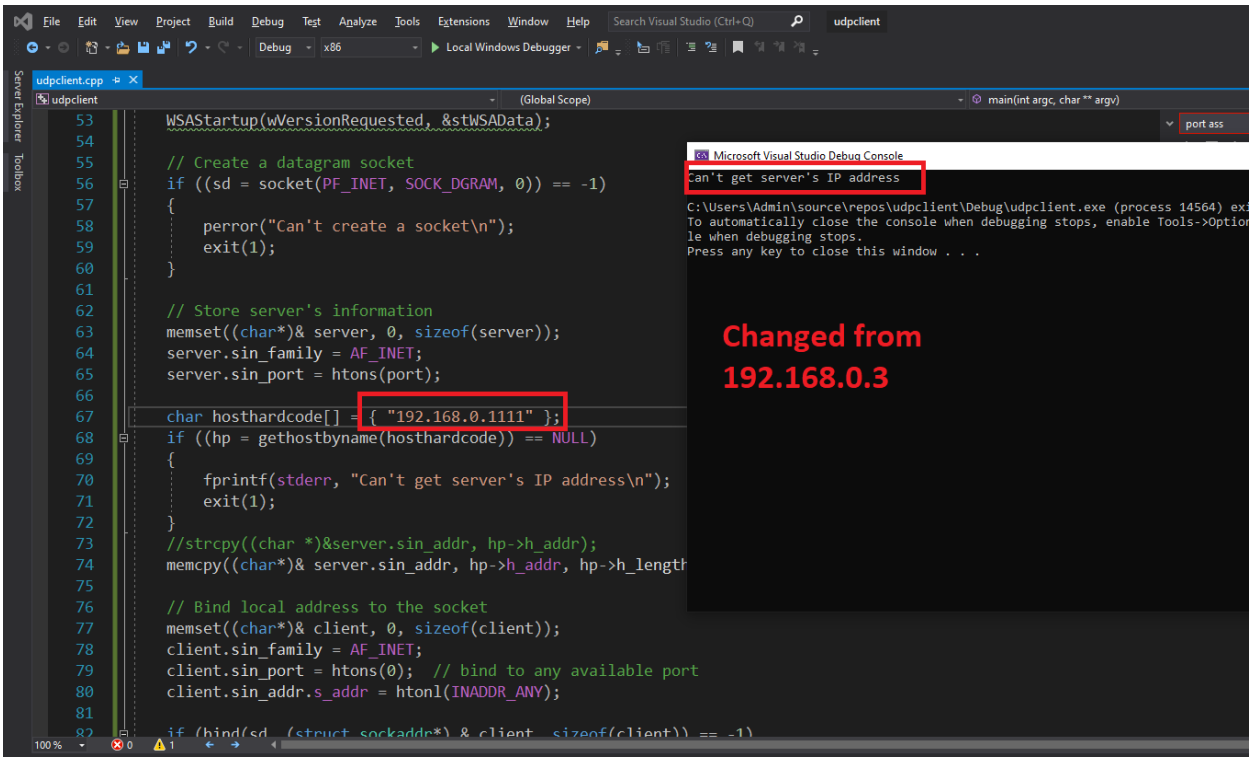


Alternate Scenario: Invalid IP

Description: If an invalid IP is sent across the network, a failure message will appear as it cannot send data out. Testing was done on a machine with IP 192.168.0.3, however the current version has a different default IP.

Success criteria: There should be no traffic to the invalid IP in wireshark

Status: Passed (the screenshot is empty as there was no UDP traffic)

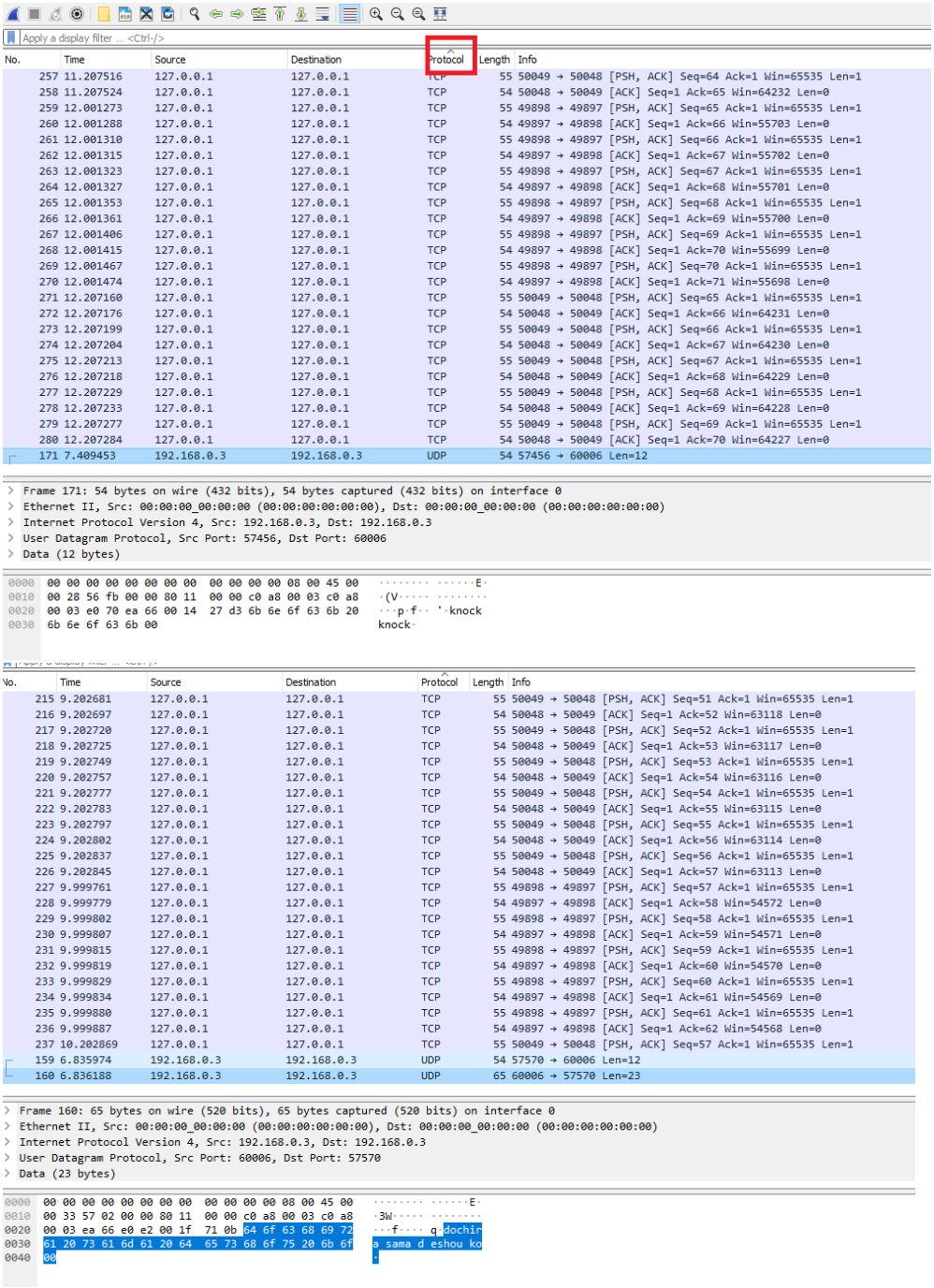


Alternate Scenario: Server does not send correct message

Description: The handshake should not complete if the correct messages are not received.

Success criteria: The handshake should end when an invalid message is received.

Status: Passed



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
257	11.207516	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=64 Ack=1 Win=65535 Len=1
258	11.207524	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=65 Win=64232 Len=0
259	12.001273	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=65 Ack=1 Win=65535 Len=1
260	12.001288	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=66 Win=55703 Len=0
261	12.001310	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=66 Ack=1 Win=65535 Len=1
262	12.001315	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=67 Win=55702 Len=0
263	12.001323	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=67 Ack=1 Win=65535 Len=1
264	12.001327	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=68 Win=55701 Len=0
265	12.001353	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=68 Ack=1 Win=65535 Len=1
266	12.001361	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=69 Win=55700 Len=0
267	12.001406	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=69 Ack=1 Win=65535 Len=1
268	12.001415	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=70 Win=55699 Len=0
269	12.001467	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=70 Ack=1 Win=65535 Len=1
270	12.001474	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=71 Win=55698 Len=0
271	12.207160	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=65 Ack=1 Win=65535 Len=1
272	12.207176	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=66 Win=64231 Len=0
273	12.207199	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=66 Ack=1 Win=65535 Len=1
274	12.207204	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=67 Win=64230 Len=0
275	12.207213	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=67 Ack=1 Win=65535 Len=1
276	12.207218	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=68 Win=64229 Len=0
277	12.207229	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=68 Ack=1 Win=65535 Len=1
278	12.207233	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=69 Win=64228 Len=0
279	12.207277	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=69 Ack=1 Win=65535 Len=1
280	12.207284	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=70 Win=64227 Len=0
171	7.409453	192.168.0.3	192.168.0.3	UDP	54	57456 → 60006 Len=12

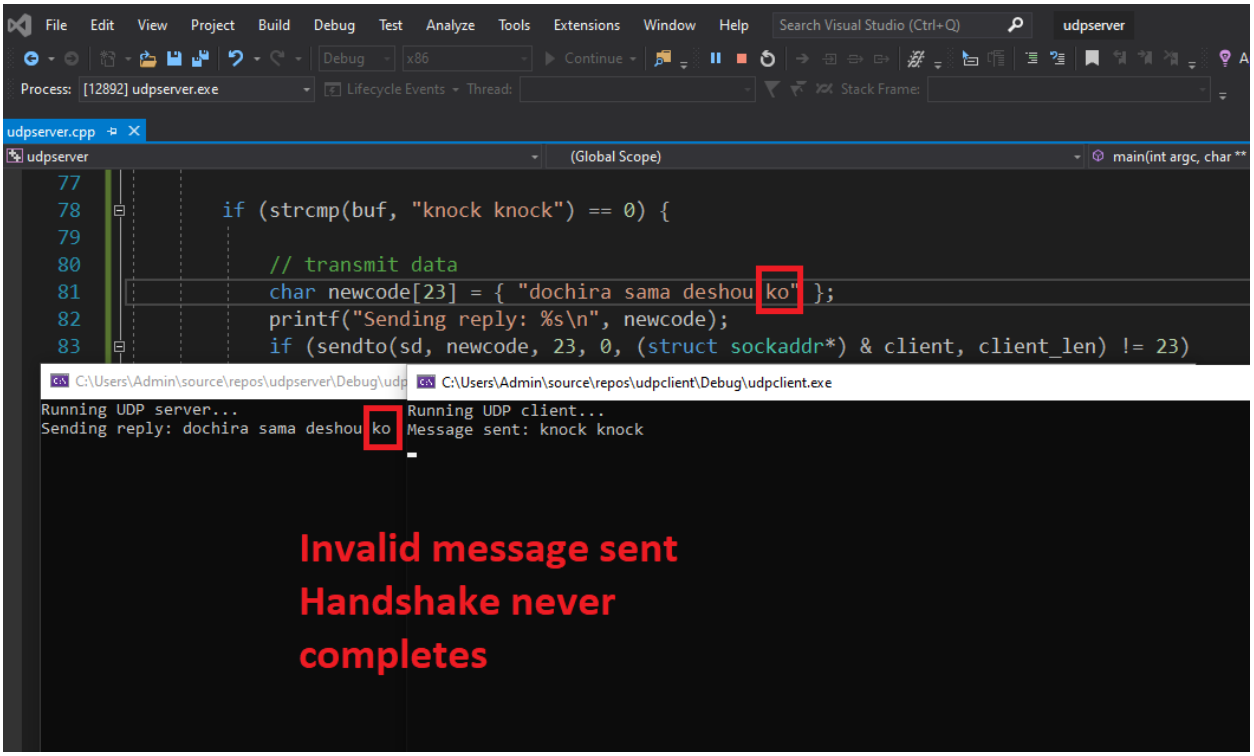
```
> Frame 171: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.0.3, Dst: 192.168.0.3
> User Datagram Protocol, Src Port: 57456, Dst Port: 60006
> Data (12 bytes)
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E·
0010  00 28 56 fb 00 00 80 11  00 00 c0 a8 00 03 c0 a8   ·(V····· ........
0020  00 03 e0 70 ea 66 00 14  27 d3 6b 6e 6f 63 6b 20   ···p·f·· '·knock
0030  6b 6e 6f 63 6b 00                                 knock·
```

No.	Time	Source	Destination	Protocol	Length	Info
215	9.202681	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=51 Ack=1 Win=65535 Len=1
216	9.202697	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=52 Win=63118 Len=0
217	9.202720	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=52 Ack=1 Win=65535 Len=1
218	9.202725	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=53 Win=63117 Len=0
219	9.202749	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=53 Ack=1 Win=65535 Len=1
220	9.202757	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=54 Win=63116 Len=0
221	9.202777	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=54 Ack=1 Win=65535 Len=1
222	9.202783	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=55 Win=63115 Len=0
223	9.202797	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=55 Ack=1 Win=65535 Len=1
224	9.202802	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=56 Win=63114 Len=0
225	9.202837	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=56 Ack=1 Win=65535 Len=1
226	9.202845	127.0.0.1	127.0.0.1	TCP	54	50048 → 50049 [ACK] Seq=1 Ack=57 Win=63113 Len=0
227	9.999761	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=57 Ack=1 Win=65535 Len=1
228	9.999779	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=58 Win=54572 Len=0
229	9.999802	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=58 Ack=1 Win=65535 Len=1
230	9.999807	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=59 Win=54571 Len=0
231	9.999815	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=59 Ack=1 Win=65535 Len=1
232	9.999819	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=60 Win=54570 Len=0
233	9.999829	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=60 Ack=1 Win=65535 Len=1
234	9.999834	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=61 Win=54569 Len=0
235	9.999880	127.0.0.1	127.0.0.1	TCP	55	49898 → 49897 [PSH, ACK] Seq=61 Ack=1 Win=65535 Len=1
236	9.999887	127.0.0.1	127.0.0.1	TCP	54	49897 → 49898 [ACK] Seq=1 Ack=62 Win=54568 Len=0
237	10.202869	127.0.0.1	127.0.0.1	TCP	55	50049 → 50048 [PSH, ACK] Seq=57 Ack=1 Win=65535 Len=1
159	6.835974	192.168.0.3	192.168.0.3	UDP	54	57570 → 60006 Len=12
160	6.836188	192.168.0.3	192.168.0.3	UDP	65	60006 → 57570 Len=23

```
> Frame 160: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface 0
> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 192.168.0.3, Dst: 192.168.0.3
> User Datagram Protocol, Src Port: 60006, Dst Port: 57570
> Data (23 bytes)
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E·
0010  00 33 57 02 00 00 80 11  00 00 c0 a8 00 03 c0 a8   ·3W····· ........
0020  00 03 ea 66 e0 e2 00 1f  71 0b 64 6f 63 68 69 72   ···f···· q·dochir
0030  61 20 73 61 6d 61 20 64  65 73 68 6f 75 20 6b 6f   a sama d eshou ko
0040  00                                                ·
```


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
udpserver
Debug
x86
Continue
Process: [12892] udpserver.exe
Lifecycle Events
Thread:
Stack Frame:
udpserver.cpp
udpserver
(Global Scope)
main(int argc, char **
77
78 if (strcmp(buf, "knock knock") == 0) {
79
80 // transmit data
81 char newcode[23] = { "dochira sama deshou ko" };
82 printf("Sending reply: %s\n", newcode);
83 if (sendto(sd, newcode, 23, 0, (struct sockaddr*) & client, client_len) != 23)
C:\Users\Admin\source\repos\udpserver\Debug\udp
C:\Users\Admin\source\repos\udpclient\Debug\udpclient.exe
Running UDP server...
Sending reply: dochira sama deshou ko
Running UDP client...
Message sent: knock knock
Invalid message sent
Handshake never
completes