

# **מטלה 2 - רשות תקשורת**

**מגישים:**

**- שלומי זכריה, 315141242,**

**- יוסיין כהן, 212733836,**

# תרגיל 1

נסביר איך השלמנו את החלקים החסרים בקובץ `server.py`  
(הסבירים בתוך הצלומי מסך)

\* נסיף, כאשר קראנו ל-(IP,PORT) איז בעם מילאנו את :

`AF_INET = IPV4 , SOCK_STREAM = TCP PORT`

```
proxy.py
80 def server(host: str, port: int) -> None:
81     # socket.socket.AF_INET, socket.SOCK_STREAM
82     # (1) AF_INET is the address family for IPv4 (Address Family)
83     # (2) SOCK_STREAM is the socket type for TCP (Socket Type) - [SOCK_DGRAM is the socket type for UDP]
84     # Note: context manager ('with' keyword) closes the socket when the block is exited
85     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server_socket:
86         # SO_REUSEADDR is a socket option that allows the socket to be bound to an address that is already in use.
87         server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
88
89         # Prepare the server socket
90         # * Fill in start (1)
91         server_socket.bind((host, port)) # set(defult) ip and port
92         server_socket.listen() # the server is ready to receive requests
93         # * Fill in end (1)
94
95         threads = []
96         print(f"Listening on [{host}]:{port}")
97
98         while True:
99             try:
100                 # Establish connection with client.
101                 # * Fill in start (2)
102                 client_socket, address = server_socket.accept()
103                 # * Server accepts the client
104                 # * Fill in end (2)
105
106                 # Create a new thread to handle the client request
107                 thread = threading.Thread(target=client_handler, args=(client_socket, address))
108                 thread.start()
109                 threads.append(thread)
110
111             except KeyboardInterrupt:
112                 print("Shutting down...")
113                 break
114
115
116             for thread in threads: # Wait for all threads to finish
117                 thread.join()
118
119
120 def client_handler(client_socket: socket.socket, client_address: tuple[str, int]) -> None:
121     """
122     Function which handles client requests
123     """
124     Client_addr = f'{client_address[0]}:{client_address[1]}'
125     Client_prefix = f'{{{{client_addr}}}}'
126     with client_socket: # closes the socket when the block is exited
127         print(f"Connection established with {client_addr}")
128         while True:
129
130             data = client_socket.recv(api.BUFFER_SIZE) # * Fill in start (3) # * Fill in end (3)
131             if not data:
132                 break
133
134             try:
135                 request = api.CalculatorHeader.unpack(data)
136             except Exception as e:
137                 raise api.CalculatorClientError(
138                     f"Error while unpacking request: {e}") from e
139
140             print(f"{Client_prefix} Got request of length {len(data)} bytes")
141
142             response = process_request(request)
143             response = response.pack()
144             print(f"{Client_prefix} Sending response of length {len(response)} bytes")
145
146             try:
147                 client_socket.sendall(response) # sends the server's response (bytes)
148             except Exception as e:
149                 print(f"Unexpected server error: {e}")
150                 client_socket.sendall(api.CalculatorHeader.STATUS_SERVER_ERROR, CACHE_POLICY, CACHE_CONTROL).pack()
151
152             # * Fill in end (4)
153
154         except Exception as e:
155             print(f"Unexpected server error: {e}")
156             client_socket.sendall(api.CalculatorHeader.STATUS_SERVER_ERROR, CACHE_POLICY, CACHE_CONTROL).pack()
157
158         print(f"{Client_prefix} Connection closed")
```

server.py

```
107         # Create a new thread to handle the client request
108         thread = threading.Thread(target=client_handler, args=(client_socket, address))
109         thread.start()
110         threads.append(thread)
111
112     except KeyboardInterrupt:
113         print("Shutting down...")
114
115
116     for thread in threads: # Wait for all threads to finish
117         thread.join()
118
119
120 def client_handler(client_socket: socket.socket, client_address: tuple[str, int]) -> None:
121     """
122     Function which handles client requests
123     """
124     Client_addr = f'{client_address[0]}:{client_address[1]}'
125     Client_prefix = f'{{{{client_addr}}}}'
126     with client_socket: # closes the socket when the block is exited
127         print(f"Connection established with {client_addr}")
128         while True:
129
130             data = client_socket.recv(api.BUFFER_SIZE) # * Fill in start (3) # * Fill in end (3)
131             if not data:
132                 break
133
134             try:
135                 request = api.CalculatorHeader.unpack(data)
136             except Exception as e:
137                 raise api.CalculatorClientError(
138                     f"Error while unpacking request: {e}") from e
139
140             print(f"{Client_prefix} Got request of length {len(data)} bytes")
141
142             response = process_request(request)
143             response = response.pack()
144             print(f"{Client_prefix} Sending response of length {len(response)} bytes")
145
146             try:
147                 client_socket.sendall(response) # sends the server's response (bytes)
148             except Exception as e:
149                 print(f"Unexpected server error: {e}")
150                 client_socket.sendall(api.CalculatorHeader.STATUS_SERVER_ERROR, CACHE_POLICY, CACHE_CONTROL).pack()
151
152             # * Fill in end (4)
153
154         except Exception as e:
155             print(f"Unexpected server error: {e}")
156             client_socket.sendall(api.CalculatorHeader.STATUS_SERVER_ERROR, CACHE_POLICY, CACHE_CONTROL).pack()
157
158         print(f"{Client_prefix} Connection closed")
```

Annotations for proxy.py:

- :`server_socket.bind((host, port))` נגיד שבודת או בוגת ה-IP,אות ה-port (לפי ה-IP)
- :`server_socket.listen(1)` השות מאושר את התקשות עם הלקוח
- :`client_socket, address = server_socket.accept()` השות מאושר את התקשות עם הלקוח

Annotations for server.py:

- :`data = client_socket.recv(api.BUFFER_SIZE)` נגיד את הגודל המקסימלי של DATA
- :`api.BUFFER_SIZE = 2^16 = 65536` במקרה שלו :
- :`client_socket.sendall(response)` נשלוח את ה- response בBITSIM , לאחר שהוא נארז בפקעות

## נסביר איך השלמנו את החלקים החסריים בקובץ ע.ד proxy (הסבירים בתוך הצלומי מס' 1)

```

69     # Cache the response if all sides agree to cache it
70     if request.cache_result and response.cache_result and (server_time_remaining > 0 and client_time_remaining > 0):
71         cached[(data, request.show_steps)] = response
72         cached = True
73
74     return response, server_time_remaining, client_time_remaining, False, was_stale, cached
75
76
77 def proxy(proxy_address: tuple[str, int], server_address: tuple[str, int]) -> None:
78     # socket(AF_INET, socket.SOCK_STREAM)
79     # (1) AF_INET is the address family for IPv4 (Address Family)
80     # (2) SOCK_STREAM is the socket type for TCP (Socket Type) . [SOCK_DGRAM is the socket type for UDP]
81     # Note: the parameter (socket_type) does not exist in the block is exited
82     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as proxy_socket:
83         # SO_REUSEADDR is a socket option that allows the socket to be bound to an address that is already in use.
84         proxy_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
85
86         # Prepare the proxy socket
87         # * Fill in start (1)
88         proxy_socket.bind(proxy_address) # defines an ip address and port number for a socket instance
89         proxy_socket.listen()          # * Fill in end (1)
90
91         threads = []
92         print(f"Listening on {proxy_address[0]}:{proxy_address[1]}")
93
94         while True:
95             try:
96                 # Establish connection with client.
97                 # * Fill in start (2)
98                 client_socket, client_address = proxy_socket.accept() # proxy accepts the client
99                 # * Fill in end (2)
100
101                 # Create a new thread to handle the client request
102                 thread = threading.Thread(target=client_handler, args=(client_socket, client_address, server_address))
103                 thread.start()
104                 threads.append(thread)
105
106             except KeyboardInterrupt:
107                 print("Shutting down...")
108                 break
109
110             for thread in threads: # Wait for all threads to finish
111                 thread.join()
112
113
114 def client_handler(client_socket: socket.socket, client_address: tuple[str, int], server_address: tuple[str, int]) -> None:
115     """
116     Function which handles client requests
117     """
118     client_prefix = f'{client_address[0]}:{client_address[1]}'
119     with client_socket: # closes the socket when the block is exited
120         print(f"[{client_prefix}] Connected established")
121         while True:
122             # Receive data from the client
123             # * Fill in start (3)
124             data = client_socket.recv(api.BUFFER_SIZE) # data size : BUFFER_SIZE = 65536 = (2^16)
125             # * Fill in end (3)
126
127             if not data:
128                 break
129
130             try:
131                 # Process the request
132                 request = api.CalculatorHeader.unpack(data)
133             except Exception as e:
134                 raise api.CalculatorClientError(
135                     f"Error while unpacking request: ({e}) from {client_prefix}")
136
137             print(f"[{client_prefix}] Got request of length {len(data)} bytes")
138
139             response, server_time_remaining, client_time_remaining, cache_hit, was_stale, cached = process_request(
140                 request, server_address)
141
142             if cache_hit:
143                 print(f"[{client_prefix}] Cache hit., end={end}")
144             elif was_stale:
145                 print(f"[{client_prefix}] Cache miss, stale response., end={end}")
146             elif cached:
147                 print(f"[{client_prefix}] Cache miss, response cached., end={end}")
148             else:
149                 print(f"[{client_prefix}] Cache miss, response not cached., end={end}")
150
151             if server_time_remaining: (server_time_remaining:.2f), client_time_remaining: (client_time_remaining:.2f))
152
153             response = response.pack()
154             print(f"[{client_prefix}] Sending response of length {len(response)} bytes")
155
156             # Send the response back to the client
157             # * Fill in start (4)
158             client_socket.sendall(response)
159             # * Fill in end (4)
160
161             except Exception as e:
162                 print(f"Unexpected server error: ({e})")
163                 client_socket.sendall(api.CalculatorHeader.from_error(api.CalculatorServerError(
164                     "Internal proxy error", e), api.CalculatorHeader.STATUS_SERVER_ERROR, False, 0).pack())
165
166             print(f"[{client_prefix}] Connection closed")
167
168
169         if name == '__main__':
170             arg_parser = argparse.ArgumentParser(
171                 description='A Calculator Server.')
172             arg_parser.add_argument('local_ip', type=str, default='localhost')
173             arg_parser.add_argument('port', type=int, default=5000)
174
175             args = arg_parser.parse_args()
176
177

```

הפונקציה מקבלת את ה-IP,PORT bind(proxy\_address)  
כלומר הוא מגדיר את השותה- bind(proxy\_address)  
ורדר(proxy\_address)  
נמלא את השותה:  
(AF\_INET(IPV4),SOCK\_STREAM(TCP))  
proxy\_socket.listen(1)  
הפרקטי "מאנז" לבקשתו, הפרקטי מוכן לקבל בקשה  
הפרקטי מאשר את התקשרות עם הלוקה

```

122     while True:
123         # Receive data from the client
124         # * Fill in start (3)
125         data = client_socket.recv(api.BUFFER_SIZE) # data size : BUFFER_SIZE = 65536 = (2^16)
126         # * Fill in end (3)
127
128         if not data:
129             break
130
131         try:
132             # Process the request
133             request = api.CalculatorHeader.unpack(data)
134         except Exception as e:
135             raise api.CalculatorClientError(
136                 f"Error while unpacking request: ({e}) from {client_prefix}")
137
138         print(f"[{client_prefix}] Got request of length {len(data)} bytes")
139
140         response, server_time_remaining, client_time_remaining, cache_hit, was_stale, cached = process_request(
141             request, server_address)
142
143         if cache_hit:
144             print(f"[{client_prefix}] Cache hit., end={end}")
145         elif was_stale:
146             print(f"[{client_prefix}] Cache miss, stale response., end={end}")
147         elif cached:
148             print(f"[{client_prefix}] Cache miss, response cached., end={end}")
149         else:
150             print(f"[{client_prefix}] Cache miss, response not cached., end={end}")
151
152         if server_time_remaining: (server_time_remaining:.2f), client_time_remaining: (client_time_remaining:.2f))
153
154         response = response.pack()
155         print(f"[{client_prefix}] Sending response of length {len(response)} bytes")
156
157         # Send the response back to the client
158         # * Fill in start (4)
159         client_socket.sendall(response)
160         # * Fill in end (4)
161
162         except Exception as e:
163             print(f"Unexpected server error: ({e})")
164             client_socket.sendall(api.CalculatorHeader.from_error(api.CalculatorServerError(
165                 "Internal proxy error", e), api.CalculatorHeader.STATUS_SERVER_ERROR, False, 0).pack())
166
167         print(f"[{client_prefix}] Connection closed")
168
169
170         if name == '__main__':
171             arg_parser = argparse.ArgumentParser(
172                 description='A Calculator Server.')
173             arg_parser.add_argument('local_ip', type=str, default='localhost')
174             arg_parser.add_argument('port', type=int, default=5000)
175
176             args = arg_parser.parse_args()
177
178

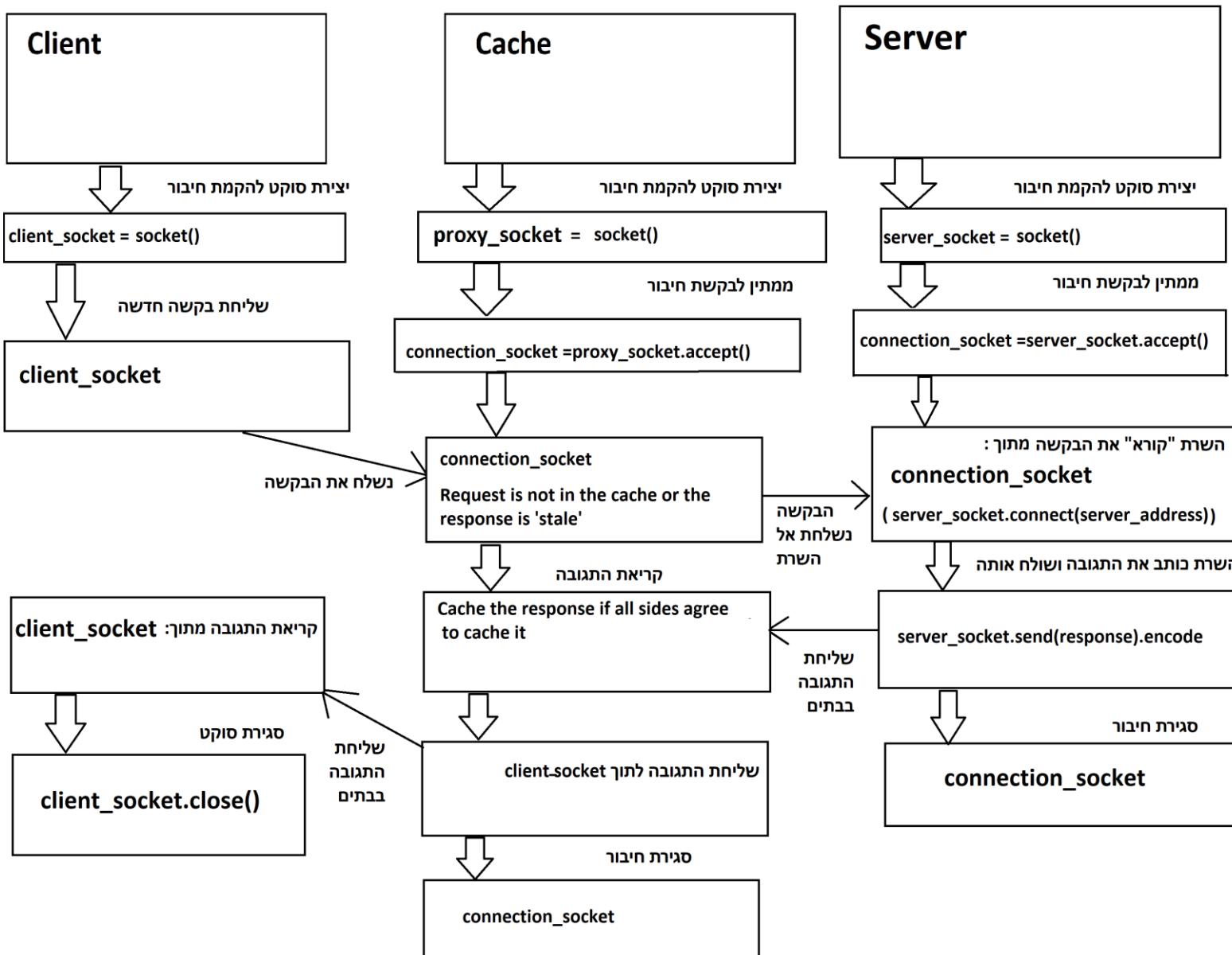
```

:data = client\_socket.recv(api.BUFFER\_SIZE)  
נדמי את גודל DATAה api.BUFFER\_SIZE=65536

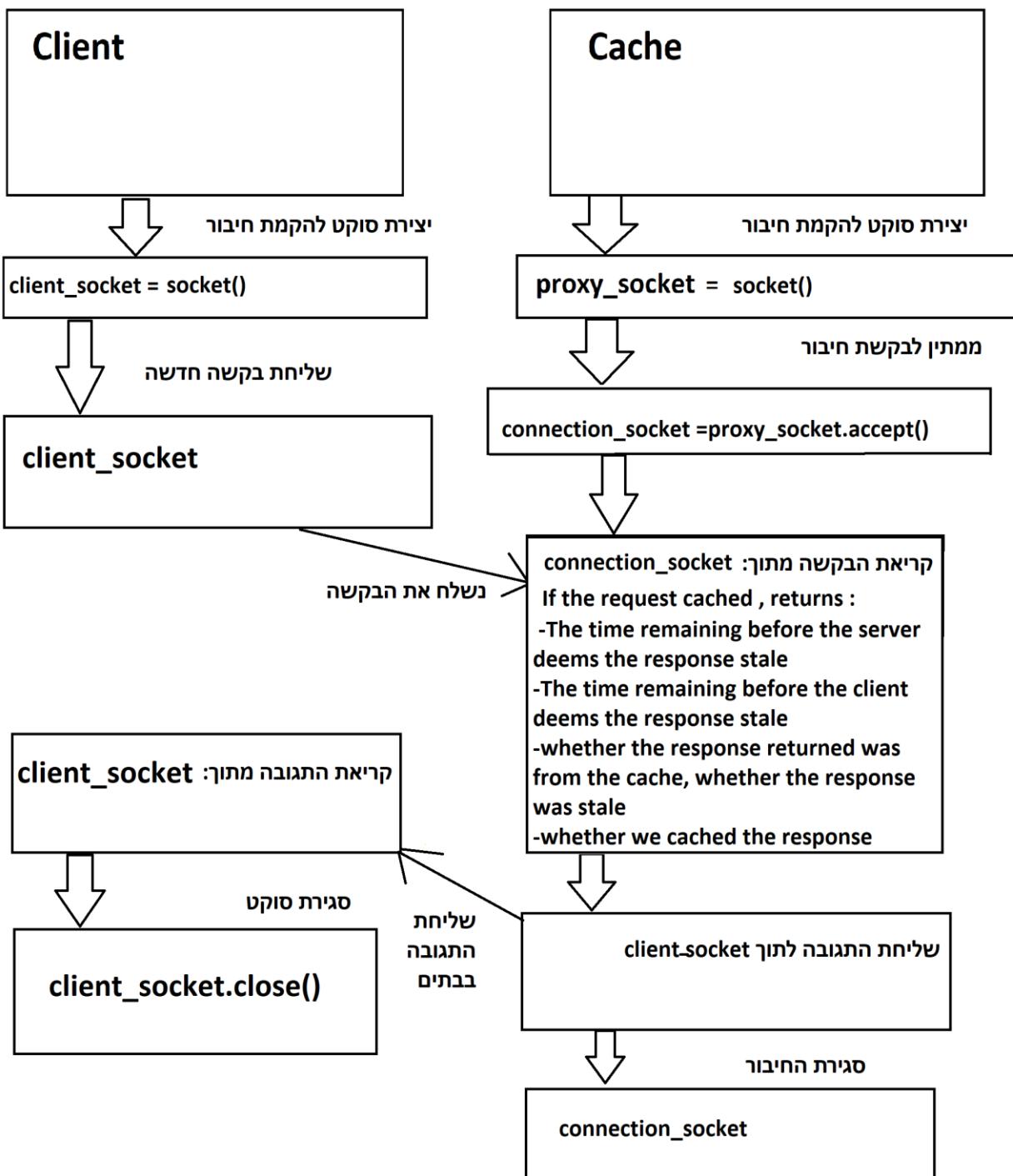
:client\_socket.sendall(response)  
לאחר שארנו את התגובה בפקטור, נשלח אותה בבייטים אל הלוקו

## תרגיל 2

### Cache System (The object is not cached)

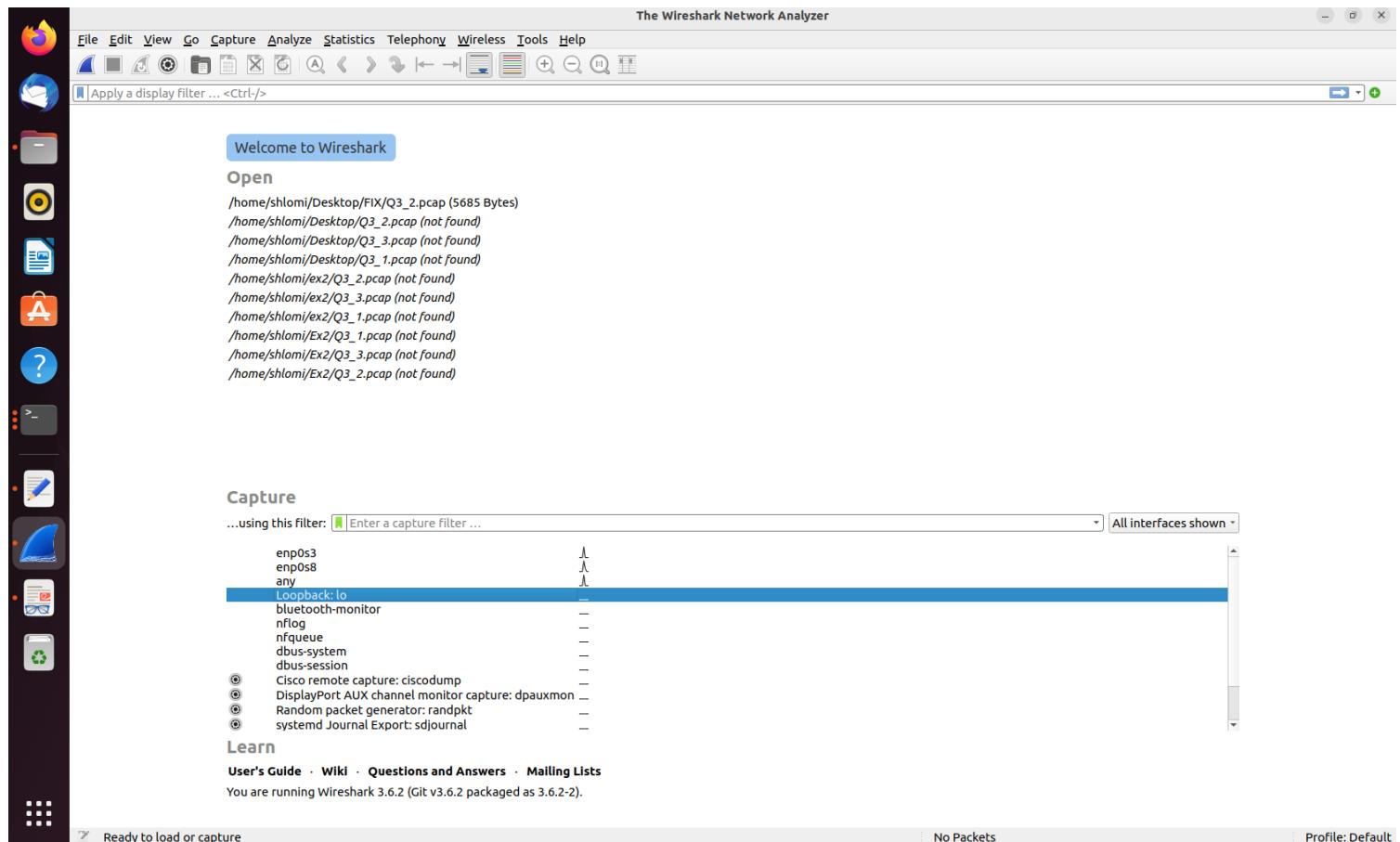


# Cache System (The object is cached)



## תרגיל 3

תחילה נכנס ל LoopBack וنبחר את הממשק או WireShark



לאחר מכן נבצע את ה filter הבא :

No.	Time	Source	Destination	Protocol	Length	Info	SYN	ACK	PSH	FIN	RESET
<pre>[tcp.port == 9999    tcp.port == 9998]</pre>											

\*נעשה filter שיסנן לנו את הפקודות הרלוונטיות : (server Port=9999,proxyPort=9998)

כדי להראות הבנה בחומר נסביר את המושגים הבאים:

- "Sequence Number" (בקיצור- seq) :

אנחנו מצפים מפרוטוקול TCP שהמידע יגיע בסדר הנכון, הרי התפקיד של שכבה התעבורה הוא לפרק את המידע לSIGMENTIM (בבתים) משכבה האפליקציה ולהעביר את המידע אל השכבות הבאות ואז המידע נשלח.

המידע שנשלח בסיגמנטים יכול להגיע לא טוב, כלומר הבטים שנשלחים יכולים להשלח לא לפי הסדר, וכך החבילה לא תגיע בשמורה.

לכן בפרוטוקול TCP כאשר המידע נשלח נוכל לדעת מה המספר הסידורי של הבית(המקום שלו), ואז נוכל לדאוג שהמידע ישלח לפי סדר של מספור הבטים.

- "ACKnowledgement" (בקיצור- ACK) :

הרי המידע מגיע ברכף מסודר של בתים, אז ACK מציין את המספר הסידורי של הבית הבא שמצופה להתקבל וכך המידע התקבל באופן תקין, המקבל (receiver) מקבל את seq מהשלוח (sender) וכדי לדעת מאיפה להתחילה את קבלת הבטים הבאים איז המქבל שולח את seq הנוכחי של השולח + גודל המידע בתים זהה בעצם המספר של ה-ACK (נסיק מכך שיש שני stream של מידע, רצף של בתים משנה הצדדים).

- "SYNchronize" (בקיצור- SYN) :

SYN היא החבילה הראשונה ביצירת הקשר, הלקוח שולח לשרת חיבור שבה הוא מבקש ליצור אליו קשר, המכילה את המספר הסידורי ההתחלתי(seq) ליצירת קשר.

\*את השאר נסביר בהמשך :)

## את הבקשות משלים בקובץ client.py (מסומן במסגרת אדומה)

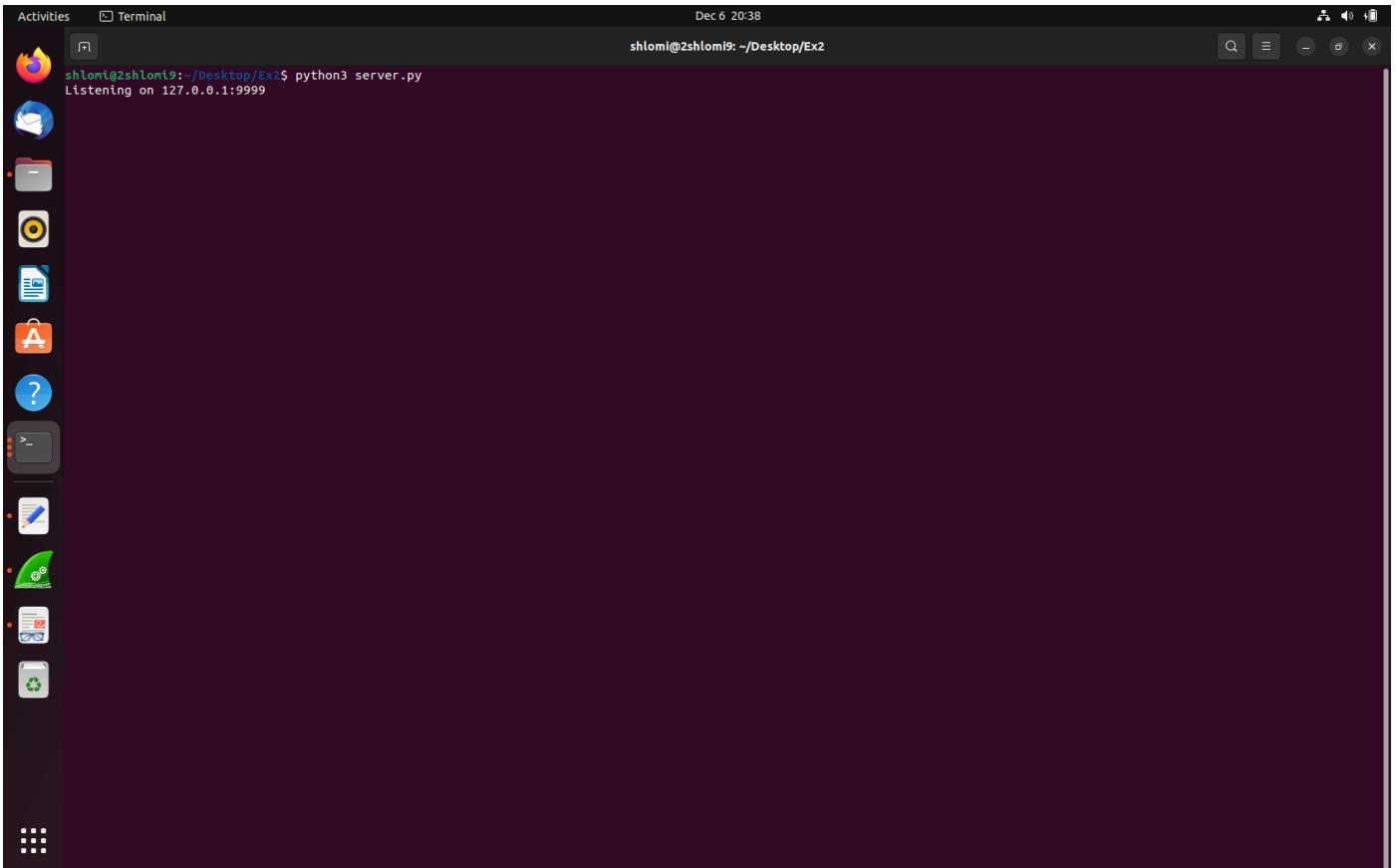
```
Activities Text Editor Dec 6 20:34
client.py
~/Desktop/Ex2
Save
client.py
proxy.py
server.py

79
80     except api.CalculatorError as e:
81         print(f'{server_prefix} Got error: {str(e)}')
82     except Exception as e:
83         print(f'{server_prefix} Unexpected error: {str(e)}')
84     print(f'{server_prefix} Connection closed')
85
86
87 if __name__ == "__main__":
88     arg_parser = argparse.ArgumentParser(description="A Calculator Client.")
89
90     arg_parser.add_argument("-p", "--port", type=int,
91                           default=api.DEFAULT_SERVER_PORT, help="The port to connect to.")
92     arg_parser.add_argument("-H", "--host", type=str,
93                           default=api.DEFAULT_SERVER_HOST, help="The host to connect to.")
94
95     args = arg_parser.parse_args()
96
97     host = args.host
98     port = args.port
99
100    # Example expressions: (uncomment one of them for your needs)
101    # (1) '(sin(max2, 3 * 4, 5, 6 * ((7 * 8) / 9), 10 / 11)) / 12) * 13' = -0.38748277824137206
102    #expr = mul_b(div_b(sin_f(max_f(2, mul_b(3, 4), 5, mul_b(
103        # 6, div_b(mul_b(7, 8), 9)), div_b(10, 11))), 12), 13) # (1)
104
105    # (2) '(max2, 3) + 3)' = 6
106    #expr = add_b(max_f(2, 3), 3) # (2)
107
108    # (3) '3 + ((4 * 2) / ((1 - 5) ** (2 ** 3)))' = 3.0001220703125
109    # expr = add_b(3, div_b(mul_b(4, 2), pow_b(sub_b(1, 5), pow_b(2, 3)))) # (3)
110
111    # (4) '((1 + 2) ** (3 * 4)) / (5 * 6)' = 17714.7
112    #expr = div_b(pow_b(add_b(1, 2), mul_b(3, 4)), mul_b(5, 6)) # (4)
113
114    # (5) '-(-(1 + (2 + 3)) ** -(4 + 5))' = 9.92290301275212e-08
115    # expr = neg_u(neg_u(pow_b(add_b(1, add_b(2, 3)), neg_u(add_b(4, 5)))))) # (5)
116
117    # (6) 'max(2, (3 * 4), log(e), (6 * 7), (9 / 8))' = 42
118    # expr = max_f(2, mul_b(3, 4), log_f(e_c), mul_b(6, 7), div_b(9, 8)) # (6)
119
120    # Change the following values according to your needs:
121
122    show_steps = True # Request the steps of the calculation
123    cache_result = True # Request to cache the result of the calculation
124    # If the result is cached, this is the maximum age of the cached response
125    # that the client is willing to accept (in seconds)
126    cache_control = 2**16 - 1
127
128    client((host, port), expr, show_steps,
129            cache_result, cache_control)
```

Python 2 Tab Width: 8 Ln 112, Col 5 INS

## תרגיל 3.1 :

בזמן שה-Wireshark מבצע הסנהה, נריץ את השרת(`python3 server.py`)



נשלח אותה בקשה פעמיים מהלקוח (`python3 client.py`)

```
shlomi@2shlomi9:~/Desktop/Ex2$ python3 client.py
address : ('127.0.0.1', 9999)
[127.0.0.1:9999] Connection established
[127.0.0.1:9999] Sending request of length 687 bytes
[127.0.0.1:9999] Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.333333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02980636755702862 * 13
= -0.38748277824137206

[127.0.0.1:9999] Connection closed
shlomi@2shlomi9:~/Desktop/Ex2$ python3 client.py
address : ('127.0.0.1', 9999)
[127.0.0.1:9999] Connection established
[127.0.0.1:9999] Sending request of length 687 bytes
[127.0.0.1:9999] Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.333333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02980636755702862 * 13
= -0.38748277824137206

[127.0.0.1:9999] Connection closed
shlomi@2shlomi9:~/Desktop/Ex2$
```

## לאחר מכון נערך את הבקשה בקובץ `client.py` כמו שהציגנו למעלה ונשלח בקשה חדשה אל השרת

```
shlomi@zshlomi9:~/Desktop/Ex$ python3 client.py
address : ('127.0.0.1', 9999)
[127.0.0.1:9999] Connection established
[127.0.0.1:9999] Sending request of length 687 bytes
[127.0.0.1:9999] Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.33333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02980636755702862 * 13
= -0.38748277824137206

[127.0.0.1:9999] Connection closed
shlomi@zshlomi9:~/Desktop/Ex$ python3 client.py
address : ('127.0.0.1', 9999)
[127.0.0.1:9999] Connection established
[127.0.0.1:9999] Sending request of length 687 bytes
[127.0.0.1:9999] Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.33333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02980636755702862 * 13
= -0.38748277824137206

[127.0.0.1:9999] Connection closed
shlomi@zshlomi9:~/Desktop/Ex$ python3 client.py
address : ('127.0.0.1', 9999)
[127.0.0.1:9999] Connection established
[127.0.0.1:9999] Sending request of length 376 bytes
[127.0.0.1:9999] Got response of length 60 bytes
Result: 6
Steps:
max(2, 3) + 3 = 3 + 3
= 6
[127.0.0.1:9999] Connection closed
shlomi@zshlomi9:~/Desktop/Ex$
```

## מכאן נחזיר אל ה-WareShark (מיינו את הפקודות של שלושת הבקשות)

WareShark Screenshot showing three captured TCP sessions (1, 2, and 3) between 127.0.0.1 and 127.0.0.1:

- Session 1:** A single TCP connection consisting of a SYN (port 9999) from 127.0.0.1 to 127.0.0.1, followed by a SYN-ACK (port 65495) from 127.0.0.1 to 127.0.0.1.
- Session 2:** A single TCP connection consisting of a SYN (port 9999) from 127.0.0.1 to 127.0.0.1, followed by a SYN-ACK (port 65495) from 127.0.0.1 to 127.0.0.1.
- Session 3:** A single TCP connection consisting of a SYN (port 9999) from 127.0.0.1 to 127.0.0.1, followed by a SYN-ACK (port 65495) from 127.0.0.1 to 127.0.0.1.

The packet details pane shows the following for each session:

- Session 1:** SYN (Seq# 0, Ack# 1, Win# 65495, Len# 0) and SYN-ACK (Seq# 1, Ack# 1, Win# 65495, Len# 0).
- Session 2:** SYN (Seq# 0, Ack# 1, Win# 65495, Len# 0) and SYN-ACK (Seq# 1, Ack# 1, Win# 65495, Len# 0).
- Session 3:** SYN (Seq# 0, Ack# 1, Win# 65495, Len# 0) and SYN-ACK (Seq# 1, Ack# 1, Win# 65495, Len# 0).

The packet list pane shows the sequence of packets for each session:

- Session 1:** 1, 2
- Session 2:** 1, 2
- Session 3:** 1, 2

The packet details for the first packet of Session 1 is as follows:

```
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
> 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: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 40198, Dst Port: 9999, Seq: 0, Len: 0
```

The packet bytes pane shows the raw hex and ASCII data for the first packet of Session 1:

```
0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 00 .-< @ ; .-.
0010 00 3c 01 b9 40 00 49 06 3b 01 f7 00 00 01 7f 00 .-> @ ; .-.
0020 00 01 9d 06 27 0f 83 0d 94 d6 00 00 00 a0 e2 .-> @ ; .-.
0030 ff d7 fe 30 00 00 02 04 ff d7 04 02 03 0a 0f e1 .-> @ ; .-.
0040 e5 c5 00 00 00 00 01 03 03 07 .-> @ ; .-
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	74	40198 → 9999 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=2413946309 TSecr=0 WS=128
2	0.000013	127.0.0.1	127.0.0.1	TCP	74	9999 → 40198 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=2413946309 TSecr=0 WS=128
3	0.000024	127.0.0.1	127.0.0.1	TCP	66	40198 → 9999 [ACK] Seq=1 Ack=1 Win=6536 Len=0 TSval=2413946309 TSecr=2413946309
4	0.000025	127.0.0.1	127.0.0.1	TCP	753	40198 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=687 TSval=2413946310 TSecr=2413946309
5	0.0000252	127.0.0.1	127.0.0.1	TCP	66	9999 → 40198 [ACK] Seq=1 Ack=688 Win=64896 Len=0 TSval=2413946310 TSecr=2413946310
6	0.001090	127.0.0.1	127.0.0.1	TCP	630	9999 → 40198 [PSH, ACK] Seq=1 Ack=688 Win=65536 Len=564 TSval=2413946311 TSecr=2413946310
7	0.001097	127.0.0.1	127.0.0.1	TCP	66	40198 → 9999 [ACK] Seq=688 Ack=565 Win=65024 Len=0 TSval=2413946311 TSecr=2413946311
8	0.001484	127.0.0.1	127.0.0.1	TCP	66	40198 → 9999 [FIN, ACK] Seq=688 Ack=565 Win=65536 Len=0 TSval=2413946311 TSecr=2413946311
9	0.001485	127.0.0.1	127.0.0.1	TCP	66	9999 → 40198 [FIN, ACK] Seq=689 Ack=689 Win=65536 Len=0 TSval=2413946311 TSecr=2413946311
10	0.001457	127.0.0.1	127.0.0.1	TCP	66	40198 → 9999 [ACK] Seq=689 Ack=566 Win=65536 Len=0 TSval=2413946311 TSecr=2413946311
11	8.588935	127.0.0.1	127.0.0.1	TCP	74	40206 → 9999 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=2413954898 TSecr=0 WS=128
12	8.588976	127.0.0.1	127.0.0.1	TCP	74	9999 → 40206 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=2413954898 TSecr=2413954898 WS=128
13	8.588993	127.0.0.1	127.0.0.1	TCP	66	40206 → 9999 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2413954898 TSecr=2413954898
14	8.589364	127.0.0.1	127.0.0.1	TCP	753	40206 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=687 TSval=2413954899 TSecr=2413954899
15	8.589373	127.0.0.1	127.0.0.1	TCP	66	9999 → 40206 [ACK] Seq=1 Ack=688 Win=64896 Len=0 TSval=2413954899 TSecr=2413954899
16	8.590416	127.0.0.1	127.0.0.1	TCP	630	9999 → 40206 [PSH, ACK] Seq=1 Ack=688 Win=65536 Len=564 TSval=2413954900 TSecr=2413954900
17	8.590437	127.0.0.1	127.0.0.1	TCP	66	40206 → 9999 [ACK] Seq=688 Ack=565 Win=65924 Len=0 TSval=2413954900 TSecr=2413954900
18	8.590613	127.0.0.1	127.0.0.1	TCP	66	40206 → 9999 [FIN, ACK] Seq=688 Ack=565 Win=65536 Len=0 TSval=2413954900 TSecr=2413954900
19	8.591087	127.0.0.1	127.0.0.1	TCP	66	9999 → 40206 [FIN, ACK] Seq=689 Ack=689 Win=65536 Len=0 TSval=2413954901 TSecr=2413954901
20	8.591126	127.0.0.1	127.0.0.1	TCP	66	40206 → 9999 [ACK] Seq=689 Ack=566 Win=65536 Len=0 TSval=2413954901 TSecr=2413954901
21	25.486218	127.0.0.1	127.0.0.1	TCP	74	34646 → 9999 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=2413971796 TSecr=0 WS=128
22	25.486259	127.0.0.1	127.0.0.1	TCP	74	9999 → 34646 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=2413971796 TSecr=2413971796 WS=128
23	25.486291	127.0.0.1	127.0.0.1	TCP	66	34646 → 9999 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2413971796 TSecr=2413971796
24	25.487079	127.0.0.1	127.0.0.1	TCP	442	34646 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=376 TSval=2413971796 TSecr=2413971796
25	25.487104	127.0.0.1	127.0.0.1	TCP	66	9999 → 34646 [ACK] Seq=1 Ack=377 Win=65152 Len=0 TSval=2413971797 TSecr=2413971796
26	25.487753	127.0.0.1	127.0.0.1	TCP	126	9999 → 34646 [PSH, ACK] Seq=1 Ack=377 Win=65533 Len=60 TSval=2413971797 TSecr=2413971796
27	25.487776	127.0.0.1	127.0.0.1	TCP	66	34646 → 9999 [ACK] Seq=377 Ack=61 Win=65536 Len=0 TSval=2413971797 TSecr=2413971797
28	25.488115	127.0.0.1	127.0.0.1	TCP	66	34646 → 9999 [FIN, ACK] Seq=377 Ack=61 Win=65536 Len=0 TSval=2413971798 TSecr=2413971797
29	25.488195	127.0.0.1	127.0.0.1	TCP	66	9999 → 34646 [FIN, ACK] Seq=61 Ack=378 Win=65536 Len=0 TSval=2413971798 TSecr=2413971798
30	25.488207	127.0.0.1	127.0.0.1	TCP	66	34646 → 9999 [ACK] Seq=378 Ack=62 Win=65536 Len=0 TSval=2413971798 TSecr=2413971798

```

0: Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
1: Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 127.0.0.1 (ff:ff:ff:ff:ff:ff)
2: Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
3: Transmission Control Protocol, Src Port: 40198, Dst Port: 9999, Seq: 0, Len: 0

```

Packets: 30 · Displayed: 30 (100.0%)

Profile: Default

\* מיינו את הבקשות לפי המסגרת האדומת בטור info

\* בירוק נראה את ה-qseq שהלך רוצה להתחיל, ואת ack שהציג השרת

\* בכתום נוכל לראות את הqseq timestamp (tsecr,tsval)

\* בצהוב נוכל לראות את אורך המידע len וגודלו ההודעה שהלך יכול לקבל בתיים

[syn]-בקשת תחילת קשר(הסבירנו ופירטנו על כך לפני תחילת התרגיל)

[syn,ack]-השרת מחייב לקלוח שקשר הוקדם בהצלחה(syn), ושולח בחזרה את המספר הסידורי ההתחלתי של הבטים שנשלחו מהשרת לקלוח(ack)

[ack]-הלך מאשר את החביבה [syn,ack] ועכשו גם הלך וgam השרת מסוכרים על המספרים הסידוריים הראשונים (הלך שולח את המספר הסידורי הבא שהוא מצפה לקבל)

[push,ack]-הלך מבקש מהשרת שייעבד את המידע ושולח לו את ack שמננו הוא מצפה שהມידע יתחליל לתקבל.

[ack]-השרת שולח לקלוח שהוא מאשר את הבקשה ואת ack הנוכחי

[push,ack]-השרת מתחילה לSEND את המידע  $ack=688$ , נוכל לראות שקיבלנו את המידע לפי אורך המידע ( $len=564$ )

[ack]-הלקוח מאשר שהוא קיבל את החבילה

[fin,ack]-שהdagל של ה-fin דלוק אז הלקוח מבקש לסגור את הקשר ושהכל התנהל כראוי, כמובן הוא שולח את זה עם המספר הסודרי (ack) שהוא מצפה לקבל את סגירת הקשר.

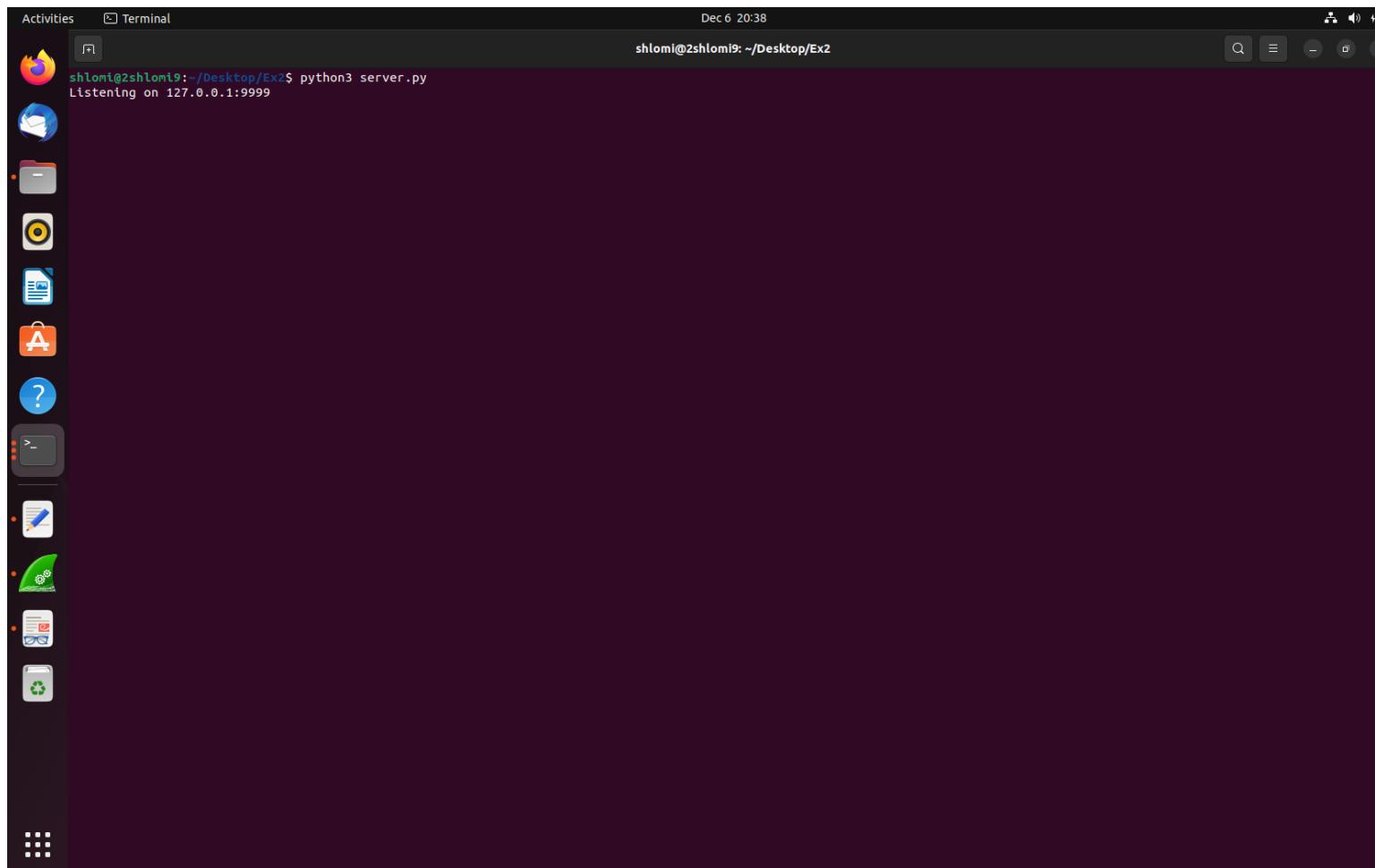
[fin,ack]-השרת מאשר את סגירת הקשר

[ack]-הלקוח שולח שהוא קיבל את האישור לסגירת הקשר.

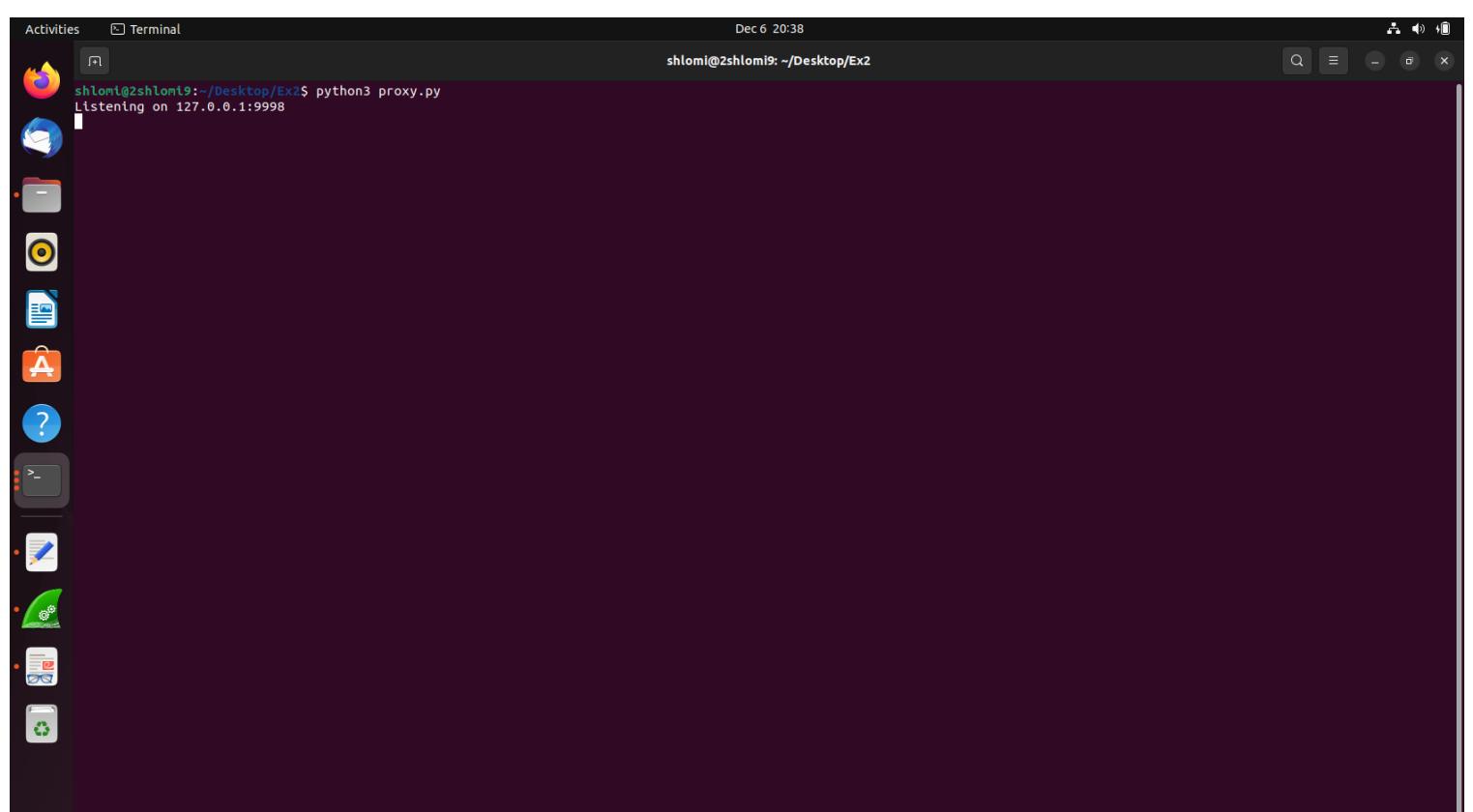
הקשר נסגר

## תרגיל 3.2

זמן שה-wireshark מבצע הסנפה נרץ את הserver



לאחר מכן נרץ את הפרויקט



## נשלח אותה בקשה פעמיים מהלך אל הפרויקט (פורט 9998)

```
Activities Terminal Dec 6 20:39
shlomi@2shlomi9:~/Desktop/Ex$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 687 bytes
{127.0.0.1:9998} Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.333333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02988636755702862 * 13
= -0.38748277824137206

{127.0.0.1:9998} Connection closed
shlomi@2shlomi9:~/Desktop/Ex$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 687 bytes
{127.0.0.1:9998} Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.333333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02988636755702862 * 13
= -0.38748277824137206

{127.0.0.1:9998} Connection closed
shlomi@2shlomi9:~/Desktop/Ex$
```

## נחזיר אל הפרויקט!

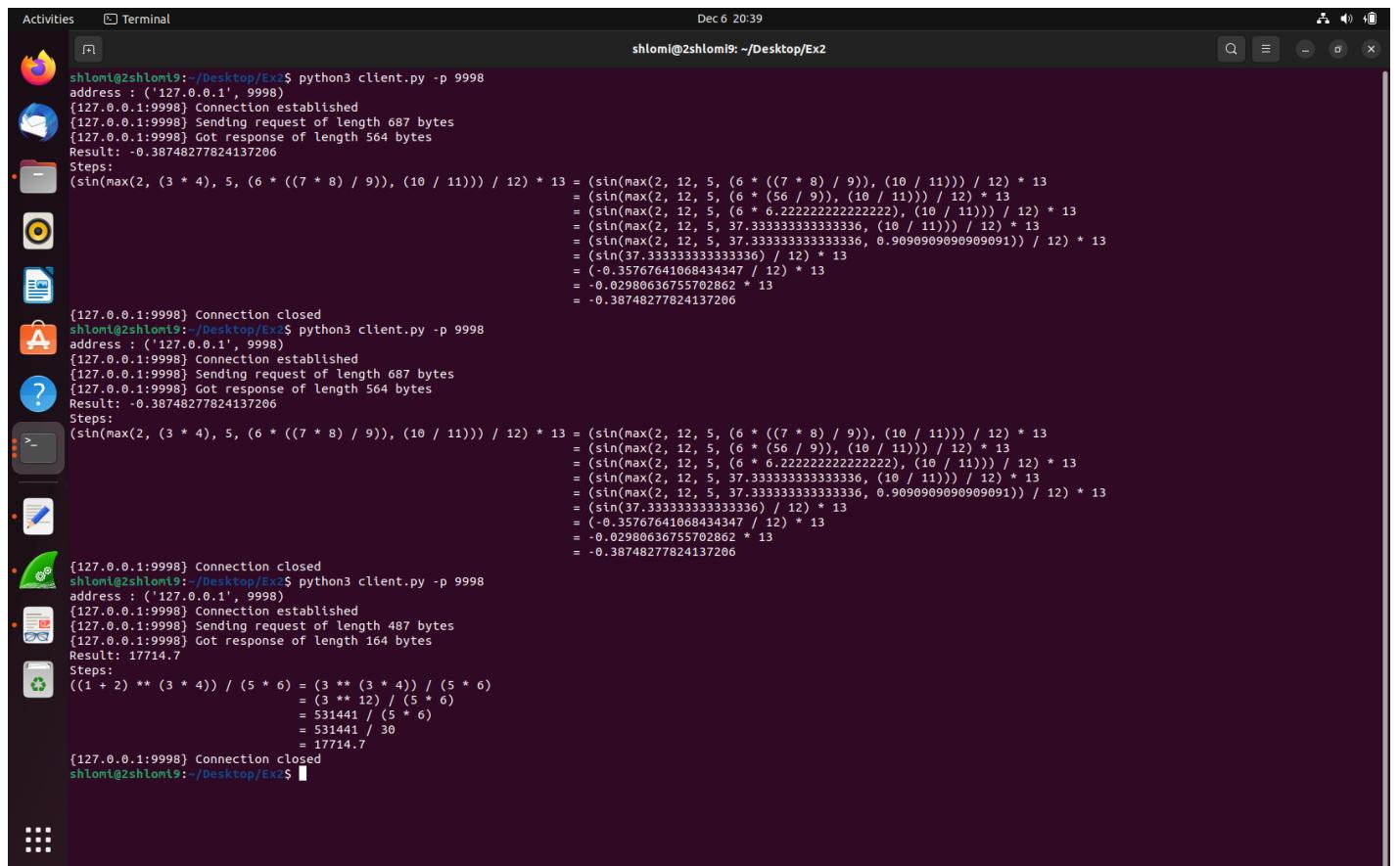
```
Activities Terminal Dec 6 20:39
shlomi@2shlomi9:~/Desktop/Ex2$ python3 proxy.py
Listening on 127.0.0.1:9998
{127.0.0.1:38964} Connected established
{127.0.0.1:38964} Got request of length 687 bytes
{127.0.0.1:38964} Cache miss, response cached ,server time remaining: inf, client time remaining: inf
{127.0.0.1:38964} Sending response of length 564 bytes
{127.0.0.1:38964} Connection closed
{127.0.0.1:58746} Connected established
{127.0.0.1:58746} Got request of length 687 bytes
{127.0.0.1:58746} Cache hit ,server time remaining: inf, client time remaining: inf
{127.0.0.1:58746} Sending response of length 564 bytes
{127.0.0.1:58746} Connection closed
```

\*הבקשה לא נמצאת במתמון - cache miss

\*הבקשה נמצאת במתמון – cache hit

בתמונה נראה שאות התגובה לבקשת הראשונה הוא הביא מהשרת ואת התגובה לבקשת השנייה הוא הביא מה-cache

עכשו נשלח לפיקוסי בקשה אחרת

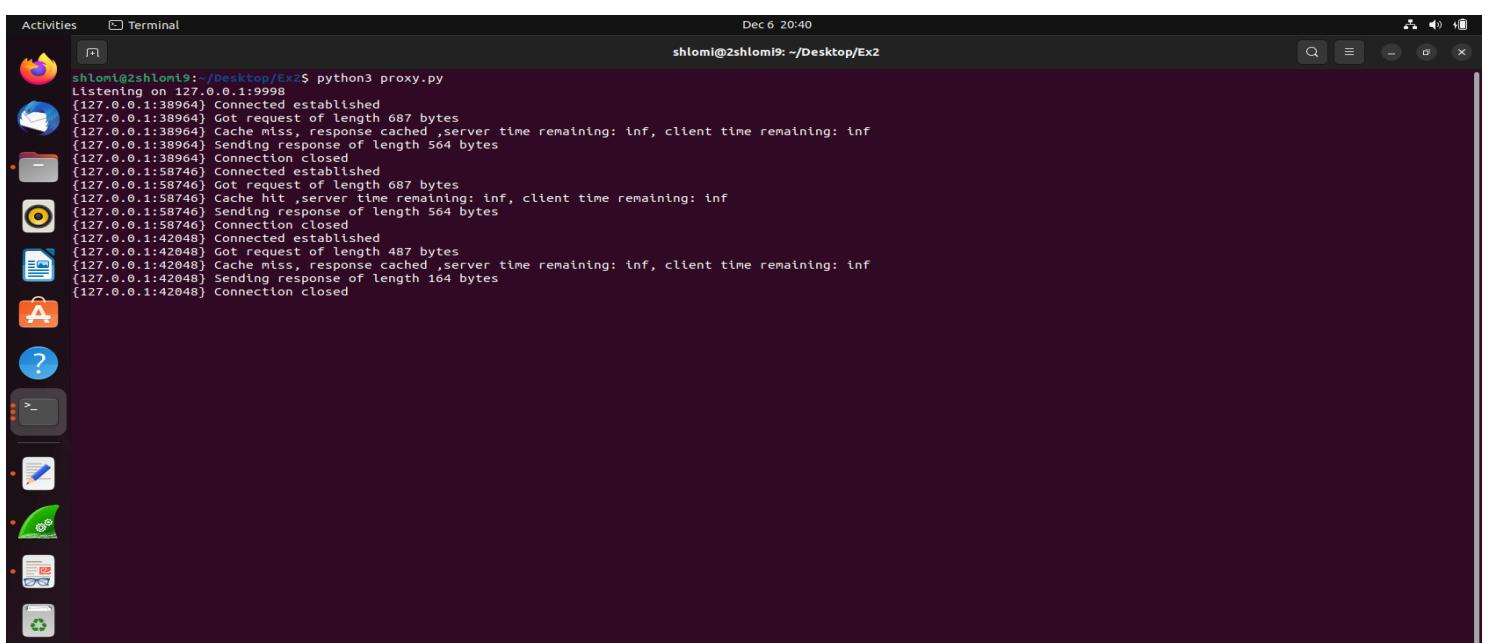


```
Activities Terminal Dec 6 20:39
shlomi@zshlomi9:~/Desktop/Ex2$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 687 bytes
{127.0.0.1:9998} Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.33333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02980636755702862 * 13
= -0.38748277824137206

{127.0.0.1:9998} Connection closed
shlomi@zshlomi9:~/Desktop/Ex2$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 687 bytes
{127.0.0.1:9998} Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.33333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02980636755702862 * 13
= -0.38748277824137206

{127.0.0.1:9998} Connection closed
shlomi@zshlomi9:~/Desktop/Ex2$
```

נחזיר אל הפיקוסי נראה שנקלט cache miss על הבקשת השלישית כי היא אכן עדין לא נשמרה במתמון



```
Activities Terminal Dec 6 20:40
shlomi@zshlomi9:~/Desktop/Ex2$ python3 proxy.py
Listening on 127.0.0.1:9998
{127.0.0.1:38964} Connected established
{127.0.0.1:38964} Got request of length 687 bytes
{127.0.0.1:38964} Cache miss, response cached, server time remaining: inf, client time remaining: inf
{127.0.0.1:38964} Sending response of length 564 bytes
{127.0.0.1:38964} Connection closed
{127.0.0.1:58746} Connected established
{127.0.0.1:58746} Got request of length 687 bytes
{127.0.0.1:58746} Cache hit, server time remaining: inf, client time remaining: inf
{127.0.0.1:58746} Sending response of length 564 bytes
{127.0.0.1:58746} Connection closed
{127.0.0.1:42048} Connected established
{127.0.0.1:42048} Got request of length 487 bytes
{127.0.0.1:42048} Cache miss, response cached, server time remaining: inf, client time remaining: inf
{127.0.0.1:42048} Sending response of length 164 bytes
{127.0.0.1:42048} Connection closed
```

## נראה את תגובת השרת

```

Activities Terminal
shlomi@2shlomi9: ~/Desktop/Ex2$ python3 server.py
Listening on 127.0.0.1:9999
Connection established with 127.0.0.1:42962
{127.0.0.1:42962} Got request of length 687 bytes
{127.0.0.1:42962} Sending response of length 564 bytes
{127.0.0.1:42962} Connection closed
Connection established with 127.0.0.1:41258
{127.0.0.1:41258} Got request of length 487 bytes
{127.0.0.1:41258} Sending response of length 164 bytes
{127.0.0.1:41258} Connection closed

```

נראה שרק פעמיים רשום connection established כלומר רק פעמיים יצרנו חיבור עם השרת, זאת מכיוון שהבקשה השנייה הייתה בטמונה

נחזיר אל wireshark

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	74	38964 + 9998 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM Tsvl=908941278 Tscr=0 WS=128
2	6.000023	127.0.0.1	127.0.0.1	TCP	74	9998 + 38964 [SYN, ACK] Seq=1 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM Tsvl=908941278 Tscr=908941278
3	0.000082	127.0.0.1	127.0.0.1	TCP	66	38964 + 9998 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsvl=908941278 Tscr=908941278
4	0.001096	127.0.0.1	127.0.0.1	TCP	753	38964 + 9998 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=687 Tsvl=908941279 Tscr=908941279
5	0.001173	127.0.0.1	127.0.0.1	TCP	66	9998 + 38964 [ACK] Seq=1 Ack=688 Win=64896 Len=0 Tsvl=908941279 Tscr=908941279
6	0.001436	127.0.0.1	127.0.0.1	TCP	74	42962 + 9998 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM Tsvl=908941280 Tscr=0 WS=128
7	0.001427	127.0.0.1	127.0.0.1	TCP	74	9999 + 42962 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM Tsvl=908941280 Tscr=908941280
8	0.001436	127.0.0.1	127.0.0.1	TCP	66	42962 + 9998 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsvl=908941280 Tscr=908941280
9	0.001481	127.0.0.1	127.0.0.1	TCP	753	42962 + 9998 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=687 Tsvl=908941280 Tscr=908941280
10	0.001485	127.0.0.1	127.0.0.1	TCP	66	9999 + 42962 [ACK] Seq=1 Ack=688 Win=64896 Len=0 Tsvl=908941280 Tscr=908941280
11	0.002646	127.0.0.1	127.0.0.1	TCP	638	9999 + 42962 [PSH, ACK] Seq=1 Ack=688 Win=65536 Len=564 Tsvl=908941281 Tscr=908941280
12	0.002671	127.0.0.1	127.0.0.1	TCP	66	42962 + 9998 [ACK] Seq=888 Ack=565 Win=6524 Len=0 Tsvl=908941281 Tscr=908941281
13	0.002792	127.0.0.1	127.0.0.1	TCP	66	42962 + 9998 [FIN, ACK] Seq=088 Ack=565 Win=65536 Len=0 Tsvl=908941281 Tscr=908941281
14	0.002864	127.0.0.1	127.0.0.1	TCP	638	9998 + 38964 [PSH, ACK] Seq=1 Ack=688 Win=65536 Len=564 Tsvl=908941281 Tscr=908941279
15	0.002874	127.0.0.1	127.0.0.1	TCP	66	38964 + 9998 [ACK] Seq=688 Ack=565 Win=6524 Len=0 Tsvl=908941281 Tscr=908941281
16	6.002395	127.0.0.1	127.0.0.1	TCP	66	9999 + 42962 [FIN, ACK] Seq=565 Ack=688 Win=65536 Len=0 Tsvl=908941281 Tscr=908941281
17	0.002916	127.0.0.1	127.0.0.1	TCP	66	42962 + 9998 [ACK] Seq=689 Win=65536 Len=0 Tsvl=908941281 Tscr=908941281
18	0.003208	127.0.0.1	127.0.0.1	TCP	66	38964 + 9998 [FIN, ACK] Seq=688 Ack=565 Win=65536 Len=0 Tsvl=908941281 Tscr=908941281
19	0.003283	127.0.0.1	127.0.0.1	TCP	66	9998 + 38964 [ACK] Seq=688 Ack=565 Win=65536 Len=0 Tsvl=908941282 Tscr=908941281
20	0.003304	127.0.0.1	127.0.0.1	TCP	66	38964 + 9998 [ACK] Seq=689 Ack=566 Win=65536 Len=0 Tsvl=908941282 Tscr=908941282
21	28.890038	127.0.0.1	127.0.0.1	TCP	74	58746 + 9998 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM Tsvl=908962168 Tscr=0 WS=128
22	28.890048	127.0.0.1	127.0.0.1	TCP	74	9998 + 58746 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM Tsvl=908962168 Tscr=908962168
23	28.890048	127.0.0.1	127.0.0.1	TCP	66	58746 + 9998 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsvl=908962168 Tscr=908962168
24	28.890258	127.0.0.1	127.0.0.1	TCP	753	58746 + 9998 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=687 Tsvl=908962169 Tscr=908962168
25	28.890263	127.0.0.1	127.0.0.1	TCP	66	9998 + 58746 [ACK] Seq=1 Ack=688 Win=64896 Len=0 Tsvl=908962169 Tscr=908962169
26	28.890557	127.0.0.1	127.0.0.1	TCP	638	9998 + 58746 [PSH, ACK] Seq=1 Ack=688 Win=65536 Len=564 Tsvl=908962169 Tscr=908962169
27	28.890565	127.0.0.1	127.0.0.1	TCP	66	58746 + 9998 [ACK] Seq=688 Ack=565 Win=6524 Len=0 Tsvl=908962169 Tscr=908962169
28	28.890954	127.0.0.1	127.0.0.1	TCP	66	58746 + 9998 [FIN, ACK] Seq=088 Ack=565 Win=65536 Len=0 Tsvl=908962169 Tscr=908962169
29	28.890995	127.0.0.1	127.0.0.1	TCP	66	9998 + 58746 [FIN, ACK] Seq=565 Ack=688 Win=65536 Len=0 Tsvl=908962169 Tscr=908962169
30	28.891001	127.0.0.1	127.0.0.1	TCP	66	58746 + 9998 [ACK] Seq=689 Ack=566 Win=65536 Len=0 Tsvl=908962169 Tscr=908962169
31	58.701448	127.0.0.1	127.0.0.1	TCP	74	42948 + 9998 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM Tsvl=908991980 Tscr=0 WS=128
32	58.701458	127.0.0.1	127.0.0.1	TCP	74	9998 + 42948 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM Tsvl=908991980 Tscr=908991980
33	58.701466	127.0.0.1	127.0.0.1	TCP	66	42948 + 9998 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsvl=908991980 Tscr=908991980
34	58.701623	127.0.0.1	127.0.0.1	TCP	553	42948 + 9998 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=487 Tsvl=908991980 Tscr=908991980
35	58.701629	127.0.0.1	127.0.0.1	TCP	66	9998 + 42948 [ACK] Seq=1 Ack=488 Win=6524 Len=164 Tsvl=908991981 Tscr=908991980
36	58.702031	127.0.0.1	127.0.0.1	TCP	74	41258 + 9998 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM Tsvl=908991980 Tscr=0 WS=128
37	58.702037	127.0.0.1	127.0.0.1	TCP	74	9999 + 41258 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM Tsvl=908991980 Tscr=908991980
38	58.702043	127.0.0.1	127.0.0.1	TCP	66	41258 + 9998 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsvl=908991980 Tscr=908991980
39	58.702065	127.0.0.1	127.0.0.1	TCP	553	41258 + 9998 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=487 Tsvl=908991980 Tscr=908991980
40	58.702068	127.0.0.1	127.0.0.1	TCP	66	9999 + 41258 [ACK] Seq=1 Ack=488 Win=6524 Len=164 Tsvl=908991980 Tscr=908991980
41	58.702776	127.0.0.1	127.0.0.1	TCP	238	9999 + 41258 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=487 Tsvl=908991980 Tscr=908991980
42	58.702786	127.0.0.1	127.0.0.1	TCP	66	41258 + 9998 [ACK] Seq=488 Ack=165 Win=65498 Len=0 Tsvl=908991981 Tscr=908991981
43	58.702848	127.0.0.1	127.0.0.1	TCP	66	41258 + 9998 [FIN, ACK] Seq=488 Ack=165 Win=65498 Len=0 Tsvl=908991981 Tscr=908991981
44	58.702885	127.0.0.1	127.0.0.1	TCP	238	9998 + 42048 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=164 Tsvl=908991981 Tscr=908991980
45	58.702892	127.0.0.1	127.0.0.1	TCP	66	42048 + 9998 [ACK] Seq=488 Ack=165 Win=65498 Len=0 Tsvl=908991981 Tscr=908991981
46	58.702925	127.0.0.1	127.0.0.1	TCP	66	9999 + 41258 [FIN, ACK] Seq=165 Ack=489 Win=65536 Len=0 Tsvl=908991981 Tscr=908991981
47	58.702931	127.0.0.1	127.0.0.1	TCP	66	411258 + 9998 [ACK] Seq=489 Ack=166 Win=65536 Len=0 Tsvl=908991981 Tscr=908991981
48	58.703108	127.0.0.1	127.0.0.1	TCP	66	42048 + 9998 [FIN, ACK] Seq=488 Ack=165 Win=65536 Len=0 Tsvl=908991981 Tscr=908991981
49	58.703186	127.0.0.1	127.0.0.1	TCP	66	9998 + 42048 [FIN, ACK] Seq=165 Ack=489 Win=65536 Len=0 Tsvl=908991981 Tscr=908991981

\*במסגרת הכתומה(הבקשה הראשונה) נראה שהליך מבקש בקשה מהפרקטי(נראה שהפורט שיוצר קשור הוא 9998 שזה הפרוקטי)

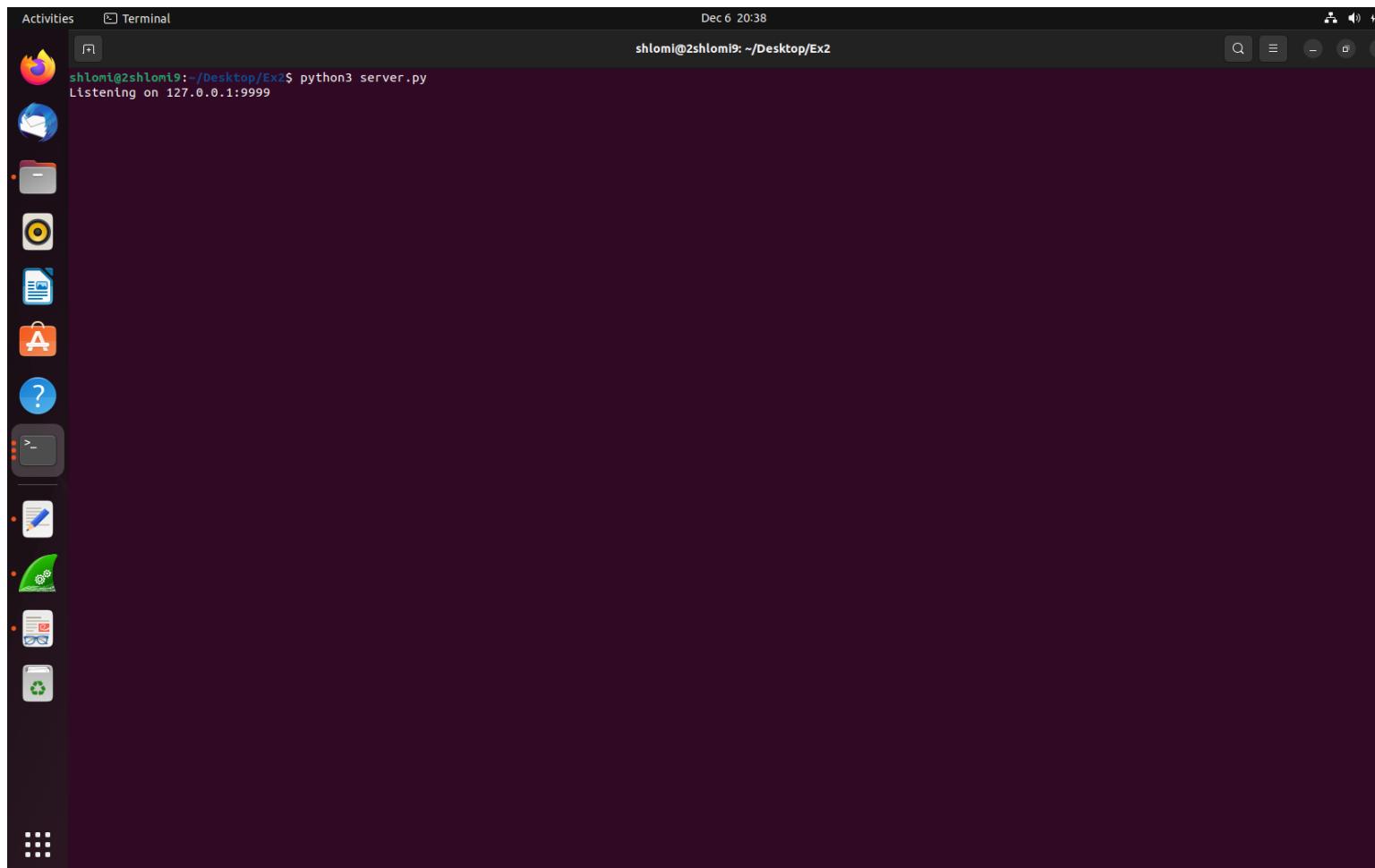
\*במסגרת הצהובה (הבקשה הראשונה) נשים לב שהפרקטי יצא קשור עם השירות ובירך ממנו את הבקשה של הלוקוח(נראה שהפורט שייצר את הקשר הינו 9999 שזה השירות)

\*המסגרת האדומה (הבקשה השנייה)

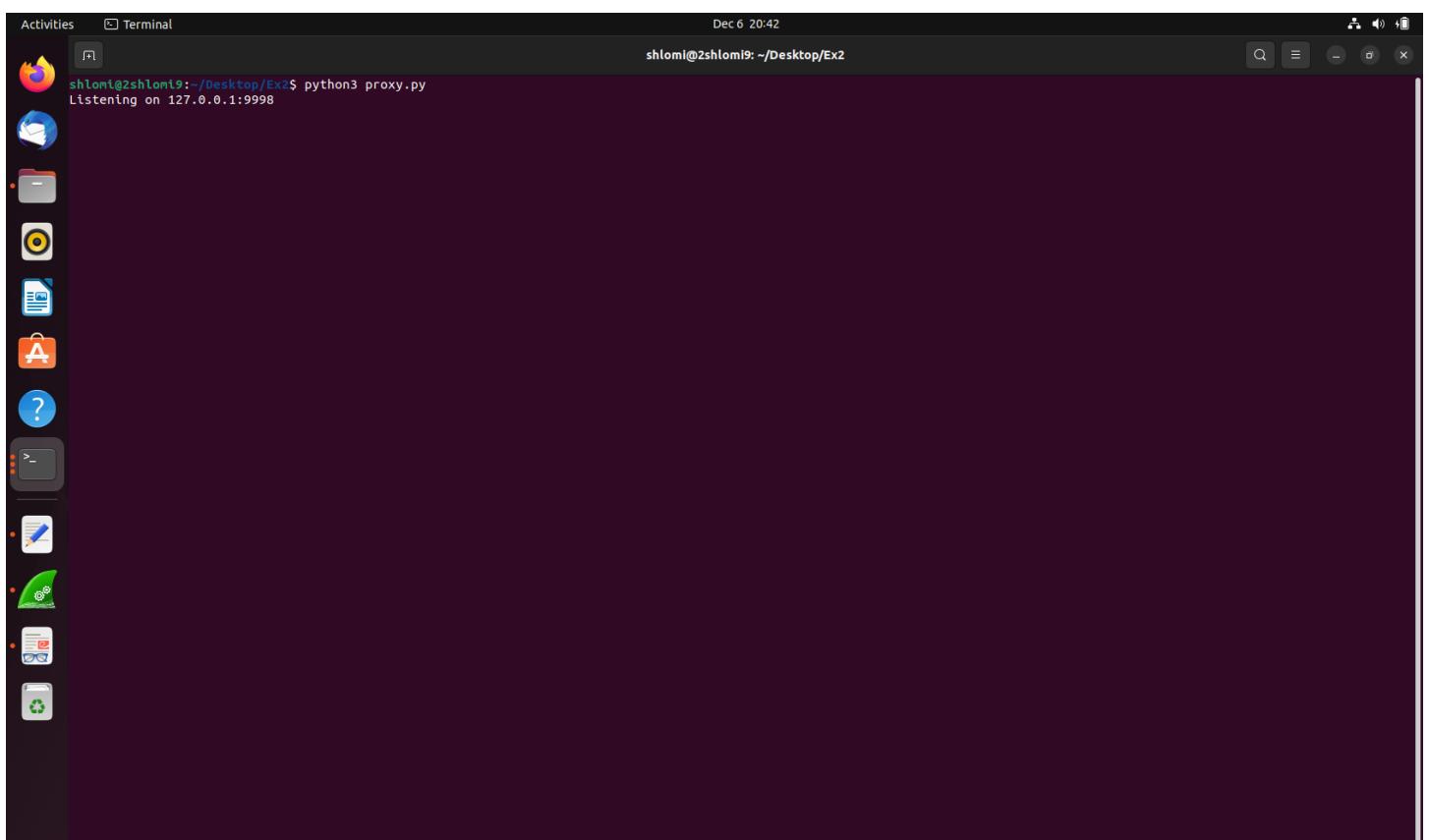
\*במסגרת הירוקה (הבקשה השנייה) נשים לב שהתקשרות היא באמצעות פорт 9998 , לא הייתה פניה מהפרקטי אל השירות(פורט 9999) ולכן נסיק מכך שהtagובה התקבלה מהcache והבקשה הראשונה נשמר במתਮון

### תרגיל 3.3

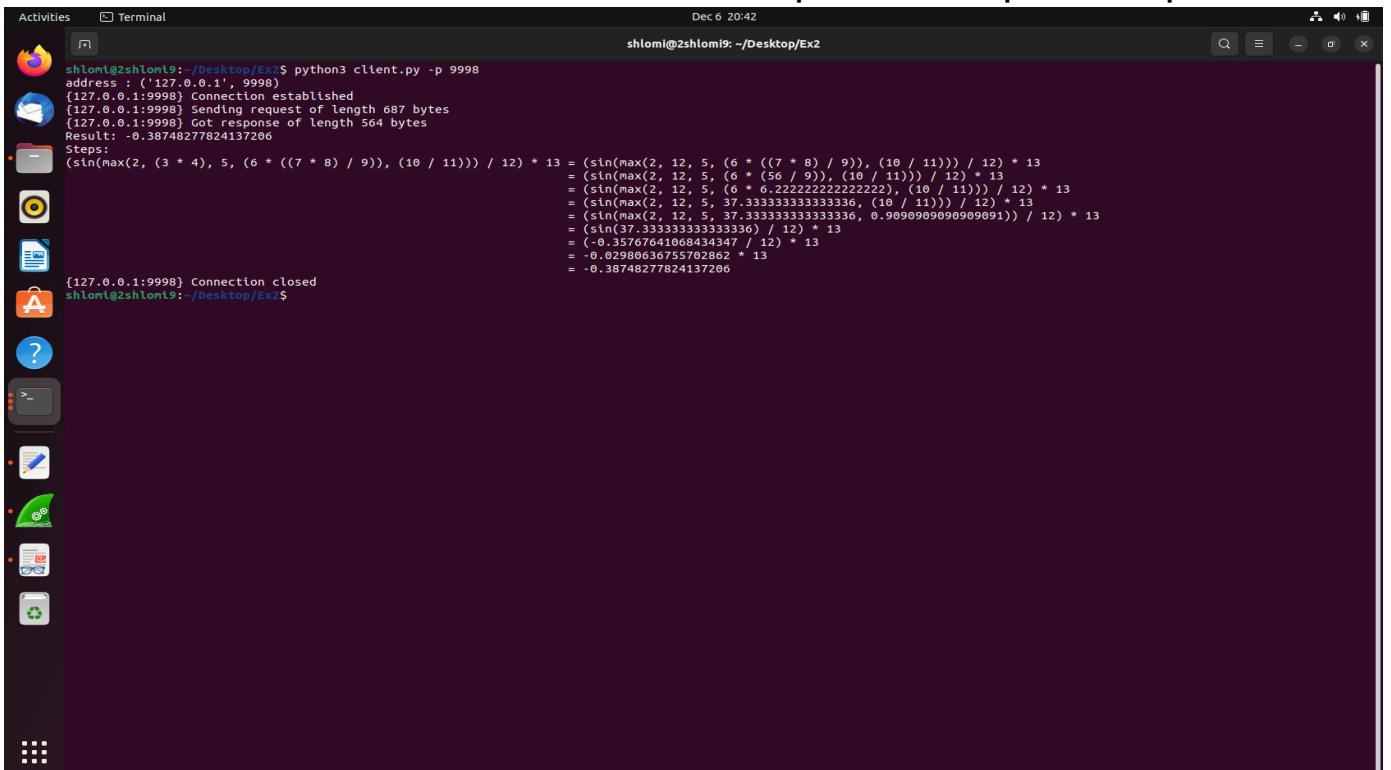
זמן שה-wireshark מבצע הסנפה נרץ את הserver



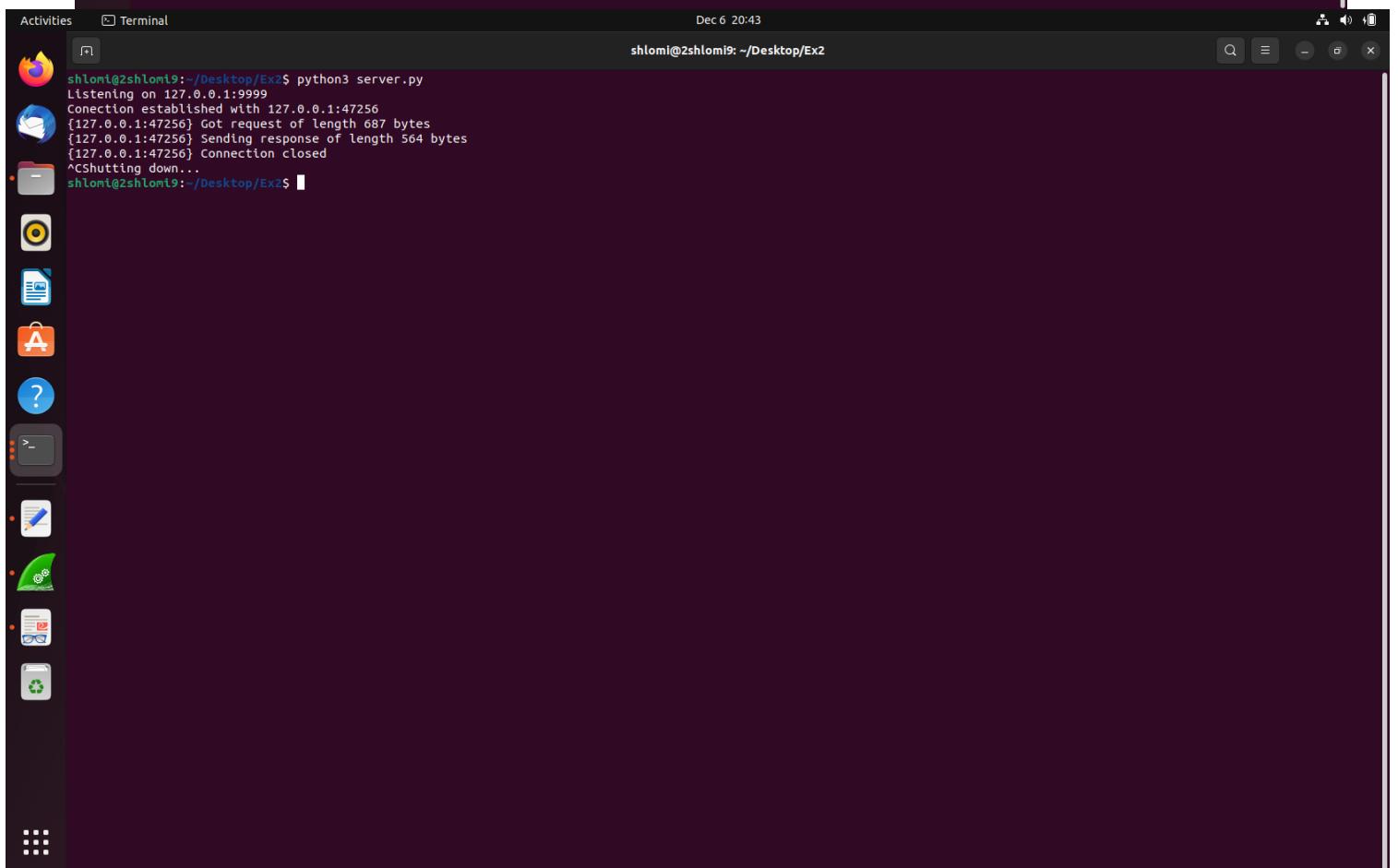
לאחר מכן נרץ את הפרויקט!



## נשלח בקשה מהלך אל הפרויקט ואז נסגור את השרת

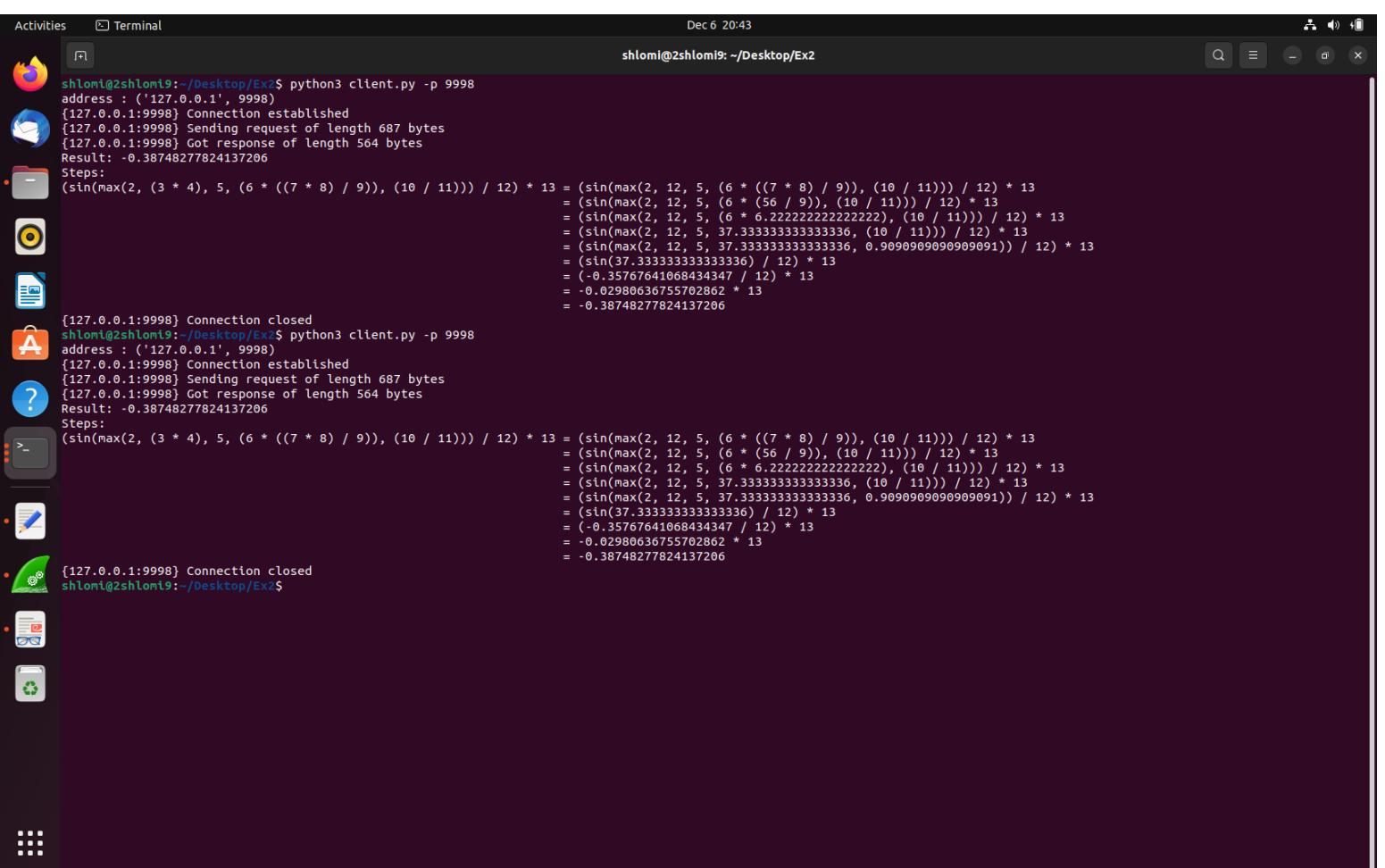


```
Activities Terminal Dec 6 20:42
shlomi@2shlomi9:~/Desktop/Ex2$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 687 bytes
{127.0.0.1:9998} Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.333333333333336) / 12) * 13
= (-0.35767641068443447 / 12) * 13
= -0.02988636755762862 * 13
= -0.38748277824137206
{127.0.0.1:9998} Connection closed
shlomi@2shlomi9:~/Desktop/Ex2$
```



```
Activities Terminal Dec 6 20:43
shlomi@2shlomi9:~/Desktop/Ex2$ python3 server.py
Listening on 127.0.0.1:9999
Connection established with 127.0.0.1:47256
{127.0.0.1:47256} Got request of length 687 bytes
{127.0.0.1:47256} Sending response of length 564 bytes
{127.0.0.1:47256} Connection closed
^CShutting down...
shlomi@2shlomi9:~/Desktop/Ex2$
```

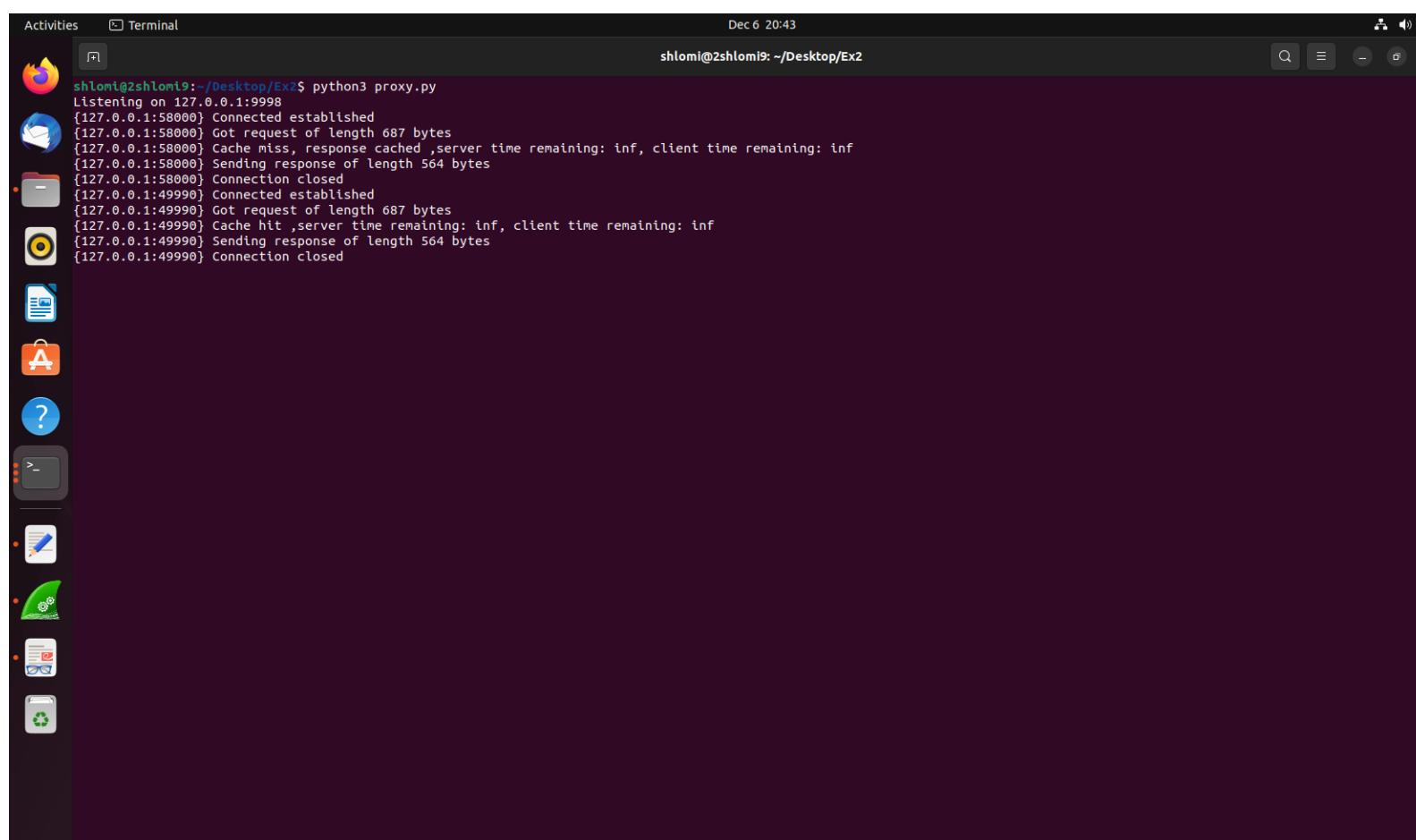
## נשלח את אותה בקשה שוב לפרוקסוי



A screenshot of an Ubuntu desktop environment. A terminal window is open in the top right corner, titled 'Terminal'. The terminal shows the output of running 'client.py' on port 9998. The logs indicate a connection established, a request sent, a response received, and a connection closed. The response content is a complex mathematical expression involving nested sine functions and constants.

```
shlomi@2shlomi9:~/Desktop/Ex2$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
[127.0.0.1:9998] Connection established
[127.0.0.1:9998] Sending request of length 687 bytes
[127.0.0.1:9998] Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.333333333333336) / 12) * 13
= (-0.35767641668434347 / 12) * 13
= -0.02988636755702862 * 13
= -0.38748277824137206
[127.0.0.1:9998] Connection closed
shlomi@2shlomi9:~/Desktop/Ex2$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
[127.0.0.1:9998] Connection established
[127.0.0.1:9998] Sending request of length 687 bytes
[127.0.0.1:9998] Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.333333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.333333333333336) / 12) * 13
= (-0.35767641668434347 / 12) * 13
= -0.02988636755702862 * 13
= -0.38748277824137206
[127.0.0.1:9998] Connection closed
shlomi@2shlomi9:~/Desktop/Ex2$
```

## נחזיר אל התגובה של הפרוקסוי

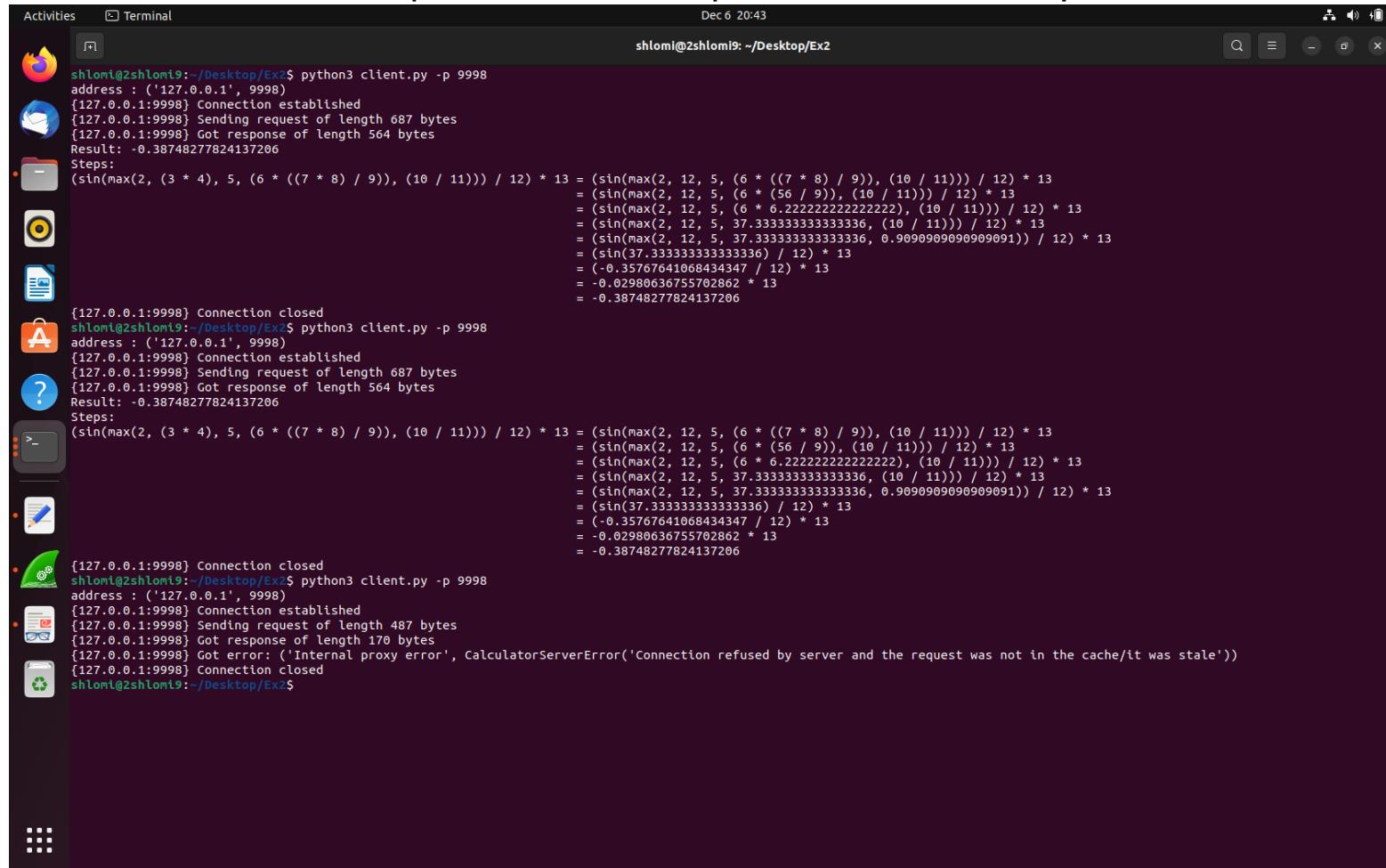


A screenshot of an Ubuntu desktop environment. A terminal window is open in the top right corner, titled 'Terminal'. The terminal shows the output of running 'proxy.py'. The logs indicate a connection established, a request received, a cache miss, a response sent, and a connection closed. The response content is identical to the one from the client, showing the same complex mathematical expression.

```
shlomi@2shlomi9:~/Desktop/Ex2$ python3 proxy.py
Listening on 127.0.0.1:9998
[127.0.0.1:58000] Connected established
[127.0.0.1:58000] Got request of length 687 bytes
[127.0.0.1:58000] Cache miss, response cached ,server time remaining: inf, client time remaining: inf
[127.0.0.1:58000] Sending response of length 564 bytes
[127.0.0.1:58000] Connection closed
[127.0.0.1:49990] Connected established
[127.0.0.1:49990] Got request of length 687 bytes
[127.0.0.1:49990] Cache hit ,server time remaining: inf, client time remaining: inf
[127.0.0.1:49990] Sending response of length 564 bytes
[127.0.0.1:49990] Connection closed
```

הבקשה הראשונה נשמרה במתماן cache miss, לאחר מכן שלחנו שוב אותה בקשה cache hit, פירטנו על זה בעמודים קודמים

עכשו בזמן שהשרת סגור נשלח בקשה אחרת אל הפרויקט!

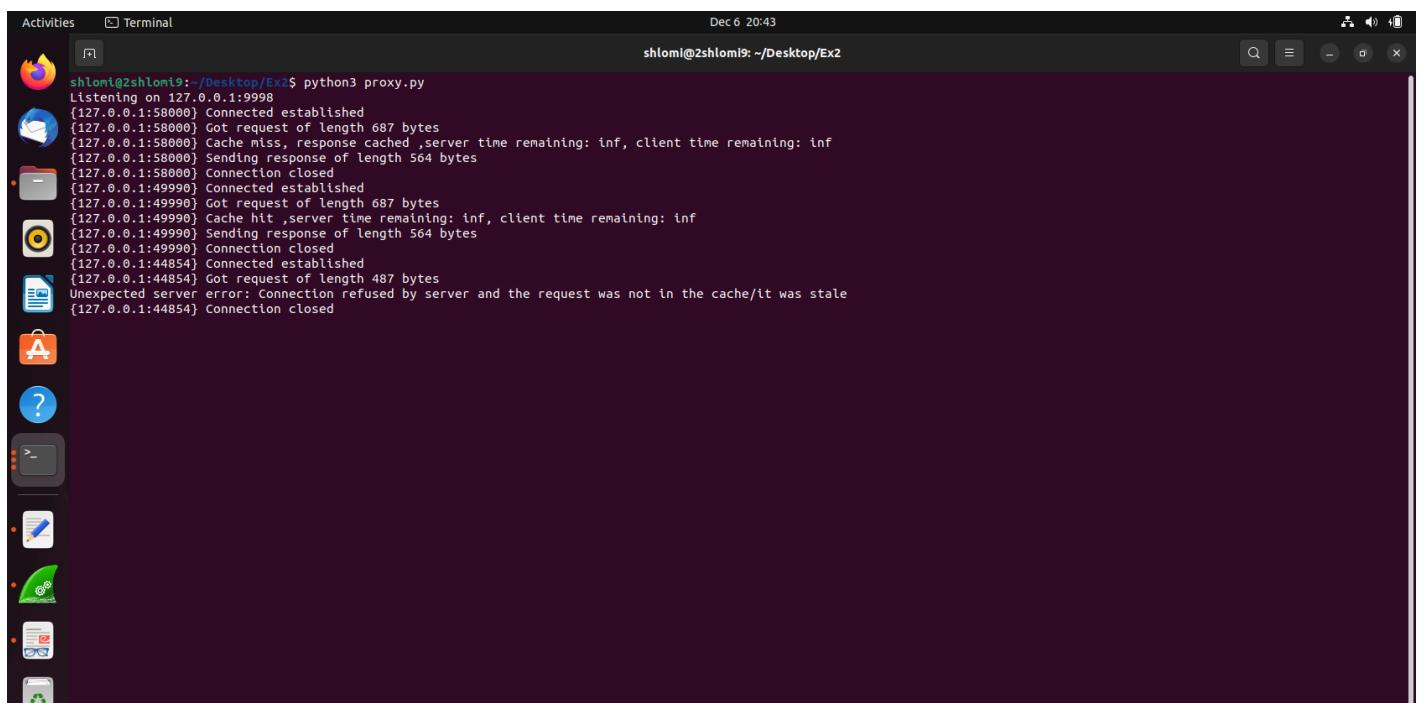


```
Activities Terminal Dec 6 20:43
shlomi@zshlomi9:~/Desktop/Ex2$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 687 bytes
{127.0.0.1:9998} Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.33333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02980636755702862 * 13
= -0.38748277824137206

{127.0.0.1:9998} Connection closed
shlomi@zshlomi9:~/Desktop/Ex2$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 687 bytes
{127.0.0.1:9998} Got response of length 564 bytes
Result: -0.38748277824137206
Steps:
(sin(max(2, (3 * 4), 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13 = (sin(max(2, 12, 5, (6 * ((7 * 8) / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * (56 / 9)), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, (6 * 6.222222222222222), (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, (10 / 11))) / 12) * 13
= (sin(max(2, 12, 5, 37.33333333333336, 0.9090909090909091)) / 12) * 13
= (sin(37.33333333333336) / 12) * 13
= (-0.35767641068434347 / 12) * 13
= -0.02980636755702862 * 13
= -0.38748277824137206

{127.0.0.1:9998} Connection closed
shlomi@zshlomi9:~/Desktop/Ex2$ python3 client.py -p 9998
address : ('127.0.0.1', 9998)
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 487 bytes
{127.0.0.1:9998} Got response of length 170 bytes
{127.0.0.1:9998} Got error: ('Internal proxy error', CalculatorServerError('Connection refused by server and the request was not in the cache/it was stale'))
{127.0.0.1:9998} Connection closed
shlomi@zshlomi9:~/Desktop/Ex2$
```

נראה גם את תגובת הפרויקט!

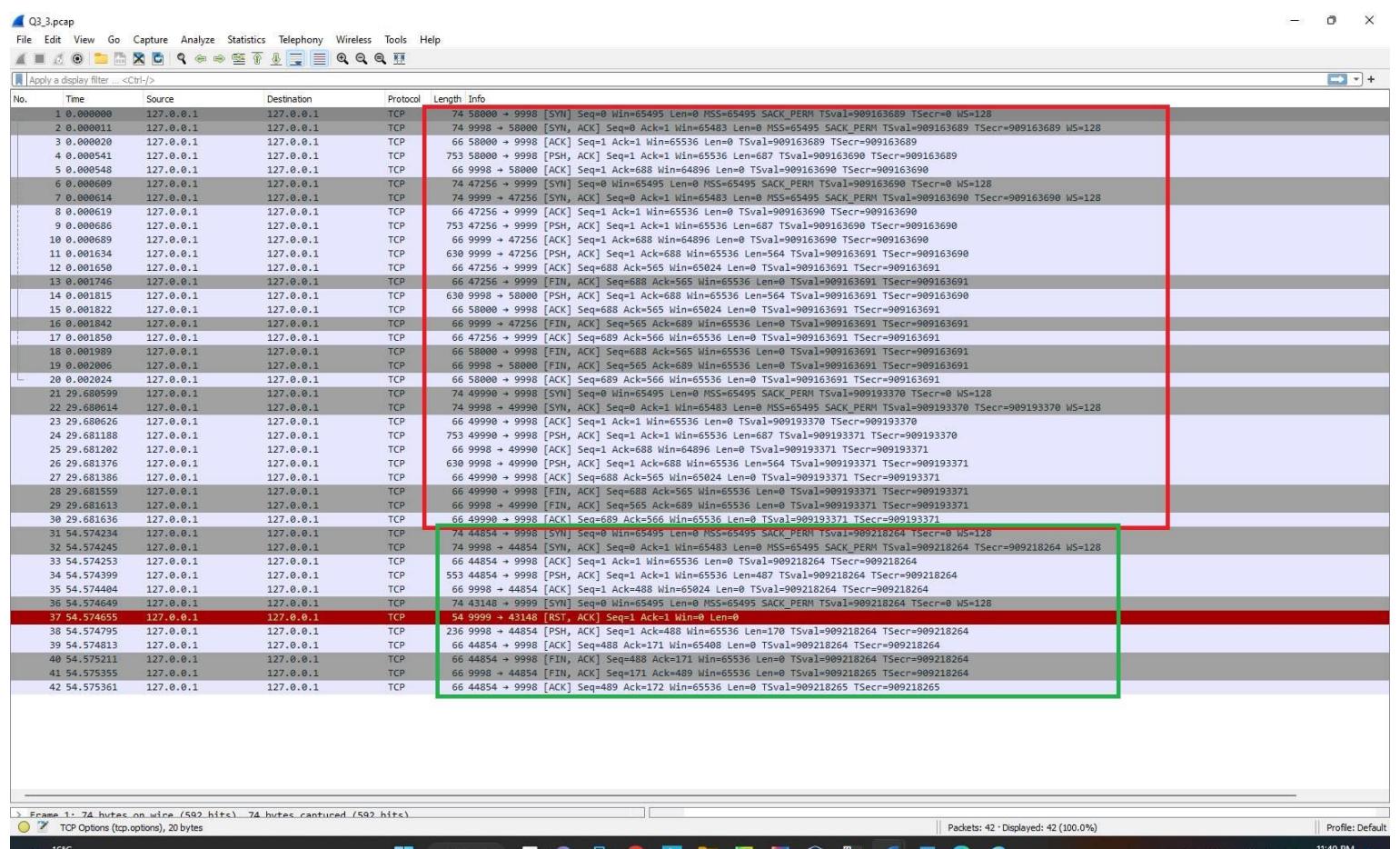


```
Activities Terminal Dec 6 20:43
shlomi@zshlomi9:~/Desktop/Ex2$ python3 proxy.py
Listening on 127.0.0.1:9998
[127.0.0.1:58000] Connected established
[127.0.0.1:58000] Got request of length 687 bytes
[127.0.0.1:58000] Cache miss, response cached, server time remaining: inf, client time remaining: inf
[127.0.0.1:58000] Sending response of length 564 bytes
[127.0.0.1:58000] Connection closed
[127.0.0.1:49990] Connected established
[127.0.0.1:49990] Got request of length 687 bytes
[127.0.0.1:49990] Cache hit, server time remaining: inf, client time remaining: inf
[127.0.0.1:49990] Sending response of length 564 bytes
[127.0.0.1:49990] Connection closed
[127.0.0.1:44854] Connected established
[127.0.0.1:44854] Got request of length 487 bytes
Unexpected server error: Connection refused by server and the request was not in the cache/it was stale
[127.0.0.1:44854] Connection closed
```

נראה שגם ב-client וגם ב-proxy קיבלנו error זהה מכיוון שביקשנו מהפרויקט' בקשה שלא נמצאת במתמון בזמן שהשרת לא רץ,

הפרויקט' יודע לשמר את המידע אבל הפעולות והчисובים מצריכים את השרת ולכן כל עוד המידע לא שומר במתמון והשרת לא רץ לא יוכל ליצור תקשורת עם השרת ולכן לא יוכל להעביר את המידע

עכשו נחזור אל ה-wireshark



במסגרת האדומה נראה את שתי הבקשות (הראשונה והשנייה)

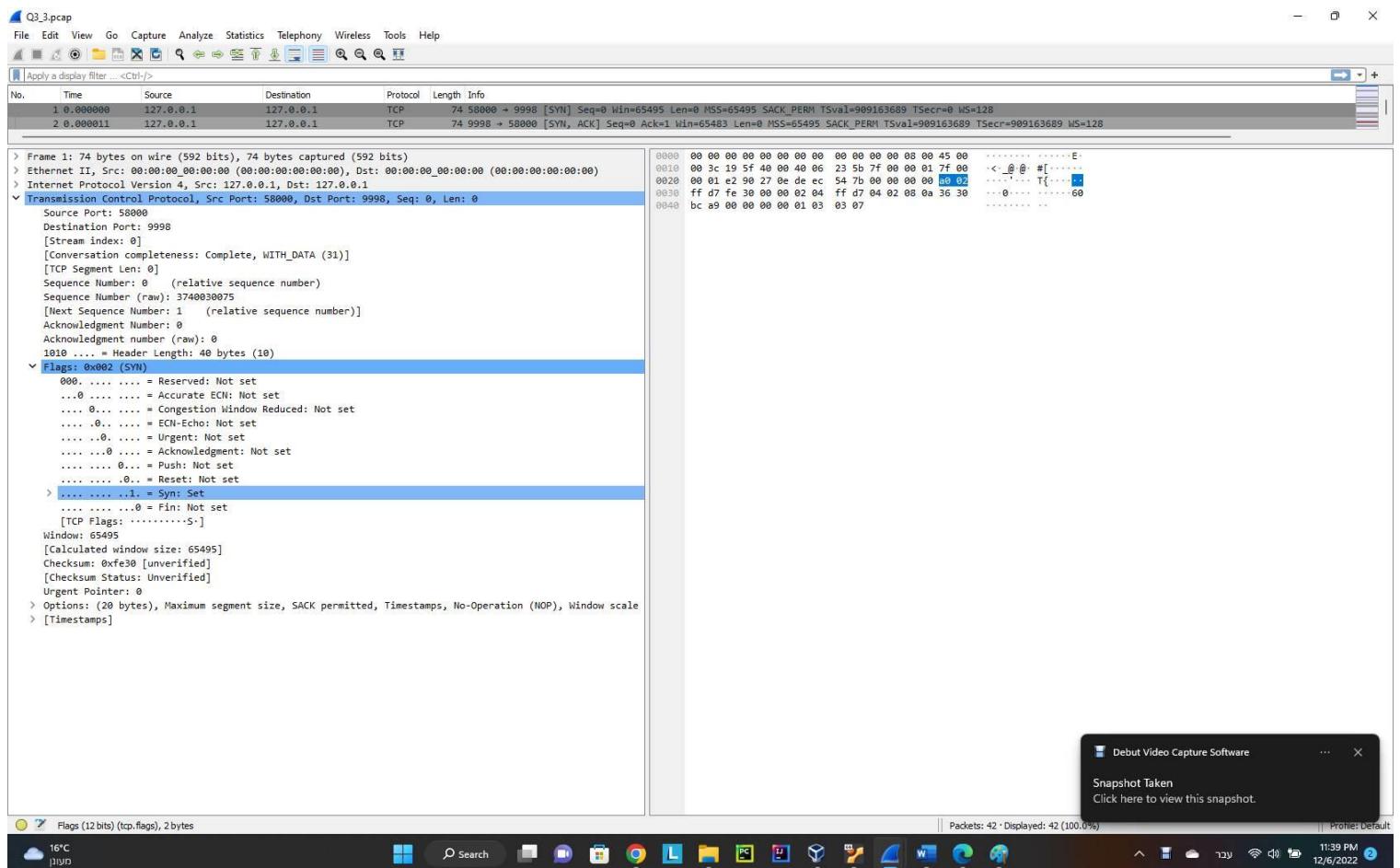
כמו שהסבירנו במפורט מוקדם, בבקשת הראשונה הלקוח יצר קשר עם הפרויקט' ואז הפרויקט' יצר קשר עם השרת, לאחר מכן התקשורת הסתיימה ושני הצדדים(הלקוח והשרת) הסכימו לפרויקט' לשמר את המתמון

את הבקשת השנייה כמו שהסבירנו מוקדם(הסבירנו עם תמונות) הפורט שאיתו אנחנו יוצרים קשר הוא 9998 ונראה שהפרויקט' לא יוצר קשר עם פורט 9999 כלומר עם השרת כי הבקשת השנייה כבר שמורה במתמון

עכשו במסגרת הירוקה שלחנו את הבקשה השלישית, נשים לב שהתקשרות בין הלקוח לפרוקס' עבודה אבל נשים לב, שהתשובה לא הייתה במתמון ולכן בזמן שהפרוקס' ניסה ליצור קשר עם השרת, הפרוקס' לא הצליח ליצור קשר מהשרת אז הוא קיבל מפורט 9999 חבילת שבת הנורות ack rst דלוקות.

על ack פירטנו והסבירנו, ולכן נסביר רק על rst:  
זמן שהנורה של rst דלוקה זה אומר שנכשל הקשר עם השרת ולכן הפרוקס' לא יכול להעביר את המידע כי הוא לא היה אצל במתמון

## אפשר לראות תחת פרוטוקול TCP



\*flag\*: בטור הדגלים נוכל לראות איזה נורוות דלוקות (false=0 true =1)

\*יש לנו אפשרות לראות את הקטגוריה שזה הזמן מקבלת החביליה

\*נראה רשום לנו שזה המספר הסידורי הבא