

מטרה 2 - רשותות תקשורת

.1

הסברים להוד

Server

```
94     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server_socket:
95         # SO_REUSEADDR is a socket option that allows the socket to be bound to an address that is already in use.
96         server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
97
98         # Prepare the server socket
99         # * Fill in start (1)
100        server_socket.bind((host, port))
101        """
102        explanation-
103            connecting the socket to the host ip and port.
104            bind method prepares the server socket to listen for connection
105            on the specific port and address (ip address) and allow clients to connect to the socket.
106        """
107
```

פקודת BIND מחברת בין הHOST לבין SOCKET ש"נפתח" ובין הפורט והIP. מכינה את הHOST להאזנה בכתובת זו, ולהתחברות לCLIENTS.

```
107             server_socket.listen(1)
108             """
109             explanation-
110                 listen method tells the server to wait for incoming connections.
111                 the numeric param tells the socket the max number of client connections
112                 he'll handle simultaneously. in this exercise the server will be handling
113                 one client at max, so we set the param to 1.
114
115
```

פקודת LISTEN אומרת לשרת להאזין לביקשות התחברות נוכחות. הפרמטר המספרי מצין את מספר החיבורים שהשרת מאפשר לנהל במקביל. בחרנו ב-1 כיון שבמטרה זו יתחבר רק לקו אחד לשרת בכל זמן נתון.

```
116             server_socket.settimeout(1) # setting a timeout for to accept method. if quit was received,
117             # timeout will make sure new thread will not open
118             # * Fill in end (1)
```

היה לנו "באג" בסגירת THREADS CIION שהשרת המשיך להאזין ולחייב ACCEPT. אך הגדרנו TIMEOUT שיפסיק לחכות לACCEPT כאשר התקבלה בקשה לסגירת SERVER (פונקציה שהוספנו בשביב נוחות בהרצאת הקורס), שלוחת הודעה עם בקשה לסגור את השרת וTERMINATE את התוכנה, על מנת שלא נצטרך לסגור בכל פעם ידנית את שלושת הקבצים).

```
157             break
158             # added lines-for quit option
159             except socket.timeout:
160                 # accept() timed out, check if we should stop
161                 if flag_quit:
162                     # QUIT was signaled by a handler thread
163                     break
164                     # If not quitting, just continue waiting for connections.
165                     pass
166             # end of added lines
```

בבקשר לפונקציית `TIMEOUT`, אם התקבלה הודעה סגירה- לא נמייך ללחכות ל`ACCEPT` ונמצא מהלולאה. במידה ולא התקבלה הודעה כזו- נישאר בולולה (הodata הסגירה היא עבור אגירת כל השירותים, ולא עבור התנטקות של לקוחות ספציפי מהשרת. כמו שצווין- משמשת עבור נוחות שלנו בלבד ולא מדרשה במתלה).

```

169          # added lines-for terminating the program
170      try:
171          print("closing socket...")
172          server_socket.close() # Close the proxy socket
173          print("terminating server.py...")
174          sys.exit(0)
175      except Exception as e:
176          print(f"Error while closing server socket: {e}")
177      # end of added lines
178

```

סגירת התוכנה כאשר התקבלה הודעה `QUIT`, וכל הTHREADS נסגרו. מודוא סגירה של `SOCKET` למרות `SHWT` אמר לודוג לך בלבד, כדי למנוע תקלות בסגירת הקובץ ולמודוא שה`SOCKET` באמת סגור.

```

76      # * Fill in start (3)
77      data = client_socket.recv(BUFFSIZE)
78      """
79      explanation -
80      after a connection is established (accept method returned the client socket and address successfully),
81      a new thread is open in order to handle the communication with this specific client.
82      the recv method is waiting to receive data from the client.
83      reads up to BUFFSIZE bytes of data from the client socket, if the client closed the connection- the recv
84      method will return an empty bytes object.
85      """
86
87
88      BUFFSIZE = api.BUFFER_SIZE # using the API buffer size to ensure consistency in data handling across all socket operations
89

```

הfonקציה `RECV` משמשת לקבלת DATA מהלקוח. הparameter המספרי שהfonקציה מקבלת מסמן את הגודל המקסימלי של מידע שהfonקציה תאפשר לקבל בכל פעם. הגדרנו אותו כמשתנה `BUFFSIZE` בראש התוכנית על מנת לאפשר את שינוי בקלות.

את המשתנה `BUFFSIZE` הגדרנו לפי גודל הבادر שהוגדר באPI, על מנת לשומר על אחידות משתנה זה בכל הקבצים הנדרשים למטרה זו. בנוסף, זה מאפשר את שינוי בקלות לכל קבצי `server.py`, `proxy.py`, `client.py` בכך צורך.

```

201      try: # checking if QUIT message was received.
202          if len(data) == 4: # implemented in this scope since request is never shorter than 12 bytes,
203              # a 4 bytes data can't be a request.
204              is_quiting = data.decode("utf-8") # decoding bytes to string
205              print(is_quiting)
206              if "QUIT" in is_quiting:
207                  flag_quit = True
208                  print(flag_quit)
209                  client_socket.close() # making sure client socket is closed
210                  break
211              except Exception as e:
212                  print(e)
213      # * Fill in end (3)

```

בלוק זה משמש עבור פונקציית הסגירה אותה ציינו קודם לכן. גודל הodata הסגירה הוא 4 והוא בהכרח לא שווה לגודל בקשה אחרת שתתקבל (כיוון שהוא קטן מהגודל המינימלי של פקחת הבקשת). כיוון שמבנה הodata שונה מבנה של בקשה, השתמשנו בפונקציית

DECODE על מנת להמיר את הודעה STRING לBYTES . לאחר קבלת הודעה זו שינינו את המשתנה הגלובלי quit_flag לTRUE, על מנת לציין לולאה הראשית שהתקבלה בקשה WITH TO TERMINATE . סגירת SOCKET הנועדה על מנת לוודא שאכן נסגר, אף שWITH מטפל בכך אוטומטית. לאחר מכן, הבצע BREAK על מנת לצאת מהלולה ולהתחל לטפל בסגירת הרצת הקובץ.

```

213     if not data: # * Change in start (1)
214         # exit loop when receiving no data (means that client closed the connection).
215         # no need to close client socket since the with method handles this automatically
216         break
217         # * Change in end (1)

```

במידה ולא התקבל מידע, זה סימן שהלוקו ניתק את הקשר עם השירות. לכן, נבצע BREAK Molalat_RECV. המשך טיפול בסגירת הקשר - בהמשך הקוד.

```

233             # * Fill in start (4)
234             client_socket.sendall(response)
235             """
236             explanation -
237                 there's two method we can choose from in order to send data to the client while using
238                 TCP protocol:
239                     send()- sends data up to some buffer size. it may not send all the data in one
240                         call, so it will require us to handle the sending of the remaining data.
241                     sendall- ensure the sending of the all the data, by internally loops, and handling the
242                         buffer limitation automatically.
243
244             we chose to use sendall method in order to make sure all the data is send to the client.
245             """
246             # * Fill in end (4)

```

.process_request. ואחרי מכן, המשכנו לפונקציית response. פונקציה זו החזירה לנו את התשובה שנשלחה בחזרה ללקוח, ואתה הגדרנו במשתנה response . את התשובה נשלח ללקוח באמצעות פונקציית sendall . בחרנו בפונקציה זו כיון שהיא מטפלת בשילוח תשובה בכל גודל בצורה אוטומטית, ולא נדרש להגדיר לה BUFFER דינית.

```

252             # * Change in start (2)
253             print(f"{client_prefix} Connection closed")
254             client_socket.close() # same as line 254
255             return
256             # * Change in end (2)
257

```

בלוק זה נמצא מחוץ לולאת_RECV ומוחוץ לבлок WITH. כיוון שכאשר יצאנו מהלולה זו סימן שהלוקו הtentek, נדפס הודעה התנטקות. הפקודה לסגירת SOCKET אינה נוחוצה, כיוון THAT מטפל בכך אוטומטית, אך הוספנו יתר בטחון (בעקבותאג בסגירת THREADS , רצינו לוודא שאכן אין תקליה בסגירת SOCKET שימושה את הפתוח).

Proxy

```
91     # Prepare the proxy socket
92     # * Fill in start (1)
93     proxy_socket.bind(proxy_address)
94     """
95         see explanation about the bind method via server.py, line 97
96     """
97     proxy_socket.listen()
98     """
99         see explanation about the listen method via server.py, line 104
100    """
101   # * Fill in end (1)
```

פקודות אלו הוסבו כבר בהסבירי הקוד של SERVER, והסבירים זהים לchlוטין גם כן.

```
109     # * Fill in start (2)
110     proxy_socket.settimeout(1) # setting a timeout for to accept method. if quit was received,
111     # timeout will make sure new thread will not open
112     client_socket, client_address = proxy_socket.accept()
113     """
114         see explanation about the accept method via server.py, line 120
115     """
116     # * Fill in end (2)
```

גם פקודות אלו הוסבו כבר בהסבירי הקוד של SERVER, והסבירים זהים לchlוטין גם כן. שינוי המיקום של תחילת הTIMEOUT אינו משנה לפקוד פונקציה זו, והוא פועל בדוק כמו SERVER על מנת להגיד TIMEOUT **לא** ACCEPT.

```
123             break
124
125     # added lines-for quit option
126     except socket.timeout:
127         # accept() timed out, check if we should stop
128         if flag_quit:
129             # QUIT was signaled by a handler thread
130             break
131         # If not quitting, just continue waiting for connections.
132         pass
133
134     # end of added lines
```

גם שמי זהה זהה לchlוטין SERVER.

```
139     if flag_quit: # after all threads where closed (finished handling the client) checking if a QUIT request was received
140         try:
141             with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server_socket:
142                 print(f'{proxy_address[0]}:{proxy_address[1]} requesting to terminate server.py')
143                 server_socket.connect((server_host, server_port))
144                 server_socket.sendall("QUIT".encode("utf-8"))
145         except Exception as e:
146             print(f"\nError while closing server: {e}")
147             print(f'{proxy_address[0]}:{proxy_address[1]} terminating proxy...')
148         try:
149             proxy_socket.close() # Close the proxy socket
150             sys.exit(0)
151         except Exception as e:
152             print(f"\nError while closing proxy socket: {e}")
```

טיפול בבקשת הסגירה וה TERMINATION של הרצת הקבצים. כאשר הPROXY מקבל בקשה לסגירה הוא מעביר את הודעה לסגירה אל השרת, ולאחר מכן מטפל בסגירתה SOCKET שלו (של שרת PROXY) ומסיים את הרצת התוכנית בעזרת פקודות (0). sys.exit(). בוצע גם טיפול בשגיאות על ידי -try- ו-except-, ובנוסף - הדפסת הודעות רלוונטיות בטרמינל בפורמט הנדרש.

```

155     # Receive data from the client
156     # * Fill in start (3) #
157     data = client_socket.recv(BUFFSIZE)
158     """
159     see explanation about the recv method via server.py, line 120
160     here we used the same buffer size as used in the server code, since both the proxy socket and the server socket are
161     handling the same data size.
162     """

```

גם הסבר פונקציית `RECV` פה זהה להסביר הפונקציה המקבילה ב`SERVER`, ולכן לא נסביר מחדש.

```

173         # checking if QUIT message was received.
174         if len(data) == 4: # implemented in this scope since request is never shorter than 12 bytes,
175             # a 4 bytes data can't be a request.
176             try:
177                 is_quiting = data.decode("utf-8") # decoding bytes to string
178                 if "QUIT" in is_quiting:
179                     # when QUIT is received, break out of the loop to close the connection, send quit req to server
180                     flag_quit = True
181                     print(f"{client_prefix} Received QUIT, closing connection") # $ Added line $"
182                     break
183             except Exception as e:
184                 print(e)
185             # * Fill in end (3)
186

```

טיפול בבקשת `QUIT`, גם כאן- ההסבר זהה להסביר הבלוק המקורי ב`SERVER`.

```

186
187             if not data: # * Change in start (1)
188                 break
189             # * Change in end (1)

```

גם הסבר הבלוק זה מקביל להסביר של בלוק השרת.

```

218
219             # Send the response back to the client
220             # * Fill in start (4)
221             client_socket.sendall(response)
222             """
223             see explanation about the accept method via server.py, line 199
224             """
225             # * Fill in end (4)
226

```

גם כאן, הסבר מקביל להסביר אצל השרת.

```

232             # * Change in start (2)
233             print(f"{client_prefix} Connection closed")
234             # * Change in end (2)
235

```

כאשר נמצא מילולאת `RECV`, ולאחר מכן מבלוק `WITH` נדפס הודעת התנטקות מהלקות.
דילגנו על שאר הפקודות שהיו בקוד של `SERVER` כיון שאין נחוצות, והוספנו אותן שם ליתר בטחון.

Client

```
62
63     def client(server_address: tuple[str, int], expressions_list: list[api.Expression], show_steps: bool = False,
64             cache_result: bool = False, cache_control: int = api.CalculatorHeader.MAX_CACHE_CONTROL) -> None:
```

שינוי המקוד המקורי - הפונקציה מקבלת רשימה ביטויים ולא ביטוי בודד. כך ניתן לשולח מספר ביטויים ולקבל עליהם תגובה מוביל להתחבר ולהתנתק משרת PROXY לאחר כל שליחת בקשה וקבלת תגובה בודדת.

```
69         for expression in expressions_list:
70             try:
```

הכנסנו לולאת FOR בתוך בלוק WITH, כל מנת לשולח את הביטויים אחד אחריו השינוי אשר החיבור בין השרת והלקוח נשמר.

```
90     # Added lines: message to close proxy and server.
91     def closing_message(server_address: tuple[str, int]):
92         with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as client_socket:
93             client_socket.connect(server_address)
94             try:
95                 message = "QUIT"
96                 print("Sending request to close server")
97                 client_socket.sendall(message.encode("utf-8"))
98             except Exception as exception:
99                 print(exception)
100
101
102     # and of added lines
```

פונקציה שהוספנו על מנת לשולח הודעה סגירה לפרקטי (והפרקטי מעביר את הודעה זו לשרת). הודעה סגירה זו אומרת לפרקטי (ולאחר מכן לשרת) לסגור את הSOCKETS ולאחר מכן TO TERMINATE את התוכנה. זה מאפשר סגירה נוחה יותר לאחר הריצה של שלושת הקבצים, וכך לא נדרש לסגור כל קובץ במקביל. לא נדרש למשוך זאת במטה.

```
148     # * Change in start (2)
149
150         """
151         List of expressions to choose from.
152         """
153         exprList = [
154             # mul_b(div_b(sin_f(max_f(2, mul_b(3, 4), 5, mul_b(div_b(mul_b(7, 8), 9)), div_b(10, 11))), 12), 13),
155             add_b(max_f(2, 3), 3),
156             add_b(3, div_b(mul_b(4, 2), pow_b(sub_b(1, 5), pow_b(2, 3)))),
157             div_b(pow_b(add_b(1, 2), mul_b(3, 4)), mul_b(5, 6)),
158             neg_u(neg_u(pow_b(add_b(1, add_b(2, 3)), neg_u(add_b(4, 5))))),
159             max_f(2, mul_b(3, 4), log_f(e_c), mul_b(6, 7), div_b(9, 8))
160         ]
161
162         """
163             making an expressions list to send to the server.
164             the quit option was implemented to make the termination of the program easier for the programmer,
165             and reducing the number of files that needs too be manually closed after each run.
166         """
```

נמצא בMAIN. הכנסנו את דוגמאות הביטויים לרשימה, וממנה הלקוח יוכל לבחור בביטויים לשילחה. שינוי זה בוצע על מנת להקל על שליחת הביטויים.

```

165
166     exit = ""
167
168     while exit != "-1":
169         expToSend = []
170         print("Enter expression numbers from list to calculate, enter -1 to finish:")
171         expr = input("EXP: ")
172         while expr != "-1":
173             try:
174                 expToSend.append(exprList[int(expr)])
175             except Exception as e:
176                 print("illegal value")
177                 print(e)
178                 expr = input("EXP: ")
179
180             try:
181                 client((api.DEFAULT_PROXY_HOST, api.DEFAULT_PROXY_PORT), expToSend, show_steps, cache_result,
182                         cache_control)
183             except Exception as e:
184                 print("illegal value")
185                 print(e)
186
187             print("\nInstructions: "
188                  "\nType -1 to close the connection, press ENTER to send a new request")
189             exit = input("")
190
191         terminate = input("Type -1 to terminate proxy.py, server.py, client.py: ")
192         if terminate == "-1":
193             closing_message((api.DEFAULT_PROXY_HOST, api.DEFAULT_PROXY_PORT))
194
195 # * Change in end (2)
196

```

לולאה שמתפלת בשילוח הביטויים. מהלך הלולאה:

1. **לולאה עוטפת**: כל עוד לא התקבלה בקשה לסתום:

א. **לולאה פנימית**: כל עוד לא התקבלה בקשה לשילוח הבקשות :

- (i) נקלט את הביטוי מהלקוח (על ידי קליטת מיקומו ברשימה)
- (ii) נוסיף את הביטוי לרשימת הביטויים לשילוח
- (iii) אם התקבל -1 , נצא מהלולאה.

לאחר קבלת בקשה לשילוח הבקשות (הוכנס 1-):

ב. נעביר את רשימת הביטויים לשילוח אל פונקציית CLIENT

ג. פונקציית הלקוח תטפל בשילוח הבקשות ותקבל התשובות.

ד. בסיום הטיפול, נשאל האם הלקוח האם ירצה לסתום את שליחת הבקשות, או האם ירצה לשלח בקשה נוספת:

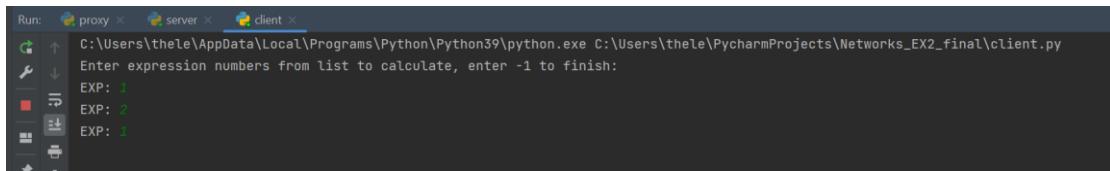
- אם מעוניין בשילוח בקשה נוספת - נישאר בלולאה (נחזיר לՍעיף א)
- אם מעוניין בסיום השליחה (הוכנס 1-) - נצא מהלולאה

2. במידה והלקוח בקש לסתום את שליחת הבקשות, נשאל האם ירצה לבצע TERMINATION לכל קבצי ההריצה של התוכנית:

- אם לא – שרת PROXY יישאר פתוח, הרצת קובץ CLIENT תסתום.
- אם כן (הוכנס 1) – נקרא לפונקציית closing_message שתשלח בקשה סגירה לSERVER (ולאחר מכן, PROXY ישלח בקשה סגירה גם לSERVER).

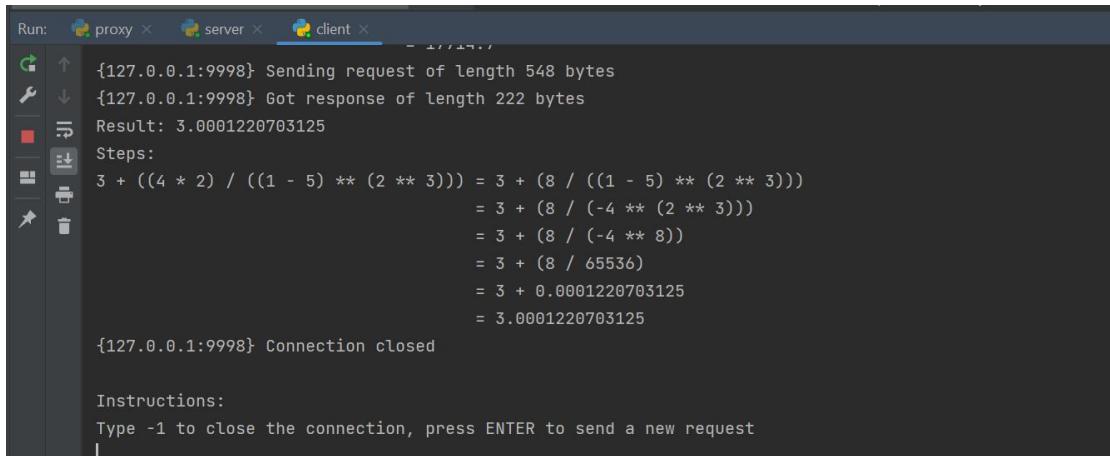
דוגמא להדפסות הטרמינל:

1. הכנסת שלושה ביטויים לשילחה:



```
Run: proxy x server x client x
C:\Users\thele\AppData\Local\Programs\Python\Python39\python.exe C:\Users\thele\PycharmProjects\Networks_EX2_final\client.py
Enter expression numbers from list to calculate, enter -1 to finish:
EXP: 1
EXP: 2
EXP: 3
```

2. הכנסת 1- לסיום קליטת הביטויים:

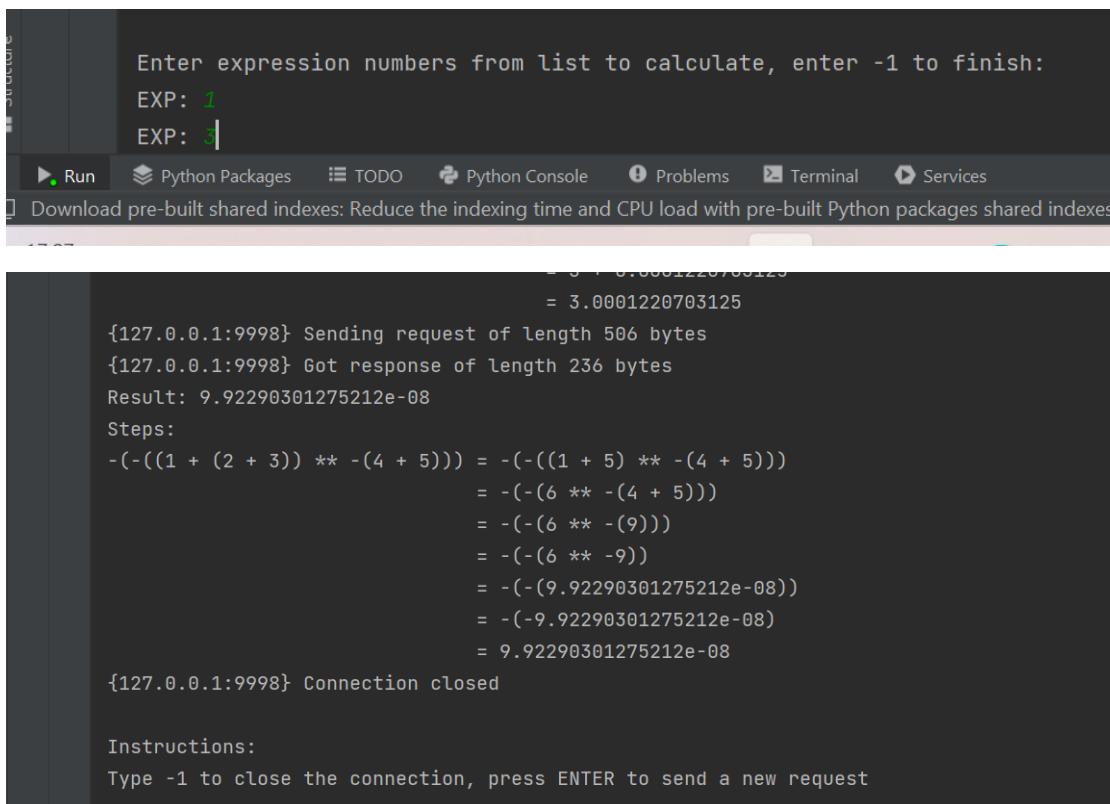


```
Run: proxy x server x client x
{127.0.0.1:9998} Sending request of length 548 bytes
{127.0.0.1:9998} Got response of length 222 bytes
Result: 3.0001220703125
Steps:
3 + ((4 * 2) / ((1 - 5) ** (2 ** 3))) = 3 + (8 / ((1 - 5) ** (2 ** 3)))
= 3 + (8 / (-4 ** (2 ** 3)))
= 3 + (8 / (-4 ** 8))
= 3 + (8 / 65536)
= 3 + 0.0001220703125
= 3.0001220703125
{127.0.0.1:9998} Connection closed

Instructions:
Type -1 to close the connection, press ENTER to send a new request
|
```

בשים לב- קיבלנו את התגובה לכל הבקשות שהוכנסו, תוך שמירה על הקשר עם השרת.
ניתוק הקשר התרחש רק לאחר שליחת כל הביטויים ברשימה וקבלת התגובה אליהם.

3. נדגים שלילה של 2 ביטויים חדשים:



```
Run: Python Packages TODO Python Console Problems Terminal Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes

Enter expression numbers from list to calculate, enter -1 to finish:
EXP: 1
EXP: 3
{127.0.0.1:9998} Sending request of length 506 bytes
{127.0.0.1:9998} Got response of length 236 bytes
Result: 9.92290301275212e-08
Steps:
-(-((1 + (2 + 3)) ** -(4 + 5))) = -(-((1 + 5) ** -(4 + 5)))
= -(-(6 ** -(4 + 5)))
= -(-(6 ** -(9)))
= -(-(6 ** -9))
= -(-(9.92290301275212e-08))
= -(-9.92290301275212e-08)
= 9.92290301275212e-08
{127.0.0.1:9998} Connection closed

Instructions:
Type -1 to close the connection, press ENTER to send a new request
```

4. כעת, נקליד 1 - שוב על מנת לציג شيئا' לנו עוד בקשה לשולחן:

```
Run: proxy x server x client x
if __name__ == "__main__":
    = -(-(6 ** -(9)))
    = -(-(6 ** -9))
    = -(-(9.92290301275212e-08))
    = -(-9.92290301275212e-08)
    = 9.92290301275212e-08
{127.0.0.1:9998} Connection closed

Instructions:
Type -1 to close the connection, press ENTER to send a new request
-1
Type -1 to terminate proxy.py, server.py, client.py: |
```

נשים לב- בשלב זה SERVER והPROXY עדין פתוחים ומאפשרים קבלה של בקשות.

: TERMINATION : 5. נקליד שוב 1 - על מנת לבקש TERMINATION

```
Run: proxy x server x client x
= 9.92290301275212e-08
{127.0.0.1:9998} Connection closed

Instructions:
Type -1 to close the connection, press ENTER to send a new request
-1
Type -1 to terminate proxy.py, server.py, client.py: -1
Sending request to close server

Process finished with exit code 0
|
```

נשים לב- כעת גם SERVER וגם PROXY סגורים, ובוצע TERMINATION להרצות שלהם
כFINE.

נזכיר גם את הדפסות SERVER:

1. שלושת הבקשות הראשונות. ניתן לראות שהחיבור נשמר עבור שלושת הבקשות,
והתנתק רק בסיום שליחת התגובה לבקשת האخונה.

```
Run: proxy x server x client x
C:\Users\thele\AppData\Local\Programs\Python\Python39\python.exe C:\Users\thele\PycharmProjects\Network
Listening on 127.0.0.1:9998
{127.0.0.1:63371} Connected established
{127.0.0.1:63371} Got request of length 548 bytes
{127.0.0.1:63371} Cache miss, response cached ,server time remaining: inf, client time remaining: inf
{127.0.0.1:63371} Sending response of length 222 bytes
{127.0.0.1:63371} Got request of length 487 bytes
{127.0.0.1:63371} Cache miss, response cached ,server time remaining: inf, client time remaining: inf
{127.0.0.1:63371} Sending response of length 164 bytes
{127.0.0.1:63371} Got request of length 548 bytes
{127.0.0.1:63371} Cache hit ,server time remaining: inf, client time remaining: inf
{127.0.0.1:63371} Sending response of length 222 bytes
{127.0.0.1:63371} Connection closed
```

2. שתי הבקשות הנוספות:

```
{127.0.0.1:63371} Connection closed
{127.0.0.1:63628} Connected established
{127.0.0.1:63628} Got request of length 548 bytes
{127.0.0.1:63628} Cache hit ,server time remaining: inf, client time remaining: inf
{127.0.0.1:63628} Sending response of length 222 bytes
{127.0.0.1:63628} Got request of length 506 bytes
{127.0.0.1:63628} Cache miss, response cached ,server time remaining: inf, client time remaining: inf
{127.0.0.1:63628} Sending response of length 236 bytes
{127.0.0.1:63628} Connection closed
```

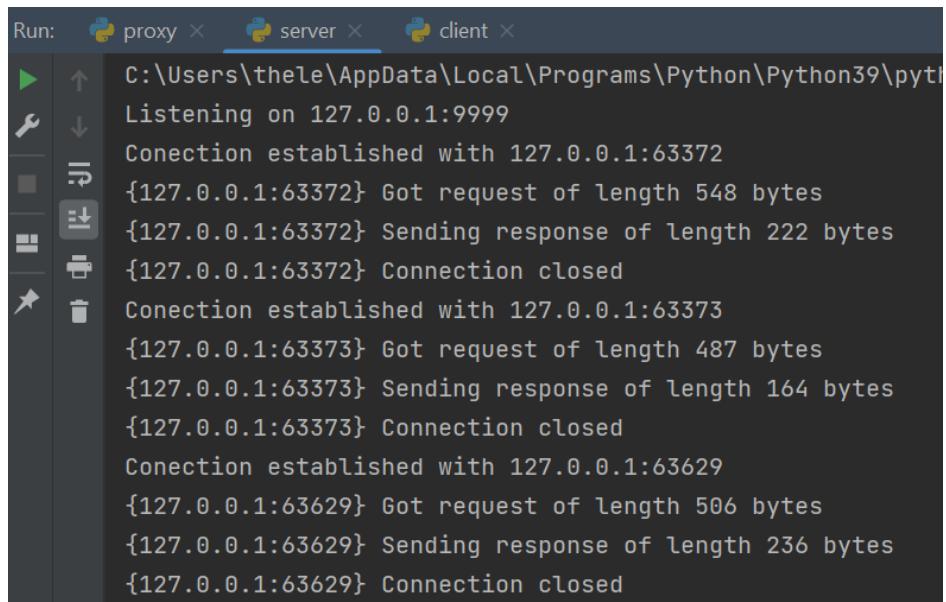
3. בקשת TERMINATION. ניתן לראות שנשלחה בקשה סגירה לשרת ורק לאחר מכן סגרנו את PROXY.

```
{127.0.0.1:63628} Connection closed
{127.0.0.1:64198} Connected established
{127.0.0.1:64198} Received QUIT, closing connection
{127.0.0.1:64198} Connection closed
127.0.0.1:9998 requesting to terminate server.py
127.0.0.1:9998 terminating proxy...

Process finished with exit code 0
```

נציג גם את הדפסות הSERVER

1. חמשת הבקשות הראשונות-



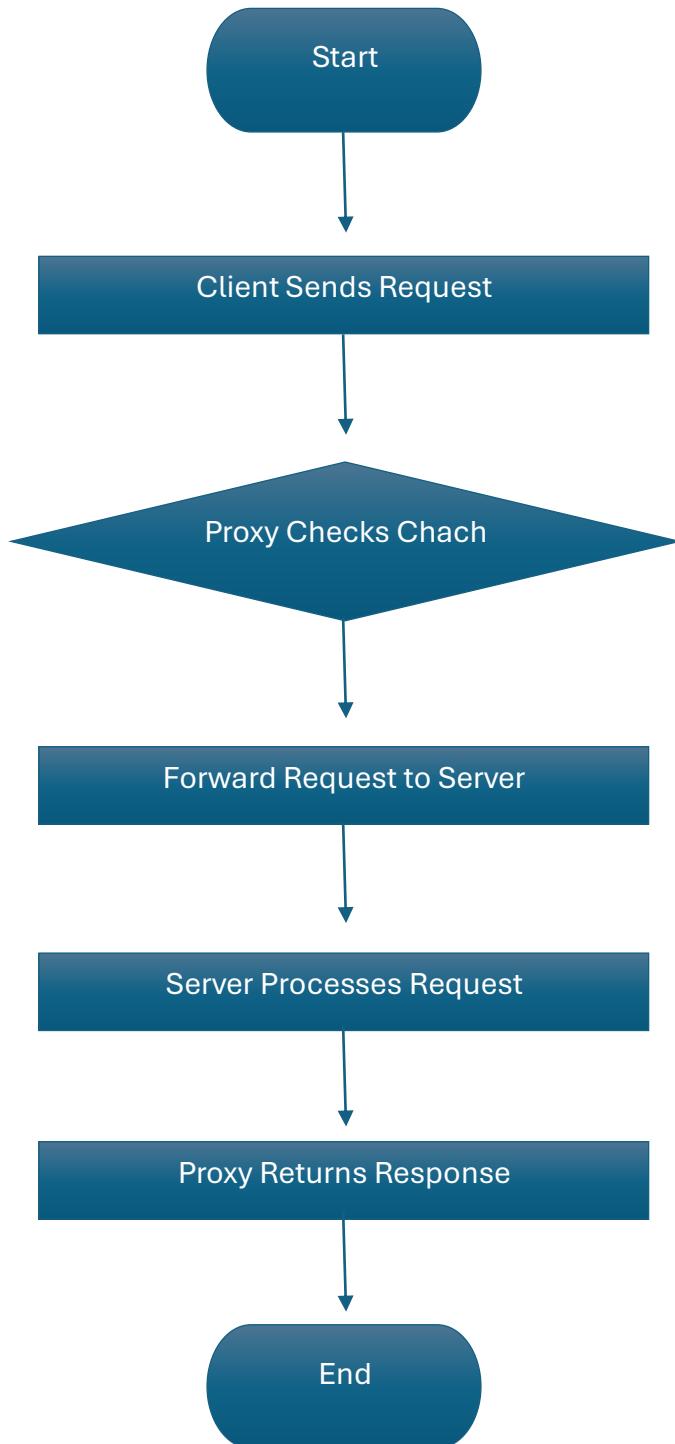
```
Run: proxy × server × client ×
C:\Users\thele\AppData\Local\Programs\Python\Python39\python.exe C:/Users/thele/Desktop/PROXY/proxy.py
Listening on 127.0.0.1:9999
Connection established with 127.0.0.1:63372
{127.0.0.1:63372} Got request of length 548 bytes
{127.0.0.1:63372} Sending response of length 222 bytes
{127.0.0.1:63372} Connection closed
Connection established with 127.0.0.1:63373
{127.0.0.1:63373} Got request of length 487 bytes
{127.0.0.1:63373} Sending response of length 164 bytes
{127.0.0.1:63373} Connection closed
Connection established with 127.0.0.1:63629
{127.0.0.1:63629} Got request of length 506 bytes
{127.0.0.1:63629} Sending response of length 236 bytes
{127.0.0.1:63629} Connection closed
```

נשים לב- אכן יצרנו קשר עם SERVER רק כאשר נשלחו בקשות חדשות שהתגובה אליה עדין לא נשמרה בCACHE.

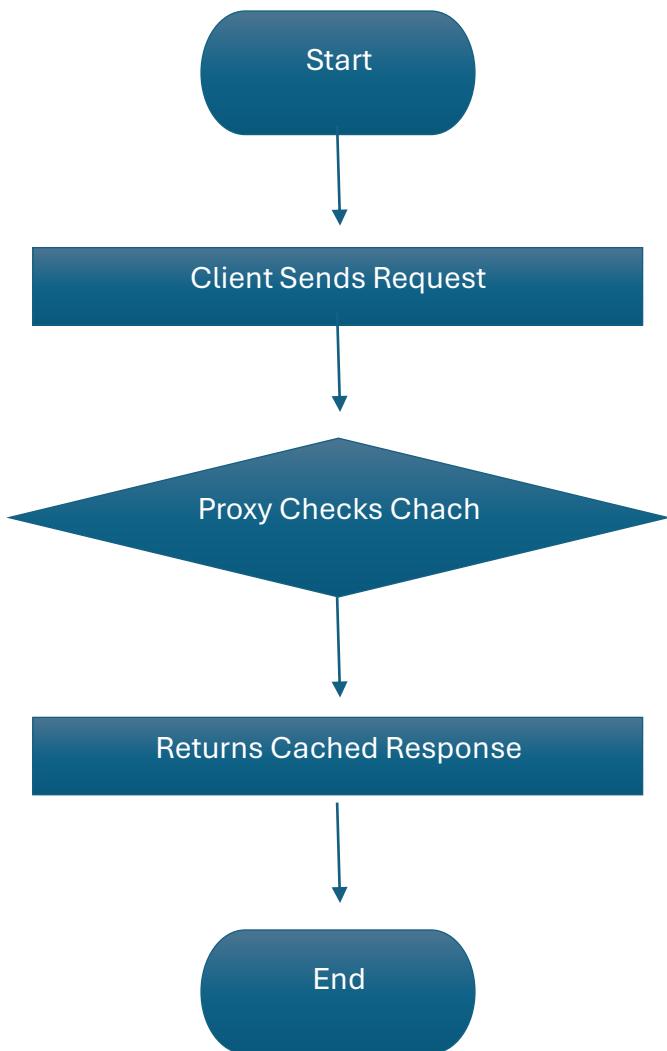
2. בקשת TERMINATION.

```
[127.0.0.1:64201] connection closed
Connection established with 127.0.0.1:64201
QUIT
{127.0.0.1:64201} Connection closed
closing socket...
terminating server.py...

Process finished with exit code 0
```

תרשים זרימה להמחשת מערכת המטען**2.1 האובייקט לא קיים במתומו**

2.2 האובייקט קיים במתמוץ



הציג הרצות ב-Wireshark

3.1 הרצה ראשונה

הרצת SERVER והREQUEST שליחת אותה הבקשה פעמיים ואז בקשה שונה.

Client

```

Run: proxy × server × client ×
{127.0.0.1:9999} Connection established
{127.0.0.1:9999} Sending request of length 548 bytes
{127.0.0.1:9999} Got response of length 222 bytes
Result: 3.0001220703125
Steps:
3 + ((4 * 2) / ((1 - 5) ** (2 ** 3))) = 3 + (8 / ((1 - 5) ** (2 ** 3)))
= 3 + (8 / (-4 ** (2 ** 3)))
= 3 + (8 / (-4 ** 8))
= 3 + (8 / 65536)
= 3 + 0.0001220703125
= 3.0001220703125

{127.0.0.1:9999} Sending request of length 548 bytes
{127.0.0.1:9999} Got response of length 222 bytes
Result: 3.0001220703125
Steps:
3 + ((4 * 2) / ((1 - 5) ** (2 ** 3))) = 3 + (8 / ((1 - 5) ** (2 ** 3)))
= 3 + (8 / (-4 ** (2 ** 3)))
= 3 + (8 / (-4 ** 8))
= 3 + (8 / 65536)
= 3 + 0.0001220703125
= 3.0001220703125

{127.0.0.1:9999} Sending request of length 487 bytes
{127.0.0.1:9999} Got response of length 164 bytes
Result: 17714.7
Steps:
((1 + 2) ** (3 * 4)) / (5 * 6) = (3 ** (3 * 4)) / (5 * 6)
= (3 ** 12) / (5 * 6)
= 531441 / (5 * 6)
= 531441 / 30
= 17714.7

{127.0.0.1:9999} Connection closed

```

Server

```

Run: proxy × server × client ×
C:\Users\thele\AppData\Local\Programs\Python\Python39\python.exe C:\Users\thele\PycharmProjects\
Listening on 127.0.0.1:9999
Connection established with 127.0.0.1:60581
{127.0.0.1:60581} Got request of length 548 bytes
{127.0.0.1:60581} Sending response of length 222 bytes
{127.0.0.1:60581} Got request of length 548 bytes
{127.0.0.1:60581} Sending response of length 222 bytes
{127.0.0.1:60581} Got request of length 487 bytes
{127.0.0.1:60581} Sending response of length 164 bytes
{127.0.0.1:60581} Connection closed

```

נבחן את היקלחת ה-SERVER - סינון לפי פורט 9999 (SERVER)

client_and_server.pcapng

	Info	Length	Protocol	Source	No.
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9999 → 60581 56			TCP	127.0.0.1	632
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 60581 → 9999 56			TCP	127.0.0.1	633
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9999 → 60581 44			TCP	127.0.0.1	634
Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9999 → 60581 592			TCP	127.0.0.1	635
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 60581 → 9999 44			TCP	127.0.0.1	636
Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 60581 → 9999 266			TCP	127.0.0.1	637
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9999 → 60581 44			TCP	127.0.0.1	638
Seq=549 Ack=223 Win=2160896 Len=548 [PSH, ACK] 9999 → 60581 592			TCP	127.0.0.1	639
Seq=223 Ack=1097 Win=2160128 Len=0 [ACK] 60581 → 9999 44			TCP	127.0.0.1	640
Seq=223 Ack=1097 Win=2160128 Len=222 [PSH, ACK] 60581 → 9999 266			TCP	127.0.0.1	641
Seq=1097 Ack=445 Win=2160640 Len=0 [ACK] 9999 → 60581 44			TCP	127.0.0.1	642
Seq=1097 Ack=445 Win=2160640 Len=487 [PSH, ACK] 9999 → 60581 531			TCP	127.0.0.1	643
Seq=445 Ack=1584 Win=2159616 Len=0 [ACK] 60581 → 9999 44			TCP	127.0.0.1	644
Seq=445 Ack=1584 Win=2159616 Len=164 [PSH, ACK] 60581 → 9999 208			TCP	127.0.0.1	645
Seq=1584 Ack=609 Win=2160640 Len=0 [ACK] 9999 → 60581 44			TCP	127.0.0.1	646
Seq=609 Ack=1585 Win=2159616 Len=0 [ACK] 60581 → 9999 44			TCP	127.0.0.1	648
Seq=609 Ack=1585 Win=2159616 Len=0 [FIN, ACK] 60581 → 9999 44			TCP	127.0.0.1	649
Seq=1585 Ack=610 Win=2160640 Len=0 [ACK] 9999 → 60581 44			TCP	127.0.0.1	650

הסבר -

יצירת החיבור בין השרת והלקוח:

Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9999 → 60581 56	TCP	127.0.0.1	632
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 60581 → 9999 56	TCP	127.0.0.1	633
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9999 → 60581 44	TCP	127.0.0.1	634

פקטה ראשונה (632) - פקטת SYN שנשלחה מהלקוח והתקבלת אצל השרת, משמשת ליצירת הקשר הראשוני בין השרת והלקוח.

פקטה שנייה (633) - פקטת SYN-ACK שנשלחה מהשרת והתקבלת אצל הלקוח. מאשרת את קבלת פקטת SYN.

פקטה שלישית (634) - ACK מהלקוח לשרת, מאשרת את קבלת SYN-ACK. לאחר התהיליך זהה מתבצעת לחיצת היד בין הלקוח והשרת.

בקשה ותגובה ראשונה -

בקשה:

Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9999 → 60581 592	TCP	127.0.0.1	635
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 60581 → 9999 44	TCP	127.0.0.1	636

פקטת הבקשה שנשלחה מהלקוח אל השרת. (365)

ACK שנשלח מהשרת אל הלקוח ו מאשרת את קבלת הבקשה. (336)

```
Frame 635: 592 bytes on wire (4736 bits), 592 bytes captured (4736 bits) on interface \Device\NPF_Loopback, id 0 <
           Null/Loopback <
           Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
           Transmission Control Protocol, Src Port: 60581, Dst Port: 9999, Seq: 1, Ack: 1, Len: 548 <
           Data (548 bytes) <
```

נשים לב - אורך הבקשה הראשונה לפני ההדפסות 548 bytes וכן, אורך הדאטה בפקטה זו הוא 548 bytes כנדרש.

תגובה -

Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 60581 → 9999 266	TCP	127.0.0.1 637
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9999 → 60581 44	TCP	127.0.0.1 638

(367) פקטת התגובה שנשלחה מהשרת אל המלצות.

(338) ACK שנשלח ממלכות אל השרת ומאשרת את קבלת התגובה.

Frame 637: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits) on interface \Device\NPF_Loopback, id 0 < Null/Loopback <
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
Transmission Control Protocol, Src Port: 9999, Dst Port: 60581, Seq: 1, Ack: 549, Len: 222 <
Data (222 bytes) <

נשים לב - אורך התגובה הראשונה לפי ההפוסות 222 bytes ואכן, אורך הדאטה בפקטה זו הוא 222 bytes כנדרש.

בקשה ותגובה שנייה + שלישית -

המבנה זהה לבקשתה הראשונה, אך לא נסביר מחדש.

בקשה שנייה:

Seq=549 Ack=223 Win=2160896 Len=548 [PSH, ACK] 9999 → 60581 592	TCP	127.0.0.1 639
Seq=223 Ack=1097 Win=2160128 Len=0 [ACK] 60581 → 9999 44	TCP	127.0.0.1 640
Seq=223 Ack=1097 Win=2160128 Len=222 [PSH, ACK] 60581 → 9999 266	TCP	127.0.0.1 641
Seq=1097 Ack=445 Win=2160640 Len=0 [ACK] 9999 → 60581 44	TCP	127.0.0.1 642

פקטת בקשה -

Frame 639: 592 bytes on wire (4736 bits), 592 bytes captured (4736 bits) on interface \Device\NPF_Loopback, id 0 < Null/Loopback <
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
Transmission Control Protocol, Src Port: 60581, Dst Port: 9999, Seq: 549, Ack: 223, Len: 548 <
Data (548 bytes) <

פקטת תגובה -

Wireshark - Packet 641 · client_and_server.pcapng	-	□	×
Frame 641: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits) on interface \Device\NPF_Loopback, id 0 < Null/Loopback <			
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <			
Transmission Control Protocol, Src Port: 9999, Dst Port: 60581, Seq: 223, Ack: 1097, Len: 222 <			
Data (222 bytes) <			

נשים לב - אורך הבקשתה השנייה שלישית לפי ההפוסות 548 bytes ואורך התגובה השלישי היא 222 bytes ואכן, אורך הדאטה בפקטוות האלו הוא 548 bytes ו- 222 bytes כנדרש.

בקשה שלישית:

Seq=1097 Ack=445 Win=2160640 Len=487 [PSH, ACK] 9999 → 60581 531	TCP	127.0.0.1 643
Seq=445 Ack=1584 Win=2159616 Len=0 [ACK] 60581 → 9999 44	TCP	127.0.0.1 644
Seq=445 Ack=1584 Win=2159616 Len=164 [PSH, ACK] 60581 → 9999 208	TCP	127.0.0.1 645
Seq=1584 Ack=609 Win=2160640 Len=0 [ACK] 9999 → 60581 44	TCP	127.0.0.1 646

פקטת בקשה -

```

Frame 643: 531 bytes on wire (4248 bits), 531 bytes captured (4248 bits) on interface \Device\NPF_Loopback, id 0 <
    Null/Loopback <
        Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
            Transmission Control Protocol, Src Port: 60581, Dst Port: 9999, Seq: 1097, Ack: 445, Len: 487 <
                Data (487 bytes) <

```

פקטת תגובה-

```

Frame 645: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface \Device\NPF_Loopback, id 0 <
    Null/Loopback <
        Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
            Transmission Control Protocol, Src Port: 9999, Dst Port: 60581, Seq: 445, Ack: 1584, Len: 164 <
                Data (164 bytes) <

```

נשים לב - אורך הבקשה השלישית 487 bytes ואורך התגובה השלישית לפי ההdfsוטה
164 bytes ואכן, אורך הדטה בפקטוות האלו הוא 487 bytes ו- 164 bytes כנדרש.

סיום הקשר בין השירות והלקוח-

Seq=1584 Ack=609 Win=2160640 Len=0 [FIN, ACK] 9999 → 60581 44	TCP	127.0.0.1	647
Seq=609 Ack=1585 Win=2159616 Len=0 [ACK] 60581 → 9999 44	TCP	127.0.0.1	648
Seq=609 Ack=1585 Win=2159616 Len=0 [FIN, ACK] 60581 → 9999 44	TCP	127.0.0.1	649
Seq=1585 Ack=610 Win=2160640 Len=0 [ACK] 9999 → 60581 44	TCP	127.0.0.1	650

(647) הלקוח שלוח לשרת בקשה לסיים את התקשרות (פקטת FIN), ככלומר- מעדכן שאין לו עוד דטה לשולח.

(648) מהשרת ללקוח שמאשר את קבלת פקטת FIN.

(649) השרת שלוח ללקוח פקטת FIN על מנת לאשר שאין לו עוד דטה לשולח.

(650) הלקוח שלוח לשרת ACK שמאשר את קבלת פקטת FIN.

לאחר התהיליך זהה הלקוח מתנתק מהשרת. נשים לב- השרת אינו "נסגר" ועודין מאשר התחברות חדשה של הלקוח או התחברות של ללקוחות אחרים.

3.2 הריצה שנייה-

הרכבת SERVER-i PROX ,CLIENT ושליחת אותה הבקשה פעמיים ואז בקשה שוניה-

Client

```
Run: proxy ✘ server ✘ client ✘
▶ {127.0.0.1:9998} Connection established
▶ {127.0.0.1:9998} Sending request of length 548 bytes
▶ {127.0.0.1:9998} Got response of length 222 bytes
Result: 3.0001220703125
Steps:
3 + ((4 * 2) / ((1 - 5) ** (2 ** 3))) = 3 + (8 / ((1 - 5) ** (2 ** 3)))
= 3 + (8 / (-4 ** (2 ** 3)))
= 3 + (8 / (-4 ** 8))
= 3 + (8 / 65536)
= 3 + 0.0001220703125
= 3.0001220703125

{127.0.0.1:9998} Sending request of length 548 bytes
{127.0.0.1:9998} Got response of length 222 bytes
Result: 3.0001220703125
Steps:
3 + ((4 * 2) / ((1 - 5) ** (2 ** 3))) = 3 + (8 / ((1 - 5) ** (2 ** 3)))
= 3 + (8 / (-4 ** (2 ** 3)))
= 3 + (8 / (-4 ** 8))
= 3 + (8 / 65536)
= 3 + 0.0001220703125
= 3.0001220703125

{127.0.0.1:9998} Sending request of length 487 bytes
{127.0.0.1:9998} Got response of length 164 bytes
Result: 17714.7
Steps:
((1 + 2) ** (3 * 4)) / (5 * 6) = (3 ** (3 * 4)) / (5 * 6)
= (3 ** 12) / (5 * 6)
= 531441 / (5 * 6)
= 531441 / 30
= 17714.7

{127.0.0.1:9998} Connection closed
```

Server

```
Run: proxy ✘ server ✘ client ✘
▶ C:\Users\thele\AppData\Local\Programs\Python\Python39\python.exe
Listening on 127.0.0.1:9999
Connection established with 127.0.0.1:63217
{127.0.0.1:63217} Got request of length 548 bytes
{127.0.0.1:63217} Sending response of length 222 bytes
{127.0.0.1:63217} Connection closed
Connection established with 127.0.0.1:63218
{127.0.0.1:63218} Got request of length 487 bytes
{127.0.0.1:63218} Sending response of length 164 bytes
{127.0.0.1:63218} Connection closed
```

Proxy

```

Run: proxy × server × client ×
C:\Users\thele\AppData\Local\Programs\Python\Python39\python.exe C:\Users\thele\PycharmProjects\Networks_Ex
Listening on 127.0.0.1:9998
{127.0.0.1:63216} Connected established
{127.0.0.1:63216} Got request of length 548 bytes
{127.0.0.1:63216} Cache miss, response cached ,server time remaining: inf, client time remaining: inf
{127.0.0.1:63216} Sending response of length 222 bytes
{127.0.0.1:63216} Got request of length 548 bytes
{127.0.0.1:63216} Cache hit ,server time remaining: inf, client time remaining: inf
{127.0.0.1:63216} Sending response of length 222 bytes
{127.0.0.1:63216} Got request of length 487 bytes
{127.0.0.1:63216} Cache miss, response cached ,server time remaining: inf, client time remaining: inf
{127.0.0.1:63216} Sending response of length 164 bytes
{127.0.0.1:63216} Connection closed

```

נבחן את היקלחת ה-Wireshark - **סינון לפי פורט SERVER (9999) ולפי פורט 9998 (PROXY)**

יצירת החיבור בין השרת PROXY והלקוח:

	Info	Length	Protocol	Source	No.
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9998 → 63216 56	TCP	127.0.0.1 221			
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 63216 → 9998 56	TCP	127.0.0.1 222			
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9998 → 63216 44	TCP	127.0.0.1 223			

זהה להרצה קודמת.

בקשה ותגובה ראשונה -

Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9998 → 63216 592	TCP	127.0.0.1 224
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 63216 → 9998 44	TCP	127.0.0.1 225
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9999 → 63217 56	TCP	127.0.0.1 226
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 63217 → 9999 56	TCP	127.0.0.1 227
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9999 → 63217 44	TCP	127.0.0.1 228
Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9999 → 63217 592	TCP	127.0.0.1 229
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 63217 → 9999 44	TCP	127.0.0.1 230
Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 63217 → 9999 266	TCP	127.0.0.1 231
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9999 → 63217 44	TCP	127.0.0.1 232
Seq=549 Ack=223 Win=2160896 Len=0 [FIN, ACK] 9999 → 63217 44	TCP	127.0.0.1 233
Seq=223 Ack=550 Win=2160640 Len=0 [ACK] 63217 → 9999 44	TCP	127.0.0.1 234
Seq=223 Ack=550 Win=2160640 Len=0 [FIN, ACK] 63217 → 9999 44	TCP	127.0.0.1 235
Seq=550 Ack=224 Win=2160896 Len=0 [ACK] 9999 → 63217 44	TCP	127.0.0.1 236
Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 63216 → 9998 266	TCP	127.0.0.1 237
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9998 → 63216 44	TCP	127.0.0.1 238

חלק ראשון - שליחת הבקשה לשרת PROXY

Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9998 → 63216 592	TCP	127.0.0.1 224
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 63216 → 9998 44	TCP	127.0.0.1 225

פרקטת הבקשה שנשלחה לproxy'ו -

Frame 224: 592 bytes on wire (4736 bits), 592 bytes captured (4736 bits) on interface \Device\NPF_Loopback, id 0 < Null/Loopback <
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
Transmission Control Protocol, Src Port: 63216, Dst Port: 9998, Seq: 1, Ack: 1, Len: 548 <
Data (548 bytes) <

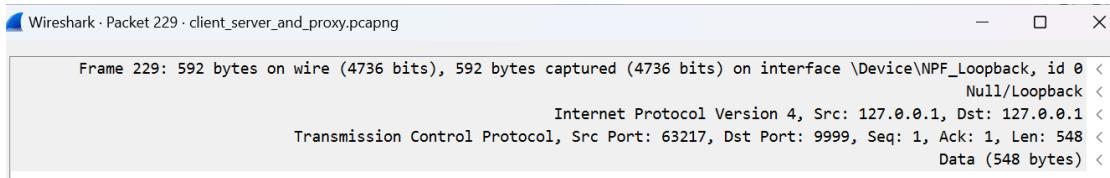
חלק שני - כיוון שהבקשה לא נמצאת במטמון, היא מועברת לSERVER .

א- יצירת החיבור עם השרת-

Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9999 → 63217 56	TCP	127.0.0.1 226
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 63217 → 9999 56	TCP	127.0.0.1 227
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9999 → 63217 44	TCP	127.0.0.1 228

ב- העברת הבקשה לשרת-

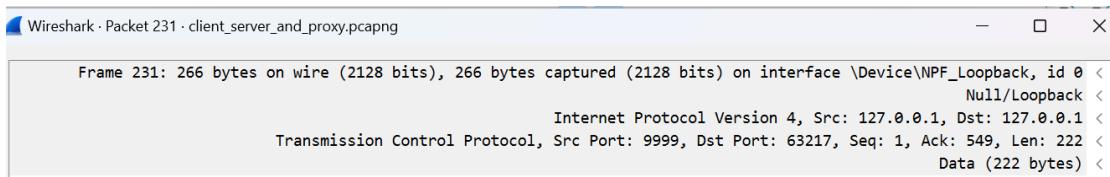
Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9999 → 63217	592	TCP	127.0.0.1	229
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 63217 → 9999	44	TCP	127.0.0.1	230



ג- קבלת תגובה מהשרת-

Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 63217 → 9999	266	TCP	127.0.0.1	231
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9999 → 63217	44	TCP	127.0.0.1	232

פקטת התגובה



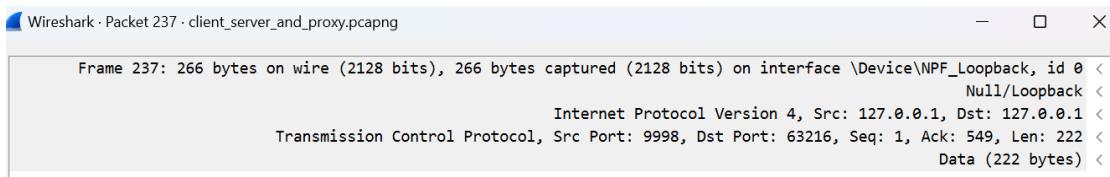
ד- סיום התקשרות עם השרת-

Seq=549 Ack=223 Win=2160896 Len=0 [FIN, ACK] 9999 → 63217	44	TCP	127.0.0.1	233
Seq=223 Ack=550 Win=2160640 Len=0 [ACK] 63217 → 9999	44	TCP	127.0.0.1	234
Seq=223 Ack=550 Win=2160640 Len=0 [FIN, ACK] 63217 → 9999	44	TCP	127.0.0.1	235
Seq=550 Ack=224 Win=2160896 Len=0 [ACK] 9999 → 63217	44	TCP	127.0.0.1	236

ה- העברת התגובה אל הלוקו-

Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 63216 → 9998	266	TCP	127.0.0.1	237
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9998 → 63216	44	TCP	127.0.0.1	238

פקטת התגובה



בקשה ותגובה שנייה-

נ筹ה כמו כן, שהתגובה תהיה במתמונן ולכן שרת הפרויקט יחזיר את התשובה ללא פניה לשרת.

Seq=549 Ack=223 Win=2160896 Len=548 [PSH, ACK] 9998 → 63216	592	TCP	127.0.0.1	239
Seq=223 Ack=1097 Win=2160128 Len=0 [ACK] 63216 → 9998	44	TCP	127.0.0.1	240
Seq=223 Ack=1097 Win=2160128 Len=222 [PSH, ACK] 63216 → 9998	266	TCP	127.0.0.1	241
Seq=1097 Ack=445 Win=2160640 Len=0 [ACK] 9998 → 63216	44	TCP	127.0.0.1	242

שליחת הבקשה לשרת הPROXY

Seq=549 Ack=223 Win=2160896 Len=548 [PSH, ACK] 9998 → 63216	592	TCP	127.0.0.1	239
Seq=223 Ack=1097 Win=2160128 Len=0 [ACK] 63216 → 9998	44	TCP	127.0.0.1	240

פקטת הבקשה

Wireshark · Packet 239 · client_server_and_proxy.pcapng						
Frame 239: 592 bytes on wire (4736 bits), 592 bytes captured (4736 bits) on interface \Device\NPF_Loopback, id 0 <						
Null/Loopback <						
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <						
Transmission Control Protocol, Src Port: 63216, Dst Port: 9998, Seq: 549, Ack: 223, Len: 548 <						
Data (548 bytes) <						

קבלת תגובה משרת PROXY

Seq=223 Ack=1097 Win=2160128 Len=222 [PSH, ACK] 63216 → 9998 266	TCP	127.0.0.1	241
Seq=1097 Ack=445 Win=2160640 Len=0 [ACK] 9998 → 63216 44	TCP	127.0.0.1	242

פקטת התגובה

Wireshark · Packet 241 · client_server_and_proxy.pcapng						
Frame 241: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits) on interface \Device\NPF_Loopback, id 0 <						
Null/Loopback <						
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <						
Transmission Control Protocol, Src Port: 9998, Dst Port: 63216, Seq: 223, Ack: 1097, Len: 222 <						
Data (222 bytes) <						

בקשה ותגובה שלישית-

Seq=1097 Ack=445 Win=2160640 Len=487 [PSH, ACK] 9998 → 63216 531	TCP	127.0.0.1	243
Seq=445 Ack=1584 Win=2159616 Len=0 [ACK] 63216 → 9998 44	TCP	127.0.0.1	244
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9999 → 63218 56	TCP	127.0.0.1	245
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 63218 → 9999 56	TCP	127.0.0.1	246
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9999 → 63218 44	TCP	127.0.0.1	247
Seq=1 Ack=1 Win=2161152 Len=487 [PSH, ACK] 9999 → 63218 531	TCP	127.0.0.1	248
Seq=1 Ack=488 Win=2160640 Len=0 [ACK] 63218 → 9999 44	TCP	127.0.0.1	249
Seq=1 Ack=488 Win=2160640 Len=164 [PSH, ACK] 63218 → 9999 208	TCP	127.0.0.1	250
Seq=488 Ack=165 Win=2161152 Len=0 [ACK] 9999 → 63218 44	TCP	127.0.0.1	251
Seq=488 Ack=165 Win=2161152 Len=0 [FIN, ACK] 9999 → 63218 44	TCP	127.0.0.1	252
Seq=165 Ack=489 Win=2160640 Len=0 [ACK] 63218 → 9999 44	TCP	127.0.0.1	253
Seq=165 Ack=489 Win=2160640 Len=0 [FIN, ACK] 63218 → 9999 44	TCP	127.0.0.1	254
Seq=489 Ack=166 Win=2161152 Len=0 [ACK] 9999 → 63218 44	TCP	127.0.0.1	255

חלק ראשון- שליחת הבקשה לשרת PROXY

Seq=1097 Ack=445 Win=2160640 Len=487 [PSH, ACK] 9998 → 63216 531	TCP	127.0.0.1	243
Seq=445 Ack=1584 Win=2159616 Len=0 [ACK] 63216 → 9998 44	TCP	127.0.0.1	244

פקטת הבקשה שנשלחה לproxy

Wireshark · Packet 243 · client_server_and_proxy.pcapng						
Frame 243: 531 bytes on wire (4248 bits), 531 bytes captured (4248 bits) on interface \Device\NPF_Loopback, id 0 <						
Null/Loopback <						
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <						
Transmission Control Protocol, Src Port: 63216, Dst Port: 9998, Seq: 1097, Ack: 445, Len: 487 <						
Data (487 bytes) <						

חלק שני- צוין שהבקשה לא נמצאת במתמונן, היא מועברת לSERVER

א- יצירת החיבור עם השרת-

Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9999 → 63218 56	TCP	127.0.0.1	245
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 63218 → 9999 56	TCP	127.0.0.1	246
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9999 → 63218 44	TCP	127.0.0.1	247

ב- העברת הבקשה לשרת-

Seq=1 Ack=1 Win=2161152 Len=487 [PSH, ACK] 9999 → 63218	531	TCP	127.0.0.1	248
Seq=1 Ack=488 Win=2160640 Len=0 [ACK] 63218 → 9999	44	TCP	127.0.0.1	249

פקטת הבקשה שנשלחה לשרת

Wireshark - Packet 248 · client_server_and_proxy.pcapng				
Frame 248: 531 bytes on wire (4248 bits), 531 bytes captured (4248 bits) on interface \Device\NPF_Loopback, id 0 < Null/Loopback < Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 < Transmission Control Protocol, Src Port: 63218, Dst Port: 9999, Seq: 1, Ack: 1, Len: 487 < Data (487 bytes) <				

ג- קבלת תגובה מהשרת-

Seq=1 Ack=488 Win=2160640 Len=164 [PSH, ACK] 63218 → 9999	208	TCP	127.0.0.1	250
Seq=488 Ack=165 Win=2161152 Len=0 [ACK] 9999 → 63218	44	TCP	127.0.0.1	251

פקטת התגובה

Wireshark - Packet 250 · client_server_and_proxy.pcapng				
Frame 250: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface \Device\NPF_Loopback, id 0 < Null/Loopback < Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 < Transmission Control Protocol, Src Port: 9999, Dst Port: 63218, Seq: 1, Ack: 488, Len: 164 < Data (164 bytes) <				

ד- סיום התקשרות עם השרת-

Seq=488 Ack=165 Win=2161152 Len=0 [FIN, ACK] 9999 → 63218	44	TCP	127.0.0.1	252
Seq=165 Ack=489 Win=2160640 Len=0 [ACK] 63218 → 9999	44	TCP	127.0.0.1	253
Seq=165 Ack=489 Win=2160640 Len=0 [FIN, ACK] 63218 → 9999	44	TCP	127.0.0.1	254
Seq=489 Ack=166 Win=2161152 Len=0 [ACK] 9999 → 63218	44	TCP	127.0.0.1	255

ה- העברת התגובה אל הלוקו-

Seq=445 Ack=1584 Win=2159616 Len=164 [PSH, ACK] 63216 → 9998	208	TCP	127.0.0.1	256
Seq=1584 Ack=609 Win=2160640 Len=0 [ACK] 9998 → 63216	44	TCP	127.0.0.1	257

פקטת התגובה

Wireshark - Packet 256 · client_server_and_proxy.pcapng				
Frame 256: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface \Device\NPF_Loopback, id 0 < Null/Loopback < Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 < Transmission Control Protocol, Src Port: 9998, Dst Port: 63216, Seq: 445, Ack: 1584, Len: 164 < Data (164 bytes) <				

סיום הקשר בין שרת הפרוקסי לлокו-

Seq=1584 Ack=609 Win=2160640 Len=0 [FIN, ACK] 9998 → 63216	44	TCP	127.0.0.1	258
Seq=609 Ack=1585 Win=2159616 Len=0 [ACK] 63216 → 9998	44	TCP	127.0.0.1	259
Seq=609 Ack=1585 Win=2159616 Len=0 [FIN, ACK] 63216 → 9998	44	TCP	127.0.0.1	260
Seq=1585 Ack=610 Win=2160640 Len=0 [ACK] 9998 → 63216	44	TCP	127.0.0.1	261

3.3 הריצה שלישית-

הריצת SERVER-PROX ושליחת בקשה, סגירת SERVER, CLIENT, שליחת הבקשה שוב ואז שליחת בקשה שונה.

Client

```

Run: proxy x client x
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 548 bytes
{127.0.0.1:9998} Got response of length 222 bytes
Result: 3.0001220703125
Steps:
3 + ((4 * 2) / ((1 - 5) ** (2 ** 3))) = 3 + (8 / ((1 - 5) ** (2 ** 3)))
= 3 + (8 / (-4 ** (2 ** 3)))
= 3 + (8 / (-4 ** 8))
= 3 + (8 / 65536)
= 3 + 0.0001220703125
= 3.0001220703125

Run: proxy x client x
Enter expression numbers from list to calculate, enter -1 to finish:
EXP: 1
EXP: 2
EXP: -1
{127.0.0.1:9998} Connection established
{127.0.0.1:9998} Sending request of length 548 bytes
{127.0.0.1:9998} Got response of length 222 bytes
Result: 3.0001220703125
Steps:
3 + ((4 * 2) / ((1 - 5) ** (2 ** 3))) = 3 + (8 / ((1 - 5) ** (2 ** 3)))
= 3 + (8 / (-4 ** (2 ** 3)))
= 3 + (8 / (-4 ** 8))
= 3 + (8 / 65536)
= 3 + 0.0001220703125
= 3.0001220703125
{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

```

Proxy

```

Run: proxy x client x
C:\Users\thele\AppData\Local\Programs\Python\Python39\python.exe C:\Users\thele\PycharmProjects\Networks_EX2_final\proxy.py
Listening on 127.0.0.1:9998
{127.0.0.1:65173} Connected established
{127.0.0.1:65173} Got request of length 548 bytes
{127.0.0.1:65173} Cache miss, response cached ,server time remaining: inf, client time remaining: inf
{127.0.0.1:65173} Sending response of length 222 bytes
{127.0.0.1:65173} Connection closed
{127.0.0.1:65185} Connected established
{127.0.0.1:65185} Got request of length 548 bytes
{127.0.0.1:65185} Cache hit ,server time remaining: inf, client time remaining: inf
{127.0.0.1:65185} Sending response of length 222 bytes
{127.0.0.1:65185} Got request of length 487 bytes
Unexpected server error: Connection refused by server and the request was not in the cache/it was stale
{127.0.0.1:65185} Connection closed

```

נבחן את הקלטת ה-Wireshark - סינון לפי פורט SERVER (9999) ולפי פורט 9998 (PROXY)

יצירת החיבור בין השרת SERVER והלקוח:

Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9998 → 65173 56	TCP	127.0.0.1 218
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 65173 → 9998 56	TCP	127.0.0.1 219
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9998 → 65173 44	TCP	127.0.0.1 220

זהה להרצה קודמת.

בקשה ותגובה ראשונה -

זהה לשאליה קודמת.

Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9998 → 65173 592	TCP	127.0.0.1 221
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 65173 → 9998 44	TCP	127.0.0.1 222
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9999 → 65174 56	TCP	127.0.0.1 223
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 65174 → 9999 56	TCP	127.0.0.1 224
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9999 → 65174 44	TCP	127.0.0.1 225
Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9999 → 65174 592	TCP	127.0.0.1 226
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 65174 → 9999 44	TCP	127.0.0.1 227
Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 65174 → 9999 266	TCP	127.0.0.1 228
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9999 → 65174 44	TCP	127.0.0.1 229
Seq=549 Ack=223 Win=2160896 Len=0 [FIN, ACK] 9999 → 65174 44	TCP	127.0.0.1 230
Seq=223 Ack=550 Win=2160640 Len=0 [ACK] 65174 → 9999 44	TCP	127.0.0.1 231
Seq=223 Ack=550 Win=2160640 Len=0 [FIN, ACK] 65174 → 9999 44	TCP	127.0.0.1 232
Seq=550 Ack=224 Win=2160896 Len=0 [ACK] 9999 → 65174 44	TCP	127.0.0.1 233
Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 65173 → 9998 266	TCP	127.0.0.1 234
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9998 → 65173 44	TCP	127.0.0.1 235

שליחת בקשה לפרקטי 221-222

חיבור לשרת 223-225

העברת בקשה וקבלת תגובה מהשרת 226-229

סיום התקשרות עם השרת 230-233

העברת התגובה על ידי שרת הפרוקטי ללקוח 234-235

מקטע נסף-

צורת השיליחה של בקשות מלהקוח בקוד של מלאצת סגירה של החיבור עם שרת הPROXY והתחברות מחדש על מנת לסגור את SERVER במהלך הרצה.

Seq=549 Ack=223 Win=2160896 Len=0 [FIN, ACK] 9998 → 65173 44	TCP	127.0.0.1 236
Seq=223 Ack=550 Win=2160640 Len=0 [ACK] 65173 → 9998 44	TCP	127.0.0.1 237
Seq=223 Ack=550 Win=2160640 Len=0 [FIN, ACK] 65173 → 9998 44	TCP	127.0.0.1 238
Seq=550 Ack=224 Win=2160896 Len=0 [ACK] 9998 → 65173 44	TCP	127.0.0.1 239
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9998 → 65185 56	TCP	127.0.0.1 507
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN, ACK] 65185 → 9998 56	TCP	127.0.0.1 508
Seq=1 Ack=1 Win=2161152 Len=0 [ACK] 9998 → 65185 44	TCP	127.0.0.1 509

בקשה ותגובה שנייה-

נכח כמובן, שהתגובה תהיה במתמונן ולכן שרת הפרוקטי יחזיר את התשובה ללא פניה לשרת. לכן סיום התקשרות עם השרת לא ישפיע על קבלת התגובה

Seq=1 Ack=1 Win=2161152 Len=548 [PSH, ACK] 9998 → 65185 592	TCP	127.0.0.1 510
Seq=1 Ack=549 Win=2160640 Len=0 [ACK] 65185 → 9998 44	TCP	127.0.0.1 511
Seq=1 Ack=549 Win=2160640 Len=222 [PSH, ACK] 65185 → 9998 266	TCP	127.0.0.1 512
Seq=549 Ack=223 Win=2160896 Len=0 [ACK] 9998 → 65185 44	TCP	127.0.0.1 513

שליחת הבקשה לשרת הPROXY

510-511

פרקטת הבקשה

Wireshark - Packet 510 · 3.3_ex2.pcapng

```
Frame 510: 592 bytes on wire (4736 bits), 592 bytes captured (4736 bits) on interface \Device\NPF_Loopback, id 0 <
    Null/Loopback <
        Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
        Transmission Control Protocol, Src Port: 65185, Dst Port: 9998, Seq: 1, Ack: 1, Len: 548 <
            Data (548 bytes) <
```

קבלת תגובה משרת PROXY

512-513

פקטת התגובה

Wireshark - Packet 512 · 3.3_ex2.pcapng

```
Frame 512: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits) on interface \Device\NPF_Loopback, id 0 <
    Null/Loopback <
        Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
        Transmission Control Protocol, Src Port: 9998, Dst Port: 65185, Seq: 1, Ack: 549, Len: 222 <
            Data (222 bytes) <
```

בקשה ותגובה שלישיית

שליחת הבקשה לשרת PROXY

Seq=549 Ack=223 Win=2160896 Len=487 [PSH, ACK] 9998 → 65185 531	TCP	127.0.0.1 514
Seq=223 Ack=1036 Win=2160128 Len=0 [ACK] 65185 → 9998 44	TCP	127.0.0.1 515

פקטת הבקשה שנשלחה לפרויקט

Wireshark - Packet 514 · 3.3_ex2.pcapng

```
Frame 514: 531 bytes on wire (4248 bits), 531 bytes captured (4248 bits) on interface \Device\NPF_Loopback, id 0 <
    Null/Loopback <
        Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <
        Transmission Control Protocol, Src Port: 65185, Dst Port: 9998, Seq: 549, Ack: 223, Len: 487 <
            Data (487 bytes) <
```

שלק שני – כיוון שהבקשה לא נמצאת במטמון, היא מועברת לSERVER. כיוון שסגרנו את השרת נצפה לשגיאה.

א- בקשה לייצרת החיבור עם השרת-

Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [SYN] 9999 → 65186 56	TCP	127.0.0.1 516
--	-----	---------------

ב- חוסר הצלחה להתחבר לשרת-

Seq=1 Ack=1 Win=0 Len=0 [RST, ACK] 65186 → 9999 44	TCP	127.0.0.1 517
SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [TCP Port numbers reused] 56	TCP	127.0.0.1 522
Seq=1 Ack=1 Win=0 Len=0 [RST, ACK] 65186 → 9999 44	TCP	127.0.0.1 523
SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [TCP Port numbers reused] 56	TCP	127.0.0.1 524
Seq=1 Ack=1 Win=0 Len=0 [RST, ACK] 65186 → 9999 44	TCP	127.0.0.1 525
SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [TCP Port numbers reused] 56	TCP	127.0.0.1 541
Seq=1 Ack=1 Win=0 Len=0 [RST, ACK] 65186 → 9999 44	TCP	127.0.0.1 542
SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM [TCP Port numbers reused] 56	TCP	127.0.0.1 543
Seq=1 Ack=1 Win=0 Len=0 [RST, ACK] 65186 → 9999 44	TCP	127.0.0.1 544

ג- העברת השגיאה משרת PROXY אל הלוקו-

Seq=223 Ack=1036 Win=2160128 Len=170 [PSH, ACK] 65185 → 9998 214	TCP	127.0.0.1 545
Seq=1036 Ack=393 Win=2160896 Len=0 [ACK] 9998 → 65185 44	TCP	127.0.0.1 546

Wireshark · Packet 545 · 3.3_ex2.pcapng

Frame 545: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface \Device\NPF_Loopback, id 0 <--> Null/Loopback

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 <--> Transmission Control Protocol, Src Port: 9998, Dst Port: 65185, Seq: 223, Ack: 1036, Len: 170 <--> Data (170 bytes)

סיום הקשר בין השירות ללקוח-

Seq=1036 Ack=393 Win=2160896 Len=0 [FIN, ACK] 9998 → 65185 44	TCP	127.0.0.1 547
Seq=393 Ack=1037 Win=2160128 Len=0 [ACK] 65185 → 9998 44	TCP	127.0.0.1 548
Seq=393 Ack=1037 Win=2160128 Len=0 [FIN, ACK] 65185 → 9998 44	TCP	127.0.0.1 549
Seq=1037 Ack=394 Win=2160896 Len=0 [ACK] 9998 → 65185 44	TCP	127.0.0.1 550

שאלות עיוניות

א. חישוב זמן טעינה הדף כולל התמונות

נתונים:

- גודל הדף: F Kbyte
- מספר תמונות: X
- גודל כל תמונה: 4F Kbyte
- זמן עיבוד של כל נתב: p
- קצב השידור: R
- אורך קו: x
- מהירות התפשטות בקווים: s

סה"כ גודל הדף והתמונות:

$$\text{גודל הדף והתמונות} = F + X \cdot 4F = F \cdot (1 + 4X)$$

נחשב קודם את ה-RTT, T_{prop} , T_{tran}

$$T_{prop} = \text{Propagation Delay} = \frac{\text{length of the link}}{\text{velocity of a bit in the link}} =$$

$$\frac{\text{אורך קו}}{\text{מהירות התפשטות}} = \frac{x}{s}$$

$$T_{tran} = \text{Transmission Delay} = \frac{\text{Packet size}}{\text{Transmission rate}} =$$

$$\frac{\text{גודל תמונה}}{\text{קצב השידור}} = \frac{4F}{R}$$

$$RTT = d_{propagation} + d_{process} + d_{queue}$$

נחשב כל אחד מהগורמים קודם:

$$d_{propagation} = 2 \times \frac{\text{אורך קו}}{\text{מהירות התפשטות}} = \frac{2x}{s} \quad (\text{נכפול ב2 בגלל ההליך חזר})$$

ב- $d_{process}$ יש 2 נתבים, ובכל אחד זמן העיבוד הוא d .
מכיוון שהחbillה עוברת את הנתבים הלווי וחזר נכפול שוב ב-2:

$$d_{process} = 2 \times 2 \times d = 4d$$

d_{queue} לא נתון אך ניתן ליחסו כזניח:

$$d_{queue} = 0$$

לכן יצא RTT:

$$RTT = \frac{2x}{s} + 4d$$

עכשו שיש את כל הנתונים אפשר לחשב עבור כל אחד מהמרקטים:

1. HTTP Non-Persistent

במקרה הזה הזמן הוא:

$$\begin{aligned} RTT + RTT + T_{prop} + X(RTT + RTT + T_{tran}) &= \\ \left(\frac{2x}{s} + 4d \right) + \left(\frac{2x}{s} + 4d \right) + \frac{x}{s} + X \left(\frac{2x}{s} + 4d + \frac{2x}{s} + 4d + \frac{4F}{R} \right) &= \\ \left(\frac{4x}{s} + 8d \right) + \frac{x}{s} + X \left(\frac{4x}{s} + 8d + \frac{4F}{R} \right) &= (X+1) \left(\frac{4x}{s} + 8d \right) + \frac{x}{s} + \frac{4Fx}{R} \end{aligned}$$

2. HTTP Persistent

כמו במקרה הראשון אך בלי תוספת RTT לכל אובייקט (כי דרך חיבור TCP אחד ניתן להעביר מספר אובייקטים)

ולכן הזמן הוא:

$$\begin{aligned} RTT + RTT + T_{prop} + X(RTT + T_{tran}) &= \\ \left(\frac{2x}{s} + 4d \right) + \left(\frac{2x}{s} + 4d \right) + \frac{x}{s} + X \left(\frac{2x}{s} + 4d + \frac{4F}{R} \right) &= \\ \left(\frac{4x}{s} + 8d \right) + \frac{x}{s} + X \left(\frac{2x}{s} + 4d + \frac{4F}{R} \right) & \end{aligned}$$

HTTP Persistent with Pipelining .3

כמו במקרה הראשון אך בלי תוספת RTT לכל אובייקט (כי דרך חיבור TCP אחד ניתן להעביר מספר אובייקטים). כיוון שאנחנו עובדים עם שידור עקב/צנרת, אז יש לנו אפשרות לשלוח את כל הבקשות אחת אחרי השניה, דבר שיוצר שידור רצוף של כלל המידע מהשרת.

ולכן הזמן יהיה:

$$\begin{aligned} RTT + RTT + T_{prop} + RTT + X \cdot T_{tran} &= 3RTT + T_{prop} + X \cdot T_{tran} = \\ 3 \cdot \left(\frac{2x}{s} + 4d \right) + \frac{x}{s} + \frac{4Fx}{R} & \end{aligned}$$

ב. הסבר ההבדלים בין השיטות

HTTP Non-Persistent: .1

- כל בקשה דורשת חיבור חדש, שכן זמן הטעינה עשוי להיות ארוך יותר.
- כל בקשה דורשת זמן של שידור, עיבוד ותקשורת מחדש עם השירות.

HTTP Persistent: .2

- אפשר לשלוח מספר בקשות על אותו חיבור.
- החיסכון הוא זמן החיבור והפחיתה העומס של ייצור חיבורים חדשים.
- אבל עדין יש עיכוב בין הבקשות (אף שהחיבורים קיימים).

HTTP Persistent with Pipelining: .3

- אפשר לשלוח מספר בקשות במקביל על אותו חיבור.
- יש שיפור משמעותי בזמן הטעינה כי יש פחתה עיכוב בין הבקשות (לא צריך לחכות לסיום כל בקשה לפני שմבקשים את הבאה בתור).

ג. ההבדל בין הودעת HTTP POST להודעת HTTP GET:

HTTP GET: •

- התפקיד שלה לשלוח בקשה כדי לקרוא מידע מהשרת.
- הפרמטרים נשלחים כחלק מהכתובת (URL) כלומר הם גלויים.
- משמשת בעיקר פעולה קריאה שאין מושנות את המידע בשרת.

HTTP POST: •

- התפקיד שלה לשלוח נתונים לשרת, בדרך כלל כדי ליזור או לעדכן מידע.
- הנתונים נשלחים בתווך גופ הבקשה ולא ב-URL-מה שמאפשר שליחה של נתונים גדולים ומורכבים יותר.
- משמשת לעדכן מצב או ביצוע פעולות שימושísticas על המידע בשרת (למשל, שליחת טופס).

2. קודם נחשב כמה זמן ייקח לשתף את כל החברים על ידי שימוש בשרת חזק:

כיוון שהשרות שלוח לכל אחד שירות, זמן ההעברה יהיה :

$$\text{Max} \left\{ \frac{\text{לדוג צבוקה * רפסם מירבחה}}{\text{בצק האלהה לש תרשה}} \right\} (1) \text{Max} = \frac{\text{לדוג צבוקה}}{\text{בצק הדורה לש לכ דחא מירבחה}}$$

נחשב קודם את (1):

$$(1) \text{Max} = \frac{\text{גודל הקובץ}}{\text{קטן מבין קצב ההודעה של החברים}}$$

הקצב הורדת ה-*C*ி קטן הוא $4MB$ לשניה, גודל הקובץ הוא $1GB$ שזה $1000MB$ לכן:

$$\frac{1000}{4} \text{ שניות} = 250$$

עכשו נחשב את (2):

$$(2) \quad 300 \text{ שניות} = \frac{30 * 1000}{100} = \frac{\text{גודל הקובץ} * \text{מספר החברים}}{\text{קצב העלאה של השרת}}$$

ניקח את המקסימלי מבין השנאים שזה (2), אך הזמן שייקח לשתף את כל החברים עם הקבצים זה 300 שניות.

עכשו נחשב כמה זמן ייקח לשתף את כל החברים על ידי שימוש ברשת P2P :

ברשת P2P החברים לא רק מורידים את הקובץ מדן, אלא גם משתפים חלקים מהקובץ זה עם זה. זה אמור להגדיל את קצב ההפצה.

דן מתחילה להעלות חלקים מהקובץ לחברים במתירות שלו. ברגע שחבר קיבל חלק מהקובץ, הוא יכול לשתף את החלק הזה עם חברים אחרים. קצב ההפצה הכלול ברשת הוא סכום של קצבים הعلاה של כל המשתתפים.

כדי למצוא את הזמן שייקח לדן לשתף את כל החברים בקובץ נדרש לחשב:

$$(1) \quad \text{Max} \left\{ \frac{\text{קצב הعلاה של כל החברים}}{\text{גודל הקובץ}} \right\} = \frac{\text{קצב הعلاה של דן}}{\text{גודל הקובץ}}$$

$$(2) \quad \text{Max} \left\{ \frac{\text{קצב הعلاה של כל החברים}}{\text{גודל הקובץ}} \right\} = \frac{\text{קצב הعلاה של דן}}{\text{גודל הקובץ}}$$

את (1) חישבנו כבר קודם ויצא **250 שניות**.

נחשב את (2), גודל הקובץ הוא $1GB = 1000MB$, קצב הعلاה של דן הוא $10MB$ לשניה, לכן:

$$(2) \quad 100 \text{ שניות} = \frac{1000MB}{10MB \text{ לשניה}}$$

נחשב את (3), מכיוון שיש 20 חברים עם קצב הعلاה של $5MB$ לשניה ו-10 חברים עם קצב הعلاה של $2MB$ לשניה.

סה"כ ביחיד זה יוצא שקצב הعلاה של כל החברים יחד הוא $120MB$ לשניה. לכן:

$$(3) \quad 8.33 \text{ שניות} = \frac{1000MB}{120MB \text{ לשניה}} = \frac{\text{גודל הקובץ}}{\text{סכום קצבים הعلاה של כל החברים}}$$

אחרי חישובנו את שלושתם יוצא:

$$\text{Max} \left\{ \text{Max} \left\{ \frac{\text{קצב הعلاה של כל החברים}}{\text{גודל הקובץ}}, \frac{\text{קצב הعلاה של דן}}{\text{גודל הקובץ}}, \frac{\text{קצב הعلاה של כל החברים}}{\text{קצב הعلاה של דן}} \right\}, \text{קצב הعلاה של דן} \right\} = \frac{\text{סכום קצבים הعلاה של כל החברים}}{\text{גודל הקובץ}} = \text{Max}\{250, 100, 8.33\}$$

לכן, כדי לדן להתחבר עם המחשב האישי שלו לרשת P2P כי הזמן שלילה של הקובץ לכל החברים יוצא יותר נמור.

ב. 1. בשרות חזק הגורמים המשפיעים הוא קצב ההורדה של החבר עם הקצב היכי איטי או קצב העלאה של השירות. במקרה שלנו, קצב העלאה של השירות מוגבל את הזמן, אז נרצה לשפר את קצב העלאה של השירות (ואפשר בנוסף גם לשפר את קצב ההורדה של החברים אם רוצים לשפר עוד יותר).

במקרה זה אמ לדוגמה קצב העלאה של השירות יהיה $1\text{GB}/\text{שנה}$:
 $\frac{30*1\text{GB}}{1\text{GB}/\text{שנה}} = 30$ שניות ואז המקסימלי מבין שני הדברים יהיה גורם מס' (1) שיצא 250. כМОון שגם אותו ניתן לשפר על ידי שיפור קצב ההורדה של כל החברים נניח $10\text{MB}/\text{שנה}$ וכן יצא זמן העברת הקובץ יהיה $1000/10$ שזה 100 שניות.

1. בשרות P2P הגורמים המשפיעים הם:
קצב העלאה של כל המשתתפים יחד- ככל שיש יותר משתתפים עם קצבים הعلاה גבוהים, כך זמן ההפוצה מתקצר.
קצב העלאה של שלוח הקובץ – דן במקרה זה.
קצב ההורדה של החבר עם קצב ההורדה היכי איטי.

כדי לשפר את הזמן הסופי אפשר להגדיל את קצב ההורדה של כל המשתתפים ל $10\text{MB}/\text{שנה}$, וכך יצא שקצב הורדת הקובץ של החבר עם הקצב המינימלי יהיה גם $10\text{MB}/\text{שנה}$. וכך:

$$\text{Max} \frac{\text{גודל הקובץ}}{\text{קצב ההורדה של כל אחד מהחברים}} = \frac{1000\text{MB}}{10\text{MB}/\text{שנה}} = 100$$

כך הקטנו את התוצאה הסופית שהיא 250 שניות במקום להיות 100 שניות.

ג. הבעיות המרכזיות בשרותות P2P ופתרונות

1. בעיות עומס על משתמשים עם קצב העלאה נמור

- **הבעיה:** בשרותות P2P המשתתפים נדרשים להעלות חלקים מהקובץ אחרים. אם להרבה משתמשים יש קצב העלאה נמור, קצב ההפוצה יהיה איטי יותר.
- **פתרון:** שימוש באלגוריתמים חכמים כמו BitTorrent שנוטנים עדיפות לשיתוף בין משתמשים עם קצבים הعلاה גבוהים ומחלקים את הקובץ לחלקים קטנים מאוד, כך שכל משתמש יכול לשתף גם חלק קטן.

2. בעיות משתמשים חמדניים

- **הבעיה:** ישנים משתמשים שמרידים חלקים מהקובץ, אך לא משתפים חזרה. זה פוגע בקצב ההפוצה הכללי.

- **פתרונות:** משתמשים יכולים להוריד חלקים רק אם הם ישתפו חלקים בעצמם. מנגנון שגורמים לרצות לשתף (לדוגמה, לתת ציון גבוה למשתתפים שתורמים יותר).

4. בעיה אבטחה ופרטיות

- **הבעיה:** בראשת P2P קשה לשלוט על התכנים שימושיים, ויש סיכון לתוךן מזויף, וירוסים, או פגעה בפרטיות.
- **הפתרון:** אימות קבצים באמצעות Hashing כדי לוודא שהתוכן שהורד תואם את המקום, שימוש בהצפנה כדי להגן על פרטיות המשתתפים.