

## מטלה 4

ת"ז מגישים: 322998287, 209326776

### חלק א:

בחלק זה של המטלה התבקשנו לכתוב את הקובץ `ping.c` ובו לממש את הפקודה "ping" בהרצת הקוד אנו מקבלים ארגומנט, הארגומנט הנ"ל מצביע לנו על הhost שאנחנו צריכים לעשות לו ping.

בתוכנית יש לנו לולאה אין סופית, התוכנית שולחת ICMP ECHO REQUEST לhost, וכאשר היא מקבלת בחזרה ICMP-ECHOREPLY היא שולחת את הICMP ECHO REQUEST הבא. וכן הלאה.

לכל חבילה שהתקבלה ביצענו הדפסה של השדות הבאים:  
packet IP, packet sequence number, time between the request and replay

### חלק ב:

בחלק זה של המטלה היה עלינו לכתוב שני קבצים `better_ping.c` ו-`watchdog.c`

הwatchdog הוא טיימר שמזהה ומשחזר תקלות, כאשר עוברת 10 שניות והwatchdog לא מקבל שום אות הוא מאתחל את המערכת.

בחלק זה שיננו את `ping.c` שכתבנו בחלק א וכתבנו את `watchdog`

הbeeter\_ping הוא גם זה שמריץ את watchdog בעזרת הפקודה `fork`. הbeeter\_ping והwatchdog פותחים סוקט ומתחברים אחד לשני כאשר הwatchdog הוא המאזין והbeeter\_ping הוא מתחבר אליו, הbeeter\_ping נכנס ללולאה שמבצעת ping ואח"כ שולחת הודעה לwatchdog אם לא מתבע Ping אז זה נתקע וגם לט שולח את ההודעה הנ"ל watch dog סופר את הזמן וכל עוד לא התקבל במשך 10 שניות הודעה עוצר את התוכנית

השתמשנו ב TCP 3001 port connction on

התק"ה שליו מכ"ה את הקבצים הבאים:

sudo ./parta <ip> <32> punkti

<ip> sudo . / part b      כי נחליף את חלקי א' וחלקי ב' בהתאמה?

make clean 3rd 5th 10th

כל פרט →  
 חשבון/פרט  
 make all  
 חשבון פרט  
 חשבון פרט  
 חשבון פרט  
 חשבון פרט  
 Admin

רואים שהפח'קים מתקבלים והם ממשיכים ככה עד שנוצר סלולית.

```

renana@renana:~/Desktop/ex4 final$ sudo ./partb 8.8.8.8
Bind() success
Waiting for incoming TCP-connections...
connected to server
ping 8.8.8.8: 19 data bytes
47 bytes from 8.8.8.8 icmp_seq= 0.ttl=118 RTT: 23.169001 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1.ttl=118 RTT: 5.853000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 2.ttl=118 RTT: 5.180000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 3.ttl=118 RTT: 5.914000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 4.ttl=118 RTT: 5.312000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 5.ttl=118 RTT: 4.828000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 6.ttl=118 RTT: 7.214000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 7.ttl=118 RTT: 9.404000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 8.ttl=118 RTT: 5.328000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 9.ttl=118 RTT: 5.534000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 10.ttl=118 RTT: 5.466000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 11.ttl=118 RTT: 6.768000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 12.ttl=118 RTT: 4.915000 milliseconds

```

⋮

```

47 bytes from 8.8.8.8 icmp_seq= 1628.ttl=118 RTT: 5.236000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1629.ttl=118 RTT: 4.408000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1630.ttl=118 RTT: 5.128000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1631.ttl=118 RTT: 11.644000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1632.ttl=118 RTT: 4.793000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1633.ttl=118 RTT: 5.258000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1634.ttl=118 RTT: 4.453000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1635.ttl=118 RTT: 5.188000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1636.ttl=118 RTT: 5.237000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1637.ttl=118 RTT: 4.527000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1638.ttl=118 RTT: 5.034000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1639.ttl=118 RTT: 7.075000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1640.ttl=118 RTT: 5.043000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1641.ttl=118 RTT: 4.855000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1642.ttl=118 RTT: 6.030000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1643.ttl=118 RTT: 4.509000 milliseconds
server 8.8.8.8 cannot be reached.
Killed

```

→ כאשר מסתים נתונים  
מחכה לו שניות  
killed SIG

```

renana@renana:~/Desktop/ex4 final$

```

30s  
: partb נתון



הקלטות מסך מתוך ה-wireshark  
חלק א'

→

```
renana@renana:~/Desktop/ex4 final$ make all
gcc ping.c -o parta
gcc watchdog.c -o watchdog
gcc better_ping.c -o partb
renana@renana:~/Desktop/ex4 final$ sudo ./parta 8.8.8
ping 8.8.8: 19 data bytes
47 bytes from 8.8.8: icmp_seq= 0.ttl=118 RTT: 15.306000 milliseconds
47 bytes from 8.8.8: icmp_seq= 1.ttl=118 RTT: 9.204000 milliseconds
47 bytes from 8.8.8: icmp_seq= 2.ttl=118 RTT: 6.846000 milliseconds
47 bytes from 8.8.8: icmp_seq= 3.ttl=118 RTT: 6.450000 milliseconds
47 bytes from 8.8.8: icmp_seq= 4.ttl=118 RTT: 7.398000 milliseconds
47 bytes from 8.8.8: icmp_seq= 5.ttl=118 RTT: 8.454000 milliseconds
47 bytes from 8.8.8: icmp_seq= 6.ttl=118 RTT: 6.692000 milliseconds
47 bytes from 8.8.8: icmp_seq= 7.ttl=118 RTT: 7.043000 milliseconds
47 bytes from 8.8.8: icmp_seq= 8.ttl=118 RTT: 6.865000 milliseconds
47 bytes from 8.8.8: icmp_seq= 9.ttl=118 RTT: 9.879000 milliseconds
47 bytes from 8.8.8: icmp_seq= 10.ttl=118 RTT: 7.396000 milliseconds
47 bytes from 8.8.8: icmp_seq= 11.ttl=118 RTT: 7.749000 milliseconds
47 bytes from 8.8.8: icmp_seq= 12.ttl=118 RTT: 5.809000 milliseconds
47 bytes from 8.8.8: icmp_seq= 13.ttl=118 RTT: 6.780000 milliseconds
47 bytes from 8.8.8: icmp_seq= 14.ttl=118 RTT: 7.417000 milliseconds
47 bytes from 8.8.8: icmp_seq= 15.ttl=118 RTT: 6.889000 milliseconds
^C
renana@renana:~/Desktop/ex4 final$
```

→ מסך של פקטות ICMP

parta.pcapng									
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help									
[icmp]									
No.	Time	Source	Destination	Protocol	Length	Info			
7	3.191905355	10.0.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 8)			
8	3.207077280	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 7)			
9	4.216158586	10.0.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 10)			
10	4.225077436	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 9)			
11	5.229688147	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 12)			
12	5.236263497	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 11)			
13	6.237776422	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 14)			
14	6.243925763	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 13)			
15	7.244771297	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 16)			
16	7.251746011	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 15)			
17	8.259953546	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 18)			
18	8.268098833	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 17)			
19	9.275003099	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 20)			
20	9.281423566	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 19)			
25	10.298215193	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 26)			
26	10.304936633	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 25)			
27	11.306212547	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 28)			
28	11.312757191	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 27)			
29	12.317638916	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 30)			
30	12.327323740	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 29)			
31	13.328156992	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 32)			
32	13.335256982	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 31)			
33	14.336042009	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 34)			
34	14.343433506	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 33)			
35	15.344523204	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 36)			
36	15.350067079	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 35)			
37	16.353158760	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 38)			
38	16.359548105	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 37)			
39	17.360135823	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 40)			
40	17.367161455	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 39)			
41	18.368139584	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 42)			
42	18.374778185	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=118 (request in 41)			

ראויים כי נשלח ping מהמחשב אל (10.0.2.15) אל שרת אולם הכתובת IP 8.8.8.8 אלוטו הנכונה זאת חזרת חקירה.  
נכון לזאת כי נשלח Echo request אל שרת האם ומזר Echo reply אל המעשב על המורה.  
נכון לזאת כי ה-ttl זהה לאלו ההקלטות אלו.

```

renana@renana:~/Desktop/ex4 final$ make all
gcc ping.c -o parta
gcc watchdog.c -o watchdog
gcc better_ping.c -o partb
renana@renana:~/Desktop/ex4 final$ sudo ./partb 8.8.8.8
Bind() success
Waiting for incoming TCP-connections...
connected to server
ping 8.8.8.8: 19 data bytes
47 bytes from 8.8.8.8 icmp_seq= 0.ttl=118 RTT: 6.824000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 1.ttl=118 RTT: 5.141000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 2.ttl=118 RTT: 5.881000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 3.ttl=118 RTT: 6.277000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 4.ttl=118 RTT: 5.932000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 5.ttl=118 RTT: 5.778000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 6.ttl=118 RTT: 5.521000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 7.ttl=118 RTT: 6.634000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 8.ttl=118 RTT: 6.162000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 9.ttl=118 RTT: 5.280000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 10.ttl=118 RTT: 5.909000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 11.ttl=118 RTT: 11.766000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 12.ttl=118 RTT: 6.480000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 13.ttl=118 RTT: 7.988000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 14.ttl=118 RTT: 5.959000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 15.ttl=118 RTT: 5.021000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 16.ttl=118 RTT: 6.249000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 17.ttl=118 RTT: 6.152000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 18.ttl=118 RTT: 7.212000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 19.ttl=118 RTT: 5.008000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 20.ttl=118 RTT: 5.105000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 21.ttl=118 RTT: 5.160000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 22.ttl=118 RTT: 6.162000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 23.ttl=118 RTT: 6.266000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 24.ttl=118 RTT: 5.135000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 25.ttl=118 RTT: 5.147000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 26.ttl=118 RTT: 5.310000 milliseconds

```

```

47 bytes from 8.8.8.8 icmp_seq= 345.ttl=118 RTT: 5.218000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 346.ttl=118 RTT: 6.235000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 347.ttl=118 RTT: 4.398000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 348.ttl=118 RTT: 4.291000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 349.ttl=118 RTT: 4.644000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 350.ttl=118 RTT: 4.433000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 351.ttl=118 RTT: 4.393000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 352.ttl=118 RTT: 4.578000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 353.ttl=118 RTT: 4.791000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 354.ttl=118 RTT: 4.644000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 355.ttl=118 RTT: 4.544000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 356.ttl=118 RTT: 5.224000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 357.ttl=118 RTT: 4.915000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 358.ttl=118 RTT: 5.118000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 359.ttl=118 RTT: 5.146000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 360.ttl=118 RTT: 5.659000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 361.ttl=118 RTT: 4.824000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 362.ttl=118 RTT: 4.926000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 363.ttl=118 RTT: 5.745000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 364.ttl=118 RTT: 5.553000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 365.ttl=118 RTT: 5.100000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 366.ttl=118 RTT: 5.232000 milliseconds
47 bytes from 8.8.8.8 icmp_seq= 367.ttl=118 RTT: 4.681000 milliseconds
server 8.8.8.8 cannot be reached.
Killed
renana@renana:~/Desktop/ex4 final$

```



→ ודעו לך  
port 3001  
אם נסן פ

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	76	56462 → 3001 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=4129829529 TSecr=0 WS=128
2	0.000022742	127.0.0.1	127.0.0.1	TCP	76	3001 → 56462 [SYN, ACK] Seq=1 Win=65536 Len=0 MSS=65536 SACK_PERM=1 TSval=4129829529 TSecr=4129829529 WS=128
3	0.000045085	127.0.0.1	127.0.0.1	TCP	68	56462 → 3001 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=4129829529 TSecr=4129829529
4	0.000497805	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=2 TSval=4129829530 TSecr=4129829529
5	0.000511523	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=3 Win=65536 Len=0 TSval=4129829530 TSecr=4129829530
6	0.000547488	8.8.8.8	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 7)
7	0.007235930	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 6)
8	0.007499596	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=3 Ack=1 Win=65536 Len=2 TSval=4129829537 TSecr=4129829530
9	0.007418410	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=5 Win=65536 Len=0 TSval=4129829537 TSecr=4129829537
10	0.007432563	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 11)
11	0.012517645	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 10)
12	0.012612410	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=5 Ack=1 Win=65536 Len=2 TSval=4129829542 TSecr=4129829537
13	0.012628387	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=7 Win=65536 Len=0 TSval=4129829542 TSecr=4129829542
14	0.012637469	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 15)
15	0.018411060	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 14)
16	0.018556416	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=7 Ack=1 Win=65536 Len=2 TSval=4129829548 TSecr=4129829542
17	0.018567241	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=9 Win=65536 Len=0 TSval=4129829548 TSecr=4129829548
18	0.018573868	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 19)
19	0.024719023	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 18)
20	0.024881415	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=9 Ack=1 Win=65536 Len=2 TSval=4129829554 TSecr=4129829548
21	0.024889735	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=11 Win=65536 Len=0 TSval=4129829554 TSecr=4129829554
22	0.024934726	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 23)
23	0.030459337	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 22)
24	0.030904486	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=11 Ack=1 Win=65536 Len=2 TSval=4129829560 TSecr=4129829554
25	0.030919296	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=13 Win=65536 Len=0 TSval=4129829560 TSecr=4129829560
26	0.030951443	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 27)
27	0.030562444	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 26)
28	0.030756305	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=13 Ack=1 Win=65536 Len=2 TSval=4129829566 TSecr=4129829560
29	0.030764135	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=15 Win=65536 Len=0 TSval=4129829566 TSecr=4129829566
30	0.030781080	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 31)
31	0.042150015	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 30)
32	0.042341597	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=15 Ack=1 Win=65536 Len=2 TSval=4129829572 TSecr=4129829566
33	0.042348961	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=17 Win=65536 Len=0 TSval=4129829572 TSecr=4129829572

נעם זאגט אז מחלות קער TCP

ראים כי נעם ping מהמחשב שלי (10.0.2.15) סוף שרת גוגל הכתובת IP 8.8.8.8 אלוט הנסן בעת חצית הפקודה.

נעם זאגט כי נעם Echo request אל שרת גוגל ואז Echo reply אל המחשב שלי.

נעם זאגט כי ה TTL זהה לאלו ההקסות שלו.

הערה 4-5 נעם זאגט שזאת חודשה לwatchdog (הפניו חשבונו חסן פורט את הטיימר ובשאר הפנימים הוא טאנס אלוט)

;

1470	2.180947954	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 1471)
1471	2.186101550	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 1470)
1472	2.186215060	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=735 Ack=1 Win=65536 Len=2 TSval=4129831715 TSecr=4129831710
1473	2.186270395	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=737 Win=65536 Len=0 TSval=4129831716 TSecr=4129831715
1474	2.186278275	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 1475)
1475	2.190090069	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=110 (request in 1474)
1476	2.190982748	127.0.0.1	127.0.0.1	TCP	70	56462 → 3001 [PSH, ACK] Seq=737 Ack=1 Win=65536 Len=2 TSval=4129831728 TSecr=4129831716
1477	2.190997466	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (no response found!)
1478	2.190997047	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=1 Ack=739 Win=65536 Len=0 TSval=4129831728 TSecr=4129831720
1479	11.424827341	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [FIN, ACK] Seq=1 Ack=739 Win=65536 Len=0 TSval=4129840954 TSecr=4129831728
1480	11.424817628	127.0.0.1	127.0.0.1	TCP	68	56462 → 3001 [FIN, ACK] Seq=739 Ack=2 Win=65536 Len=0 TSval=4129840954 TSecr=4129840954
1481	11.424827375	127.0.0.1	127.0.0.1	TCP	68	3001 → 56462 [ACK] Seq=2 Ack=749 Win=65536 Len=0 TSval=4129840954 TSecr=4129840954

נעם זאגט כי מחילוף הסמכות שנעשה הוא Echo request ואז אל הדיוו השניה נעם אל שרת חתבצה היצאת טכנית

ואז חזר Echo reply חתבה.

נעם זאגט אלוט קער TCP

**חלק א:**

בחלק זה של המטלה התבקשנו לכתוב את הקובץ ping.c ובו לממש את הפקודה "ping" בהרצת הקוד אנו מקבלים ארגומנט, הארגומנט הנ"ל מצביע לנו על הhost שאנחנו צריכים לעשות לו ping.

בתוכנית יש לנו לולאה אין סופית, התוכנית שולחת ICMP ECHO REQUEST לhost, וכאשר היא מקבלת בחזרה ICMP-ECHOREPLY היא שולחת את הICMP ECHO REQUEST הבא. וכן הלאה.

לכל חבילה שהתקבלה ביצענו הדפסה של השדות הבאים:  
packet IP, packet sequence number, time between the request and replay

בחלק זה נעזרנו בקוד שעלה למודל.

```
int main(int argc, char ** argv)
{
    if(argc!=2){
        printf("usage: %s <address>\n" ,argv[1]);
        exit(1);
    }
    char * destination_ip = argv[1];
```

עשינו בדיקה בif שבאמת התקבל מס הארגומנטים שרצינו בפקודה בטרמינל  
בקטע קוד זה אנו מקבלים מהמשתמש את ה- destination\_ip לו נעשה ping.

```
dest_in.sin_addr.s_addr = inet_addr(destination_ip);
int check= inet_pton(AF_INET, destination_ip, &(dest_in.sin_addr.s_addr));
if(check<=0){
    printf("Invalid IP Address\n");
    return 0;
}
```

בקטע קוד זה מתבצעת המרה של הכתובת ב destination\_ip לבינארי.  
מבצעים בדיקה: אם הdestination\_ip תקין אזי נקבל מספר חיובי ממש, במקרה שהוא לא תקין נקבל 0-1.

פתיחת raw socket: (ובצוע בדיקה)

```
int sock = -1;
if ((sock = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)) == -1)
{
    fprintf(stderr, "socket() failed with error: %d", errno);
    fprintf(stderr, "To create a raw socket, the process needs to be run by Admin/root user.\n\n");
    return -1;
}

86 while(1){
```

נכנסים ללולאה אין סופית, ששולחת ICMP ECHO REQUEST לhost, וכאשר היא מקבלת בחזרה ICMP-ECHOREPLY היא שולחת את הICMP ECHO REQUEST הבא.

```
//=====
// ICMP header
//=====

// Message Type (8 bits): ICMP_ECHO_REQUEST
icmphdr.icmp_type = ICMP_ECHO;

// Message Code (8 bits): echo request
icmphdr.icmp_code = 0;

// Identifier (16 bits): some number to trace the response.
// It will be copied to the response packet and used to map response to the request sent earlier.
// Thus, it serves as a Transaction-ID when we need to make "ping"
icmphdr.icmp_id = 18;

// Sequence Number (16 bits): starts at 0
icmphdr.icmp_seq = 0;

// ICMP header checksum (16 bits): set to 0 not to include into checksum calculation
icmphdr.icmp_cksum = 0;

// Combine the packet
char packet[IP_MAXPACKET];

// Next, ICMP header
memcpy((packet), &icmphdr, ICMP_HDRLen);

// After ICMP header, add the ICMP data.
memcpy(packet + ICMP_HDRLen, data, datalen);

// Calculate the ICMP header checksum
icmphdr.icmp_cksum = calculate_checksum((unsigned short*)(packet), ICMP_HDRLen + datalen);
memcpy((packet), &icmphdr, ICMP_HDRLen);
```

התחלת מדידת זמנים לפני שליחה:

```
struct timeval start, end;
gettimeofday(&start, 0);
```

שליחת ICMP ECHO REQUEST לhost

```
int bytes_sent = sendto(sock, packet, ICMP_HDRLen + datalen, 0, (struct sockaddr*)&dest_in, sizeof(dest_in));
if (bytes_sent == -1)
{
    fprintf(stderr, "sendto() failed with error: %d", errno);
    return -1;
}
```

קבלה בחזרה ICMP-ECHOREPLY

```
bzero(packet, IP_MAXPACKET);
socklen_t len = sizeof(dest_in);
size_t bytes_received = -1;
while ((bytes_received = recvfrom(sock, packet, sizeof(packet), 0, (struct sockaddr*)&dest_in, &len))
{
    if (bytes_received > 0)
    {
        // Check the IP header
        struct iphdr *iphdr = (struct iphdr *)packet;
        struct icmphdr *icmphdr = (struct icmphdr*)(packet + (iphdr->ihl * 4));
```

סיום מדידת זמנים:

```
gettimeofday(&end, 0);
```



```
char reply[IP_MAXPACKET];
memcpy(reply, packet + ICMP_HDRLEN + IP4_HDRLEN, datalen);
// printf("ICMP reply: %s\n", reply);

float milliseconds = (end.tv_sec - start.tv_sec) * 1000.0f + (end.tv_usec - start.tv_usec) / 1000.0f;
unsigned long microseconds = (end.tv_sec - start.tv_sec) * 1000.0f + (end.tv_usec - start.tv_usec);
printf("\n%d bytes from %s icmp_seq= %d. RTT: %f milliseconds (%ld microseconds)\n", bytes_received, inet_ntoa(dest_in.sin_addr), icmpSeq, milliseconds, microseconds);
sleep(1);
icmpSeq++;
```

יציאה מהwhile:

סגירת הraw socket:

```
// Close the raw socket descriptor.
close(sock);
```

הסברנו על החלק השני בצורה הכללית.. על קשר tcp בו השתמשנו הרחבנו בפרוטוקול מטלה קודמת והסבר על ping הסברנו לעיל