

PING (ICMP & TCP)

ליאור וינמן יועד תמר
213081763 213451818

2 בינואר 2023

תוכן עניינים

1	פרק א' - תיאור המערכת	2
1.1	הרצת הפרויקט	2
1.2	הקובץ היוצר - <i>Makefile</i>	3
1.3	מימוש פינג - <i>ping.c</i>	4
1.4	מימוש פינג משוכלל - <i>better_ping.c</i>	6
1.5	מימוש טיימר - <i>watchdog.c</i>	8
2	פרק ב' - תעבורה ופלט	11
2.1	פינג	11
2.1.1	תעבורת <i>ICMP</i>	11
2.1.2	פלט התוכנית	12
2.2	פינג משוכלל ללא עצירה	13
2.2.1	תעבורת <i>ICMP</i>	13
2.2.2	תעבורת <i>TCP</i>	14
2.2.3	פלט התוכנית	16
2.3	פינג משוכלל עם עצירה	17
2.3.1	תעבורת <i>ICMP</i>	17
2.3.2	תעבורת <i>TCP</i>	18
2.3.3	פלט התוכנית	19
3	ביבליוגרפיה	20

1 פרק א' - תיאור המערכת

1.1 הרצת הפרויקט

הקבצים הניתנים להרצה הם `ping.c` ו-`better_ping.c` בלבד. (שאר הקבצים הם קבצי עזר).
על מנת להריץ את הפרויקט - עלינו קודם כל להוריד את כל ארבעת הקבצים לתוך תיקיה משותפת על המחשב (יש להשתמש במערכת הפעלה `Linux Ubuntu LTS 22.04` בלבד).
לאחר שהורדנו את הקבצים, נפתח תרמינל על התיקיה הנתונה עם הקבצים ונכתוב את הפקודה "`make all`"
הדבר יגרום ליצירת קבצי ההרצה של התוכניות הניתנות להרצה. (לאחר מכן מומלץ להריץ את הפקודה "`clear`"
כדי לנקות את הפלט הסטנדרטי מהטרמינל).
כעת, אם ברצוננו להריץ את `ping.c` - נכתוב בטרמינל את הפקודה "`./parta < ip >`" (כאשר במקום `< ip >` יש
לרשום כתובת אליה אנחנו מעוניינים לשלוח את הבקשות). אם אנו מעוניינים ואם ברצוננו להריץ את `better_ping.c`
נכתוב בטרמינל את הפקודה "`./partb < ip >`".

1.2 הקובץ היוצר - Makefile

הקובץ הנ"ל הוא הקובץ שממיר את קבצי הקוד (הכתובים בשפת C) ומייצר מהם קבצי הרצה על מנת שנוכל להריץ בפועל את התוכניות שאנו כותבים. נעבור על הקוד:

```
1 CC = gcc
2 FLAGS = -Wall -g
3
4 .PHONY: all clean
5
6 all: ping better_ping watchdog
7
8 ping: ping.o
9     $(CC) $(FLAGS) -o parta ping.o
10
11 better_ping: better_ping.o
12     $(CC) $(FLAGS) -o partb better_ping.o
13
14 watchdog: watchdog.o
15     $(CC) $(FLAGS) -o watchdog watchdog.o
16
17 ping.o: ping.c
18     $(CC) $(FLAGS) -c ping.c
19
20 better_ping.o: better_ping.c
21     $(CC) $(FLAGS) -c better_ping.c
22
23 watchdog.o: watchdog.c
24     $(CC) $(FLAGS) -c watchdog.c
25
26 clean:
27     rm -f *.o parta partb watchdog
```

בקובץ זה ראשית אנו מייצרים מקבצי הקוד קבצי אובייקטים בינאריים (שורות 17 – 24) תהליך זה הוא תהליך ההידור. לאחר מכן מהקבצים הבינאריים אנו מייצרים קבצי הרצה שבעזרתם אנו למעשה מריצים את הפרוייקט (שורות 8 – 15) תהליך זה הוא תהליך הלינקוג'.

1.3 מימוש פינג - *ping.c*

נעבור על הקובץ *ping.c*:
בכל חבילה הנשלחת יש תא המיועד לביטים אשר מאמתים האם החבילה הגיעה בשלמותה ביטים אלו נקראים *checksum*. כאן כתבנו פונקציה לחישוב ולהשמה של ה-*checksum*:

```
18 void calculate_checksum(struct icmp_hdr *icmp)
19 {
20     unsigned long sum = 0;
21     unsigned short *ptr = (unsigned short *)icmp, checksum = 0;
22
23     for (int i = 0; i < sizeof(struct icmp_hdr) / 2; i++)
24     {
25         sum += *ptr++;
26     }
27
28     sum = (sum >> 16) + (sum & 0xffff);
29     sum += (sum >> 16);
30     checksum = ~sum;
31
32     icmp->checksum = checksum;
33 }
```

כאן וידאנו שהמשתמש אכן הריץ את התוכנית כנדרש, ז"א נתן לתוכנית כתובת IP אליה לבצע *ping*:

```
58 if (argc != 2) // checking that the user has specified an IP address
59 {
60     printf("usage: ./partb <ip>\n");
61     exit(EXIT_FAILURE);
62 }
```

כאן פתחנו שקע שדרכו תעבור תעבורת פינג (*rawsocket*):

```
64 sock = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP); // creating an RAW socket for ICMP communication
65 if (sock <= 0) // checking if socket created
66 {
67     perror("socket() failed");
68     close(sock);
69     exit(errno);
70 }
71
72 addr_ping.sin_family = AF_INET; // setting up the struct for ICMP communication
73 if(inet_aton(argv[1], &addr_ping.sin_addr) < 0)
74 {
75     perror("inet_aton() failed");
76     exit(errno);
77 }
```

כאן אנו מסדרים את ההגדרות של החבילות הנשלחות:

```
81     memset(&icmp, 0, sizeof(icmp)); // setting up the struct for ICMP communication
82     icmp.type = ICMP_ECHO;
83     icmp.code = 0;
84     icmp.un.echo.sequence = seq;
85     icmp.un.echo.id = getpid();
86     calculate_checksum(&icmp);
```

כאן אנו שולחים בקשת פינג לכתובת ולאחר מכן מקבלים תשובת פינג. בנוסף, אנו מחשבים את הזמנים של כל התהליך:

```
88     gettimeofday(&start, NULL); // starting counting ping-time
89
90     if (sendto(sock, &icmp, sizeof(icmp), 0, (struct sockaddr*)&addr_ping, sizeof(addr_ping)) < 0) // sending ICMP-ECHO-REQUEST
91     {
92         perror("sendto() failed");
93         close(sock);
94         exit(errno);
95     }
96
97     addr_len = sizeof(addr_ping); // receiving ICMP-ECHO-REPLY
98     len = recvfrom(sock, buffer, BUFSIZ, 0, (struct sockaddr*)&addr_ping, &addr_len);
99     if (len < 0)
100     {
101         perror("recvfrom() failed");
102         close(sock);
103         exit(errno);
104     }
105
106     gettimeofday(&end, NULL); // ending counting ping-time
107
108     time = (end.tv_sec - start.tv_sec) * 1000.0 + (end.tv_usec - start.tv_usec) / 1000.0; // saving the time in milli-seconds
```

כאן אנו מקבלים את הכתובת ממנה החבילה נשלחה כתגובה ומדפיסים למשתמש, בנוסף מבצעים המתנה של שניה:

```
110     ip = (struct iphdr*)buffer;
111
112     printf("%d bytes from %s: icmp_seq=%d ttl=%d time=%.2f ms\n",
113           len, inet_ntoa(*(struct in_addr*)&ip->saddr), icmp.un.echo.sequence, ip->ttl, time);
114
115     seq++;
116
117     sleep(1);
```

לבסוף אנו סוגרים את השקע שפתחנו ויוצאים בהצלחה מהתוכנית.

1.4 מימוש פינג משוכלל - *better_ping.c*

נעבור על הקובץ *better_ping.c*:

קובץ זה הוא הרחבה לקובץ הקודם, לכן רוב הדברים זהים לחלוטין למה שכבר ראינו (אזי לא נפרט עליהם שוב). נציג רק את התוספות החדשות של הקוד:
כאן אנו פותחים שקע לתעבורת *TCP* (המתבצעת עם ה-*watchdog* בפורט 3000) ומגדירים את האובייקט שיחזיק את הכתובת של התקשורת:

```
90     sockfd = socket(AF_INET, SOCK_STREAM, 0); // creating socket for TCP communcation
91     if(sock <= 0) // checking if socket created
92     {
93         perror("socket() failed");
94         close(sock);
95         close(sockfd);
96         exit(errno);
97     }
98
99     memset(&addr_server, '\0', sizeof(addr_server)); // setting up the struct for TCP communication
100    addr_server.sin_family = AF_INET;
101    addr_server.sin_port = htons(PORT);
102    addr_server.sin_addr.s_addr = inet_addr(IP);
```

כאן אנו מריצים את ה-*watchdog* (מריצים כעוד תהליך - במקביל):

```
121         pid = fork(); // starting the watchdog proccess
122         if (pid == 0)
123         {
124             execvp(args[0], args);
125         }
```

כאן אנו ממתינים שניה עד שה-*watchdog* יהיה מוכן לקבלת תקשורת, לאחר מכן אנו מתחברים אליו ושולחים אליו את כתובת ה-*IP* שהמשתמש הזין:

```
128     sleep(1); // delay until watchdog is setted up
129
130     if(connect(sockfd, (struct sockaddr*)&addr_server, sizeof(addr_server)) < 0) // connecting to watchdog
131     {
132         perror("connect() failed");
133         close(sock);
134         close(sockfd);
135         exit(errno);
136     }
137
138     if(send(sockfd, argv[1], strlen(argv[1]) + 1, 0) <= 0) // sending IP to watchdog
139     {
140         perror("send() failed");
141         close(sock);
142         close(sockfd);
143         exit(errno);
144     }
```

כאן אנו אמורים לקבל הודעת OK מה-*watchdog* שזהו סימן שיש תקשורת:

```
146         if(recv(sockfd , buffer , BUFSIZ , 0) <= 0) // receiving an OK message from watchdog
147         {
148             perror("recv() failed");
149             close(sock);
150             close(sockfd);
151             exit(errno);
152         }
153         if(strcmp(buffer, OK) != 0) // checking that OK received
154         {
155             printf("error occurred!");
156             close(sock);
157             close(sockfd);
158             exit(EXIT_FAILURE);
159         }
```

כאן אנו שולחים בחזרה הודעת OK וכמו כן בודקים מה מצב התהליך שרץ ברקע:

```
175         if(send(sockfd, OK, strlen(OK) + 1 , 0) < 0) // sending OK message
176         {
177             perror("send() failed");
178             close(sock);
179             close(sockfd);
180             exit(errno);
181         }
182
183         wait(&status);
184         if(status == -1)
185         {
186             printf("process status failed\n");
187             close(sock);
188             close(sockfd);
189             exit(EXIT_FAILURE);
190         }
```

1.5 מימוש טיימר - watchdog.c

הקובץ הנ"ל הוא למעשה קובץ בדיקת תקינות כתובת IP שהמשתמש הזין. הבדיקה מתרחשת על ידי הפעלת טיימר של 01 שניות של המתנה לתשובה מהכתובת (אחרי ששלחנו פינג בקשה) ואם לא התקבלה תשובה, אזי התוכנית תיעצר.

נעבור על הקובץ *watchdog.c*:

כאן כתבנו פונקציה אשר בודקת האם עבר הזמן והאם צריך לעצור או לא:

```
24 void timer_callback()
25 {
26     if (!received_echo_reply)
27     {
28         printf("server <%s> cannot be reached.\n" , buffer); // printing unreachable message
29
30         kill(0 , SIGKILL); // killing all processes and exiting
31     }
32 }
```

כאן פתחנו את השקע שדרכו תעבור התעבורה (TCP):

```
46     server_sock = socket(AF_INET, SOCK_STREAM, 0); // creating the listener socket
47     if(server_sock <= 0) // checking if socket created
48     {
49         perror("socket() failed");
50         close(server_sock);
51         exit(errno);
52     }
```

כאן אנו מאפשרים שימוש חוזר בפורט ובכתובת ומגדירים מבנה שיחזיק את הכתובת והפורט לתקשורת:

```
54     if(setsockopt(server_sock, SOL_SOCKET, SO_REUSEADDR, &er, sizeof(er)) < 0) // checking if ip and port
55     {
56         perror("setsockopt() failed");
57         close(server_sock);
58         exit(errno);
59     }
60
61     server_addr.sin_family = AF_INET; // setting up the struct for TCP communication
62     server_addr.sin_port = htons(PORT);
63     server_addr.sin_addr.s_addr = inet_addr(IP);
```


כאן אנו מקשרים את הכתובת והפורט לשקע שפתחנו ומאזינים לתקשורת נכנסת, אם יש כזו מאשרים:

```
65     if(bind(server_sock, (struct sockaddr*)&server_addr, sizeof(server_addr)) < 0) // binding socket with settings
66     {
67         perror("bind() failed");
68         close(server_sock);
69         exit(errno);
70     }
71
72     if(listen(server_sock, CONNECTIONS) < 0) //listen to incoming connections
73     {
74         perror("listen() failed");
75         close(server_sock);
76         exit(errno);
77     }
78
79     addr_size = sizeof(client_addr);
80     client_sock = accept(server_sock, (struct sockaddr*)&client_addr, &addr_size); //accept a connection
81     if(client_sock <= 0) // checking if accepted
82     {
83         perror("accept() failed");
84         close(client_sock);
85         close(server_sock);
86         exit(errno);
87     }
```

כאן אנו מקבלים מהמשתמש את כתובת ה-IP ושולחים הודעת OK:

```
89     if(recv(client_sock , buffer , BUFSIZ , 0) < 0) // receiving IP
90     {
91         perror("recv() failed");
92         close(client_sock);
93         close(server_sock);
94         exit(errno);
95     }
96
97     if(send(client_sock, OK , strlen(OK) + 1 , 0) < 0) // sending OK message
98     {
99         perror("send() failed");
100         close(client_sock);
101         close(server_sock);
102         exit(errno);
103     }
```

כאן אנו מפעילים טיימר ל-01 שניות, לאחר מכן בודקים האם יש צורך לעצור את התוכנית:

```
105     timer.it_value.tv_sec = 10; // setting up timer for 10 seconds
106     timer.it_value.tv_usec = 0;
107     timer.it_interval.tv_sec = 0;
108     timer.it_interval.tv_usec = 0;
109
110     printf("waiting for ping response...\n");
111
112     setitimer(ITIMER_REAL, &timer, NULL); // running the timer
113
114     signal(SIGALRM, timer_callback);
```

כמוכן שאם כן, אנו חוזרים לפונקציה למעלה שם מדפיסים הודעת שגיאה ויוצאים. אבל אם אין צורך, אז אנו מקבלים הודעת OK ויוצאים מהתוכנית בהצלחה:

```
116     if(recv(client_sock , buff , BUFSIZ , 0) < 0) // receiving OK message
117     {
118         perror("recv() failed");
119         close(client_sock);
120         close(server_sock);
121         exit(errno);
122     }
123     if(strcmp(buff, OK) != 0) // checking that OK received
124     {
125         printf("error occurred!");
126         close(client_sock);
127         close(server_sock);
128         exit(EXIT_FAILURE);
129     }
130
131
132     close(client_sock);
133     close(server_sock);
134     exit(EXIT_SUCCESS);
```

2 פרק ב' - תעבורה ופלט

2.1 פינג

כאן אנו רואים את התעבורה והפלט של "ping.c".

2.1.1 תעבורת ICMP

כאן אנו יכולים לראות את התעבורה של הפינג, שליחה סטנדרטית של חבילות בקשה ולאחר מכן קבלה של חבילות תגובה.
חבילת בקשה:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=0/0, ttl=64 (reply in 2)
2	0.006500273	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=0/0, ttl=115 (request in 1)
3	1.007030258	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=256/1, ttl=64 (reply in 4)
4	1.012183349	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=256/1, ttl=115 (request in 3)
5	2.012764421	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=512/2, ttl=64 (reply in 6)
6	2.019604914	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=512/2, ttl=115 (request in 5)
9	3.020292667	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=768/3, ttl=64 (reply in 10)
10	3.032714444	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=768/3, ttl=115 (request in 9)
11	4.033405028	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=1024/4, ttl=64 (reply in 12)
12	4.037976415	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=1024/4, ttl=115 (request in 11)
13	5.039123803	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=1280/5, ttl=64 (reply in 14)
14	5.053682174	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=1280/5, ttl=115 (request in 13)
17	6.055219584	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=1536/6, ttl=64 (reply in 18)
18	6.060186502	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=1536/6, ttl=115 (request in 17)

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_08:d3:0e (08:00:27:80:d3:0e), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.8.8.8
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 28
Identification: 0xd7a6 (55206)
Flags: 0x40, Don't Fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: ICMP (1)
Header Checksum: 0x471c [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.0.2.15
Destination Address: 8.8.8.8
Internet Control Message Protocol

0000 52 54 00 12 35 02 08 00 27 80 d3 0e 08 00 45 00 RT..5... '....E-
0010 00 1c d7 a6 40 00 00 01 47 1c 0a 00 02 0f 08 08@.G.
0020 08 08 08 00 0b 75 ec 8a 00 00U.

חבילת תגובה:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=0/0, ttl=64 (reply in 2)
2	0.006500273	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=0/0, ttl=115 (request in 1)
3	1.007030258	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=256/1, ttl=64 (reply in 4)
4	1.012183349	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=256/1, ttl=115 (request in 3)
5	2.012764421	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=512/2, ttl=64 (reply in 6)
6	2.019604914	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=512/2, ttl=115 (request in 5)
9	3.020292667	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=768/3, ttl=64 (reply in 10)
10	3.032714444	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=768/3, ttl=115 (request in 9)
11	4.033405028	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=1024/4, ttl=64 (reply in 12)
12	4.037976415	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=1024/4, ttl=115 (request in 11)
13	5.039123803	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=1280/5, ttl=64 (reply in 14)
14	5.053682174	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=1280/5, ttl=115 (request in 13)
17	6.055219584	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0xec8a, seq=1536/6, ttl=64 (reply in 18)
18	6.060186502	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0xec8a, seq=1536/6, ttl=115 (request in 17)

Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp0s3, id 0
Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_08:d3:0e (08:00:27:80:d3:0e)
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.0.2.15
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x18 (DSCP: Unknown, ECN: Not-ECT)
Total Length: 28
Identification: 0x63a7 (25511)
Flags: 0x00
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 115
Protocol: ICMP (1)
Header Checksum: 0xc803 [validation disabled]
[Header checksum status: Unverified]
Source Address: 8.8.8.8
Destination Address: 10.0.2.15
Internet Control Message Protocol

0010 00 1c 63 a7 00 00 73 01 c8 03 08 08 08 08 0a 00 ..C...S... ..
0020 02 0f 00 00 13 75 ec 8a 00 00 00 00 00 00 00 00U.....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00

2.1.2 פלט התוכנית

פלט התוכנית המתקבל הוא פלט סטנדרטי כמו של התוכנית *ping* הסטנדרטית של לינוקס. כאן הרצנו את התוכנית 21 פעמים:

```
yoad@yoad-VirtualBox:~/Desktop/Ex4_cn/cnc_assignment4-1$ sudo ./parta 8.8.8.8
28 bytes from 8.8.8.8: icmp_seq=0 ttl=115 time=6.61 ms
28 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=5.27 ms
28 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=7.03 ms
28 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=12.56 ms
28 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=4.66 ms
28 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=14.74 ms
28 bytes from 8.8.8.8: icmp_seq=6 ttl=115 time=5.07 ms
28 bytes from 8.8.8.8: icmp_seq=7 ttl=115 time=5.04 ms
28 bytes from 8.8.8.8: icmp_seq=8 ttl=115 time=4.87 ms
28 bytes from 8.8.8.8: icmp_seq=9 ttl=115 time=157.47 ms
28 bytes from 8.8.8.8: icmp_seq=10 ttl=115 time=8.93 ms
28 bytes from 8.8.8.8: icmp_seq=11 ttl=115 time=15.08 ms
28 bytes from 8.8.8.8: icmp_seq=12 ttl=115 time=8.43 ms
28 bytes from 8.8.8.8: icmp_seq=13 ttl=115 time=5.12 ms
28 bytes from 8.8.8.8: icmp_seq=14 ttl=115 time=8.62 ms
28 bytes from 8.8.8.8: icmp_seq=15 ttl=115 time=5.84 ms
28 bytes from 8.8.8.8: icmp_seq=16 ttl=115 time=267.39 ms
28 bytes from 8.8.8.8: icmp_seq=17 ttl=115 time=4.54 ms
28 bytes from 8.8.8.8: icmp_seq=18 ttl=115 time=194.48 ms
28 bytes from 8.8.8.8: icmp_seq=19 ttl=115 time=5.10 ms
28 bytes from 8.8.8.8: icmp_seq=20 ttl=115 time=6.04 ms
^C
yoad@yoad-VirtualBox:~/Desktop/Ex4_cn/cnc_assignment4-1$
```

2.2 פינג משוכלל ללא עצירה

כאן אנו רואים את התעבורה והפלט של "better_ping.c" ללא עצירת ה-watchdog.

2.2.1 תעבורת ICMP

כאן אנו רואים חבילה של בקשת פינג:

No.	Time	Source	Destination	Protocol	Length	Info
7	0.962793699	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=0/0, ttl=64 (reply in 8)
8	0.974854711	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=0/0, ttl=115 (request in 7)
9	2.963942340	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=256/1, ttl=64 (reply in 10)
10	3.078168600	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=256/1, ttl=115 (request in 9)
11	4.965391736	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=512/2, ttl=64 (reply in 12)
12	4.969888103	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=512/2, ttl=115 (request in 11)
13	6.970162324	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=768/3, ttl=64 (reply in 14)
14	6.993286404	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=768/3, ttl=115 (request in 13)
15	8.973598952	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=1024/4, ttl=64 (reply in 16)
16	8.980647209	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=1024/4, ttl=115 (request in 15)
17	10.9751177...	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=1280/5, ttl=64 (reply in 18)
18	10.9802741...	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=1280/5, ttl=115 (request in 17)

Frame 7: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_08:d3:0e (08:00:27:80:d3:0e), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.8.8.8
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 28
Identification: 0x40b4 (16564)
Flags: 0x40, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: ICMP (1)
Header Checksum: 0xde0e [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.0.2.15
Destination Address: 8.8.8.8
Internet Control Message Protocol

0000 52 54 00 12 35 02 08 00 27 80 d3 0e 08 00 45 00 RT...5... '.....E-
0010 00 1c 40 b4 40 00 40 01 de 0e 0a 00 02 0f 08 08 ..0.0.
0020 08 08 00 00 a5 70 52 8f 00 00pR...
Internet Control Message Protocol: Protocol

Packets: 18 - Discarded: 17 (66.7%)

כאן אנו רואים חבילה של תגובת פינג:

No.	Time	Source	Destination	Protocol	Length	Info
7	0.962793699	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=0/0, ttl=64 (reply in 8)
8	0.974854711	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=0/0, ttl=115 (request in 7)
9	2.963942340	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=256/1, ttl=64 (reply in 10)
10	3.078168600	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=256/1, ttl=115 (request in 9)
11	4.965391736	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=512/2, ttl=64 (reply in 12)
12	4.969888103	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=512/2, ttl=115 (request in 11)
13	6.970162324	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=768/3, ttl=64 (reply in 14)
14	6.993286404	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=768/3, ttl=115 (request in 13)
15	8.973598952	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=1024/4, ttl=64 (reply in 16)
16	8.980647209	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=1024/4, ttl=115 (request in 15)
17	10.9751177...	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x528f, seq=1280/5, ttl=64 (reply in 18)
18	10.9802741...	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x528f, seq=1280/5, ttl=115 (request in 17)

Frame 8: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp0s3, id 0
Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_08:d3:0e (08:00:27:80:d3:0e)
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.0.2.15
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x18 (DSCP: Unknown, ECN: Not-ECT)
Total Length: 28
Identification: 0x6561 (25953)
Flags: 0x00
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 115
Protocol: ICMP (1)
Header Checksum: 0xc649 [validation disabled]
[Header checksum status: Unverified]
Source Address: 8.8.8.8
Destination Address: 10.0.2.15
Internet Control Message Protocol

0010 00 1c 65 61 00 00 73 01 c6 49 08 08 08 08 0a 00 ..ea..s..I....
0020 02 0f 00 00 ad 70 52 8f 00 00 00 00 00 00 00 00pR...
0030 00 00 00 00 00 00 00 00 00 00 00 00
Time to Live (ip.ttl): 1 byte

Packets: 18 - Discarded: 17 (66.7%)

2.2.2 תעבורת TCP

קעת בגלל שיש לנו *watchdog* ישנה גם תעבורת *TCP*. נתבונן בה.
כאן אנו רואים פתיחת קשר:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	39112 → 3000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3400486088 TSecr=0 WS=128
2	0.000013257	127.0.0.1	127.0.0.1	TCP	74	3000 → 39112 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3400486088 TSecr=3...
3	0.000025543	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
4	0.000067025	127.0.0.1	127.0.0.1	TCP	74	39112 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=3400486088 TSecr=3400486088
5	0.000098818	127.0.0.1	127.0.0.1	TCP	66	3000 → 39112 [ACK] Seq=1 Ack=9 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
6	0.000157467	127.0.0.1	127.0.0.1	TCP	69	3000 → 39112 [PSH, ACK] Seq=1 Ack=9 Win=65536 Len=3 TSval=3400486088 TSecr=3400486088
7	0.000162125	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=9 Ack=4 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
8	0.000210478	127.0.0.1	127.0.0.1	TCP	69	39112 → 3000 [PSH, ACK] Seq=9 Ack=4 Win=65536 Len=3 TSval=3400486088 TSecr=3400486088
9	0.000239665	127.0.0.1	127.0.0.1	TCP	66	3000 → 39112 [FIN, ACK] Seq=4 Ack=12 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
10	0.049433050	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=12 Ack=5 Win=65536 Len=0 TSval=3400486138 TSecr=3400486088
11	2.006423395	127.0.0.1	127.0.0.1	TCP	74	39116 → 3000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3400488095 TSecr=0 WS=128
12	2.006437677	127.0.0.1	127.0.0.1	TCP	74	3000 → 39116 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3400488095 TSecr=3...
13	2.006448407	127.0.0.1	127.0.0.1	TCP	66	39116 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3400488095 TSecr=3400488095
14	2.006475589	127.0.0.1	127.0.0.1	TCP	74	39116 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=3400488095 TSecr=3400488095

כאן אנו רואים קבלה של כתובת ה-IP:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	39112 → 3000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3400486088 TSecr=0 WS=128
2	0.000013257	127.0.0.1	127.0.0.1	TCP	74	3000 → 39112 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3400486088 TSecr=3...
3	0.000025543	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
4	0.000067025	127.0.0.1	127.0.0.1	TCP	74	39112 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=3400486088 TSecr=3400486088
5	0.000098818	127.0.0.1	127.0.0.1	TCP	66	3000 → 39112 [ACK] Seq=1 Ack=9 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
6	0.000157467	127.0.0.1	127.0.0.1	TCP	69	3000 → 39112 [PSH, ACK] Seq=1 Ack=9 Win=65536 Len=3 TSval=3400486088 TSecr=3400486088
7	0.000162125	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=9 Ack=4 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
8	0.000210478	127.0.0.1	127.0.0.1	TCP	69	39112 → 3000 [PSH, ACK] Seq=9 Ack=4 Win=65536 Len=3 TSval=3400486088 TSecr=3400486088
9	0.000239665	127.0.0.1	127.0.0.1	TCP	66	3000 → 39112 [FIN, ACK] Seq=4 Ack=12 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
10	0.049433050	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=12 Ack=5 Win=65536 Len=0 TSval=3400486138 TSecr=3400486088
11	2.006423395	127.0.0.1	127.0.0.1	TCP	74	39116 → 3000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3400488095 TSecr=0 WS=128
12	2.006437677	127.0.0.1	127.0.0.1	TCP	74	3000 → 39116 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3400488095 TSecr=3...
13	2.006448407	127.0.0.1	127.0.0.1	TCP	66	39116 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3400488095 TSecr=3400488095
14	2.006475589	127.0.0.1	127.0.0.1	TCP	74	39116 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=3400488095 TSecr=3400488095

[Next Sequence Number: 9 (relative sequence number)]
 Acknowledgment Number: 1 (relative ack number)
 Acknowledgment number (raw): 3215676898
 1000 = Header Length: 32 bytes (8)
 • Flags: 0x018 (PSH, ACK)
 Window: 512
 [Calculated window size: 65536]
 [Window size scaling factor: 128]
 Checksum: 0xfe30 [unverified]
 [Checksum Status: Unverified]
 Urgent Pointer: 0
 • Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 • [Timestamps]
 • [SEQ/ACK analysis]
 TCP payload (8 bytes)
 • Data (8 bytes)
 Data: 382e382e382e3800
 Length: 81
 0020 00 01 98 c8 0b b8 dc 48 d5 bf bf ab 55 e2 80 18H....U...
 0030 02 00 fe 30 00 00 01 01 08 0a ca af 4c c8 ca af ...0.....L...
 0040 4c c8 38 2e 38 2e 38 2e 38 00 L:8,8,8,8.

כאן אנו רואים שיש שליחה של הודעת OK:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	39112 → 3000 [SYN, Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3400486088 TSecr=0 WS=128
2	0.000013257	127.0.0.1	127.0.0.1	TCP	74	3000 → 39112 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3400486088 TSecr=3...
3	0.000025543	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
4	0.000067025	127.0.0.1	127.0.0.1	TCP	74	39112 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=3400486088 TSecr=3400486088
5	0.000098818	127.0.0.1	127.0.0.1	TCP	66	3000 → 39112 [ACK] Seq=1 Ack=9 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
6	0.000157467	127.0.0.1	127.0.0.1	TCP	69	3000 → 39112 [PSH, ACK] Seq=1 Ack=9 Win=65536 Len=3 TSval=3400486088 TSecr=3400486088
7	0.000162125	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=9 Ack=4 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
8	0.000210478	127.0.0.1	127.0.0.1	TCP	69	39112 → 3000 [PSH, ACK] Seq=9 Ack=4 Win=65536 Len=3 TSval=3400486088 TSecr=3400486088
9	0.000239665	127.0.0.1	127.0.0.1	TCP	66	3000 → 39112 [FIN, ACK] Seq=4 Ack=12 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
10	0.049433050	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=12 Ack=5 Win=65536 Len=0 TSval=3400486138 TSecr=3400486088
11	2.006423395	127.0.0.1	127.0.0.1	TCP	74	39116 → 3000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3400488095 TSecr=0 WS=128
12	2.006437677	127.0.0.1	127.0.0.1	TCP	74	3000 → 39116 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3400488095 TSecr=3...
13	2.006448407	127.0.0.1	127.0.0.1	TCP	66	39116 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3400488095 TSecr=3400488095
14	2.006475589	127.0.0.1	127.0.0.1	TCP	74	39116 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=3400488095 TSecr=3400488095

[Next Sequence Number: 4 (relative sequence number)]
Acknowledgment Number: 9 (relative ack number)
Acknowledgment number (raw): 3695760839
1000 ... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window: 512
[Calculated window size: 65536]
[Window size scaling factor: 128]
Checksum: 0xfe2b [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
[SEQ/ACK analysis]
TCP payload (3 bytes)
Data (3 bytes)
Data: 6f6b00
[Length: 3]
0020 00 01 0b b8 98 c8 bf ab 55 e2 dc 48 d5 c7 00 18U..H....
0030 02 00 fe 2b 00 00 01 01 08 0a ca af 4c c8 ca af+.....L....
0040 4c c8 6f 6b 00L..ok.

כאן אנו רואים שיש קבלה של הודעת OK:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	39112 → 3000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3400486088 TSecr=0 WS=128
2	0.000013257	127.0.0.1	127.0.0.1	TCP	74	3000 → 39112 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3400486088 TSecr=3...
3	0.000025543	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
4	0.000067025	127.0.0.1	127.0.0.1	TCP	74	39112 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=3400486088 TSecr=3400486088
5	0.000098818	127.0.0.1	127.0.0.1	TCP	66	3000 → 39112 [ACK] Seq=1 Ack=9 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
6	0.000157467	127.0.0.1	127.0.0.1	TCP	69	3000 → 39112 [PSH, ACK] Seq=1 Ack=9 Win=65536 Len=3 TSval=3400486088 TSecr=3400486088
7	0.000162125	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=9 Ack=4 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
8	0.000210478	127.0.0.1	127.0.0.1	TCP	69	39112 → 3000 [PSH, ACK] Seq=9 Ack=4 Win=65536 Len=3 TSval=3400486088 TSecr=3400486088
9	0.000239665	127.0.0.1	127.0.0.1	TCP	66	3000 → 39112 [FIN, ACK] Seq=4 Ack=12 Win=65536 Len=0 TSval=3400486088 TSecr=3400486088
10	0.049433050	127.0.0.1	127.0.0.1	TCP	66	39112 → 3000 [ACK] Seq=12 Ack=5 Win=65536 Len=0 TSval=3400486138 TSecr=3400486088
11	2.006423395	127.0.0.1	127.0.0.1	TCP	74	39116 → 3000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3400488095 TSecr=0 WS=128
12	2.006437677	127.0.0.1	127.0.0.1	TCP	74	3000 → 39116 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3400488095 TSecr=3...
13	2.006448407	127.0.0.1	127.0.0.1	TCP	66	39116 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3400488095 TSecr=3400488095
14	2.006475589	127.0.0.1	127.0.0.1	TCP	74	39116 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=3400488095 TSecr=3400488095

[Next Sequence Number: 12 (relative sequence number)]
Acknowledgment Number: 4 (relative ack number)
Acknowledgment number (raw): 3215676901
1000 ... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window: 512
[Calculated window size: 65536]
[Window size scaling factor: 128]
Checksum: 0xfe2b [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
[SEQ/ACK analysis]
TCP payload (3 bytes)
Data (3 bytes)
Data: 6f6b00
[Length: 3]
0020 00 01 98 c8 0b b8 dc 48 d5 c7 bf ab 55 e5 00 18H....U...
0030 02 00 fe 2b 00 00 01 01 08 0a ca af 4c c8 ca af+.....L....
0040 4c c8 6f 6b 00L..ok.

2.2.3 פלט התוכנית

פלט התוכנית שהתקבל:

```
⊙ yoad@yoad-VirtualBox:~/Desktop/Ex4_cn/cnc_assignment4-1$ sudo ./partb 8.8.8.8
waiting for ping response...
28 bytes from 8.8.8.8: icmp_seq=0 ttl=115 time=0.61 ms
waiting for ping response...
28 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=0.90 ms
waiting for ping response...
28 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=4.25 ms
waiting for ping response...
28 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=1.88 ms
waiting for ping response...
28 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=0.60 ms
waiting for ping response...
28 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=0.45 ms
^C
⊙ yoad@yoad-VirtualBox:~/Desktop/Ex4_cn/cnc_assignment4-1$
```


2.3 פינג משוכלל עם עצירה

כאן אנו רואים את התעבורה והפלט של "better_ping.c" עם עצירת ה-watchdog.

2.3.1 תעבורת ICMP

כאן אנו רואים חבילה בודדת של בקשת פינג, נשים לב שאין תגובה:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.0.0.8	ICMP	42	Echo (ping) request id=0x3592, seq=0/0, ttl=64 (no response found!)
Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface enp0s3, id 0						
Ethernet II, Src: PcsCompu_08:d3:0e (08:00:27:80:d3:0e), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)						
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.0.0.8						
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0xc26d [correct]						
[Checksum Status: Good]						
Identifier (BE): 13714 (0x3592)						
Identifier (LE): 37429 (0x9235)						
Sequence Number (BE): 0 (0x0000)						
Sequence Number (LE): 0 (0x0000)						
[No response seen]						
[Expert Info (warning/Sequence): No response seen to ICMP request]						
0000	52 54 00 12 35 02 08 00	27 00 d3 0e 08 00 45 00	RT: 5... ..E-			
0010	00 1c 83 60 40 00 01 a3	6a 00 02 0f 08 00	...@...j.....			
0020	00 08 08 00 c2 6d 35 92	00 00m5..			

זו כל התעבורה כיוון שלאחר החבילה הזו ה-watchdog עצר את התוכנית.

2.3.2 תעבורת TCP

כאן אנו רואים יצירת קשר וסגירת קשר אחת ורואים את כלל התהליכים שכבר ראינו קודם לכן:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	52350 → 3000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3401025700 TSecr=0 WS=128
2	0.000012559	127.0.0.1	127.0.0.1	TCP	74	3000 → 52350 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3401025700 TSecr=34010...
3	0.000021913	127.0.0.1	127.0.0.1	TCP	66	52350 → 3000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3401025700 TSecr=3401025700
4	0.000052906	127.0.0.1	127.0.0.1	TCP	70	52350 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4 TSval=3401025700 TSecr=3401025700
5	0.000056114	127.0.0.1	127.0.0.1	TCP	66	3000 → 52350 [ACK] Seq=1 Ack=5 Win=65536 Len=0 TSval=3401025700 TSecr=3401025700
6	0.000133142	127.0.0.1	127.0.0.1	TCP	69	3000 → 52350 [PSH, ACK] Seq=1 Ack=5 Win=65536 Len=3 TSval=3401025700 TSecr=3401025700
7	0.000139074	127.0.0.1	127.0.0.1	TCP	66	52350 → 3000 [ACK] Seq=5 Ack=4 Win=65536 Len=0 TSval=3401025700 TSecr=3401025700
8	10.0043563	127.0.0.1	127.0.0.1	TCP	66	3000 → 52350 [FIN, ACK] Seq=4 Ack=5 Win=65536 Len=0 TSval=3401035704 TSecr=3401025700
9	10.0043563	127.0.0.1	127.0.0.1	TCP	66	52350 → 3000 [FIN, ACK] Seq=5 Ack=4 Win=65536 Len=0 TSval=3401035704 TSecr=3401025700
10	10.0043795	127.0.0.1	127.0.0.1	TCP	66	3000 → 52350 [ACK] Seq=5 Ack=6 Win=65536 Len=0 TSval=3401035704 TSecr=3401035704
11	10.0043797	127.0.0.1	127.0.0.1	TCP	66	52350 → 3000 [ACK] Seq=6 Ack=5 Win=65536 Len=0 TSval=3401035704 TSecr=3401035704

....	..0.	= Urgent: Not set
....	...1	= Acknowledgment: Set
....	0...	= Push: Not set
....	0...	= Reset: Not set
....0.	= Syn: Not set
*1		= Fin: Set
* [Expert Info (Chat/Sequence): Connection finish (FIN)]			
* [TCP Flags:A...F]			
* [Expert Info (Note/Sequence): This frame undergoes the connection closing]			
Window: 512			
[Calculated window size: 65536]			
[Window size scaling factor: 128]			
Checksum: 0xfe28 [unverified]			
[Checksum Status: Unverified]			
Urgent Pointer: 0			
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps			
* [Timestamps]			

0000	00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00E.
0010	00 34 dc 22 40 00 40 06 60 9f 7f 00 00 01 7f 00	.4."0@.'.....
0020	00 01 cc 7e 0b b8 e8 bc 33 7f c3 77 02 2a 80 113..w*..

2.3.3 פלט התוכנית

פלט התוכנית שהתקבל:

```
⊙ yoad@yoad-VirtualBox:~/Desktop/Ex4_cn/cnc_assignment4-1$ sudo ./partb 8.8
waiting for ping response...
server <8.8> cannot be reached.
Killed
⊙ yoad@yoad-VirtualBox:~/Desktop/Ex4_cn/cnc_assignment4-1$
```

3 ביבליוגרפיה

רשימת מקורות בהם השתמשנו בעת מימוש וכתובת הפרוייקט:

1. <https://www.geeksforgeeks.org/fork-system-call/>

2. <https://www.digitalocean.com/community/tutorials/execvp-function-c-plus-plus>

3. <https://www.techtarget.com/searchnetworking/definition/ICMP>

4. מצגות הקורס "תכנות מערכות 1" של ד"ר חוגי אסף.

5. מצגות וחוברת הקורס "רשתות תקשורת".