מגישים: 204420590
314078023

חלק 1 – myping

1. כלל הקוד במטלה נוהל ותועד ב-GitHub של הפרויקט:
https://github.com/ShmuelLa/Communication_Ex4

2. בחלק זה כתבנו קוד C בשם myping.c אשר שולח ומקבל פאקטת ICMP ובודק את זמן ה RTT שלה[1]. RTT, ראשי תיבות של Round Trip Time, הינו הזמן אשר לוקח לשלוח את פאקטת ה-ICMP שלנו ולקבל תגובה בחזרה.

3. להלן חלק מהפיצ'רים מרכזיים אשר מומשו/שונו בקוד זה:

   a. שינינו את תוכן הפאקטה כך שתהיה הודעה שנכתבה על ידינו.
   b. מחקנו לחלוטין את ה- IP-Header ואת הגדרת כתובת השליחה אנו מבצעים ע"י:
   dest_in.sin_addr.s_addr = inet_addr; ("8.8.8.8")
   c. שינינו את הקוד כך שימדוד את הזמן אך ורק כאשר הוא מקבל תשובה משרת היעד ע"י recvfrom.

4. מצ"ב דוגמת ההרצה לחלק א':

```
1: sam@samVB: ~/Git/Communication_Ex4
sam@samVB:~/Git/Communication_Ex4$ sudo ./ping.o
Ping received with RTT: 82.7 milliseconds |  82703.1 microsoeconds
sam@samVB:~/Git/Communication_Ex4$ sudo ./ping.o
Ping received with RTT: 83.3 milliseconds |  83347.1 microsoeconds
sam@samVB:~/Git/Communication_Ex4$ sudo ./ping.o
Ping received with RTT: 83.1 milliseconds |  83097.1 microsoeconds
sam@samVB:~/Git/Communication_Ex4$
```

   a. התאמה להסנפה ב-Wireshark (נבחר לדוג' את הבקשה הראשונה):

ניתן לראות כאן שה-RTT שלנו תואם ברמת ה-0.2 מילישניות לנתונים ב-Wireshark וכמו כן שה-IP מקור ויעד גם תואמים בהתאם להגדרות שלנו בקוד. ביקשנו לשלוח את הבקשה לשרת של Google והיא אכן נשלחה לשם (התוכן בתמונה הוא של פאקטת ה REPLY בכדי שנואל לראות את יצוג הזמנים).

```
1 0.000000000   10.0.2.15   8.8.8.8    ICMP   84 Echo (ping) request  id=0x1200, seq=0/0, ttl=64 (reply in 2)
2 0.082502549   8.8.8.8    10.0.2.15   ICMP   84 Echo (ping) reply    id=0x1200, seq=0/0, ttl=112 (request in 1)
> Frame 2: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface any, id 0
> Linux cooked capture v1
v 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: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 68
    Identification: 0x124d (4685)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 112
    Protocol: ICMP (1)
    Header Checksum: 0x1c4e [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 8.8.8.8
    Destination Address: 10.0.2.15
v Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x4097 [correct]
    [Checksum Status: Good]
    Identifier (BE): 4608 (0x1200)
    Identifier (LE): 18 (0x0012)
    Sequence Number (BE): 0 (0x0000)
    Sequence Number (LE): 0 (0x0000)
    [Request frame: 1]
    [Response time: 82.503 ms]
  > Data (40 bytes)
```

[1] https://en.wikipedia.org/wiki/Round-trip_delay

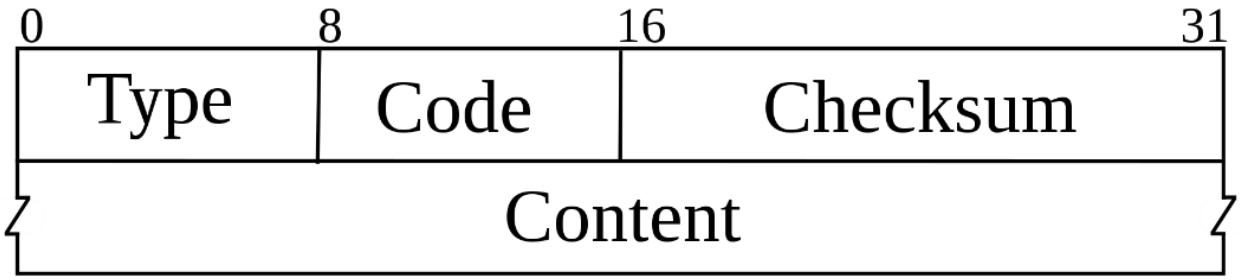
מצ״ב הרצת ifconfig המראה שה-IP זהה:

```
sam@samVB:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::5f23:265b:b6d9:4b13  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f5:6b:33  txqueuelen 1000  (Ethernet)
        RX packets 57  bytes 7680 (7.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 114  bytes 11585 (11.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 149  bytes 12349 (12.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 149  bytes 12349 (12.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

חלק 2 – Sniffing

1. בחלק זה כתבנו קוד C בשם sniff.c אשר מסניף את תעבורת ה-ICMP מכרטיס הרשת ומדפיס את התוכן שלה בצורה מסודרת. הפרויקט מומש ע״י ספריית PCAP.

2. בחלק זה במטלה הדפסנו את המרכיבים השונים של פאקטת ICMP בהתאם להדרים שלה[2]:



3. בקוד שלנו אנו מדפיסים את הנתונים הבאים:
   a. מס׳ פאקטה (ע״פ סדר מתחילת ההרצה של התכנית עד לסגירה שלה)
   b. פרוטוקול (במקרה שלנו נראה רק ICMP)
   c. IP מקור.
   d. IP יעד.
   e. סוג הפאקטה, במקרה של ICMP יהיו לנו שני סוגים:
      i. בקשה
      ii. תגובה
   f. קוד הפאקטה[3], בפרוטוקול ICMP מס׳ הקוד נותן מידע ראשונה על מה יכול היה לקרות עם הפאקה, רלוונטי בעיקר במקרה של שגיאות. במקרה שלנו נרצה תמיד לקבל קוד 0.
   g. Checksum[4] – בדיקת נכונות של הפאקטה.
   h. תוכן הפאקטה Content – בהספקה שלנו נקרא DATA.

---

[2] https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
[3] https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol#header_code
[4] https://tools.ietf.org/html/rfc1071

מגישים: 204420590
314078023

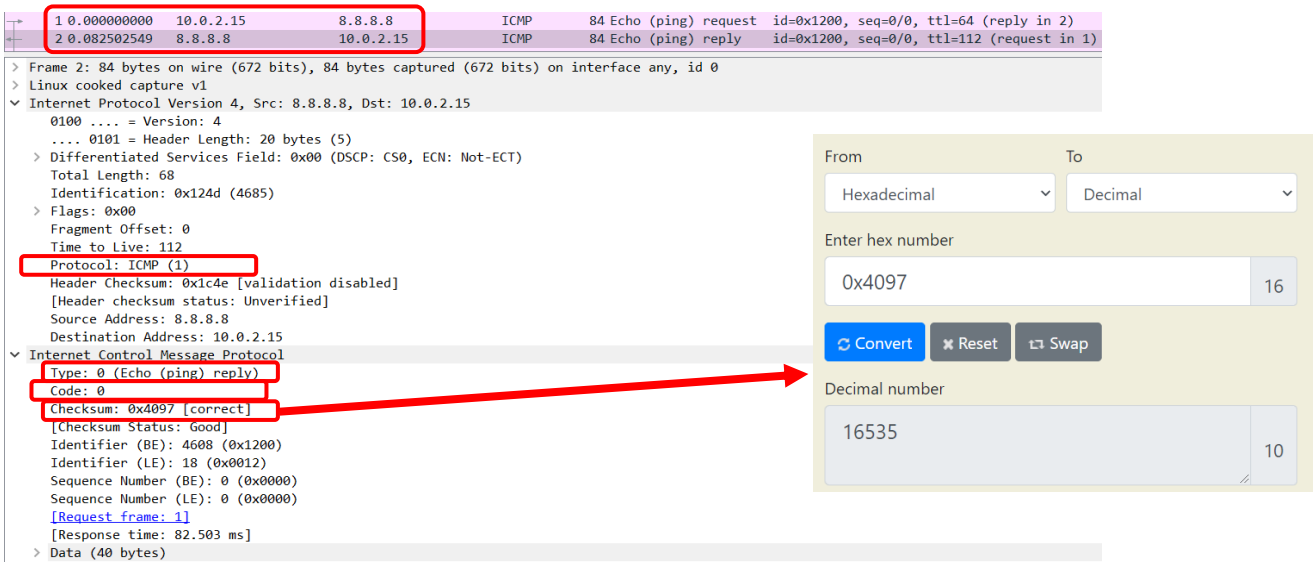
4. מצ"ב דוגמת הרצה התואמת את הדוג' בשאלה 1:

```
2: sam@samVB: ~/Git/Communication_Ex4
sam@samVB:~/Git/Communication_Ex4$ sudo ./sniff.o
No.: 1 | Protocol: ICMP | SRC_IP: 10.0.2.15 | DST_IP: 8.8.8.8 | Type: Request | Code: 0 | Checksum 14487
Data: This is a custom Gidon_Shmuel Ping :)

No.: 2 | Protocol: ICMP | SRC_IP: 8.8.8.8 | DST_IP: 10.0.2.15 | Type: Reply | Code: 0 | Checksum 16535
Data: This is a custom Gidon_Shmuel Ping :)

No.: 3 | Protocol: ICMP | SRC_IP: 10.0.2.15 | DST_IP: 8.8.8.8 | Type: Request | Code: 0 | Checksum 14487
Data: This is a custom Gidon_Shmuel Ping :)

No.: 4 | Protocol: ICMP | SRC_IP: 8.8.8.8 | DST_IP: 10.0.2.15 | Type: Reply | Code: 0 | Checksum 16535
Data: This is a custom Gidon_Shmuel Ping :)

No.: 5 | Protocol: ICMP | SRC_IP: 10.0.2.15 | DST_IP: 8.8.8.8 | Type: Request | Code: 0 | Checksum 14487
Data: This is a custom Gidon_Shmuel Ping :)

No.: 6 | Protocol: ICMP | SRC_IP: 8.8.8.8 | DST_IP: 10.0.2.15 | Type: Reply | Code: 0 | Checksum 16535
Data: This is a custom Gidon_Shmuel Ping :)
```

5. נבחן את התוכן הנ"ל בהסנפה ב-Wireshark:
   a. כפי שניתן לראות בתוכן פאקטת ה REPLY שהראינו בחלק הראשון של המטלה:
      i. ניתן לראות כי כתובת היעד והמקור זהים.
      ii. פרוטוקול תקין.
      iii. סוג הבקשה (REPLY) תקין
      iv. קוד, 0, תקין.
      v. כפי שניתן לראות בסוף הפאקטה התוכן תואם את התוכן שהכנסנו ב-PING:

```
0020  08 08 08 08 08 00 38 97  12 00 00 00 54 68 69 73   ......8· ····This
0030  20 69 73 20 61 20 63 75  73 74 6f 6d 20 47 69 64    is a cu stom Gid
0040  6f 6e 5f 53 68 6d 75 65  6c 20 50 69 6e 67 20 3a   on_Shmue l Ping :
0050  29 20 0a 00                                        ) ··
```

      vi. CHECKSUM, זהה למספר שהדפסנו. בקוד אנו הדפסנו את המספר הדצימלי 16535 שהמרה ל-Hexadecimal כמו שמוצג ב-Wireshark שווה ל-0x4097:



מגישים: 204420590
314078023

6. מצ״ב הדפסה של התוכנות בזמן אמת במקביל אחת לשנייה:

```
1: sam@samVB: ~/Git/Communication_Ex4
sam@samVB:~/Git/Communication_Ex4$ sudo ./ping.o
Ping received with RTT: 82.7 milliseconds |  82703.1 microsoeconds
sam@samVB:~/Git/Communication_Ex4$ sudo ./ping.o
Ping received with RTT: 83.3 milliseconds |  83347.1 microsoeconds
sam@samVB:~/Git/Communication_Ex4$ sudo ./ping.o
Ping received with RTT: 83.1 milliseconds |  83097.1 microsoeconds
sam@samVB:~/Git/Communication_Ex4$
```

```
2: sam@samVB: ~/Git/Communication_Ex4
sam@samVB:~/Git/Communication_Ex4$ sudo ./sniff.o
No.: 1 | Protocol: ICMP | SRC_IP: 10.0.2.15 | DST_IP: 8.8.8.8 | Type: Request | Code: 0 | Checksum 14487
Data: This is a custom Gidon_Shmuel Ping :)

No.: 2 | Protocol: ICMP | SRC_IP: 8.8.8.8 | DST_IP: 10.0.2.15 | Type: Reply | Code: 0 | Checksum 16535
Data: This is a custom Gidon_Shmuel Ping :)

No.: 3 | Protocol: ICMP | SRC_IP: 10.0.2.15 | DST_IP: 8.8.8.8 | Type: Request | Code: 0 | Checksum 14487
Data: This is a custom Gidon_Shmuel Ping :)

No.: 4 | Protocol: ICMP | SRC_IP: 8.8.8.8 | DST_IP: 10.0.2.15 | Type: Reply | Code: 0 | Checksum 16535
Data: This is a custom Gidon_Shmuel Ping :)

No.: 5 | Protocol: ICMP | SRC_IP: 10.0.2.15 | DST_IP: 8.8.8.8 | Type: Request | Code: 0 | Checksum 14487
Data: This is a custom Gidon_Shmuel Ping :)

No.: 6 | Protocol: ICMP | SRC_IP: 8.8.8.8 | DST_IP: 10.0.2.15 | Type: Reply | Code: 0 | Checksum 16535
Data: This is a custom Gidon_Shmuel Ping :)
```

7. ניתן לראות שהרצנו את תוכנת ה PING (צד שמאל) שלוש פעמים מה שכולל שליחה וקבלה בכל אחד מהם וניתן לראות בצד ימין את השליחה והקבלה בהתאמה (6 פאקטות סה״כ). ניתן לראות שמס׳ הפאקטות והסדר תואם לחלוטין את ההסנפה שעשינו במקביל ב-Wireshark:

icmp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.8.8	ICMP	84	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 2)
2	0.082502549	8.8.8.8	10.0.2.15	ICMP	84	Echo (ping) reply id=0x1200, seq=0/0, ttl=112 (request in 1)
3	0.712080333	10.0.2.15	8.8.8.8	ICMP	84	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 4)
4	0.794980982	8.8.8.8	10.0.2.15	ICMP	84	Echo (ping) reply id=0x1200, seq=0/0, ttl=112 (request in 3)
5	1.173682693	10.0.2.15	8.8.8.8	ICMP	84	Echo (ping) request id=0x1200, seq=0/0, ttl=64 (reply in 6)
6	1.256657533	8.8.8.8	10.0.2.15	ICMP	84	Echo (ping) reply id=0x1200, seq=0/0, ttl=112 (request in 5)