

רשתות - תרגיל 1

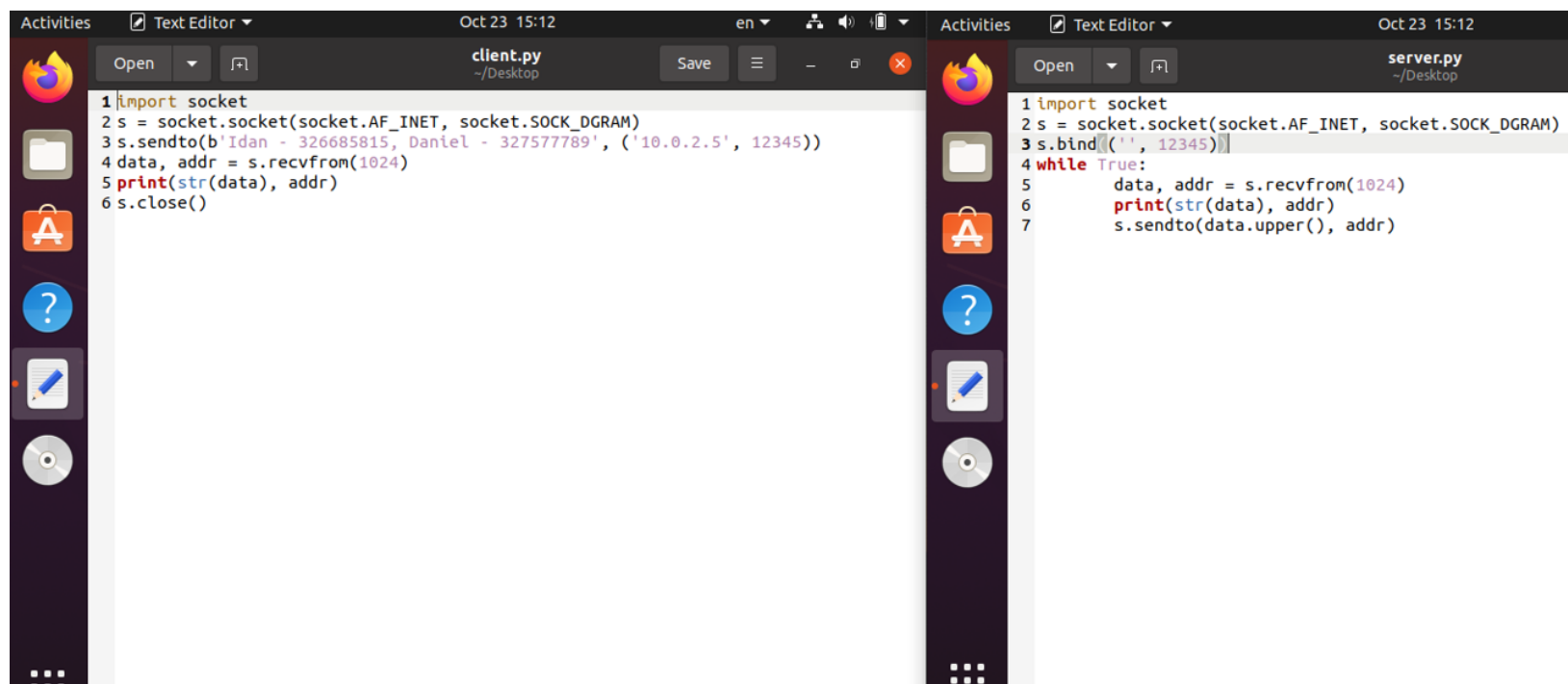
מגישים

- דניאל פוטר
- עידן טורקניץ



חלק א

1. העתקנו את קוד השרת וקוד הלקוח מהמודל ושינינו את קוד הלקוח על מנת שישלח לשרת את השמות ותעודות הזהות שלנו :

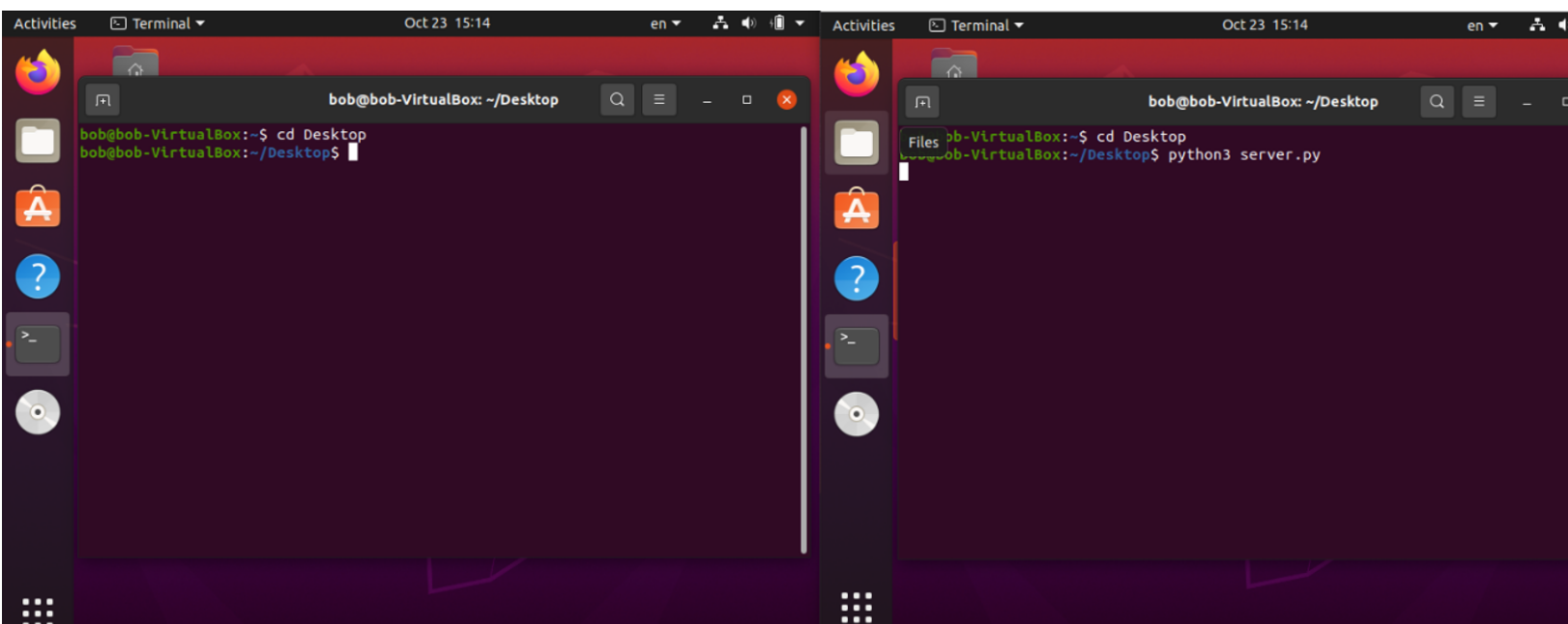


The image shows two side-by-side screenshots of a text editor window. The left window is titled 'client.py' and the right window is titled 'server.py'. Both windows show Python code for a socket-based communication.

```
client.py
1 import socket
2 s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
3 s.sendto(b'Idan - 326685815, Daniel - 327577789', ('10.0.2.5', 12345))
4 data, addr = s.recvfrom(1024)
5 print(str(data), addr)
6 s.close()

server.py
1 import socket
2 s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
3 s.bind(('', 12345))
4 while True:
5     data, addr = s.recvfrom(1024)
6     print(str(data), addr)
7     s.sendto(data.upper(), addr)
```

הרצנו את הקוד של השרת:

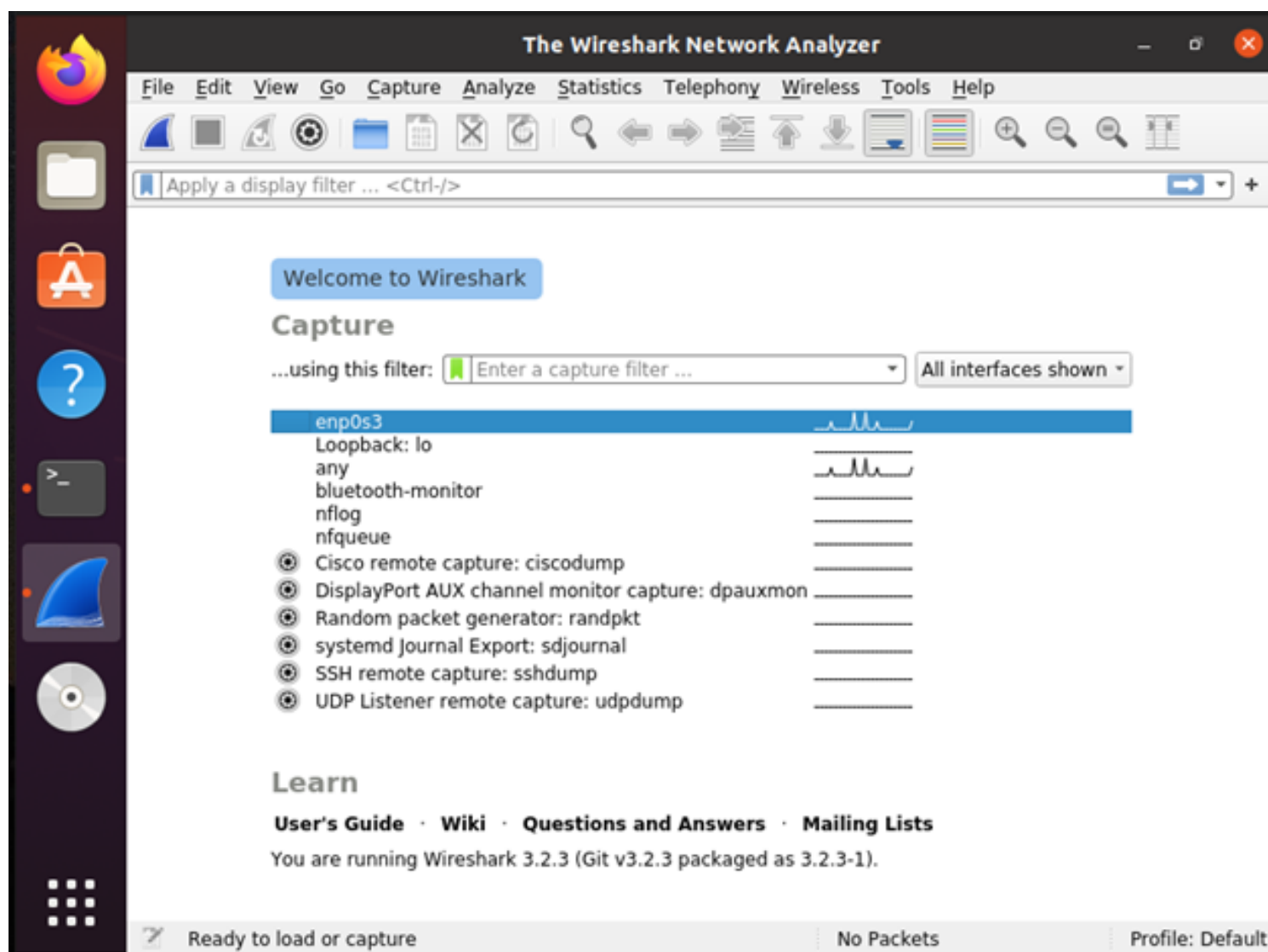


The image shows two side-by-side screenshots of a terminal window. The left window shows the execution of the client code, and the right window shows the execution of the server code.

```
bob@bob-VirtualBox: ~/Desktop
bob@bob-VirtualBox:~$ cd Desktop
bob@bob-VirtualBox:~/Desktop$

Files bob-VirtualBox:~$ cd Desktop
bob@bob-VirtualBox:~/Desktop$ python3 server.py
```

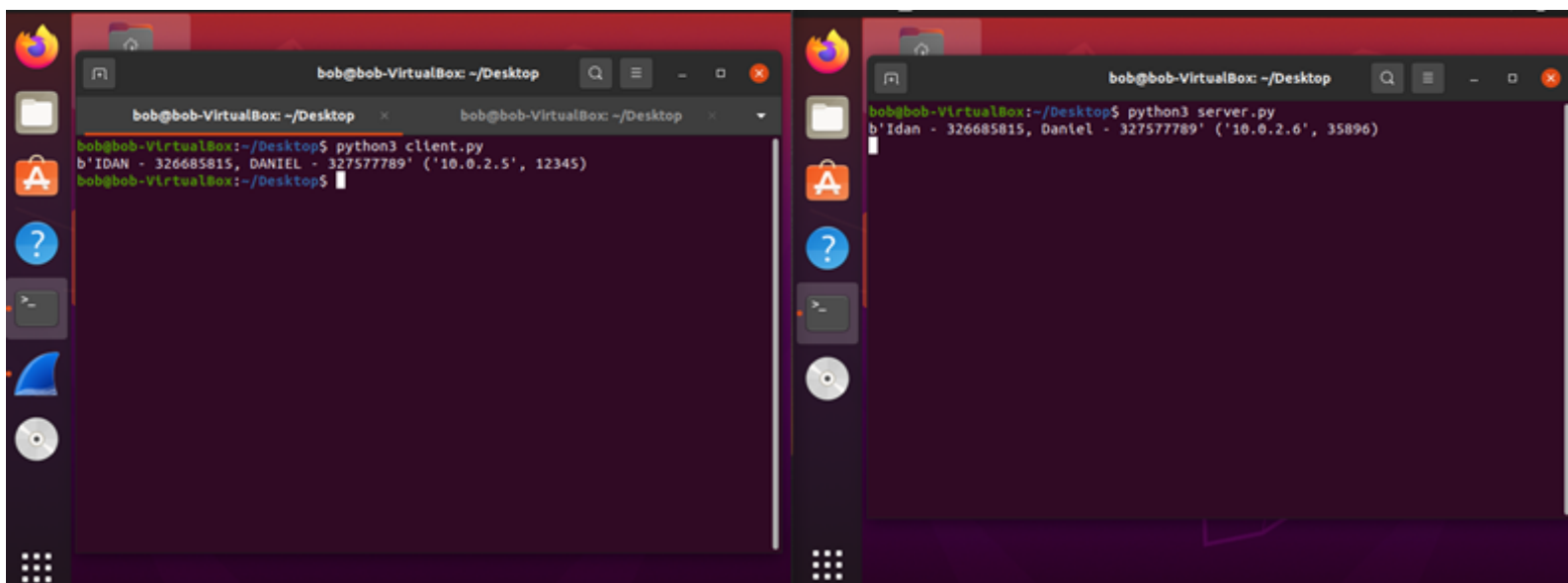
לפני שהרצנו את הקוד של הלקוח, נכנסנו מהמכונה שמריצה את הקוד של הלקוח לwireshark על מנת לתפוס את התעבורה. בחרנו את כרטיס הרשת enp0s3 מכיוון שהלקוח והשרת רצים כל אחד על מכונה נפרדת משלו ולכן המידע אמור לעבור בכרטיס הרשת ה"פיזי" של המכונה ולא בכרטיס הרשת הווירטואלי. תמונה מצורפת:



2. סיננו את התעבורה לפי כתובת ה־ip של השרת (וגם לפי שימוש בפרוטוקול `udp`) , וכך תפסנו את כל התקשורות המכילות את ה־ip של השרת:

```
udp && ip.addr == 10.0.2.5
```

הרצנו את הקוד של הלקוח (בזמן שהסנפנו את התעבורה ב־wireshark)



וקיבלנו את המידע הנשלח בין הלקוח לשרת ובחזרה:

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help							
udp && ip.addr == 10.0.2.5							
No.	Time	Source	Destination	Protocol	Length	Info	
15	10851.278347...	10.0.2.6	10.0.2.5	UDP	78	35896 → 12345	Len=36
16	10851.278950...	10.0.2.5	10.0.2.6	UDP	78	12345 → 35896	Len=36

נשים לב כי באמת datan שנשלח בחבילות הוא datan הרצוי:

בחבילת הלקוח שנשלחה לשרת:

15	10851.278347...	10.0.2.6	10.0.2.5	UDP	78	35896 → 12345	Len=36
16	10851.278950...	10.0.2.5	10.0.2.6	UDP	78	12345 → 35896	Len=36

```
Frame 15: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_f8:41:eb (08:00:27:f8:41:eb), Dst: PcsCompu_9e:b7:ac (08:00:27:9e:b7:ac)
Internet Protocol Version 4, Src: 10.0.2.6, Dst: 10.0.2.5
User Datagram Protocol, Src Port: 35896, Dst Port: 12345
Data (36 bytes)
Data: 4964616e202d2033332363638353831352c2044616e69656c...
[Length: 36]
```

```
0000 08 00 27 9e b7 ac 08 00 27 f8 41 eb 08 00 45 00  ..'.A...E.
0010 00 40 8f 73 40 00 40 11 93 2f 0a 00 02 06 0a 00  .@.s@.@. ./.....
0020 02 05 8c 38 30 39 00 2c 18 48 49 64 61 6e 20 2d  ...809.,.HIdan -
0030 20 33 32 36 36 38 35 38 31 35 2c 20 44 61 6e 69  3266858 15, Dani
0040 65 6c 20 2d 20 33 32 37 35 37 37 37 38 39      el - 327 577789
```

בחבילת השרת שנשלחה ללקוח (נשים לי כי המידע הוא אותו מידע שנשלח לשרת מהלקוח רק באותיות גדולות, שזה בדיוק מה שרצינו):

15	10851.278347...	10.0.2.6	10.0.2.5	UDP	78	35896 → 12345	Len=36
16	10851.278950...	10.0.2.5	10.0.2.6	UDP	78	12345 → 35896	Len=36

```
Frame 16: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_9e:b7:ac (08:00:27:9e:b7:ac), Dst: PcsCompu_f8:41:eb (08:00:27:f8:41:eb)
Internet Protocol Version 4, Src: 10.0.2.5, Dst: 10.0.2.6
User Datagram Protocol, Src Port: 12345, Dst Port: 35896
Data (36 bytes)
Data: 4944414e202d2033332363638353831352c2044414e49454c...
[Length: 36]
```

```
0000 08 00 27 f8 41 eb 08 00 27 9e b7 ac 08 00 45 00  ..'.A...E.
0010 00 40 a1 38 40 00 40 11 81 6a 0a 00 02 05 0a 00  .@.8@.@. .j.....
0020 02 06 30 39 8c 38 00 2c 76 15 49 44 41 4e 20 2d  ...09.8.,.vIDAN -
0030 20 33 32 36 36 38 35 38 31 35 2c 20 44 41 4e 49  3266858 15, DANI
0040 45 4c 20 2d 20 33 32 37 35 37 37 37 38 39      EL - 327 577789
```

3. נוכל למצוא את השימוש במספרי פורט בשכבת הtransport (התעבורה), שכן היא משמשת להעברת המידע לאפליקציה הנכונה :

בחבילת הלקוח שנשלחה לשרת :

No.	Time	Source	Destination	Protocol	Length	Info
15	10851.278347...	10.0.2.6	10.0.2.5	UDP	78	35896 → 12345 Len=36
16	10851.278950...	10.0.2.5	10.0.2.6	UDP	78	12345 → 35896 Len=36

▶ Frame 15: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface enp0s3, id 0

▶ Ethernet II, Src: PcsCompu_f8:41:eb (08:00:27:f8:41:eb), Dst: PcsCompu_9e:b7:ac (08:00:27:9e:b7:ac)

▶ Internet Protocol Version 4, Src: 10.0.2.6, Dst: 10.0.2.5

▼ User Datagram Protocol, Src Port: 35896, Dst Port: 12345

Source Port: 35896

Destination Port: 12345

Length: 44

Checksum: 0x1848 [unverified]

[Checksum Status: Unverified]

[Stream index: 6]

▶ [Timestamps]

בחבילה מהשרת שנשלחה חזרה ללקוח (נשים לב כי הפורט המקור ופורט היעד התחלפו)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp && ip.addr == 10.0.2.5

No.	Time	Source	Destination	Protocol	Length	Info
15	10851.278347...	10.0.2.6	10.0.2.5	UDP	78	35896 → 12345 Len=36
16	10851.278950...	10.0.2.5	10.0.2.6	UDP	78	12345 → 35896 Len=36

Frame 16: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface enp0s3, id 0

- Ethernet II, Src: PcsCompu_9e:b7:ac (08:00:27:9e:b7:ac), Dst: PcsCompu_f8:41:eb (08:00:27:f8:41:eb)
- Internet Protocol Version 4, Src: 10.0.2.5, Dst: 10.0.2.6
- User Datagram Protocol, Src Port: 12345, Dst Port: 35896
 - Source Port: 12345
 - Destination Port: 35896
 - Length: 44
 - Checksum: 0x7615 [unverified]
 - [Checksum Status: Unverified]
 - [Stream index: 6]
 - [Timestamps]
- Data (36 bytes)

4. כתובת הִקו של הלקוח הוא 10.0.2.6 ושל השרת הוא 10.0.2.5. השימוש בכתובות אלו נמצא בשכבת הרשת.
נסתכל על שכבת הרשת בשליחת ההודעה מהלקוח לשרת ונוכל למצוא את כתובות אלו:

No.	Time	Source	Destination	Protocol	Length	Info
15	10851.278347...	10.0.2.6	10.0.2.5	UDP	78	35896 → 12345 Len=36
16	10851.278950...	10.0.2.5	10.0.2.6	UDP	78	12345 → 35896 Len=36

<p>Frame 15: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface enp0s3, id 0</p> <p>Ethernet II, Src: PcsCompu_f8:41:eb (08:00:27:f8:41:eb), Dst: PcsCompu_9e:b7:ac (08:00:27:9e:b7:ac)</p> <p>Internet Protocol Version 4, Src: 10.0.2.6, Dst: 10.0.2.5</p> <p>0100 = Version: 4</p> <p>.... 0101 = Header Length: 20 bytes (5)</p> <p>Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)</p> <p>Total Length: 64</p> <p>Identification: 0x8f73 (36723)</p> <p>Flags: 0x4000, Don't fragment</p> <p>Fragment offset: 0</p> <p>Time to live: 64</p> <p>Protocol: UDP (17)</p> <p>Header checksum: 0x932f [validation disabled]</p> <p>[Header checksum status: Unverified]</p> <p>Source: 10.0.2.6</p> <p>Destination: 10.0.2.5</p> <p>User Datagram Protocol, Src Port: 35896, Dst Port: 12345</p> <p>Data (36 bytes)</p>
--

שכבת הרשת בשליחת הודעה מהשרת ללקוח:

No.	Time	Source	Destination	Protocol	Length	Info
15	10851.278347...	10.0.2.6	10.0.2.5	UDP	78	35896 → 12345 Len=36
16	10851.278950...	10.0.2.5	10.0.2.6	UDP	78	12345 → 35896 Len=36

<p>Frame 16: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface enp0s3, id 0</p> <p>Ethernet II, Src: PcsCompu_9e:b7:ac (08:00:27:9e:b7:ac), Dst: PcsCompu_f8:41:eb (08:00:27:f8:41:eb)</p> <p>Internet Protocol Version 4, Src: 10.0.2.5, Dst: 10.0.2.6</p> <p>0100 = Version: 4</p> <p>.... 0101 = Header Length: 20 bytes (5)</p> <p>Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)</p> <p>Total Length: 64</p> <p>Identification: 0xa138 (41272)</p> <p>Flags: 0x4000, Don't fragment</p> <p>Fragment offset: 0</p> <p>Time to live: 64</p> <p>Protocol: UDP (17)</p> <p>Header checksum: 0x816a [validation disabled]</p> <p>[Header checksum status: Unverified]</p> <p>Source: 10.0.2.5</p> <p>Destination: 10.0.2.6</p>

נוכל לאשר שאלו הם כתובות ה־ip ע"י ביצוע פקודת ifconfig בterminal של השרת והלקוח (פקודה זאת מביאה את כתובות ה־ip של כרטיסי הרשת)

אצל הלקוח:

```
bob@bob-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.6 netmask 255.255.255.0 broadcast 10.0.2.255
```

אצל השרת:

```
bob@bob-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.5 netmask 255.255.255.0 broadcast 10.0.2.255
```

חלק ב

החלטנו להריץ את הקוד על אותה מכונה

הIP והPort של שלושת התוכנות הם :

client	foo	server	
random	55555	12345	Port
127.0.0.1	127.0.01	127.0.0.1	IP

הסבר על הפרוטוקול:

כל חבילה בגודל 100 בתים, כאשר העשרה בתים הראשונים משמשים למספר(אנחנו משתמשים בתווים לייצוג המספר, כלומר ישנם 9,999,999,999 מספרים שונים מבשיל החבילות, הרבה יותר ממה שזקוקים לו בשביל התרגיל)
המחשה:

Header - Packet number (10 Bytes)
Data - 90 Bytes

פעולתו של השרת:

- השרת שומר מילון (printed) אשר מכיל את מספרי החבילות שהודפסו (על מנת לא להדפיס פעמיים)

- כל פעם שהשרת מקבל חבילה, הוא בודק האם מספרה נמצא במילון printed, אם לא, אז הוא מוסיף אותה למילון ומדפיס אותה. בכל מקרה אח"כ הוא מחזיר את החבילה חזרה ללקוח על מנת שידע שהשרת קיבל אותה. אנחנו עושים זאת על מנת שהשרת לא ידפיס פעמיים את אותה חבילה.

פעולתו של הלקוח:

- כאשר הלקוח שולח חבילה, הוא מחכה עד שהיא תחזור ורק אז שולח את החבילה הבאה, וכמו בשלב 2, אם הלקוח מחכה יותר מפרק זמן מסוים, אז הוא שולח שוב את החבילה.

אנחנו בעצם נוקטים בשיטת *Stop&Wait* ובכך לא שולחים את החבילה הבאה עד שקיבלנו אישור שהחבילה הקודמת הגיעה לשרת, ובכך תמיד כל החבילות יגיעו לשרת, ובסדר הרצוי

שלב 1 - foo משחק נחמד:

הרצנו את הקוד :

(1)

```
idan@idan-HP-Pavilion-x360-Convertible:~/Desktop/Networks/ComputerNetworks-Ex1/pa
rtB$ python3 server.py 12345 127.0.0.1
```

(2)

```
idan@idan-HP-Pavilion-x360-Convertible:~/Desktop/Networks/ComputerNetworks-Ex1/pa
rtB$ python3 foo.py 55555 127.0.0.1 12345 1
Playing nice
```

(3)

```
idan@idan-HP-Pavilion-x360-Convertible:~/Desktop/Networks/ComputerNetworks-Ex1/pa
rtB$ python3 client.py 55555 127.0.0.1 test.txt
```

כאשר test.txt מכילה רק את המשפט

Hello World!

נריץ את הקוד ונסניף את התעבורה בwireshark (כמובן נסתכל על כרטיס הרשת loopback) ונסנן על מנת לתפוס רק את הפקטות הקשורות

udp && udp.port == 55555 && ip.addr == 127.0.0.1						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	ADwin ...	64	
2	0.000118689	127.0.0.1	127.0.0.1	ADwin ...	64	
3	0.000203931	127.0.0.1	127.0.0.1	ADwin ...	64	
4	0.000247283	127.0.0.1	127.0.0.1	ADwin ...	64	

(נתעלם מכך שהwireshark לא מראה את הפרוטוקול הנכון כאשר נעבור על השכבות נוכל לראות כי שכבת התעבורה משתמשת בפרוטוקול UDP).

נוכל לראות כי מופיעות 4 תעבורות, וזאת מכיוון שמתרחש הדבר הבא:

- פקטה 1 : מועברת חבילה (המכילה את המשפט Hello World) מהclient לfoo
- פקטה 2 : foo מעביר את החבילה לserver
- פקטה 3 : הסרבר מעביר את החבילה לfoo
- פקטה 4 : foo מחזיר את החבילה לclient

נסתכל על השכבות השונות (למשל של פקטה 1):

1	0.000000000	127.0.0.1	127.0.0.1	ADwin ...	64	
2	0.000118689	127.0.0.1	127.0.0.1	ADwin ...	64	
3	0.000203931	127.0.0.1	127.0.0.1	ADwin ...	64	
4	0.000247283	127.0.0.1	127.0.0.1	ADwin ...	64	

▶ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface lo, id 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ User Datagram Protocol, Src Port: 48593, Dst Port: 55555
▶ ADwin configuration protocol

ונוכל לראות כי באמת שכבת התעבורה עובדת עם User Datagram Protocol, הלוא הוא UDP.

נסתכל על שכבת התעבורה - על מנת לראות את הsourcePort והdestPort :

▼ User Datagram Protocol, Src Port: 48593, Dst Port: 55555
Source Port: 48593
Destination Port: 55555

אפשר לראות כי הsource port נבחר רנדומלית ע"י מערכת ההפעלה והdestination port הוא של foo, כלומר 55555.

בשכבת הרשת נוכל לראות כי גם הsource ip וגם הdestination ip הם 127.0.0.1 מכיוון שאנחנו מריצים את שלושת התוכנות על אותה מכונה.

Source: 127.0.0.1
Destination: 127.0.0.1

נסתכל על שכבת האפליקציה, בשביל למצוא את הdata הנשלח בחבילה:

udp && udp.port == 55555 && ip.addr == 127.0.0.1						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	ADwin ...	64	
2	0.000118689	127.0.0.1	127.0.0.1	ADwin ...	64	
3	0.000203931	127.0.0.1	127.0.0.1	ADwin ...	64	
4	0.000247283	127.0.0.1	127.0.0.1	ADwin ...	64	

▶ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface lo, id 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ User Datagram Protocol, Src Port: 48593, Dst Port: 55555
▶ ADwin configuration protocol

0000	00 00 00	00 00 00 00 00 00	00 00 00 00 08 00 45 00E
0010	00 32 a9	a1 40 00 40 11	93 17 7f 00 00 01 7f 00	2..@.
0020	00 01 bd	d1 d9 03 00 1e	fe 31 30 30 30 30 30 30	1000000
0030	30 30 30	30 48 65 6c 6c	6f 20 57 6f 72 6c 64 21	0000Hello	o World!

בחבילות האחרות המצב דומה - ה data אותו דבר, אך בשכבת התעבורה ה source port ו ה dest port משתנים בהתאם למי שולח למי.

שלב 2 - foo זורק אחוז מסויים מהחבילות:

הדרך שלנו להתמודד עם הבעיה הזו הייתה שברגע שהלקוח שולח חבילה מסויימת, הוא מפעיל טיימר (במקרה שלנו של 0.01 שניות), אם החבילה חוזרת אליו בזמן הזה אז הוא יודע שהחבילה הגיעה לשרת ששלח אותה בחזרה אליו, ואם היא לא מגיעה אז הוא לא יודע אם החבילה הגיעה לשרת, ולכן הוא שולח אותה שוב לשרת.
נריץ שוב את הקוד רק ש foo מכוון למוד 2

```
ldan@ldan-HP-Pavilion-x360-Convertible:~/Desktop/Networks/ComputerNetworks-Ex1  
rtB$ python3 foo.py 55555 127.0.0.1 12345 2  
Dropping 28%
```

(את השרת והלקוח נריץ כרגיל, עם הקובץ טקסט כמו בשלב 1)

נסניף את התעבורה ב wireshark (שוב בכרטיס ה loop back) ונסנן

udp && udp.port == 55555 && ip.addr == 127.0.0.1					
	Time	Source	Destination	Protocol	Length Info
1	0.000000000	127.0.0.1	127.0.0.1	ADwin ...	64
2	0.000132760	127.0.0.1	127.0.0.1	ADwin ...	64
3	0.000228994	127.0.0.1	127.0.0.1	ADwin ...	64
4	1.001251627	127.0.0.1	127.0.0.1	ADwin ...	64
5	1.001551337	127.0.0.1	127.0.0.1	ADwin ...	64
6	1.001730359	127.0.0.1	127.0.0.1	ADwin ...	64
7	2.002585550	127.0.0.1	127.0.0.1	ADwin ...	64
8	2.002894570	127.0.0.1	127.0.0.1	ADwin ...	64
9	2.003076058	127.0.0.1	127.0.0.1	ADwin ...	64
10	2.003250922	127.0.0.1	127.0.0.1	ADwin ...	64

נשים לי כי הפעם ישנן 10 חבילות במקום ה-4 שראינו כאשר foo התנהגה יפה.
 נכנסנו לשכבת התעבורה (שעובדת עם פרוקוטול UDP) וראינו את הsource port והdest
 port של כל חבילה :

Packet num	Source port	Destination port
1	37501	55555
2	55555	12345
3	12345	55555
4	37501	55555
5	55555	12345
6	12345	55555
7	37501	55555
8	55555	12345
9	12345	55555
10	55555	37501

מה נוכל להסיק מהמידע הזה:

- שורות 1-3 : אפשר לראות כי החבילה עברה מהלקוח, לfoo ואז לשרת ובחזרה לfoo, אך לא רואים שfoo החזירה ללקוח (כלומר זרקה את החבילה), לכן ללקוח נגמר זמן הtimeout והוא שלח בחזרה את החבילה לשרת (לכן שורה 4 מראה כי הלקוח שוב פעם מעביר לfoo את החבילה).
- שורות 4:6 אותו מקרה כמו שורות 1-3 - החבילה הגיעה לשרת, אך בחזרה מהשרת foo זרקה אותה, לכן בשורה 7 הלקוח שוב פעם שולח את החבילה.

- שורות 7:10 - החבילה עברה בהצלחה את המסלול מהלקוח foo, לשרת, בחזרה לfoo ובחזרה ללקוח מבלי שfoo תזרוק אותה.

(במידה והיו יותר מחבילה אחת שהלקוח היה צריך לשלוח הוא היה יכול לעבור הלאה ולשלוח אותה).

הפלט של foo במקרה זה:

```
Phiiii, no drop.... 18
Yay, no sleep.... 62 100
Forwarded b'0000000000Hello World!' from ('127.0.0.1', 37501) to ('127.0.0.1', 12345)
Dropped! randomly... 99
Phiiii, no drop.... 5
Yay, no sleep.... 66 100
Forwarded b'0000000000Hello World!' from ('127.0.0.1', 37501) to ('127.0.0.1', 12345)
Dropped! randomly... 71
Phiiii, no drop.... 36
Yay, no sleep.... 16 100
Forwarded b'0000000000Hello World!' from ('127.0.0.1', 37501) to ('127.0.0.1', 12345)
Phiiii, no drop.... 39
Yay, no sleep.... 27 100
Forwarded b'0000000000Hello World!' from ('127.0.0.1', 12345) to ('127.0.0.1', 37501)
```


שלב 3 : foo משנה את סדר שליחת החבילות

למעשה הפתרון שלנו לשלב 2, פותר גם את שלב 3. מכיוון שהלקוח לא ממשיך לשלוח את החבילה הבאה, עד שהוא בטוח שהחבילה הקודמת הגיעה לשרת. (הוא יודע זאת כי השרת מחזיר לו אישור), ולכן בוודאות כל החבילות יגיעו לפי הסדר אל השרת.

נפעיל את foo על מוד 3 ונסניף את התעבורה בwireshark (הפעם test.txt שלנו יהיה)

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
```

על מנת להדגים שליחת 2 פקטות.

נריץ את הקוד

```
idan@Thunder:~/Desktop/Network/ComputerNetworks-Ex1/partB$ python3 foo.py 55555 127.0.0.1 12345 3  
Delaying 84%
```

נסניף בwireshark ונקבל

No.	Time	Source	Destination	Source Port	Dest Port	Protocol	Length	Info	New Column
1	0.000000	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		1
2	1.000244	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		2
3	1.000432	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		3
4	1.000580	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		4
5	2.001467	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		5
6	3.002679	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		6
7	3.848389	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		7
8	3.848577	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		8
9	4.004007	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		9
10	4.432003	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		10
11	4.432259	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		11
12	5.004406	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		12
13	5.238083	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		13
14	5.238261	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		14
15	5.249647	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		15
16	5.249823	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		16
17	6.250077	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		17
18	6.436287	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		18
19	6.436444	127.0.0.1	127.0.0.1	50769	55555	ADwin ...	142		19
20	6.436673	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		20
21	6.436836	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		21
22	6.436971	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		22
23	6.566482	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		23
24	6.566685	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		24
25	6.566959	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		25
26	6.579118	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		26
27	6.579311	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		27
28	6.708513	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		28
29	6.708676	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		29
30	6.717230	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		30
31	7.705768	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		31
32	7.705939	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		32
33	7.984300	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		33
34	7.984462	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		34
35	8.271458	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		35
36	10.580268	127.0.0.1	127.0.0.1	55555	12345	ADwin ...	142		36
37	10.580457	127.0.0.1	127.0.0.1	12345	55555	ADwin ...	142		37
38	10.910799	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		38
39	11.796324	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		39
40	13.268257	127.0.0.1	127.0.0.1	55555	50769	ADwin ...	142		40

(הוספנו את הsource port והdestination port משכבת התעבורה על מנת שיהיה יותר קל לראות מי שולח ולאן)

כאשר כרגיל:

פורט השרת הוא 12345

פורט foo הוא 55555

פורט הלקוח נקבע רנדומלית ע"י מערכת ההפעלה, ובמקרה זה הוא 50769

ישנן הרבה חבילות שעוברות בתהליך, אך הדבר החשוב שיש לזכור הוא שאנחנו עובדים בשיטת הStop&Wait, ולכן עד שחבילה לא מגיעה ללקוח מהשרת, הלקוח לא עובר לשלוח את החבילה הבאה, מה שמובטח שכל החבילות יגיעו לשרת בסדר נכון. במקרה זה, חבילות 1-15 שימשו לשליחת החבילה הראשונה, בחלק מן החבילות foo עיכבה את השליחה של החבילה ולכן זמן ההמתנה של הלקוח עבר (במקרה שלנו 1 שנייה), והלקוח שלח שוב. בחבילה 15, החבילה הראשונה הגיעה ללקוח וכך הלקוח יודע בוודאות כי היא הגיעה לשרת והשרת הדפיס אותה, ולכן הלקוח יכול לשלוח את החבילה הבאה. וכן האלה. נשים לב כי חלק מהפקטות הבאות (אחרי מספר 15) יכולות להיות מורכבות מaaa... ולא רק מbbb... וזאת בגלל שfoo עיכבה את החבילות של aaa.. ובקרבם הם ישלחו ללקוח/לשרת תלוי איפה foo עיכבה אותן. זה לא משנה מכיוון שהשרת יודע לא להדפיס פעמיים את אותן חבילות, וללקוח לא משנה שחבילות באות פעמיים.

שלב 4 : foo משנה את סדר שליחת החבילות וגם זורקת חבילות בדבר

ראינו שהפתרון שלנו פותר את שלבים 2 ו-3, כאשר בשלב 2 צריך לוודא כי שהחבילות מגיעות אל השרת, ובשלב 3 צריך לוודא כי השרת ידמיס את החבילות בסדר הנכון. שלב זה משלב את שתי הבעיות משתי השלבים. מכיוון שהפתרון שלנו פותר את שתיהן אז הוא פותר גם את השלב הזה.

הרצנו את הקוד:

```
Ldan@Thunder:~/Desktop/Network/ComputerNetworks-Ex1/partB$ python3 foo.py 55555 127.0.0.1 12345 4
Dropping 76%
Delaying 18%
```

תפסנו את התעבורה בwireshark וקיבלנו:

No.	Time	Source	Destination	Protocol	Length	Info	source port	dest port
2	1.261771939	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
3	2.262974211	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
4	3.264301825	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
5	4.264829471	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
6	4.265079034	127.0.0.1	127.0.0.1	ADW1n ...	142		55555	12345
7	4.265427482	127.0.0.1	127.0.0.1	ADW1n ...	142		12345	55555
8	5.265950871	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
9	6.266791718	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
10	7.268043561	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
11	8.268717649	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
12	8.268965293	127.0.0.1	127.0.0.1	ADW1n ...	142		55555	12345
13	8.269087806	127.0.0.1	127.0.0.1	ADW1n ...	142		12345	55555
14	9.269807190	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
15	10.271005286	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
16	10.271317267	127.0.0.1	127.0.0.1	ADW1n ...	142		55555	12345
17	10.271572914	127.0.0.1	127.0.0.1	ADW1n ...	142		12345	55555
18	11.272268144	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
19	11.272593807	127.0.0.1	127.0.0.1	ADW1n ...	142		55555	12345
20	11.272768989	127.0.0.1	127.0.0.1	ADW1n ...	142		12345	55555
21	11.272968993	127.0.0.1	127.0.0.1	ADW1n ...	142		55555	59061
22	11.273185108	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
23	12.274461843	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
24	13.275634413	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
25	14.276822643	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
26	15.277974983	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
27	16.279134403	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
28	17.280339698	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
31	18.248127625	127.0.0.1	127.0.0.1	ADW1n ...	142		55555	12345
32	18.248309555	127.0.0.1	127.0.0.1	ADW1n ...	142		12345	55555
33	18.281305917	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
34	19.282563404	127.0.0.1	127.0.0.1	ADW1n ...	142		59061	55555
35	19.282874499	127.0.0.1	127.0.0.1	ADW1n ...	142		55555	12345
36	19.283051737	127.0.0.1	127.0.0.1	ADW1n ...	142		12345	55555
37	19.581120404	127.0.0.1	127.0.0.1	ADW1n ...	142		55555	59061

כאשר כרגיל:

פורט השרת הוא 12345

פורט foo הוא 55555

פורט הלקוח נקבע רנדומלית ע"י מערכת ההפעלה, ובמקרה זה הוא 59061

ישנן הרבה חבילות שוב, אך שוב, באמצעות שיטת stop & wait אנחנו יכולים להבטיח שהחבילות תמיד יגיעו לשרת ובסדר הנכון.

הפלט של foo במקרה זה היה:

```
Dropped! randomly... 96
Dropped! randomly... 38
Dropped! randomly... 48
Phiiii, no drop.... 21
Yay, no sleep.... 14 82
Forwarded b'0000000000aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n' from ('127.0.0.1', 59061) to ('127.0.0.1', 12345)
Dropped! randomly... 81
Dropped! randomly... 53
Dropped! randomly... 43
Dropped! randomly... 58
Phiiii, no drop.... 19
Yay, no sleep.... 45 82
Forwarded b'0000000000aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n' from ('127.0.0.1', 59061) to ('127.0.0.1', 12345)
Dropped! randomly... 59
Dropped! randomly... 89
Phiiii, no drop.... 7
Yay, no sleep.... 73 82
Forwarded b'0000000000aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n' from ('127.0.0.1', 59061) to ('127.0.0.1', 12345)
Dropped! randomly... 60
Phiiii, no drop.... 13
Yay, no sleep.... 46 82
Forwarded b'0000000000aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n' from ('127.0.0.1', 59061) to ('127.0.0.1', 12345)
Phiiii, no drop.... 9
Yay, no sleep.... 60 82
Forwarded b'0000000000aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n' from ('127.0.0.1', 12345) to ('127.0.0.1', 59061)
Dropped! randomly... 31
Dropped! randomly... 56
Phiiii, no drop.... 6
Good night.... 97 82
Delaying! for 4.967 seconds...
Dropped! randomly... 88
Dropped! randomly... 83
Dropped! randomly... 72
Dropped! randomly... 74
Forwarded b'0000000001bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb\n' from ('127.0.0.1', 59061) to ('127.0.0.1', 12345)
Dropped! randomly... 65
Dropped! randomly... 28
Phiiii, no drop.... 20
Yay, no sleep.... 12 82
Forwarded b'0000000001bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb\n' from ('127.0.0.1', 59061) to ('127.0.0.1', 12345)
Phiiii, no drop.... 19
Good night.... 91 82
Delaying! for 0.297 seconds...
Forwarded b'0000000001bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb\n' from ('127.0.0.1', 12345) to ('127.0.0.1', 59061)
```