

פרויקט סיום תקשורת

אור 209163054

גיל 304871106

בפרויקט זה נציג מערכת שליחת מסרים והורדת קבצים בין משתמש יחיד למקור, בין משתמשים רבים למקור, בין משתמש יחיד לרבים ובין משתמשים רבים למשתמשים רבים.

שלב ראשון

בנינו מערכת הבנויה משרת ולקוחות הבנויה על שימוש בTCP

פרוטוקול תקשורת אמין המהווה בסיס לרשתות רבות

חילקנו את אופן התוכנית ל2 שכבות

האחת השכבה העליונה היא בשביל בקרת השליחה – החלק שמנהל את חלוקת הקבצים לפאקטות קטנות הנשלחות בחתיכות קטנות מתקבלות ובמקום הקבלה נאספות מחדש עד להרכבת ההודעה המקורית.

השרת - SERVER

השרת מכיל בתוכו רשימה של כל מי שמחובר והוא למעשה שולט במערכת כך שהוא יוצר את החיבור ומאפשר אותו למי שנמצא ברשימה שלו

לכל משתמש סיפקנו מזהה – שם (אחר כך נתאם לו בהתאם גם את הסוקט) שולח בקשות לשרת

השכבה התחתונה – השליחה למיקום הנכון לפי הIP המתאים. לכל כתובת הגדרנו לקוח, כל משתמש לפי שם משלו שיאפיין אותו וכך השרת ידע לאן לשלוח את המידע המבוקש או ההודעה.

שלב שני

בחלק זה נדרשנו להוסיף שכבת קוד להעברת הקבצים מעל UDP שיטה אשר נקרא לה - FAST reliable UDP

בחרנו לעשות זאת בעזרת

Reliable Data Transfer over UDP = העברת נתונים אמינה באמצעות UDP – היות וUDP הינו פרוטוקול לא אמין אז בכדי לחפות על זה נוסיף מהימנות בעזרת פרוטוקול חזרה הפעול סלקטיבית.

הפעלת השרת והאזנה ללקוחות :

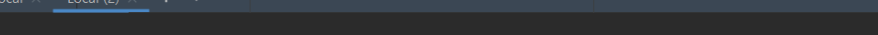
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\גורבסקו\ליג גורבסקו\Desktop\ליג\סידומיל\אירא\פשת\לאירא\oop\code\Assignments\finalProjectNetwork> ./server
Listening...
```

הלקוחות/ המשתמשים

התחברות לשרת




```
Terminal: Local x Local (2) x + v

Try the new cross-platform PowerShell https://aka.ms/pscore6

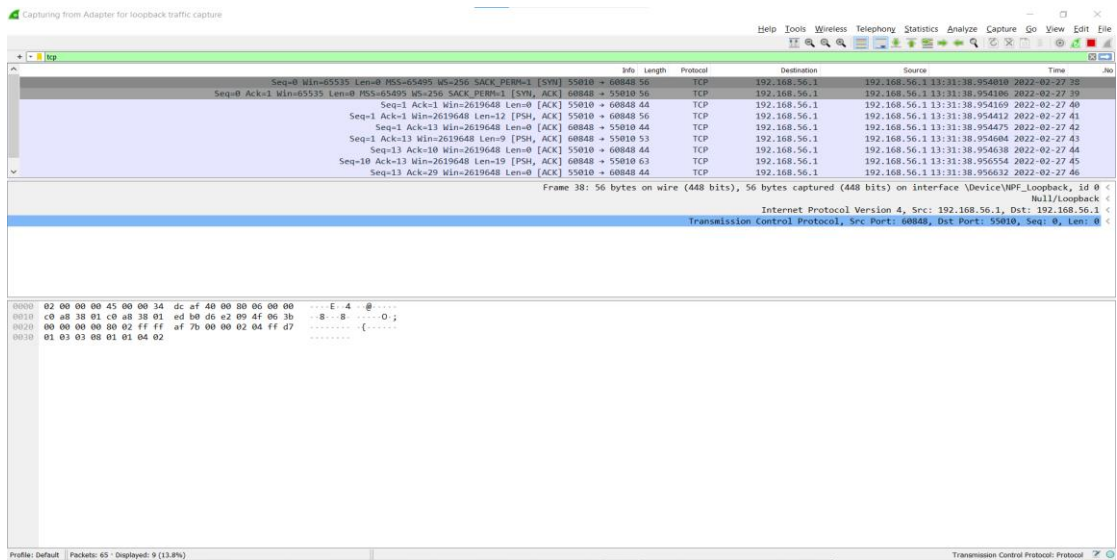
PS C:\Users\סוים\finalProjectNetwork> ./Client
connect client1
connected
new Client: client1
```

אימות החיבור מצד השרת והוספת הלקוח לרשימת ה'מנויים'

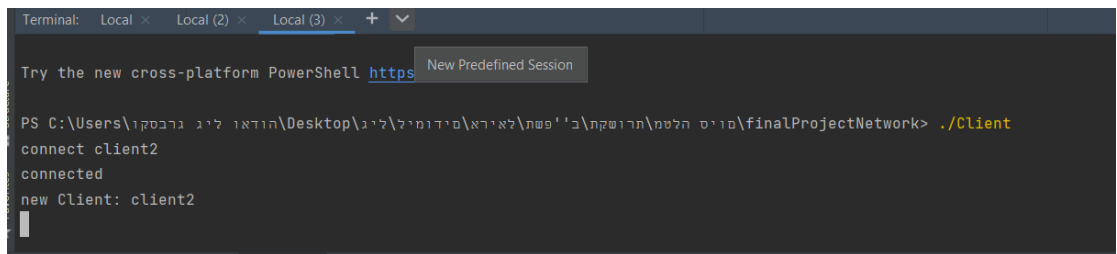


```
Terminal: Local x Local (2) x + v
Try the new cross-platform PowerShell https://aka.ms/pscore6

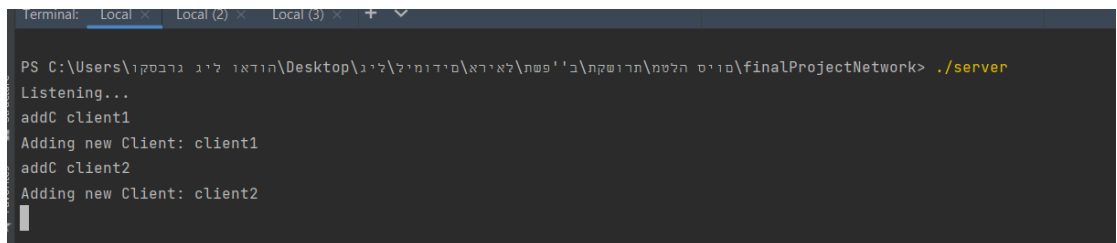
PS C:\Users\גורבסקו\ליג הודאו\Desktop\ליג הודאו\סיידומיל\ליג הודאו\לאירא'פשת'פשת\תרושקת\ב'פשת'פשת> ./server
Listening...
addC client1
Adding new Client: client1
```



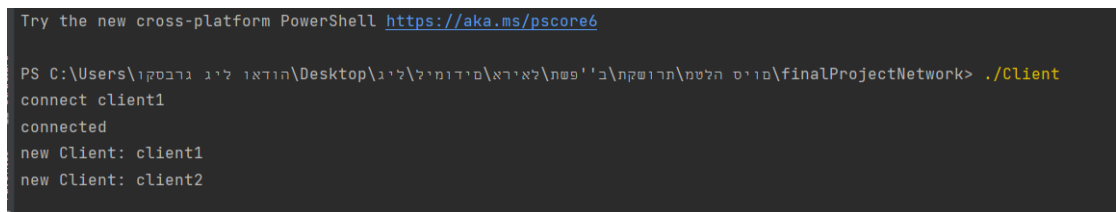
הוספת משתמש נוסף



הוספת המשתמש הנוסף מצד של השרת



עדכון על משתמש חדש אצל המשתמש/ים הקיימ/ים



Capturing from Adapter for loopback traffic capture

Help Tools Wireless Telephony Statistics Analyze Capture Go View Edit File

tcp

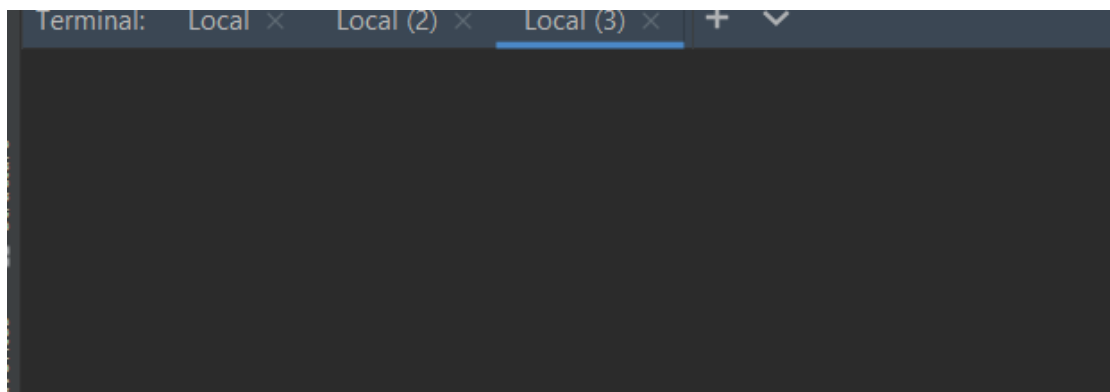
Info	Length	Protocol	Destination	Source	Time
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 [SYN]	55010 → 60848 56	TCP	192.168.56.1	192.168.56.1 13:31:38.954010	2022-02-27 38
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 [SYN, ACK]	60848 → 55010 56	TCP	192.168.56.1	192.168.56.1 13:31:38.954106	2022-02-27 39
Seq=1 Ack=1 Win=2619648 Len=0 [ACK]	55010 → 60848 44	TCP	192.168.56.1	192.168.56.1 13:31:38.954169	2022-02-27 40
Seq=1 Ack=1 Win=2619648 Len=12 [PSH, ACK]	55010 → 60848 56	TCP	192.168.56.1	192.168.56.1 13:31:38.954412	2022-02-27 41
Seq=1 Ack=13 Win=2619648 Len=0 [ACK]	60848 → 55010 44	TCP	192.168.56.1	192.168.56.1 13:31:38.954475	2022-02-27 42
Seq=1 Ack=13 Win=2619648 Len=9 [PSH, ACK]	60848 → 55010 53	TCP	192.168.56.1	192.168.56.1 13:31:38.954604	2022-02-27 43
Seq=13 Ack=10 Win=2619648 Len=0 [ACK]	55010 → 60848 44	TCP	192.168.56.1	192.168.56.1 13:31:38.954638	2022-02-27 44
Seq=10 Ack=13 Win=2619648 Len=19 [PSH, ACK]	60848 → 55010 63	TCP	192.168.56.1	192.168.56.1 13:31:38.956554	2022-02-27 45
Seq=13 Ack=29 Win=2619648 Len=0 [ACK]	55010 → 60848 44	TCP	192.168.56.1	192.168.56.1 13:31:38.956632	2022-02-27 46
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 [SYN]	60862 → 60863 56	TCP	127.0.0.1	127.0.0.1 13:32:51.495464	2022-02-27 86
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 [SYN, ACK]	60863 → 60862 56	TCP	127.0.0.1	127.0.0.1 13:32:51.495559	2022-02-27 87
Seq=1 Ack=1 Win=2619648 Len=0 [ACK]	60862 → 60863 44	TCP	127.0.0.1	127.0.0.1 13:32:51.495620	2022-02-27 88
Seq=1 Ack=1 Win=2619648 Len=1 [PSH, ACK]	60863 → 60862 45	TCP	127.0.0.1	127.0.0.1 13:33:51.518306	2022-02-27 113
Seq=1 Ack=2 Win=2619648 Len=0 [ACK]	60862 → 60863 44	TCP	127.0.0.1	127.0.0.1 13:33:51.518449	2022-02-27 114
Seq=2 Ack=1 Win=2619648 Len=0 [FIN, ACK]	60863 → 60862 44	TCP	127.0.0.1	127.0.0.1 13:33:51.518827	2022-02-27 115
Seq=1 Ack=3 Win=2619648 Len=0 [ACK]	60862 → 60863 44	TCP	127.0.0.1	127.0.0.1 13:33:51.518856	2022-02-27 116
Seq=1 Ack=3 Win=2619648 Len=0 [FIN, ACK]	60862 → 60863 44	TCP	127.0.0.1	127.0.0.1 13:33:51.518886	2022-02-27 117
Seq=3 Ack=2 Win=2619648 Len=0 [ACK]	60863 → 60862 44	TCP	127.0.0.1	127.0.0.1 13:33:51.518921	2022-02-27 118
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 [SYN]	55010 → 60870 56	TCP	192.168.56.1	192.168.56.1 13:34:07.629238	2022-02-27 125
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 [SYN, ACK]	60870 → 55010 56	TCP	192.168.56.1	192.168.56.1 13:34:07.629338	2022-02-27 126
Seq=1 Ack=1 Win=2619648 Len=0 [ACK]	55010 → 60870 44	TCP	192.168.56.1	192.168.56.1 13:34:07.629429	2022-02-27 127
Seq=1 Ack=1 Win=2619648 Len=12 [PSH, ACK]	55010 → 60870 56	TCP	192.168.56.1	192.168.56.1 13:34:07.629668	2022-02-27 128
Seq=1 Ack=13 Win=2619648 Len=0 [ACK]	60870 → 55010 44	TCP	192.168.56.1	192.168.56.1 13:34:07.629728	2022-02-27 129
Seq=1 Ack=13 Win=2619648 Len=9 [PSH, ACK]	60870 → 55010 53	TCP	192.168.56.1	192.168.56.1 13:34:07.629803	2022-02-27 130
Seq=13 Ack=10 Win=2619648 Len=0 [ACK]	55010 → 60870 44	TCP	192.168.56.1	192.168.56.1 13:34:07.630935	2022-02-27 131
Seq=29 Ack=13 Win=2619648 Len=19 [PSH, ACK]	60848 → 55010 63	TCP	192.168.56.1	192.168.56.1 13:34:07.630491	2022-02-27 132
Seq=13 Ack=48 Win=2619648 Len=0 [ACK]	55010 → 60848 44	TCP	192.168.56.1	192.168.56.1 13:34:07.630538	2022-02-27 133
Seq=10 Ack=13 Win=2619648 Len=19 [PSH, ACK]	60870 → 55010 63	TCP	192.168.56.1	192.168.56.1 13:34:07.630583	2022-02-27 134
Seq=13 Ack=29 Win=2619648 Len=0 [ACK]	55010 → 60870 44	TCP	192.168.56.1	192.168.56.1 13:34:07.630617	2022-02-27 135

Frame 38: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_{...} id 0 <
 Null/Loopback <
 Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1 <
 Transmission Control Protocol, Src Port: 60848, Dst Port: 55010, Seq: 0, Len: 0 <

0000 02 00 00 00 45 00 00 34 dc af 40 00 80 06 00 00 ---E:4 --@---
 0010 c0 a8 38 01 c0 a8 38 01 ed b0 d6 e2 09 4f 06 3b --8--8: ---0-;
 0020 00 00 00 00 00 00 ff ff af 7b 00 00 02 04 ff d7 ---:--- -(---
 0030 01 03 03 08 01 01 04 02 ---:---

Profile: Default Packets: 172 Displayed: 29 (16.9%) Transmission Control Protocol: Protocol

התנתקות client2(Local2) מהשרת מצד הלקוח שהתנתק:



התנתקות הלקוח client2(Local2) מצד השרת

```
disconnect
disconnecting: client2
Exception in thread Thread-2:
Traceback (most recent call last):
```

התנתקות client2(Local2) מצד המשתמשים הנוספים שעדיין מחוברים

```
Terminal: Local x Local (2) x
Windows PowerShell
Copyright (C) Microsoft Corp

Try the new cross-platform P

PS C:\Users\גורבסקו\ליג הודאו
connect client1
connected
new Client: client1
new Client: client2
disconnected: client2
```

Seq=10 Ack=13 Win=2619648 Len=19 [PSH, ACK] 60870 → 55010 63	TCP	192.168.56.1	192.168.56.1 13:34:07.630583	2022-02-27 134
Seq=13 Ack=29 Win=2619648 Len=0 [ACK] 55010 → 60870 44	TCP	192.168.56.1	192.168.56.1 13:34:07.630617	2022-02-27 135
Seq=13 Ack=29 Win=2619648 Len=18 [PSH, ACK] 55010 → 60870 62	TCP	192.168.56.1	192.168.56.1 13:35:44.029534	2022-02-27 184
Seq=29 Ack=31 Win=2619648 Len=0 [ACK] 60870 → 55010 44	TCP	192.168.56.1	192.168.56.1 13:35:44.029592	2022-02-27 185
Seq=48 Ack=13 Win=2619648 Len=21 [PSH, ACK] 60848 → 55010 65	TCP	192.168.56.1	192.168.56.1 13:35:44.029909	2022-02-27 186
Seq=13 Ack=69 Win=2619648 Len=0 [ACK] 55010 → 60848 44	TCP	192.168.56.1	192.168.56.1 13:35:44.029954	2022-02-27 187
Seq=29 Ack=31 Win=2619648 Len=21 [PSH, ACK] 60870 → 55010 65	TCP	192.168.56.1	192.168.56.1 13:35:44.029989	2022-02-27 188
Seq=31 Ack=50 Win=2619648 Len=0 [ACK] 55010 → 60870 44	TCP	192.168.56.1	192.168.56.1 13:35:44.030032	2022-02-27 189
Seq=50 Ack=31 Win=2619648 Len=0 [FIN, ACK] 60870 → 55010 44	TCP	192.168.56.1	192.168.56.1 13:35:44.030119	2022-02-27 190
Seq=31 Ack=51 Win=2619648 Len=0 [ACK] 55010 → 60870 44	TCP	192.168.56.1	192.168.56.1 13:35:44.030152	2022-02-27 191
Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 [SYN] 60898 → 60899 56	TCP	127.0.0.1	127.0.0.1 13:35:51.526522	2022-02-27 194
Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 [SYN, ACK] 60899 → 60898 56	TCP	127.0.0.1	127.0.0.1 13:35:51.526804	2022-02-27 195
Seq=1 Ack=1 Win=2619648 Len=0 [ACK] 60898 → 60899 44	TCP	127.0.0.1	127.0.0.1 13:35:51.527267	2022-02-27 196

Frame 38: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_{...} id 0 <
Null/Loopback <
Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1 <
Transmission Control Protocol, Src Port: 60848, Dst Port: 55010, Seq: 0, Len: 0 <

02 00 00 00 45 00 00 34 dc af 40 00 00 06 00 00E..4..@....
c0 a8 38 01 c0 a8 38 01 ed b0 d6 e2 09 4f 06 3b ..8..8.....0.;
00 00 00 00 00 02 ff ff af 7b 00 00 02 04 ff d7{.....
03 03 08 01 01 04 02|.....

שליחת הודעה לכלל המחוברים מצד הלקוח השולח:

```
Terminal: Local x Local (2) x Local (4) x + v
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\גורבסקו\ליג הודאו\Desktop\ליג הודאו\סימולומים\פשת\לאירא\מידומיל\ליג\finalProjectNetwork> ./Client
connect client3
connectednew Client: client3
set_msg_all "hi evreyone"
client3: "hi evreyone"
```

קבלת הודעה שנשלחה לכלל המחוברים מצד הלקוח המקבל:

```

Terminal: Local x Local (2) x Local (4) x + v
connect client1
connected
new Client: client1
new Client: client2
disconnected: client2
new Client: client3
client3: "hi evreystone"

```

שליחת הודעה לכלל המחברים מצד השרת:

```

Terminal: Local x Local (2) x Local (4) x + v
File "C:\Users\מיקי\Documents\לימודים\פיתוח\איראן\סימוליון\ליב\הודאו ליב גרסאות\finalProject.py":
print("Error: " + e)
TypeError: can only concatenate str (not "RuntimeError") to str
addC client3
Adding new Client: client3
set_msg_all "hi evreystone"
sending broadcast message: client3: "hi evreystone"

```

Seq	Ack	Win	Len	Flags	Source	Destination	Timestamp	Window
Seq=13	Ack=29	Win=2619648	Len=26	[PSH, ACK]	192.168.56.1	192.168.56.1	13:37:46.218275	2022-02-27 295
Seq=29	Ack=39	Win=2619648	Len=0	[ACK]	192.168.56.1	192.168.56.1	13:37:46.218354	2022-02-27 296
Seq=88	Ack=13	Win=2619648	Len=23	[PSH, ACK]	192.168.56.1	192.168.56.1	13:37:46.218685	2022-02-27 297
Seq=13	Ack=111	Win=2619648	Len=0	[ACK]	192.168.56.1	192.168.56.1	13:37:46.218749	2022-02-27 298
Seq=29	Ack=39	Win=2619648	Len=23	[PSH, ACK]	192.168.56.1	192.168.56.1	13:37:46.218801	2022-02-27 299
Seq=39	Ack=52	Win=2619648	Len=0	[ACK]	192.168.56.1	192.168.56.1	13:37:46.218845	2022-02-27 300

Frame 295: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{...} id 0 <
 Null/Loopback <
 Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1 <
 Transmission Control Protocol, Src Port: 60914, Dst Port: 55010, Seq: 13, Ack: 29, Len: 26 <
 Data (26 bytes) <

```

0000  02 00 00 00 45 00 00 dc d7 40 00 80 06 00 00 ....E..B..@....
0010  c0 a8 38 01 c0 a8 38 01 ed f2 d6 e2 0c d3 b8 f4 ..8...8-.....
0020  78 ee 5c d6 50 18 27 f9 50 74 00 00 73 65 74 5f x.\.P.'..Pt..set
0030  6d 73 67 5f 61 6c 20 22 20 68 69 20 65 76 72 msg_all "hi evr

```

שליחת הודעה ללקוח יחיד מלקוח :

```

Terminal: Local x Local (2) x Local (4) x + v
connected
new Client: client1
new Client: client2
disconnected: client2
new Client: client3
client3: "hi evreystone"
set_msg client3 "hi there"

```

קבלת הודעה שנשלחה ללקוח יחיד מלקוח :

```
Terminal: Local x Loca
PS C:\Users\גרבסקו\
connect client3
connectednew Client:
set_msg_all "hi evre
client3: "hi evreyon
client1: "hi there"
```

הודעה שנשלחה ללקוח יחיד מלקוח מהצד של השרת:

```
Terminal: Local x Local (2) x Local (4) x + v
Adding new Client: client3
set_msg_all "hi evryone"
sending broadcast massage: client3: "hi evreyo
set_msg client3 "hi there"
sending to client3: client1: "hi there"
```

Seq=13	Ack=111	Win=2619648	Len=30	[PSH, ACK]	55010 → 60848	74	TCP	192.168.56.1	192.168.56.1	13:39:07.861413	2022-02-27 344
Seq=111	Ack=43	Win=2619648	Len=0	[ACK]	60848 → 55010	44	TCP	192.168.56.1	192.168.56.1	13:39:07.861472	2022-02-27 345
Seq=52	Ack=39	Win=2619648	Len=23	[PSH, ACK]	60914 → 55010	67	TCP	192.168.56.1	192.168.56.1	13:39:07.861748	2022-02-27 346
Seq=39	Ack=75	Win=2619648	Len=0	[ACK]	55010 → 60914	44	TCP	192.168.56.1	192.168.56.1	13:39:07.861797	2022-02-27 347

Frame 344: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{...} id 0 <
Null/Loopback <
Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1 <
Transmission Control Protocol, Src Port: 60848, Dst Port: 55010, Seq: 13, Ack: 111, Len: 30 <
Data (30 bytes) <

0000	02 00 00 00 45 00 00 46	dc dd 40 00 80 06 00 00E..F..@....
0010	c0 a8 38 01 c0 a8 38 01	ed b0 d6 e2 09 4f 06 48	--8...8-....0.H
0020	7b 0b 1c 2b 50 18 27 f9	af 9d 00 00 73 65 74 5f	{...+P.'....set_
0030	6d 73 67 20 63 6c 69 65	6e 74 33 20 22 68 65 6c	msg clie nt3 "hel
0040	6c 6f 20 66 72 69 65 6e	64 22	lo frien d"

קבלת רשימת משתמשים מחוברים :

```
Terminal: Local x Local (2) x Local (4) x
disconnected: client2
new Client: client3
client3: "hi evryone"
set_msg client3 "hi there"
set_msg client3 "hi there"
get_users
client1,client3
```

קבלת רשימת משתמשים מחוברים מצד השרת:

```
Terminal: Local x Local (2) x Local (4) x + v
set_msg client3 "hi there"
sending to client3: client1: "hi there"
set_msg client3 "hi there"
sending to client3: client1: "hi there"
get_users
client1,client3
returning the online people: client1,client3
```

No.	Time	Source	Destination	Protocol	Length	Info
3	13:39:51.652087	127.0.0.1	127.0.0.1	TCP	44	Seq=3 Ack=2 Win=2619648 Len=0 [ACK] 60943 → 60942
39	13:40:16.986312	192.168.56.1	192.168.56.1	TCP	53	Seq=39 Ack=75 Win=2619648 Len=9 [PSH, ACK] 55010 → 60914
75	13:40:16.986373	192.168.56.1	192.168.56.1	TCP	44	Seq=75 Ack=48 Win=2619648 Len=0 [ACK] 60914 → 55010
75	13:40:16.987015	192.168.56.1	192.168.56.1	TCP	59	Seq=75 Ack=48 Win=2619648 Len=15 [PSH, ACK] 60914 → 55010
48	13:40:16.987070	192.168.56.1	192.168.56.1	TCP	44	Seq=48 Ack=90 Win=2619648 Len=0 [ACK] 55010 → 60914

Frame 382: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface \Device\NPF_{...} id 0 <
Null/Loopback <
Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1 <
Transmission Control Protocol, Src Port: 60914, Dst Port: 55010, Seq: 39, Ack: 75, Len: 9 <
Data (9 bytes) <

0000 02 00 00 00 45 00 01 dc e1 40 00 80 06 00 00 ...E:1 ..@...
0010 c0 a8 38 01 c0 a8 38 01 ed f2 d6 e2 0c d3 b9 0e ..8...8
0020 78 ee 5d 04 50 18 27 f9 0c 22 00 00 67 65 74 5f x-]P-.-.-.get_
0030 75 73 65 72 73 users

קבלת רשימת קבצים הקיימת בשרת :

```
Terminal: Local x Local (2) x Local (4) x + v
connect client3
connectednew Client: client3
set_msg_all "hi evreyone"
client3: "hi evreyone"
client1: "hi there"
get_list_file
['.git', '.idea', 'Client.py', 'README.md', 'server.py', 'v.txt', '__pycache__', 'קלמ'ב.docx', 'קלמ'ג.docx']
```

קבלת רשימת קבצים הקיימת בשרת מצד השרת :

Terminal: Local × Local (2) × Local (4) × + ▾

```

set_msg client3 "hi there"
sending to client3: client1: "hi there"
get_users
client1,client3
returning the online people: client1,client3
get_list_file
The files are: ['.git', '.idea', 'Client.py', 'README.md', 'server.py', 'v.txt', '__pycache__', 'ב קליח.docx', 'ג קליח.docx']

```

No.	Time	Source	Destination	Protocol	Length	Info
43	0.0000000	192.168.56.1	192.168.56.1	TCP	57	Seq=43 Ack=111 Win=2619648 Len=13 [PSH, ACK] 55010 → 60848 57
111	0.0000000	192.168.56.1	192.168.56.1	TCP	44	Seq=111 Ack=56 Win=2619648 Len=0 [ACK] 60848 → 55010 44
111	0.0000000	192.168.56.1	192.168.56.1	TCP	160	Seq=111 Ack=56 Win=2619648 Len=116 [PSH, ACK] 60848 → 55010 160
56	0.0000000	192.168.56.1	192.168.56.1	TCP	44	Seq=56 Ack=227 Win=2619392 Len=0 [ACK] 55010 → 60848 44

Frame 443: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface \Device\NPF_{...} id 0 <

Null/Loopback <

Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1 <

Transmission Control Protocol, Src Port: 60848, Dst Port: 55010, Seq: 43, Ack: 111, Len: 13 <

Data (13 bytes) <

```

0000  02 00 00 00 45 00 00 35  dc e5 40 00 80 06 00 00  ....E..5..@....
0010  c0 a8 38 01 c0 a8 38 01  ed b0 d6 e2 09 4f 06 66  ..8..8.....0.f
0020  7b 0b 1c 2b 50 18 27 f9  41 7e 00 00 67 65 74 5f  {...P.'A...get_
0030  6c 69 73 74 5f 66 69 6c  65                          list_file

```

Profile: Default | Packets: 451 | Displayed: 85 (18.8%)
Transmission Control Protocol: Protocol ? 0

הורדת קובץ מהשרת :

```
download v.txt
receiving...
it is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a v
it is a vit is a vit is a vit is a vit is a v
it is a vit is a vit is a v
it is a vit is a vit is a v
it is a vit is a vi
You downloaded 100.28% out of file. Last byte is: 70246.
t is a v
it is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a v
it is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a v
it is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a vit is a v
it is a vit is a vit is a vit is a vit is a vit is a vit is a v
it is a vit is a vit is a vit is a vit is a vit is a vit is a v
You downloaded 100.86% out of file. Last byte is: 70651.
```

הורדת קובץ מהשרת מצד השרת:

[illegible]

	Seq=1 Ack=1 Win=2619648 Len=1 [PSH, ACK] 60953 → 60952 45	TCP	127.0.0.1	127.0.0.1[13:42:51.664091 2022-02-27 585]
	Seq=1 Ack=2 Win=2619648 Len=0 [ACK] 60952 → 60953 44	TCP	127.0.0.1	127.0.0.1[13:42:51.664137 2022-02-27 586]
	Seq=2 Ack=1 Win=2619648 Len=0 [FIN, ACK] 60953 → 60952 44	TCP	127.0.0.1	127.0.0.1[13:42:51.664563 2022-02-27 587]
	Seq=1 Ack=3 Win=2619648 Len=0 [ACK] 60952 → 60953 44	TCP	127.0.0.1	127.0.0.1[13:42:51.664597 2022-02-27 588]
	Seq=1 Ack=3 Win=2619648 Len=0 [FIN, ACK] 60952 → 60953 44	TCP	127.0.0.1	127.0.0.1[13:42:51.664639 2022-02-27 589]
v	Seq=3 Ack=2 Win=2619648 Len=0 [ACK] 60953 → 60952 44	TCP	127.0.0.1	127.0.0.1[13:42:51.664696 2022-02-27 590]

Frame 585: 45 bytes on wire (360 bits), 45 bytes captured (360 bits) on interface \Device\NPF_{loopback}, id 0
Null/Loopback

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 60952, Dst Port: 60953, Seq: 1, Ack: 1, Len: 1
Data (1 byte)

```

0000 02 00 00 00 45 00 00 29 fa 38 40 00 00 06 00 00 ....E...)..8@...
0010 7f 00 00 01 7f 00 00 01 ee 18 ee 19 1c 6c 97 e4 .....L...
0020 53 55 ed 12 50 18 27 f9 b7 e4 00 00 01          SUI-P'.....

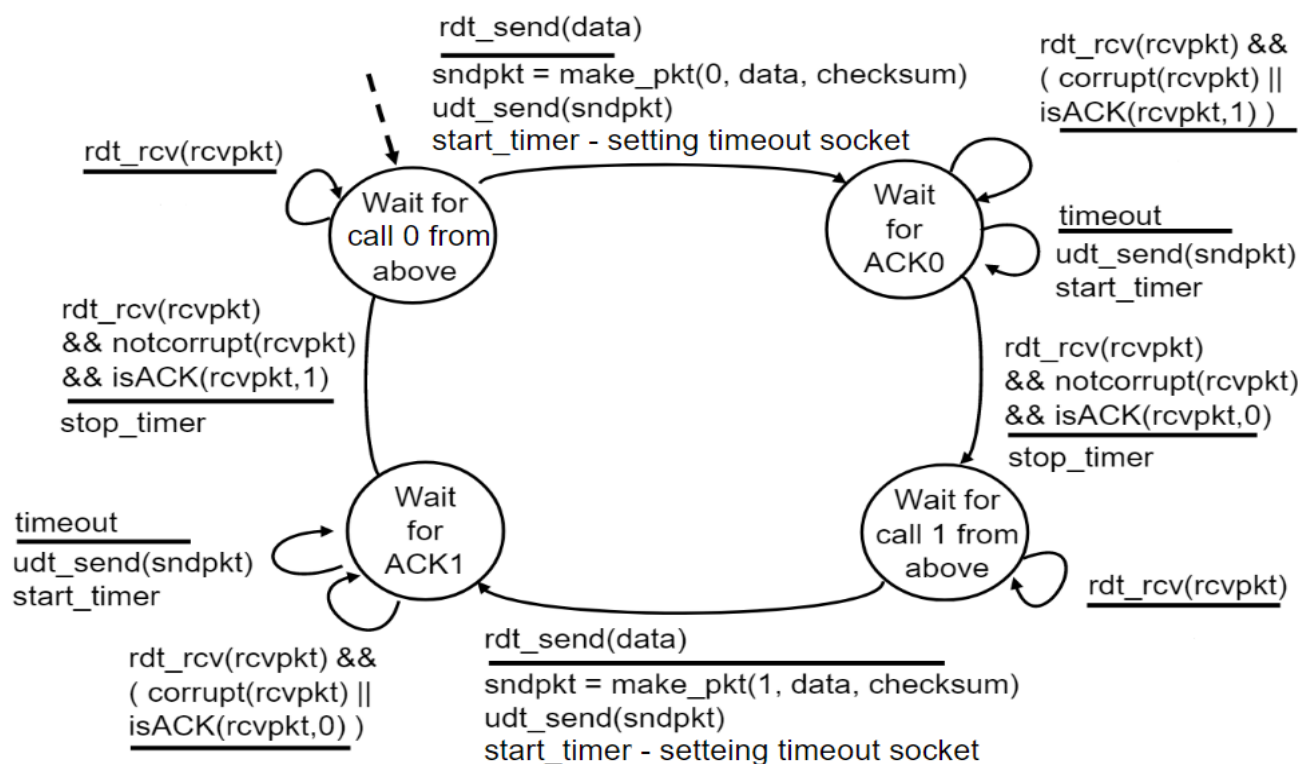
```

בחלק זה של המטלה נתבקשנו להוסיף שכבה חדשה המשתמשת בפרוטוקול מהיר `udp` אך יחד עם זאת אמין

בחרנו לעשות זאת בעזרת RDT over UDP

הפרוטוקול רץ כשכבת אפליקציה בשימוש בפרוטוקול `UDP` שבד"כ הוא בעצמו לא מוסיף הרבה פונקציות נוספות ל

S דיאגרמת המצבים:



`notcorrupt()` - return true if the checksum is valid
`corrupt()` - returns true if the checksum is not valid

המערכת מתגברת על איבוד חבילות כך:

כל חבילה שנשלחת נחכה ל-ACK מתאים (0 או 1) בהתחלה שולחים ACK 0 ולאחר מכן את ACK 1 ואז שוב פעם ACK 0 וכך הלאה. לאחר שהגיע ה-ACK המתאים וגם ה-`checksum` תקני נשלח את החבילה הבאה. כל פעם שנקבל חבילה תקינה נגדיל את הבאפר פי 2 ואם לא נקטין את הבאפר פי 2 כל עוד לא הגענו לערכים המינימליים או המקסימליים הנתונים.

המערכת מתגברת על בעיות latency כך:

כל חבילה שנשלחת יש לה זמן מוגדר `TIMEOUT` ואם עובר זמן זה ועדיין לא קיבלנו את ACK מתאים אז נשלח שוב את החבילה ונקטין את גודל הבאפר (הקטנה פי 2) כל עוד לא הגענו לגודל המינימלי של הבאפר.

חלק ג

בהינתן מחשב חדש המתחבר לרשת אנא תארו את כל ההודעות שעוברות החל מהחיבור הראשוני ל switch ועד שההודעה מתקבלת בצד השני של הצאט. אנא פרטו לפי הפורמט הבא. a: סוג הודעה, פירוט הודעה והשדות הבאים. כתובת IP מקור/יעד, כתובת פורט מקור/יעד, כתובת MAC מקור/יעד, פרוטוקול שכבת התעבורה

ההודעות המתקבלות בעת תהליך התחברות מחשב חדש ל switch

סוג ההודעה	פירוט ההודעה	כתובת IP מקור/יעד	כתובת פורט מקור/יעד	כתובת MAC מקור/יעד	פרוטוקול שכבת התעבורה
unicast	הכתובת לא מוכרת				Ethernet
unicast	הכתובת מוכרת למתג בטבלת ה-MAC				

2. הסבירו מה זה CRC ?

בתחילת ההתקשרות בין המודם השולח והקולט מסכימים ביניהם שני אלו על סוג פרוטוקול השידור בהתאם אליו נקבעת גם סוג הבדיקה – בדיקת שלימות המידע והעברתו CRC או Cheacsum לפיה יוודא המקבל את שלמות המידע שהתקבל.

2. **Cyclic redundancy check - CRC** טכניקה לבקרת שגיאות מאפשרת לגלות שגיאות בשידור, הטכניקה מבוססת על קוד מחזורי הכוונה שימוש בקוד שיטתי, עניין המחזוריות גם מפשט את כתיבתו של הקוד וגם בעיות העברה נפוצות בערוצי תקשורת

מנגון זה הוא סוג של בקרת שגיאות הנקבע בהתאם לפרוטוקול השידור שהוסכם בין המודמים המתקשרים

ה CRC מחושב בצורה מתמטית. ונחשבת למאוד יעילה באחוז בוודאות של המידע שעבר.

3.

QUIC – הוא פרוטוקול תקשורת (פיתוח של google) HTTP-Over-Quic אשר נועד למעשה לקבל את המהירות של UDP מחד ומצד שני לספק אמינות מידע כמו ב TCP. למעשה הוא שולט באופן שבו המידע מועבר ממחשב אחד לאחר.

חשוב ביותר הוא גם שהשליטה על אחוז העברת המידע שפה הוא נעשה בצורה מהירה יותר מאשר השימוש ב TCP

Quic מהיר יותר מ-TCP גם ביצירת קשרים מוצפנים .

http

פרוטוקול בו משתמשים בweb 2 סוגי בקשות בפרוטוקול זה ישנם 2 סוגי הודעות בקשה ותגובה ובעצם הוא מדמה מודל של 'שרת- לקוח' בו השרת הוא שרת web תפקידו לשלוח אובייקטים ללקוח בהתאם לבקשותיו. הלקוח – הוא הדפדפן תפקידו להציג אובייקטים בהתאם לבקשות

נדבר על 2 גרסאות

http1.1-משתמשים בו בכדי להציג שיטות ניהול מטמון מתוככמות יותר

פה TCP לא נסגר לאחר פעולה כזו או אחרת אלא רק סגר את העסקה

בקשת האובייקטים וקבלתה היא רציפה

http1.0-משתמשים בו לרוב בשביל header . כל חיבור מעבירים אובייקט 1 בלבד לשליחצ רובייקט
נוס = אז יש לפתוח JDS clirbt

4.למה צריך מספרי port ?

פורט נותן לכל כתובת או פרוטוקול מסוים מספר באורך 16 ביטים שזה למעשה מספר הפורט הוא מאפשר העברת נתונים באופן ישיר על ידי שימוש בו הכוונה: על ידי פורט מוכר – (מספר פורט מוכר)

ניתן לאפשר סטנדרט אחיד לפרוטוקולים בהתחברות לשרתים המספקים שירותים מסויימים על מנת שהאפליקציה תבצע משהו בצורה מסוימת היא צריכה לעשות זאת בעזרת הפורט למשל כמו דפדפן כשהוא פונה לאתר אינטרנט בHTTP הוא צריך לפנות לפורט פתוח על השרת כדי שיקבל את הפניות אליו ויטפל בהן – פורט זה הוא פורא המוכר לתעבורת HTTP

5.מה זה subnet ולמה צריך את זה?

Subnet הינו למעשה NETWORK כלומר כתובת הרשת. אך מושג זה מתכוון למעשה ל"תת רשת".

משתמשים בו כאשר יש רשת גדולה הפרושה על מספר גדול של אתרים ורוצה לחלק אותה לתתי רשתות בעזרת נתבים

יכול להיווצר בעיה הגורמת לניתוק הקשר בין האתרים ובשביל זה משתמשים ב subnet mask

שהוא שיטה בה מסמנים עבור המחשב מהו חלק הרשת בכתובת שלו באמצעות מספר דמוי כתובת IP הקרוי subnet mask.

הוא בנוי ממספר בינארי בעל 32 סיביות המורכב מרצף של "1" שלאחריו יופיע רצף של "0" תפקידו היא בעצם לקבוע את כתובת הרשת על ידי הגדרה של מספר הסיביות בכתובת הIP שלה. כתובת הרשת זהה לאורך של רצף האחדות ב subnet למעשה גם את SUBNET ניתן הציג בצורה דיצמלית כמו כתובת הIP כתובת אלו הם למעשה יוצרות מעין תת רשתות המחלקות את הרשתות לקטנות יותר זאת במטרה

א. להימנע ממספר רב של כתובות שאינן נחוצות דבר המבזבז כתובות רבות לשווא – יעילו
ב. מספר רב של כתובות גורם לבעית אבטחה חמורות

6. למה צריך כתובות mac ומדוע לא מספיק לעבוד עם כתובות ip ?

כתובת mac היא מזהה יחודי המוטבע על כל רכיב תקשורת בעת הייצור. בדרך כלל מוטבעת בכרטיס הרשת של המחשב או במודם והוא נחשב כחלק מהשכבה הפיזית של מודל tcp/ip

למעשה כתובת MAC היא 'תעודת הזהות' של המחשב היות וכתובת IP איננה קבועה וגם יתכן כי 2 מחשבים שונים עם אותה כתובת IP לכן יש צורך במזהה יחודי כמו הכתובת MAC אשר מאפיין בצורה ייחודית ובלעדית כל מחשב מרגע היווצרו.

7. מהם ההבדלים בין switch לRouter

ROUTER	SWITCH
מחבר switch מרובים ואת הרשתות המתאימות שלהם	מחבר מכשירי רשת מרובים ברשת
עובד בשכבת network	עובד בשכבת data-link
משתמש ב LAN או ב MAN	משתמש ב LAN
מנהל החלטות ניתוב מהר יותר	מנהל החלטות ניתוב לאט יותר
יכול לעבוד עם רשתות חוטיות ואלחוטיות	יכול לעבוד רק עם רשתות חוטיות – חיבור חוטי
מספק אמצעי אבטחה להגנה על הרשת מפני איומי אבטחה(כללים)	מבחינת הגנה נותן הגנה רק על port
מוגדר כמכשיר רשת חכם	מוגדר כמכשיר קשת חצי חכם

8. שיטות להתגבר על המחסור ב Pv4 ולפרט?

IPv4 היא רשת אינטרנט שבו כתובת IP מוגדרת בת 32 סיביות. החסרון ברשת הוא בכמות הכתובות שהיא מאפשרת, מכיוון שכשהומצא הipv4 לא חשבו שיהיו כל כך הרבה מכשירים שיצטרכו כתובות IP לכן התחילה להיווצר מחסור בכמות הכתובות.

על מנת להתמודד עם בעיה זו הוגדר תקן חדש לכתובות IP גרסה זו אומרת בה כל כתובת IP מורכבת מ 8 קבוצות של 16 סיביות דבר המאפשר מספר עצום של כתובות שונות ובכך פותר את הבעיה

9

e. בעזרת איזה פרוטוקול לומד הנתב 3c על תת רשת x?

תחילה נאבחן את המיקומים והדרך אותה יש לעשות מהנתב לתת הרשת

C3 נמצא בAS3 וx נמצא בAS4 לכן כפי שנתון: במעבר בין AS3 ל AS4 מריצים פרוטוקול BGP – BGP

המשתמש בפרוטוקול TCP לכן כעת נותר לקבל את המידע בין C4 מx ולזה c4 המידע מגיע מתוך אותה רשת פנימית ולכן זה מתבצע בעזרת פרוטוקול RIP שמשתמש בUDP כפי שנתון שAS4 מפעיל RIP

f. בעזרת איזה פרוטוקול לומד הנתב 3a על תת רשת x?

תחילה מיקומים ואיפיונים:

3c הוא Border Gateway ראוטר בAS3 x הוא תת רשת הנמצא בAS4 לכן ביניהם רץ פרוטוקול BGP כנתון, כפי שנאמר כבר פרוטוקול זה משתמש בTCP כעת קבלת המידע מx לc4 וזה נעשה בעזרת הפרוטוקול הפנימי של AS4 שזה RIP שמשתמש בפרוטוקול UDP ומה שנותר הוא קבלת המידע/העברת המידע בין 3c ל3a וזה נעשה כפי הנתון לפי פרוטוקול OSPF

g. 1c לומד על תת הרשת x בעזרת :

חיבור שלו ל3a זאת אומרת - 1c הוא Border Gateway ראוטר של AS1 ולכן קבלת המידע שלו על x היא מ3a וכדי להגיע אליו מופעל פרוטוקול BGP כפי שנתון והיתר זה כמו הסעיף הקודם.

h. 2c לומד על תת הרשת x כך:

ל2c אין חיבור פיזי ישיר ל4AS כפי שנתון ולכן הוא עושה את כל ה'סיבוב'

עד 1c המידע יגיע כמו שפירטנו בסעיפים קודמים משם הוא יועבר ל1b לפי פרוטוקול RIP שמופעל כפי שנתון ב1AS ומשם יועבר בעזרת BGP ל2a. 2a יעביר את המידע ל2c בעזרת OSPF