

# פרוייקט סיום – רשתות תקשורת

## שמות המגשים:

- איתמר גואטה – 206616690
- אביתר יוסף – 207467820

## תוכן עניינים

3	מטרת הפרויקט
4	אופן הרצת התוכנית
6	תשובות
8	דיאגרמת מצבים
9	ביבליוגרפיה

## מטרת הפרויקט

מטרת הפרויקט היא לבנות מערכת שמממשת מערכת תקשורתית בין לקוח לבין מספר שרתים נוספים.

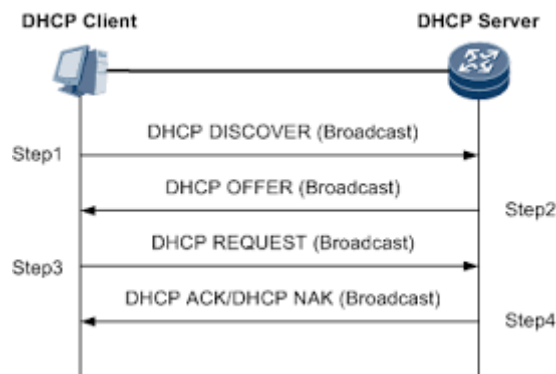
בהתחלה, הלקוח ייגש אל שרת ה-DHCP ויבקש ממנו כתובת IP פנויה.

לאחר-מכן הוא ייגש אל ה-DNS ויבקש את ה-IP לפי הדומיין.

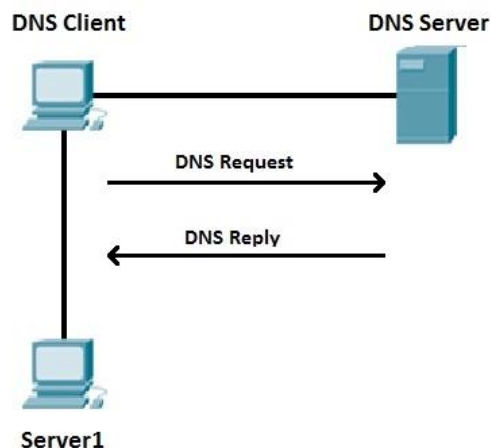
ולבסוף הלקוח ייגש אל האפליקציה. אנו בחרנו באפליקציה מסוג **http redirect**. מה שיקרה זה שהלקוח ייגש אל האפליקציה ויבקש קובץ טקסט, האפליקציה תפנה לסרבר ותבקש את הנתיב לאותו קובץ טקסט. לאחר שסרבר יחזיר לאפליקציה את הנתיב לקובץ, האפליקציה תחזיר ללקוח את הנתיב ובכך יסתיים התהליך.

השרתים שלנו הם:

- **DHCP** - זה השרת שבעצם אחראי על הקצאת כתובות ה-IP. בסך-הכל התקשורת מתבצעת בצורה הבאה:



- **DNS** - שרת ה-DNS אחרי על התאמת דומיינים לכתובות IP. הלקוח מבקש מהשרת בקשת DNS Query והשרת מחזיר לו תגובת DNS Response עם הדומיין המתאים. בסך-הכל התקשורת מתבצעת בצורה הבאה:



## אופן הרצת התוכנית

הקוד שלנו מחולק לשני חלקים. (הקוד רץ על סביבת UBUNTU)

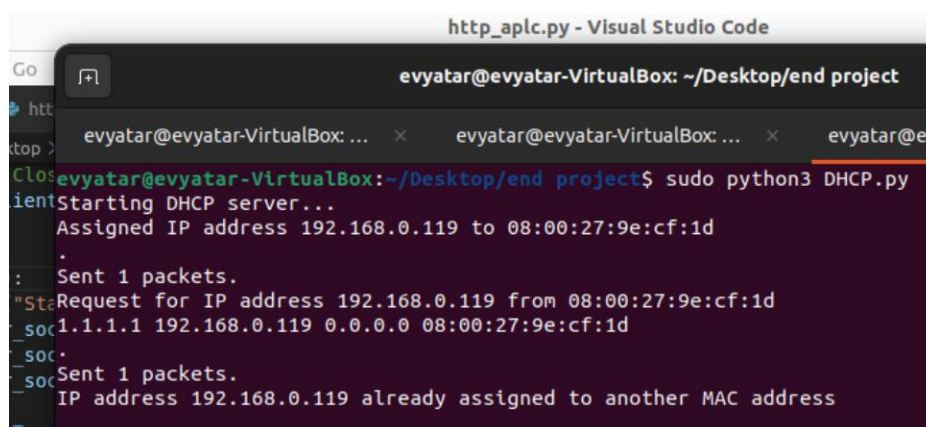
החלק הראשון:

**Client.py**

**DHCP.py**

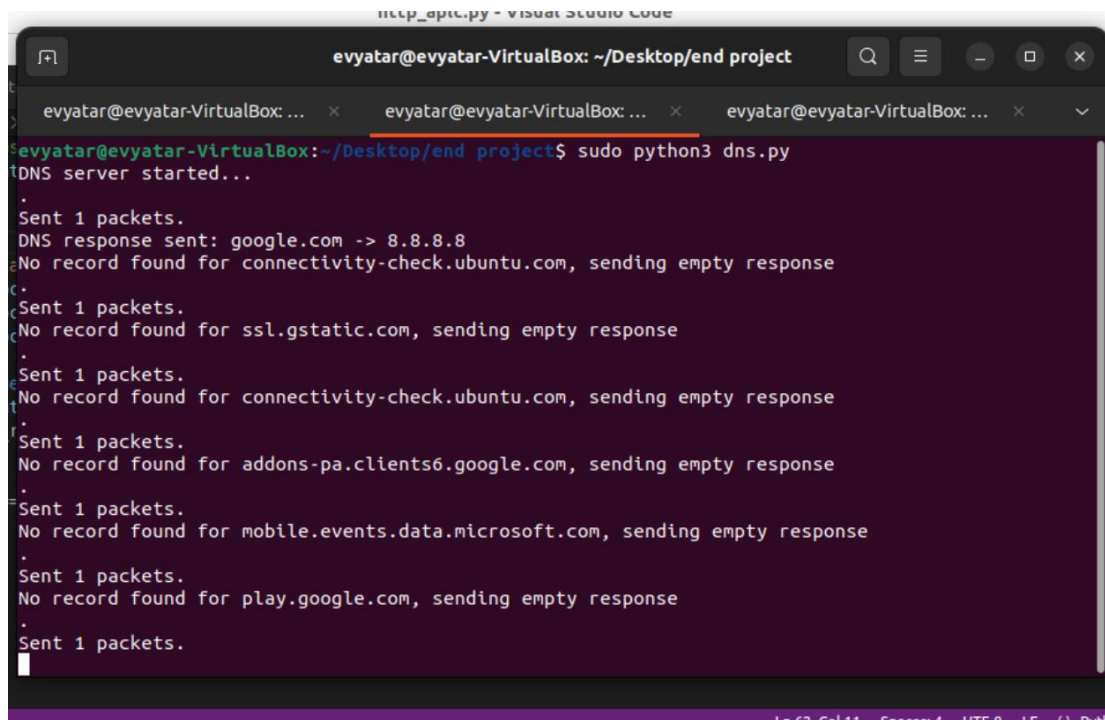
**dns.py**

- נפתח 3 טרמינלים שונים.
- בכל טרמינל נריץ את אחד משלושת הקבצים שמוצגים לעיל ( את **Client.py** נריץ אחרון)



The screenshot shows a terminal window titled "http\_aplc.py - Visual Studio Code" with the command prompt "evyatar@evyatar-VirtualBox: ~/Desktop/end project". The user has executed the command "sudo python3 DHCP.py". The output shows the DHCP server starting, assigning the IP address 192.168.0.119 to the MAC address 08:00:27:9e:cf:1d, and sending 1 packet. It then receives a request for the same IP address from the same MAC address, and responds with "IP address 192.168.0.119 already assigned to another MAC address".

```
evyatar@evyatar-VirtualBox: ~/Desktop/end project$ sudo python3 DHCP.py
Starting DHCP server...
Assigned IP address 192.168.0.119 to 08:00:27:9e:cf:1d
.
Sent 1 packets.
Request for IP address 192.168.0.119 from 08:00:27:9e:cf:1d
1.1.1.1 192.168.0.119 0.0.0.0 08:00:27:9e:cf:1d
.
Sent 1 packets.
IP address 192.168.0.119 already assigned to another MAC address
```



The screenshot shows a terminal window titled "http\_aplc.py - Visual Studio Code" with the command prompt "evyatar@evyatar-VirtualBox: ~/Desktop/end project". The user has executed the command "sudo python3 dns.py". The output shows the DNS server starting, sending 1 packet, and responding with "google.com -> 8.8.8.8". It then receives requests for "connectivity-check.ubuntu.com", "ssl.gstatic.com", "connectivity-check.ubuntu.com", "addons-pa.clients6.google.com", "mobile.events.data.microsoft.com", and "play.google.com", all of which are not found, and sends empty responses.

```
evyatar@evyatar-VirtualBox: ~/Desktop/end project$ sudo python3 dns.py
DNS server started...
.
Sent 1 packets.
DNS response sent: google.com -> 8.8.8.8
No record found for connectivity-check.ubuntu.com, sending empty response
.
Sent 1 packets.
No record found for ssl.gstatic.com, sending empty response
.
Sent 1 packets.
No record found for connectivity-check.ubuntu.com, sending empty response
.
Sent 1 packets.
No record found for addons-pa.clients6.google.com, sending empty response
.
Sent 1 packets.
No record found for mobile.events.data.microsoft.com, sending empty response
.
Sent 1 packets.
No record found for play.google.com, sending empty response
.
Sent 1 packets.
```

```
evyatar@evyatar-VirtualBox: ~/Desktop/end project$ sudo python3 Client.py
Enter 1 to start sending DHCP discovery messages, 2 to ask for domain, or 0 to stop: 1
.
Sent 1 packets.
.
Sent 1 packets.
DHCP ACK received:
IP address: 192.168.0.119
Subnet mask: 255.255.255.0
Default gateway: 196.0.0.11
DNS server: n
press 1 to send dns request or 0 to leave 1
Enter a domain name: google.com
.
Sent 1 packets.
The IP address of google.com is b'google\x03com\x00\x00\x01\x00\x01\x00\x00\x00\x00\x00\x04\x08\x08\x08\x08'
press 1 for send a dhcp request or 0 to leave
```

החלק השני:

http\_client.py

http\_aplc.py

redirect\_server.py

- נפתח 3 טרמינלים שונים.
- בכל טרמינל נריץ את אחד משלושת הקבצים שמוצגים לעיל ( את http\_client.py נריץ אחרון)

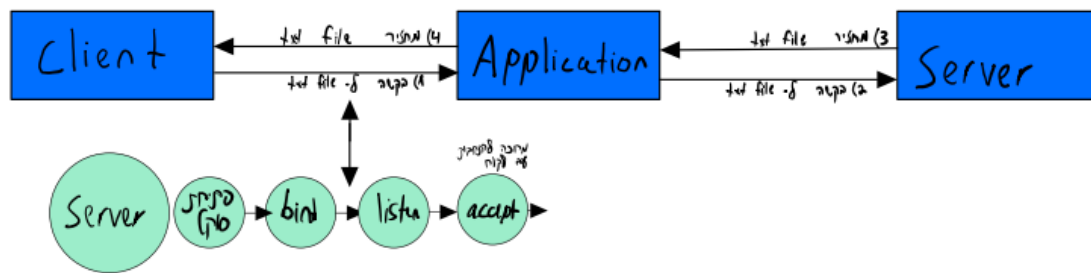
```
evyatar@evyatar-VirtualBox: ~/Desktop/end project$ sudo python3 http_aplc.py
[sudo] password for evyatar:
Starting HTTP server...
Received connection from: ('127.0.0.1', 50488)
Received GET request: GET /redirect HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0
```

```
evyatar@evyatar-VirtualBox: ~/Desktop/end project$ sudo python3 redirect_server.py
[sudo] password for evyatar:
starting up on 127.0.0.1 port 8001
waiting for a connection
connection from ('127.0.0.1', 33808)
done sending file
waiting for a connection
```

```
evyatar@evyatar-VirtualBox: ~/Desktop/end project$ sudo python3 http_client.py
[sudo] password for evyatar:
Sending request to HTTP server...
Received response from HTTP server.

printing the text:
evyatar - 207467820
itamar - 206616690
```

## דיאגרמת מצבים



## תשובות

(1) ההבדלים העיקריים בין TCP ל-QUIC הם:

- יצירת חיבור: ב-TCP, יצירת חיבור היא לחיצת יד תלת כיוונית, בעוד שב-QUIC, מדובר בתהליך של סיבוב אחד או שניים. המשמעות היא ש-QUIC יכול ליצור חיבורים מהר יותר מ-TCP.
- בקרת גודש: לשני הפרוטוקולים יש מנגנונים למנוע עומס ברשת ולהבטיח מסירה אמינה של נתונים. עם זאת, QUIC משתמש באלגוריתם בקרת גודש מודרני יותר שלוקח בחשבון את תנאי הרשת ומסתגל מהר יותר לשינויים בעומס.
- מספר משתמשים - QUIC מאפשר שליחת זרמי נתונים מרובים בחיבור יחיד, בעוד ש-TCP דורש חיבור נפרד לכל זרם.
- אבטחת שכבת התעבורה - גם TCP וגם QUIC תומכים בהצפנה, אבל QUIC משתמש באבטחת שכבת תחבורה (TLS) כברירת מחדל, בעוד ש-TCP משתמש בה באופן אופציונלי.
- הבדל בגודל הפקטות: ב-TCP, חבילות יכולות להיות בגודל של עד 65,535 בתים. לעומת פקטות של QUIC שלהן אין הגבלה.

(2) ההבדלים העיקריים בין CUBIC ל-VEGAS הם:

- וגאס הוא אלגוריתם בקרת גודש המבוסס על עיכוב (delay), בעוד ש-קיוביק הוא אלגוריתם בקרת גודש מבוסס אובדן פקטות (packet loss) וגאס משתמשת ב-Round-Trip Time (RTT) כדי לקבוע את גודל חלון הגודש המתאים, בעוד קיוביק עוקב אחר אובדן מנות כדי להתאים את חלון הגודש.
- CUBIC שומר על חלוקה שווה של רוחב פס בין זרמים שונים של מידע לעומת VEGAS שאינה מבטיחה חלוקה שווה אלא נותנת עדיפות לזמני עיכוב נמוכים ואובדן פקטות נמוך מה שעלול לגרום לאי אחידות ברוחב הפס.
- היענות לתנאי הרשת: קיוביק נחשבת מגיבה יותר לגודש ברשת מאשר וגאס מכיוון שהיא מגיבה מהר יותר לאובדן מנות. וגאס נוטה להיות שמרנית יותר ואיטית יותר כדי להגדיל את גודל חלון הגודש שלה, מה שעלול לגרום לתפוקה נמוכה יותר.

(3) BGP (Border Gateway Protocol) הוא פרוטוקול ניתוב המשמש להחלפת מידע ניתוב בין מערכות אוטונומיות שונות (AS) באינטרנט. הוא נועד לספק ניתוב ניתן להרחבה, יציב וגמיש ברחבי האינטרנט העולמי. בניגוד ל-OSPF (Open Shortest Path First), שהוא פרוטוקול שער פנימי המשמש לניתוב בתוך מערכת אוטונומית אחת, BGP הוא פרוטוקול שער חיצוני המשמש לניתוב בין מערכות אוטונומיות שונות. OSPF נועד למצוא את הנתיב הקצר ביותר בתוך רשת, בעוד BGP נועד לבחור את הנתיב הטוב ביותר בין רשתות שונות. אחד ההבדלים העיקריים בין OSPF ל-BGP הוא הדרך שבה הם קובעים את הדרך הטובה ביותר ליעד. OSPF משתמש באלגוריתם ניתוב מצב קישור, שלוקח בחשבון את הנתיב הקצר ביותר בין צמתים בהתבסס על עלות הקישורים ביניהם. BGP, לעומת זאת, משתמש באלגוריתם של וקטור נתיב, שלוקח בחשבון גורמים כמו מספר הקפצות, איכות הרשת והמדיניות של המערכות האוטונומיות השונות. BGP בדרך כלל מנסה למצוא את הנתיב הקצר ביותר ליעד, אך הוא לוקח בחשבון גם גורמים אחרים כגון זמינות הנתיב והאמינות. בנוסף BGP מאפשר למנהלי רשת לשלוט

בהחלטות הניתוב על ידי קביעת מדיניות המבוססת על הדרישות העסקיות שלהם. מבחינת האם BGP עובד לפי מסלולים קצרים, התשובה היא בדרך כלל כן, אבל זה תלוי בתצורת הרשת הספציפית ובמדיניות שנקבעה על ידי מנהל הרשת. BGP מנסה למצוא את הדרך הקצרה ביותר ליעד, אך היא יכולה לקחת בחשבון גם גורמים נוספים כמו איכות הניתוב ומהימנותו. בנוסף, המדיניות שנקבעה על ידי מנהל הרשת יכולה להשפיע על החלטות הניתוב שהתקבלו על ידי BGP, כך שניתן ל-BGP לבחור נתיב ארוך יותר בהתבסס על מדיניות זו.

(4)

Application	Port Src	Port Des	IP Src	IP Des	Mac Src	Mac Des
Dhcp	67	68	10.0.2.2	192.168.0.136	08:00:27:9E:CE:1D	08:00:27:9E:CE:1D

Application	Port Src	Port Des	IP Src	IP Des	Mac Src	Mac Des
DNS	20820	53	10.0.2.15	5.24.1436	08:00:27:9E:CE:1D	08:00:27:9E:CE:1D

Application	Port Src	Port Des	IP Src	IP Des	Mac Src	Mac Des
App	20820	30820	127.0.0.1	127.0.0.1	08:00:27:9E:CE:1D	08:00:27:9E:CE:1D

Application	Port Src	Port Des	IP Src	IP Des	Mac Src	Mac Des
Server	30820	20820	127.0.0.1	127.0.0.1	08:00:27:9E:CE:1D	08:00:27:9E:CE:1D

אם היה NAT אז הוא ישנה כתובת ה-IP של ה-SRC בכתובת ציבורית. ואת כתובת ה-IP של ה-DES הוא יחליף בכתובת הפרטית של הרשת. אם נשתמש בפרוטוקול QUIC אז ה-porta src של ה-TCP ישתנה לפורט הדיפולטיבי 443.

(5) ARP (Address Resolution Protocol) ו-DNS (Domain Name System) הם שני פרוטוקולים שונים המשמשים ברשתות מחשבים למטרות שונות. ARP משמש למיפוי כתובת רשת (כגון כתובת IP) לכתובת פיזית (כגון כתובת MAC). הוא משמש כדי להקל על תקשורת בין מכשירים ברשת מקומית. כאשר מכשיר רוצה לשלוח נתונים למכשיר אחר באותה רשת, הוא צריך לדעת את הכתובת הפיזית (MAC) של מכשיר היעד. ARP משמש כדי לפתור את כתובת ה-IP של מכשיר היעד לכתובת ה-MAC שלו. DNS, לעומת זאת, משמש לתרגום שמות דומיינים הניתנים לקריאה אנושית (כגון www.AMITDVIR.com) לכתובות IP שבהן מחשבים יכולים להשתמש כדי לאתר שרתים ושירותים באינטרנט. כאשר משתמש מזין שם תחום בדפדפן אינטרנט, הדפדפן שולח שאילתת DNS לשרת DNS כדי למצוא את כתובת ה-IP המשויכת לשם תחום זה. לאחר מכן שרת ה-DNS מחזיר את כתובת ה-IP לדפדפן, אשר יכול להשתמש בה כדי ליצור חיבור לשרת המארח את האתר. לסיכום, ARP משמש לתקשורת רשת מקומית למיפוי כתובות IP לכתובות MAC, בעוד DNS משמש לתקשורת אינטרנט למיפוי שמות דומיין לכתובות IP.



## ביבליוגרפיה

**Geeks for geeks:**

<https://www.geeksforgeeks.org/socket-programming-cc/Wikipedia:>

**Moodle of the course:**

You know what the url is..

**Witestlab:**

<https://witestlab.poly.edu/blog/tcp-congestion-control-basics>

**CS.Princeton.edu:**

<https://www.cs.princeton.edu/courses/archive/fall16/cos561/papers/Cubic08.pdf>

**TCP.Lafibre:**

[https://lafibre.info/images/doc/201207\\_TCP\\_Congestion\\_Control\\_Comparison.pdf](https://lafibre.info/images/doc/201207_TCP_Congestion_Control_Comparison.pdf)

[https://en.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://en.wikipedia.org/wiki/Transmission_Control_Protocol)

[https://en.wikipedia.org/wiki/TCP\\_Vegas](https://en.wikipedia.org/wiki/TCP_Vegas)

[https://en.wikipedia.org/wiki/CUBIC\\_TCP](https://en.wikipedia.org/wiki/CUBIC_TCP)