פרויקט גמר בתקשורת

מגישים: אלמוג דוד – 207749441 ורוני פיק – 206794075

- **חלק א:**

- **כיצד להפעיל את המערכת:**

תחילה יש להפעיל את ה- SERVER ע"י פתיחת והפעלת Server.py דרך הטרמינל או דרך IDE נבחר כרצונכם.

לאחר הפעלת הסרבר, נפעיל את GUI.py גם כן דרך הטרמינל או דרך IDE נבחר כרצונכם. בעת הפעלת ה-GUI תתבקשו להזין את השם איתו תתחברו לשרת הצ'אט.

בחלון הצ'אט יופיעו לכם מספר כפתורים שונים:

Disconnect- אם ברצונכם להתנתק מהמערכת, לחצו על הכפתור והוא ינתק אתכם מהחיבור לשרת. לאחר שתתנתקו, לא תוכלו לבצע שום פעולות במסך ותצטרכו לצאת ממסך הצ'אט.

Get users – בעת לחיצת הכפתור תקבלו את שמות כל המשתמשים אשר מחוברים לשרת/ מחוברים לחדר הצ'אט.

Get files – בעת לחיצת הכפתור תקבלו את שמות כל הקבצים שקיימים בשרת אותם תוכלו להוריד.

Download – בצמוד לכפתור ה-download מופיעה תיבת טקסט קטנה. הזינו בתיבת הטקסט את שם הקובץ שברצונכם להוריד. במידה והקובץ אינו קיים בשרת, תתקבל שגיאה.

@ - אם ברצונכם לשלוח הודעה פרטית לאדם מסוים בחדר הצ'אט רשמו כך: @ אחריו את שם המשתמש אליו תרצו לשלוח הודעה ':' ואז את ההודעה שברצונכם לשלוח.

לדוגמה: @alon: hey alon! Only you can get this message

צילום התעבורה (pcap) מצורף בקבצי המטלה (תקשורת ה-TCP וה-UDP).

- **חלק ב:**

הלקוח מבקש להוריד קובץ מהשרת ע"י שליחת ההודעה:download: (file name)

השרת בודק האם הקובץ קיים לו ברשימת הקבצים שנשמרו בשרת.

אם הקובץ לא קיים לנו – נשלח הודעה דרך סוקט הTCP שהקובץ לא קיים.

אם הקובץ כן קיים – נבדוק אם הורידו את הקובץ בעבר (שכן הוא שמור לנו במילון הקבצים שהורדו)

אם הקובץ לא קיים במילון הקבצים שהורדו – ניצור חלוקת פקטות ומספרי זיהוי עבור המידע של הקובץ.

לאחר התהליך, נשלח ללקוח דרך TCP הודעה שאומרת שהקובץ מוכן להורדה

במידה וקיבלנו מהשרת את ההודעה: ready for download, נתחיל בפעולת 3 hand handshake דרך UDP באמצעות שליחת הודעת connected

ברגע שקיבלנו את הודעת connected של הלקוח, נחזיר לו הודעת connected (לחיצת יד שנייה)

ברגע שקיבלנו את הודעת connected של השרת, נחזיר לו הודעת start על מנת להגיד לו להתחיל את ההורדה (לחיצת יד שלישית)

ברגע שקיבלנו את הודעת start של הלקוח, נתחיל בשליחת הפקטות שמכילות את המידע של הקובץ.

נפרק את ההודעה שקיבלנו מהשרת למספר זיהוי ומידע. אם לא קיבלנו כבר את המידע הזה, נשמור אותו במילון (מפתח – מספר זיהוי, ערך – המידע של הפקטה).

לאחר מכן, נשלח לשרת את מספר הזיהוי שקיבלנו ממנו על מנת שלא ישלח שוב את הפקטה הזו (הודעת ack שמאשרת קבלת מידע)

אם לא קיבלנו הודעה מהלקוח בפרק זמן מסוים (שנייה), איבדנו פקטה ונשלח לו את המידע מחדש בשביל שנקבל ממנו הכרה על ההודעה.

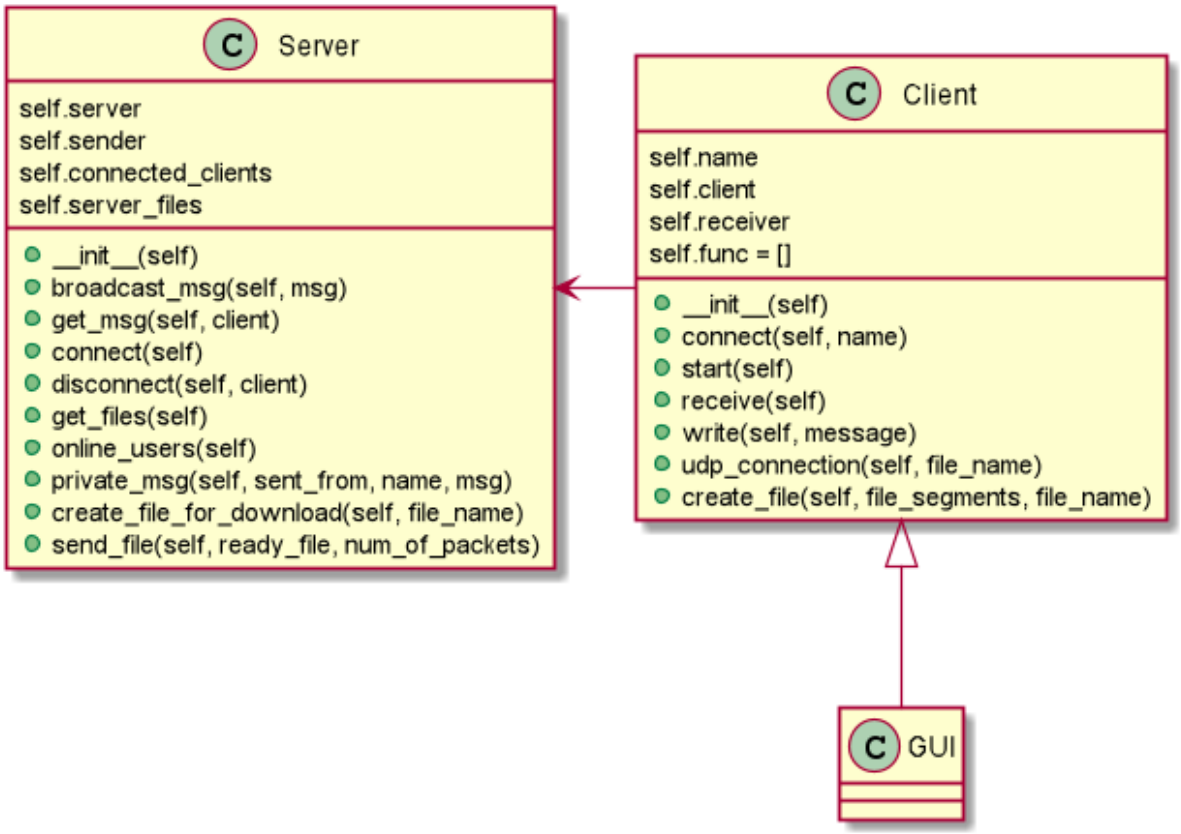
לאחר שקיבלנו הודעה מהלקוח, נמחק את הפקטה מהמילון בשביל שלא נצטרך לשלוח אותה שוב, זאת מאחר וקיבלנו מהלקוח אישור לכך שהוא אכן קיבל את הפקטה.

ברגע שנקבל את הודעת done ע"י השרת שאומרת שהוא שלח לנו את כל המידע, נשלח לו הודעת got all data על מנת לסגור את הקשר.

לאחר ששלחנו את ההודעה, נתחיל ליצור את הקובץ (create file) על בסיס המידע שקיבלנו ושמרנו במילון (מספר זיהוי ומידע)

ברגע שקיבלנו את הודעת got all data מהלקוח, נסגור את קשר הUDP יחד עם הלקוח.

דיאגרמת UML של המחלקות



- **חלק ג:**

<u>שאלה 1:</u>

תחילה, כאשר מחשב מנסה להתחבר לאינטרנט נוצרת בקשת DHCP - שרת DHCP הוא למעשה פרוטוקול תקשורת אשר משמש להקצאה של כתובות IP ייחודיות למחשבים ברשת המקומית. שרת זה מספק את כתובות ה-IP ובנוסף מספק נתונים נוספים כמו כתובת שרת ה-DNS ו-subnet, על מנת שהמחשב יוכל להתחיל לתפקד ברשת ללא צורך בחיפוש אחר נתונים נוספים. שרת זה עובד מעל פרוטוקול UDP ומועבר כשידור ברשת LAN. כך שבתום שלב זה מחשב הלקוח מקבל כתובת IP וכתובת MAC : FF:FF:FF:FF אשר מציינת שידור, כתובת subnet וכו׳.

לאחר מכן שרת ה-DHCP יוצר DHCP ACK אשר יכיל את כתובת ה-IP של הלקוח, את ה-subnet mask, את ה-default gateway (שער ברירת מחדל מספק את האפשרות להתקנים ברשת המקומית לצאת ממנה אל עבר רשתות מקומיות אחרות אשר נמצאות באותו תחום פיזי או מרוחקות יותר), כתובת שרת ה-DNS ואת הזמן במהלכו הלקוח יוכל להתשמש באותה כתובת IP אשר הוקצתה לו על ידי שרת DHCP.

לאחר מכן נשתמש בפרוטוקול ARP על מנת שנוכל לקבל את כתובת ה-MAC עבור ה- default gateway שלנו. הודעת ה-ARP נשלחת דרך LAN בעזרת ה-Ethernet אשר מוגדר לו היעד אליו הוא יבצע את ה-broadcast.

כעת כאשר הגדרנו כתובת IP משלנו ויש לנו את כתובת ה-IP וכתובת ה-MAC של ה- default gateway שלנו, נוכל ליצור בקשת DNS בכדי לקבל את כתובת ה-IP של היעד הסופי שלנו.

בקשת DNS נשלחת באמצעות LAN דרך ה-default gateway, הפקטה הנשלחת מכילה את כתובת ה- IP שלנו, כלומר של המקור, ואת כתובת ה-IP של היעד של שרת ה-DNS אשר הונפק לנו במהלך בקשת ה- DHCP, בנוסף מכילה את שם השרת שאליו אנו מנסים להגיע, אם לשרת ה-DNS שלנו יש את התשובה לשאילתת ה-DNS שלנו, הוא מחזיר פקטת DNS דרך UDP עם המידע המבוקש, אחרת הוא יבדוק מול שרתי DNS אחרים עד שנמצא הכתובת המבוקשת שלנו או עד שנבין שכתובת כזו לא קיימת.

השלב האחרון הוא השלב בו יש לנו את כתובת ה-IP של היעד ואנו מוכנים ליצור תקשורת ראשונית עם ה-host בעזרת שימוש בפרוטוקול TCP, פרוטוקול זה מבטיח לנו שקיים חיבור אמין ולכן נשתמש בו, ושימוש בתהליך 3 way handshake עם השרת בו אנו שולחים תחילה בקשת SYN על מנת לראות שהצד השני ״מקשיב לנו״ ומחכים לתגובת SYNACK אשר בה אנו מקבלים אישור מהשרת שאכן יש לנו חיבור TCP, נאשר שקיבלנו את ה- SYNACK ע״י תגובת ACK ונתחיל בשליחת בקשות HTTP לשרת המכילות את הנתונים הדרושים על מנת להציג את הודעות הצ'אט שלנו.

שאלה 2:

CRC - שיטה לגילוי שגיאות בעת העברת נתונים. השיטה מסתמכת על חילוק פולינומים.

כיצד השיטה פועלת: בהינתן פולינום בדרגה N וההודעה שברצוננו לקדד, נוסיף N אפסים מימין להודעה שברצוננו לקדד כדרגת הפולינום לדאטה ונבצע חילוק פולינומים על בסיס מקדמי הפולינום יחד עם הדאטה.

לאחר החלוקה, נחליף את N האפסים בשארית שהתקבלה. כאשר הלקוח / הצד השני מקבל את המידע הוא מחלק את המידע באותו פולינום. אם תוצאת החילוק היא מלאה / ללא שארית – עבר את בדיקת הCRC ז"א שלא התגלו שגיאות בעת העברת הנתונים. (שיטה אשר דומה במטרתה לשיטת CHECKSUM אשר בודקת האם קיימות שגיאות בעת העברת נתונים אך בשיטת CHECKSUM אין מנגנון לתיקון שגיאות).

שאלה 3:

נציין את ההבדלים בין סוגי הפרוטוקולים השונים:

- HTTP 1 – NONE PERSISTANCE HTTP – חיבור שמאפשר לנו ביצוע בקשה אחת בלבד. לאחר קבלת מענה הבקשה, החיבור נסגר ויש צורך לפתוח קשר חדש על מנת לבצע בקשה חדשה.
- HTTP 1.1 – PERSISTANCE HTTP – חיבור עקבי שמאפשר לנו לבקש יותר מבקשה אחת ללא צורך בפתיחת קשר חדש לאחר שקיבלנו מענה עבור הבקשה.
  ל-HTTP 1.1 יש שני סוגים: NOT PIPELINE & PIPELINE.
  PIPELIINE מאפר לנו לבצע מספר בקשות במקביל ללא הצורך להמתין עבור קבלת מענה על הבקשה הקודמת שביצענו.
  NONE PIPELINE - בכל פעם נוכל לבצע רק בקשה/שאילתה אחת. לא נוכל לבצע שאילתה נוספת עד שנקבל מענה על השאילתה שביצענו תחילה.
- HTTP 2 - בשונה מ-HTTP 1.1 שבו כאשר אנחנו שולחים בקשה למספר קבצים, השרת שולח לנו קובץ קובץ מה שגורם לכך שקבצים קטנים יכולים להתעכב הרבה זמן בעקבות קובץ גדול שנשלח לפניהם. ב- HTTP 2 השרת מעביר מכל בקשה כמות מסוימת של פקטות כך שמכל בקשה נקבל כמות חתיכות מסוימת במקביל. במצב זה קבצים קטנים יגיעו מהר יותר בלי להתעכב בגלל בקשות קודמות (גדולות) שהיו לפניהם. בנוסף, HTTP 2 "מגדיל ראש" ושולח לנו את כל המידע אותו אנחנו צפויים לבקש. לדוגמה במידה ונבקש לקבל דף HTML אשר מכיל בתוכו תמונות, השרת ישלח לנו את התמונות בנוסף לדף ה-HTML וכך יחסוך לנו את זמן גילוי התמונות בדף ואת פתיחת הבקשות להורדה עבור התמונות הללו.

- QUIC – בשונה מהפרוטוקולים שציינו לעיל אשר פועלים באמצעות פרוטוקול TCP, QUIC מעביר מידע על ידי פרוטוקול UDP מה שהופך אותו למהיר יותר. מאחר ולפרוטוקולי UDP אין אמינות במה גבוהה, ב-QUIC יש HEADER נוסף אשר משמש להשגת האמינות בעת מעבר הפקטות. נוסף על כך, ב-QUIC נפתחים מספר סטרימים (stream) של UDP שמאפשרים העברת מידע במקביל ויוצרים זמן המתנה מינימלי לקבלת המידע. עוד יתרון של QUIC הוא שניהול איבוד הפקטות מתבצע פר פקטה ובכך מאפשר שליחת מידע, גם אם אבדה פקטה במהלך העברה, מבלי לתקוע את העברת המידע (seq יותר גדול).

שאלה 4:

מספר הפורט הוא חלק ממידע הכתובת שמאפשר לנו לזהות באופן ייחודי שולחים/מקבלי מידע ולאיזה תהליך מיועדת ההודעה המגיעה אליו. (דרך האינטרנט או הרשת). קיימים מספרי פורטים רבים, אשר חלקם שמורים במערכת עבור פרוטוקולים ספציפיים, אך זו רק מוסכמה וניתן להעביר דרך פורטים אלה פרוטוקולים שונים.

שאלה 5:

subnet – רשת משנה או "רשת בתוך רשת" אשר מכילה אוסף כתובות בעלי התחלה של X ביטים משותפת (X בין 0 ל32) אשר מייצגים את מספר הביטים של כתובת הIP שאינם ניתנים לשינוי. כלומר, נגיד וניתן לנו subnet מסוים לדוגמא 140.2.3.1/24 כלומר רק 8 הביטים האחרונים ניתנים לשינוי, ולכן יש 2^8 כתובות ב-subnet הניתן. וכולם ייוצגו כך 140.23.x כך ש x מספר בין 0-255.

שימוש ברשתות משנה הופכת את הרשתות ליעילות יותר מאחר וכאשר אנו משתמשים ברשת משנה, תעבורת הרשת יכולה לעבור מרחק קצר יותר מבלי לעבור דרך נתבים מיותרים בשביל להגיע ליעד שלה. מאחר ויכולים להיות לנו מיליוני מכשירים שמחוברים לנו לראוטר בעלי כתובת IP דומה יכול לקחת לנו הרבה מאוד זמן עד שנמצא את המכשיר המתאים. כאשר נשתמש ברשת משנה נוכל לצמצם את טווח החיפוש של כתובות ה-IP על מנת למצוא את המכשיר אותו אנו מחפשים. הצמצום יאפשר העברת הודעות בצורה מהירה יותר מאחר ולא נצטרך לעשות מעבר על כל הכתובות אלא רק על כתובות אשר נמצאות בטווח המסוים.

שאלה 6:

כתובת MAC היא מזהה חומרה בעלת 48 בייטים המזהה באופן ייחודי כל מכשיר ברשת אשר נמצאת על כרטיס הרשת של החשב/המכשיר.

הסיבה לכך שאנו משתמשים בכתובת MAC היא שכתובות ה-MAC הן כתובות קבועות שלא משתנות לעומת כתובות IP שהן כתובות דינמיות אשר משתנות מעת לעת או על סמך הגדרות הרשת. בנוסף, למכשירים יכול להיות יותר מכתובת IP אחת בכרטיס הרשת. השימוש בכתובת

MAC עוזר בגילוי אלו מכשירים מחוברים לאיזה פורטים בתוך הרשת ועוזר בניתוב תוך רשתי, מאחר וחיפוש כתובת קבועה (כתובת MAC) עדיפה מאשר חיפוש כתובת זמנית (כתובת IP) או כתובת שלא בהכרח שייכת למכשיר אותו נחפש.

שאלה 7:

Nat – היא טכניקת ניתוב ברשת מחשבים ,בה נכתבות מחדש כתובות ה IP-של הפקטות שעוברות בנתב או בחומת אש ,כלומר, כתובת ה IP-שתצא משימוש תחזור אל מאגר הכתובות ותינתן למחשב אחר בשעת הצורך.

הניתוב יתבצע באופן הבא:

כאשר מתקבלת פקטה מהרשת הפנימית אשר מיועדת לרשת האינטרנט, כתובת המקור (IP SOURCE) של הפקטה מתורגמת כאילו שהפקטה נשלחה מהנתב עצמו לאינטרנט.

מספר הפורט המקורי ממנו נשלחה הפקטה מתורגם לפורט מקור גבוה הניתן ע"י הנתב. פורט זה משמש כאינדקס בטבלת ה-NAT כדי שהנתב ידע להתאים את התשובה ל-IP הפנימי אליה היא שייכת.

כאשר מתקבלת התשובה עבור הפקטה שנשלחה עבור אחד ממחשבי הרשת הפנימית, פורט היעד משמש כאינדקס לטבלת ה-NAT בשביל למצוא את ה-IP הפנימי אליו התשובה צריכה להיות מנותבת.

פורט היעד מתורגם לפורט הפנימי ממנו יצאה הפקטה. כתובת היעד מתורגמת לכתובת ה-IP הפנימי של המחשב אליו התשובה צריכה להגיע והמידע מועבר/ התשובה מועברת אל החיבור של הרשת הפנימית בצירוף מספר הפורט שאיתו יצאה הפקטה מהרשת הפנימית.

או בקצרה: הניתוב מתבצע ע"י שינוי המידע ב-IP HEADER של הפקטה כך שהתשובה לפקטה הנשלחת תוחזר לנתב והוא ישנה את כתובת ה-IP DESTINATION לכתובת של מי שקרא לו (שולח הפקטה) ובכך התשובה תנותב חזרה על סמך המסלול שביצעה.

Switch – תפקידו של ה-switch הוא העברת הודעות בין מכשירים בתוך הרשת הפנימית. ה-switch עובד בשכבת הניק ובשכבה הפיזית ומעביר הודעות ע"י בקשות arp. אם נצטרך להעביר הודעה לכתובת שלא נמצאת הטבלת הכתובות שלו, הוא ישאל את כלל החיבורים שלו אצל מי נמצא הכתובת של X תעבירו את ההודעה ל-Y (יבצע הודעת broadcast לכל המחוברים שלו). כאשר תתקבל תשובה, ה-switch יוסיף את הכתובת לטבלת הכתובות שלו.

Router – הוא התקן רשת המעביר מנות נתונים(שאילתות/תשובות/פקטות) בין רשתות מחשבים. נתבים מבצעים את פונקציות הכוונת התנועה באינטרנט (עובדים בשכבת הרשת). נתונים הנשלחים דרך האינטרנט, כגון דף אינטרנט או דואר אלקטרוני, הם בצורה של מנות נתונים. מנה מועברת

בדרך כלל מנתב אחד לנתב אחר דרך הרשתות המהוות רשת אינטרנט (למשל האינטרנט) עד שהיא מגיעה לצומת היעד שלה.

שאלה 8:

נציין את הפתרונות למצוקת כתובות ה-IP :

- **חלוקה לכתובות פנימיות וחיצוניות** - יש הרבה רשתות בעלות מכשירים רבים שנמצאות באותה רשת (לפעמים רשת סגורה). מאחר וכולם פועלים באותה רשת סגורה, ניתן לחזור על אותן כתובות בתוך הרשתות האלה, וכך נוכל לחסוך בכתובות חיצוניות. דוגמה: משרד גדול בעל מכשירים רבים. אם לכל מכשיר היה כתובת IP חיצונית משלו, היו נתפסים כתובות IP רבים ואין בכך צורך מאחר וכולם עובדים באותה רשת.
- **nat gateway** - כיום בכל משרד/בית ישנ כמות גדולה של מכשירים אשר ניגשים לאינטרנט, וכולן צריכות כתובת. ולכן נוצר הפתרון nat gateway אשר אומר שכל המחשבים תחת אותה רשת יצרו לאינטרנט דרך כתובת אחת , ולכן היום שבכל בית יש מספר רב של מכשירים חכמים כולם יוצאים עם אותה כתובת, ובמשרדים בעלי רשתות רבות כולן יוצאות לאינטרנט עם כמות כתובות קטנה ביחס לכמות המכשירים הקיימים.
- **ipv6 -** פרוטוקול ipv6 הוא הוספת ביטים לכתובת, כלומר כתובת בעלת 128 ביטים וכך לא נגיע בחיים לכמות הכתובות האלו, לפתרון זה יש יתרון מובהק מאחר ומדובר בכמות כתובות גדולה במיוחד. החיסרון לפתרון זה הוא שלא כל המערכות (נכון לעתה) תומכות בפרוטוקול זה ולכן לא ידעו להתמודד עם המידע של ה-Ipv6.

שאלה 9:

1. הנתב C3 לומד על תת הרשת X באמצעות פרוטוקול BGP, מאחר והוא הראוטר היוצא אל AS4.
2. הנתב A3 לומד על תת הרשת X באמצעות פרוטוקול OSPF מאחר והוא מקבל את המידע בקשר לתת הרשת X דרך C3. בתוך הרשת של AS3 המידע מועבר באמצעות פרוטוקול OSPF.
3. הנתב C1 לומד על תת הרשת X באמצעות פרוטוקול EBGP, מאחר והוא הראוטר היוצא אך AS3. A3 לומד על X באמצעות OSPF (סעיף 2) וC1 לומד על תת הרשת X באמצעות שימוש בפרוטוקול EBGP.
4. הנתב C2 לומד על תת הרשת X באמצעות פרוטוקול OSPF, מאחר והוא מקבל את המידע מA2 אחרי שהמידע הועבר בתוך AS2 באמצעות פרוטוקול OSPF.