

תקשורת מחשב

פרויקט גמר

הוראות להגשת המטלה:

1. **הגשה ביחידים או בזוגות.**
2. שם קובץ ההגשה (מקווץ) חייב להכיל את ת"ז של הסטודנטים.
3. שמו לב, חובה שהקוד יירוץ בכל מחשב ולכון כל החלטה מקומית (**path**) חיבת להיות כללית.
4. כתיבת קוד נכונה כולל שמות משתנים, אובייקטים ופונקציות קטנות שצורך, אין מספרי קסם.
5. חלק נכבד מהציוון (לפחות 40%) ניתן על כתיבת קוד נכונה, מבנה, בדיקות וכו'
6. יש לכווץ את כל הקבצים באחד מהפורמטים הבאים: ZIP, RAR, 7-ZIP, בילב'.
 - a. יש להגיש מסמך pdf שסביר מה עשייתם.
 - b. יש להגיש קובץ הקלטה של התעבורה שהלקיים ממנו יפורטו במסמך ה-pdf
 - c. חובה להשתמש ב-unit testing.
7. עליכם להתייחס למסמך זה כאפיון המערכת. חובה עליכם להתייחס למקרי קצה ולטיפול באגדים גם אם לא מופיע במסמך באופן מדויק מה המקירה. אי טיפול באגדים ובמקרים קצה יגרור הורדה של ניקוד.
8. אופן/צורת/אלגוריתם שתבחרו (לדוגמה RDT, congestion control) יהווה חלק מהציוון. אל תנסו לשאול אותנו אם משחו מספיק אלא על פי מרכיבות הפתרון ניתן הציוון.
9. אסור להעתיק. זאת אומרת:
 - מותר לדבר אחד עם השני בנוגע למטלה, להתייעץ איך כדאי למש, ולשתף בבעיות שצצו, כל זמן שהוא נעשה בע"פ. אסור לעזר ע"י העברת של קוד קטע קוד כלשהו מאחד לשני, אפילו לא פונקציה אחת. במידה ומחלגת העתקה 2 הצדדים יקבלו 0 במטלה וייכשלו בקורס, ללא תלות במי העתיק ממי.
 - מותר להיעזר באינטרנט, אבל אסור להעתיק קוד שמצאתם כמות שהוא - כתבו את הקוד בעצמכם. בפרט, יש אסור להעתיק קוד מ-github .
 - מי שעבוד עם github חייב להגדיר repository private .
 - מותר להיעזר בחונך או במורה פרטי, אבל אסור שהם יכתבו לכם את הקוד או חילקו

תיאור התרגיל: שפת תיבנות היא **Python**

חלק א' – על בסיס תיאור המערכת:
עליכם לבנות מערכת מסרים מיידיים פרימיטיבית (בדומה ל- messenger) מבוססת על תקשורת .

חלק ב' – על בסיס תיאור המערכת:
להוסיף אותה מערכת שכבה חדשה (קוד נוסף) להעברת קבצים מעל UDP אשר נקרא לה FAST reliable UDP
ענו על השאלות הבאות:

- ציירו דיאגרמת מצבים בהם המערכת עובדת
- כיצד המערכת מתגברת על עיבוד חבילות latency
- כיצד המערכת מתגברת על בעיות

חלק ג': בתוספת לשני חלקים הקוד. ענו על השאלות הבאות:

1. בהינתן מחשב חדש המתחבר לרשota אנה תארו את כל ההודעות שעוברות הח' מהחיבור הראשוני ל switch ועד שההודעה מתבלטת בצד השני של היצא. אנה פרטו לפי הפורמט הבא:
a. סוג הודעה, פירוט הודעה והשדות הבאים

.i. כתובת IP מקור/יעד, כתובת פורט מקור/יעד, כתובת MAC מקור/יעד,
פרוטוקול שכבת התעבורה.

2. הסבירו מה זה CRC
3. מה ההבדל בין http 1.0 ,http 1.1, http 2.0, QUIC
4. למה צריך מספרי port ?
5. מה זה subnet ולמה צריך את זה ?
6. למה צריך כוחות mac ומה לא מספיק לעובד עם כוחות ip ?
7. מה ההבדל בין Router Switch Nat ?
8. שיטות להתגבר על המחשב בIPv4 ולפרט ?
9. נזונה הרשות הבאה.

OSPF מרכיבים AS2, AS3 .a

RIP מרכיבים AS1, AS4 .b

Between Routers .c

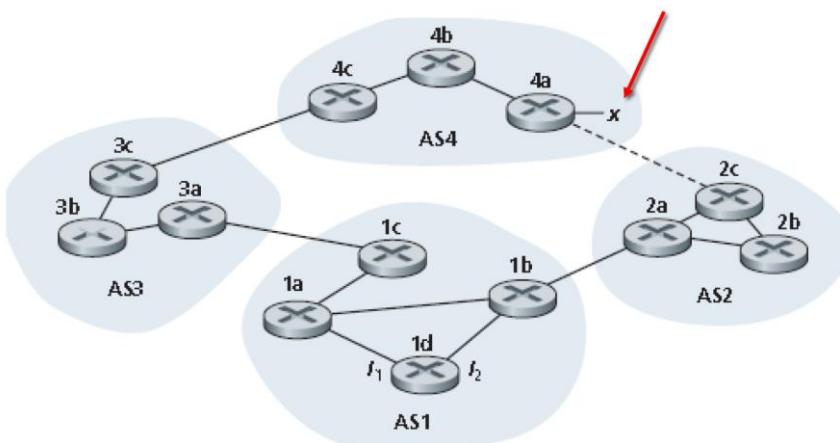
AS4, AS2 .d

Without physical connection between them .e

Without physical connection between them .f

Without physical connection between them .g

Without physical connection between them .h



תיאור המערכת:

המערכת תהיה בנוייה משרת ולקוחות כאשר:

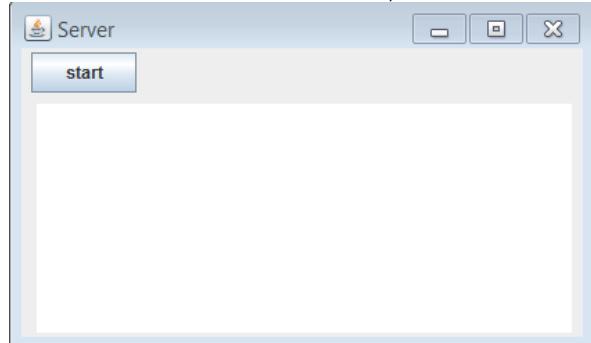
השרת (server) מתחילה את עצמו ו"מקשיב". השרת רץ על גבי שרת IP מסוים ומקשיב ללקוחות בפורט מסוים הידוע ללקוחות לדוגמה 55000, כאשר הוא מאפשר להתחבר במספר לקוחות בו זמנית (לפחות 5). בשליל הסדר הטוב נסכם שבתרגילים זה נעשה הצד של השרת שימוש בפורטים מסוימים 55000-55015. לאחר שלקווה מתנתך הפורט שהתפנה חזר להיות משאב פנו.

כל לקוח יכול לשולח הודעה לכל מחוברים ב-chat והודעה פרטית למחובר ספציפי תוך ניטור העברת באמצעות קשור TCP. כל לקוח יכול ליזום בקשה להורדת קובץ מהשרת באמצעות קשור TCP במקביל לערוץ העברת קובץ ע"י קשור UDP שלו לב שהעברת הקובץ צריכה להיות אמינה ולקחת בחשבון עומס ברשות ולכך את צרכיים לממשה FAST reliable UDP (במילים אחרות מימוש RDT מעל היבור UDP בתוספת control). בעת סיום, יש להוציאו הודעה מתאימה, בצווף ערך הבית האחרון שנשלח. שימושו לב שניתו להוריד בו זמנית הקובץ כמו להקות שוונים. כמובן, יתכן שבעת המתנה לקבלת אישור להמשך הורדת מלוקה אחד, הלוקה الآخر מבקש ומתחיל לקבל את הקובץ במקביל.

שימוש לב, מטרת התרגיל היא הצד התקשורתי ולכך אין צורך לפתח את ה-GUI. קבוצות שיפתחו GUI משליהם או על בסיס הדוגמאות בתרגיל זה יקבלו בונוס.

שימוש לב מס' 2, כל הפיתוח צריך להיות שלכם ולא להתבסס על קוד פתוח מהרשת.

דוגמא לעיצוב המסך של השרת:

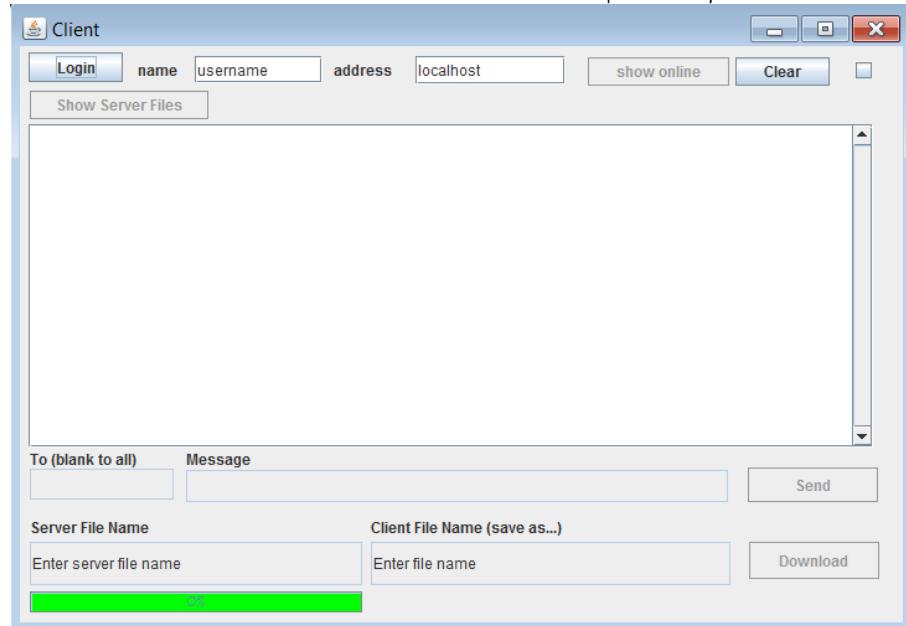


לקוח (client) יכול לבצע את הפעולות הבאות:

- (1) להתחבר לשרת,
- (2) להתנתך מהשרת,
- (3) לשולח הודעה ללקוח אחר,
- (4) לשולח הודעה לכל הלוקחות המוחברים לשרת כתע,
- (5) לקבל את שמות הלוקחות המוחברים לשרת,
- (6) לקבל רשימת קבציםקיימים בשרת,
- (7) לשולח בקשה להורדת קובץ מהשרת,
- (8) להוריד קובץ מתוך הרשת.

כמו כן, במידה והצטרכ לךו חדש או תנתן לךו קיים יש לשולח הודעה מתאימה לכל המשתתפים.

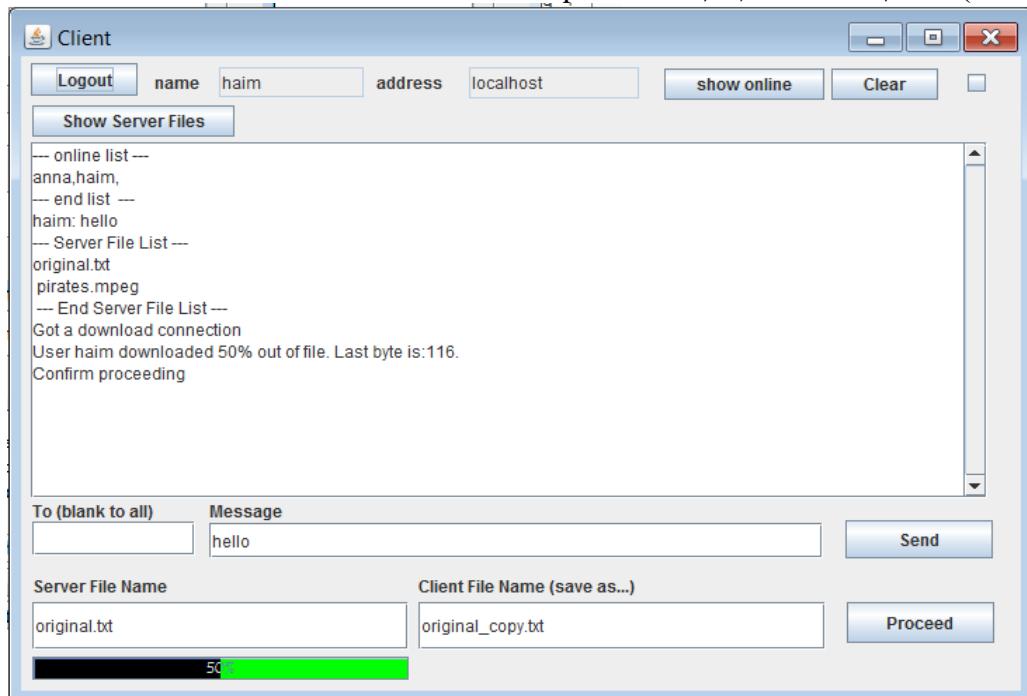
דוגמא לעיצוב המסך של הלקוח:



מומלץ להגדיר פרוטוקול (שפה משותפת) עבור הודעות בין השירות לקוחות. לדוגמה:

רשימת הודעות שנשלחות מהלקוח (לשרת):

- (1) אני רוצה להתחבר: <connect><name>
- (2) אני רוצה לקבל את רשימת המחוברים: <get_users>
- (3) אני רוצה להתנתק: <disconnect>
- (4) אני רוצה לשולח הודעה ללקוח: <set_msg><Name>
- (5) אני רוצה לשולח הודעה לכלום: <set_msg_all>
- (6) אני רוצה לקבל את רשימת הקבצים שיש בשרת: <get_list_file>
- (7) אני רוצה להוריד קובץ <test.txt": <download> < test.txt > <id" <proceed>
- (8) המשך בהעברת הקובץ: <proceed>



רשימת ההודעות שמתקבלות מהשרת (לקוח):

- (1) התחברת <connected>
- (2) התנתקת <disconnected>
- (3) ההודעות עבורה <msg_lst><num_of_msgs><"...">....<"..."><end>
- (4) רשימת המחוברים <users_lst><num_of_users><"...">...<"..."><end>
- (5) רשימת הקבצים <file_lst><"...">...<"..."><end>
- (6) קיבלה 100% מהקובץ:

User <name> downloaded 100% out of file. Last byte is: yyy.

הערות כלליות:

- ניתן להניח שכמויות ההודעות של כל לקוח לא תעלה על 100 הודעות.
- **שם לב, גודל מסימלי לחיבור (datagram) להעברה באמצעות UDP הינו .kB64. אך יש להגביל גודל הקובץ להורדה.**
- קודם יש לנסות על מחשב אחד גם את הל庫 ו גם את שרת. כתובות מקומית של כל מחשב היא localhost או 127.0.0.1
- תשובה בסגנון: "לא כתבתם במללה שצורך לטפל בזה" לא תתקבל.
- אתם אמורים ליצור איבוד חבילות + השהיות על מנת לבדוק את המערכת שלכם

בצלחה!