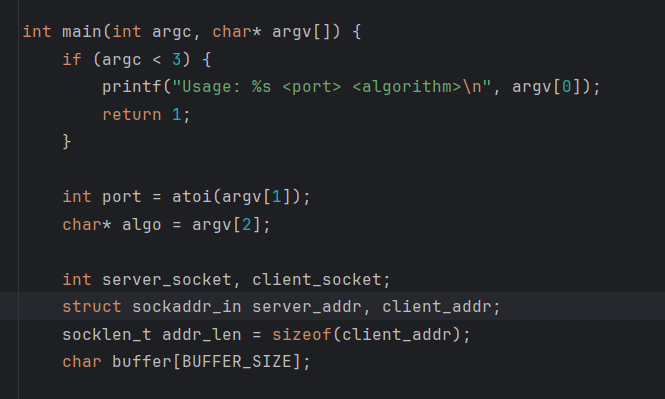
מטלה 3 רשתות תקשורת – 214797367,326065646

חלק ראשון תיעוד :

בתוכנית זאת בנינו 2 קבצים , SENDER\_TCP, RECIVER\_TCP

אחד ישלח לשני קובץ בגודל 2MB והשני יהיה פתוח לקבל אותו עד שהמשתמש לא ירצה להמשיך לשלוח

RECIVER\_TCP



האתחול של הכול. לוקחים את הפרמטרים שהוכנסו מהמשתמש ושומרים, מייצרים את המשתנים שצריך

תמונה שמכילה טקסט, גופן, צילום מסך

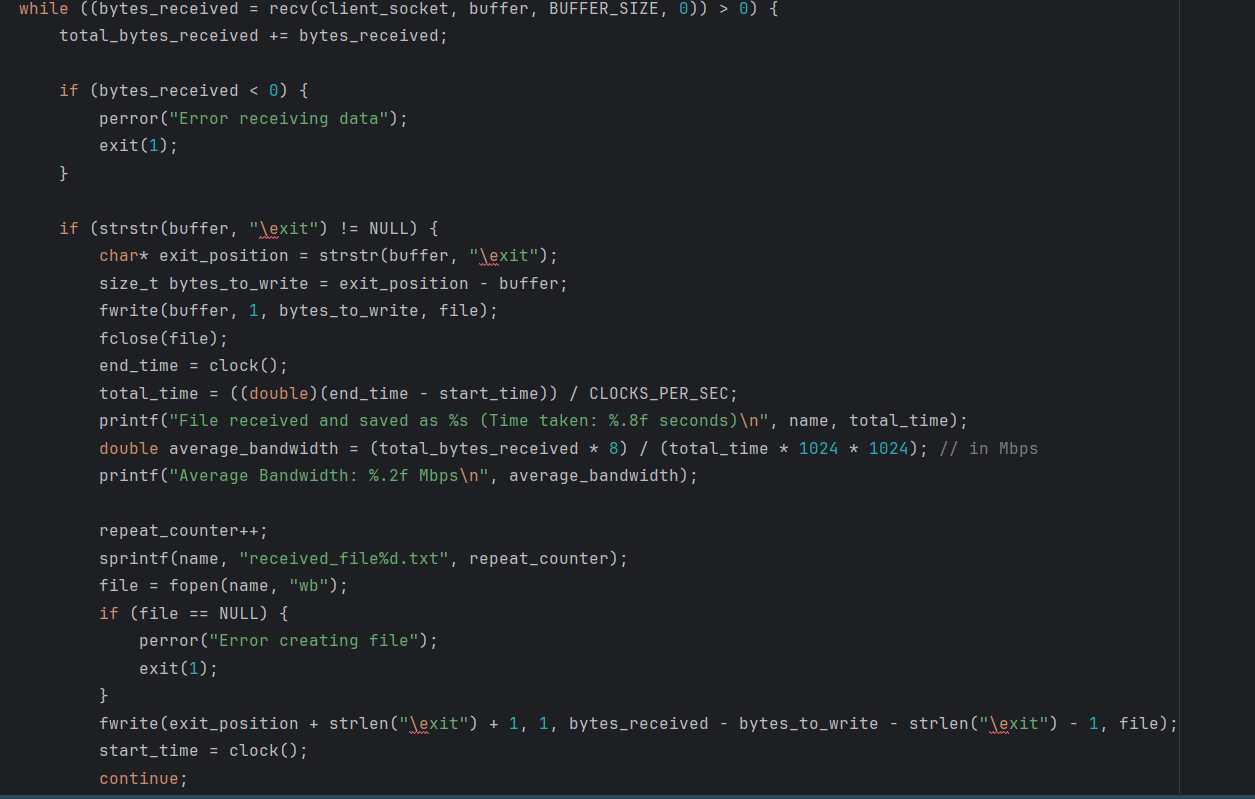
התיאור נוצר באופן אוטומטי

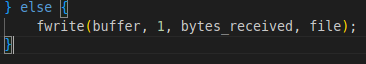
בודקים לפי איזה אלגוריתם אנחנו נשלח את הפקטות

תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטי

מחברים את reciver , server שמים במצב האזנה ומחכים לאישור התחברות





מקבל את הפאקטות, עד אשר מקבל פאקטה של “\exit, ואז הוא מסיים את את הכתיבה וסוגר את הקובץ. מחשב את הסטטים ופותח קובץ חדש. לאחר מכן מתחיל ספירת זמן מחדש (שתבוא לידי ביטוי אם נשלח עוד קובץ) וממשיך.

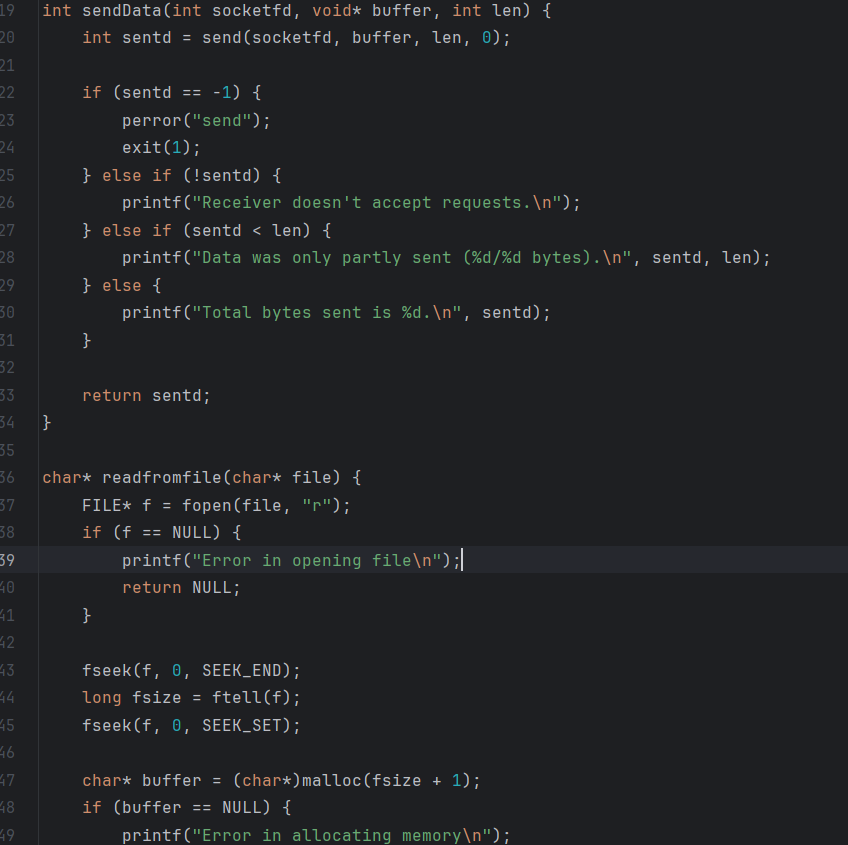
אם הוא לא מקבל exit הוא פשוט ממשיך לכתוב לקובץ שכבר פתוח

תמונה שמכילה טקסט, גופן, צילום מסך, עיצוב

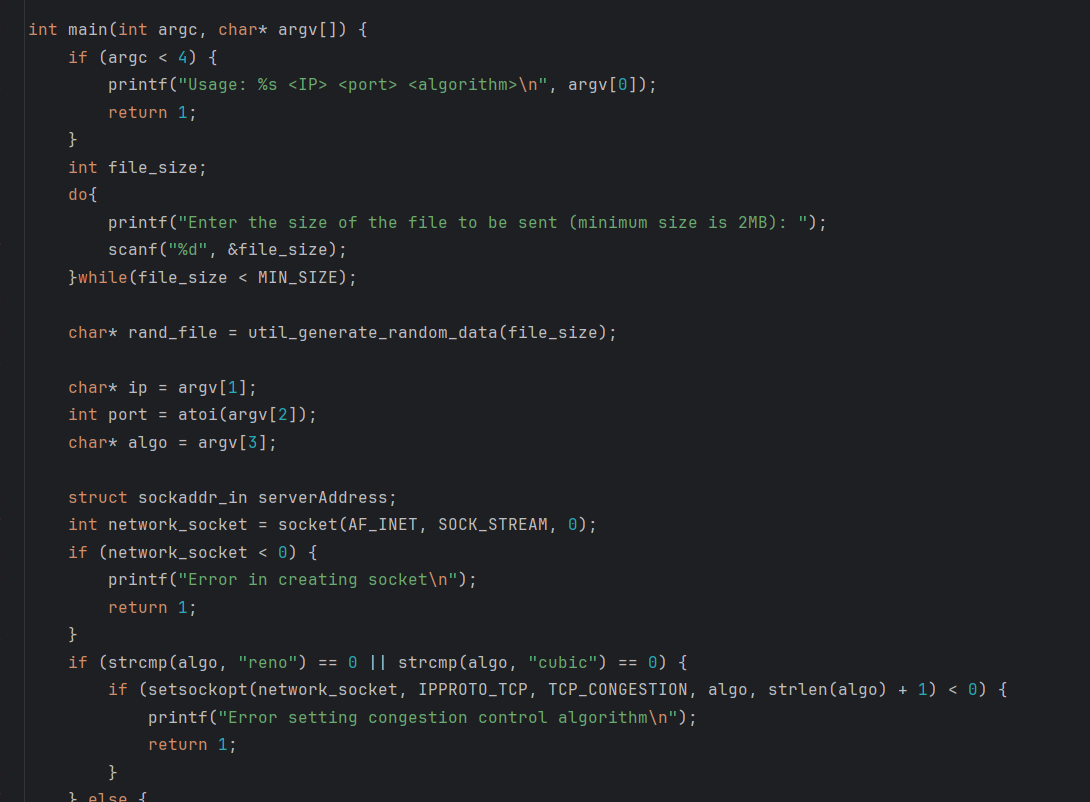
התיאור נוצר באופן אוטומטי

סוגר את הסוקטים ואת הקובץ

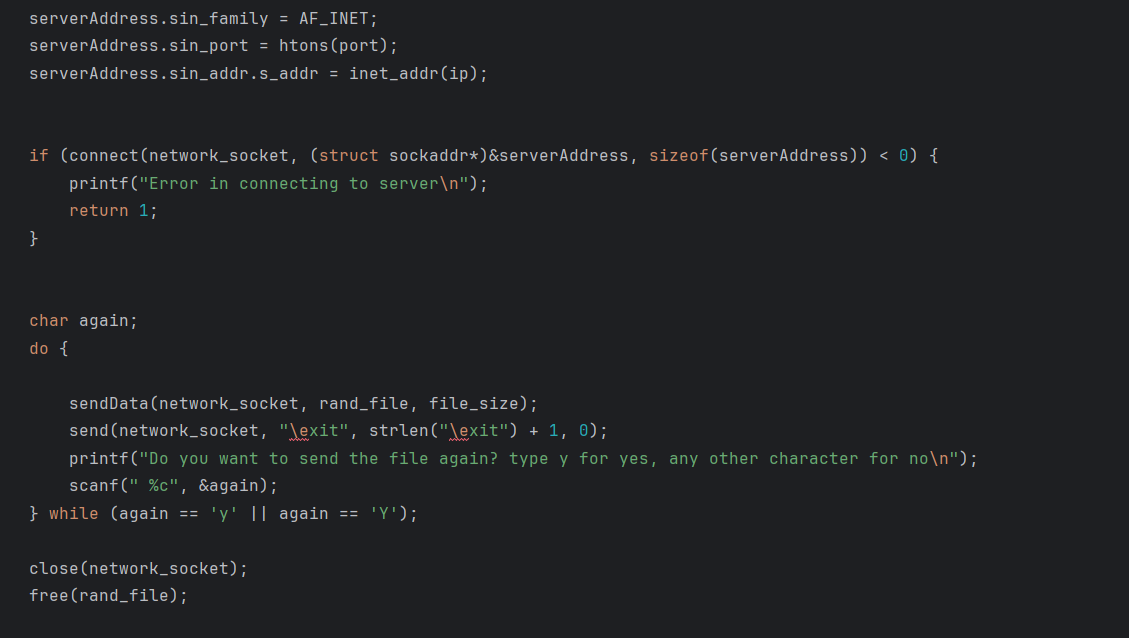
SENDER\_TCP



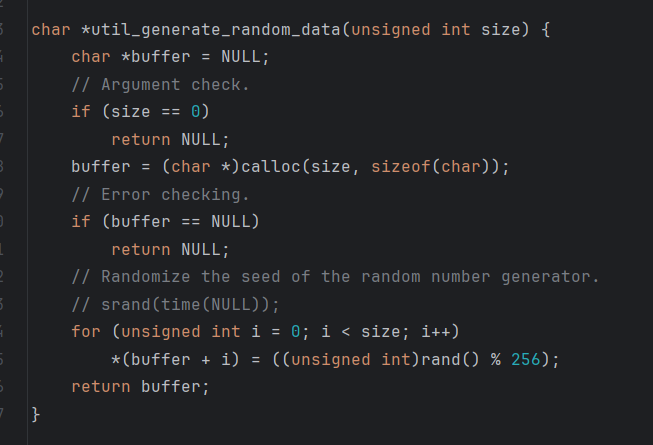
הפונקציות שקוראות מהקובץ הרנדומלי שאנחנו יוצרים (נראה בהמשך ) ושושלחת אותו



בדומה מאוד לreciver רק שפה גם מאתחלים את הקובץ הרנדומלי ובודקים איזה מהאלגוריתמים משתמשים

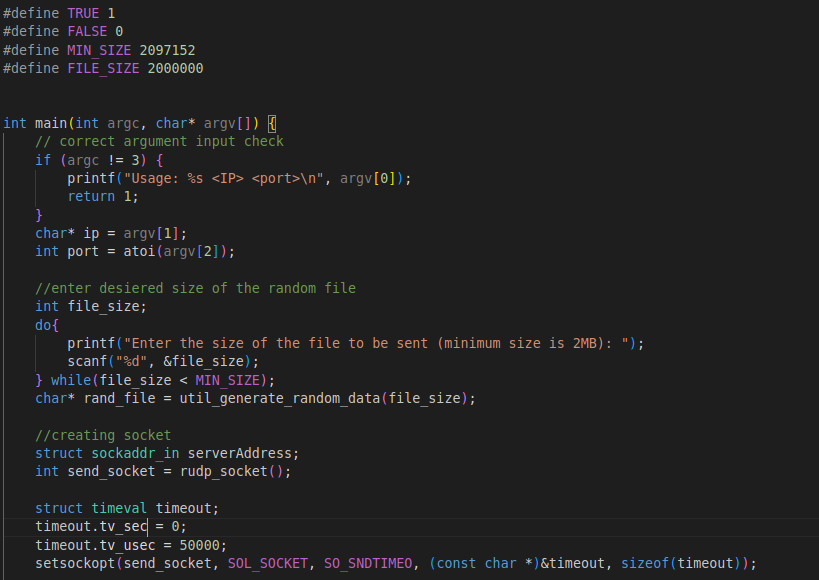


תהליך שליחת המידע ל reciver שמחכה לקלוט. בסוף שיחת הקובץ שולח exit ושואל האם לשלוח שוב. אם לא אז סוגר את הסוקט והקובץ

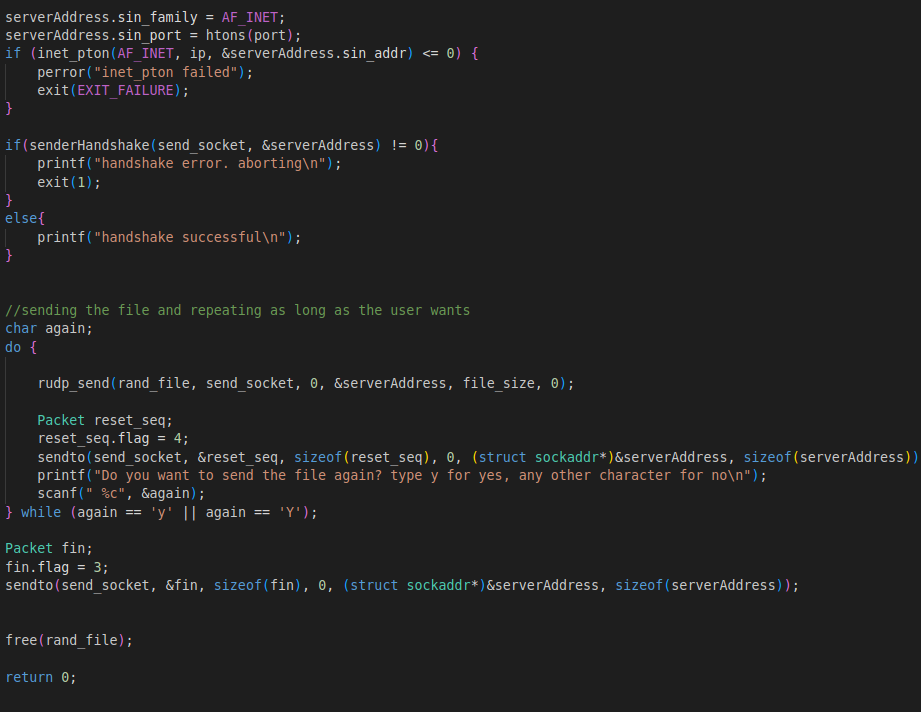
יוצר קובץ רנדומלי 

חלק 2 תיעוד :

RUDP SENDER



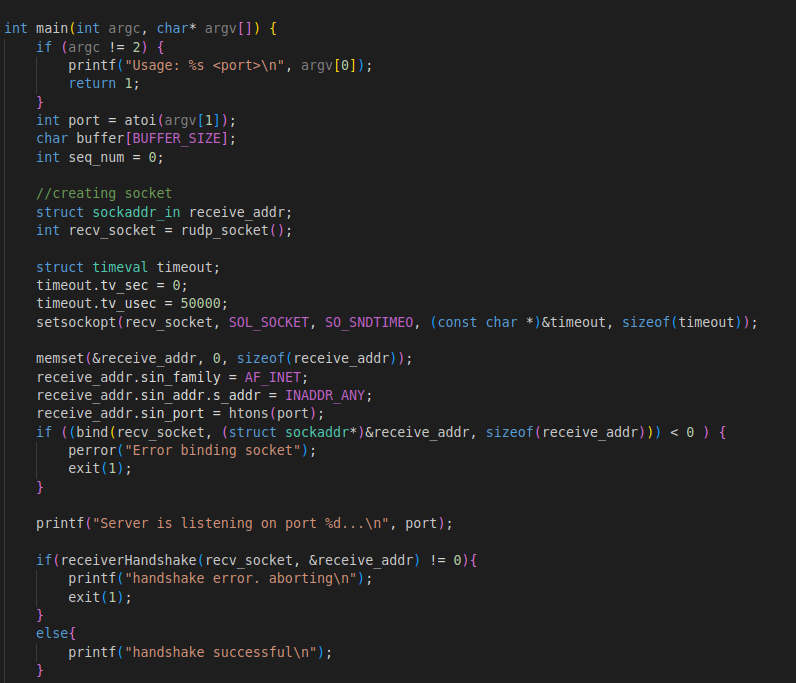
מקבל את הדברים מהקריאה לפונקציה, קולט את גודל הקובץ הרצוי ויוצר קובץ רנדומלי. פותח סוקט וקובע את הגדרות הזמנים



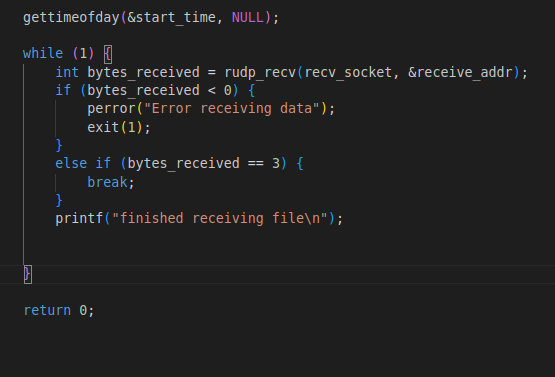
עושה את הגדרות הסוקט וקורא ללחיצת ידיים. אחר כך שולח קובץ ושואל אם לשלוח שוב בדומה לחלק 1. כאשר אחרי שליחת הקובץ שולחים פקטה שאומרת לאפס את seqnum (בשביל מקרה של קובץ חדש).

במידה ולא יוצרים קובץ חגש שולחים פקטת FIN, סוגרים את הקובץ ומסיימים את ההתקשרות.

RUDP RECEIVER



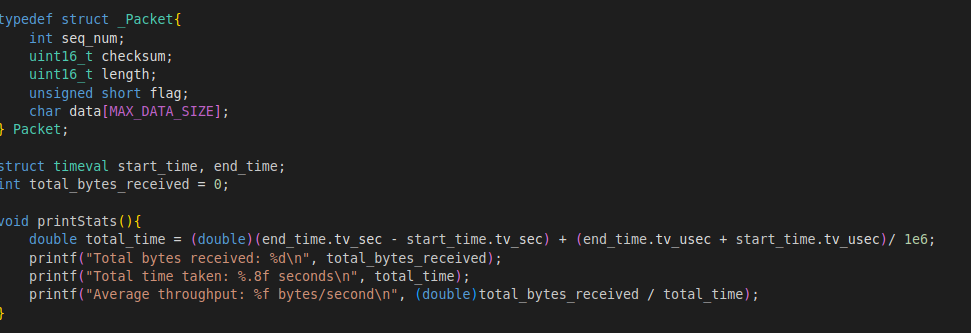
שוב מקבל פרמטרים מקריאה לפונקציה וקובע את הגדרות הזמנים והסוקט. לאחר מכן מבצע לחיצת יד.



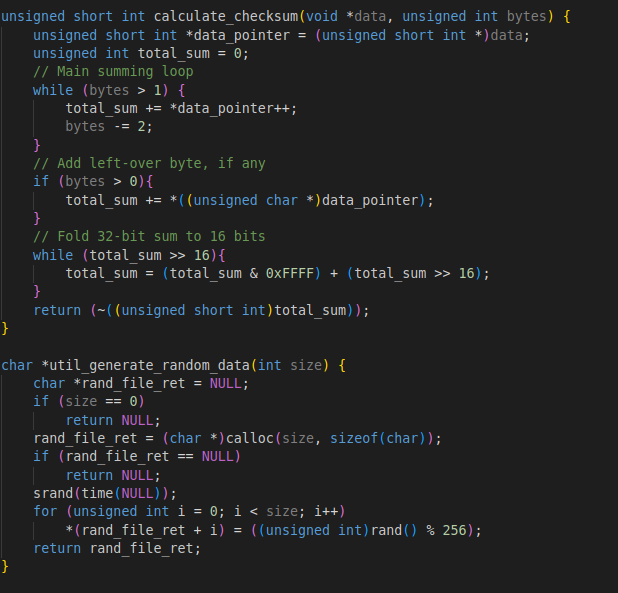
מקבל את הפאקטות אחד אחד, עד ש-RUDP\_RECV מחזיר 3 (קוד לסיום הקליטה)

נוסף על כך הוא מגדיר את זמן ההתחלה לחישוב הסטטים בהמשך.

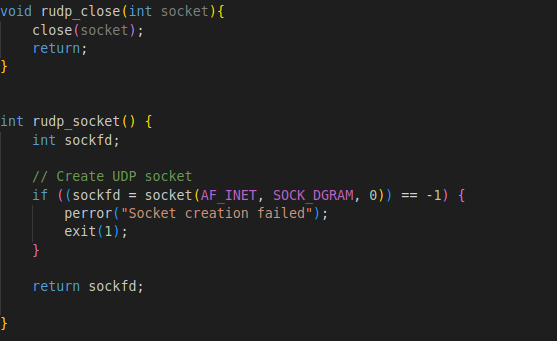
RUDP API



הגדרת הסטראקט של הפקטה של RUDP וחישוב סטטים לפי הגדרות זמן שהגדרנו במהלך ריצת התוכנית (קוראים לחישוב הסטטים בסוף)



פונקציות עזר כמו שהובאו לנו בהוראות המשימה



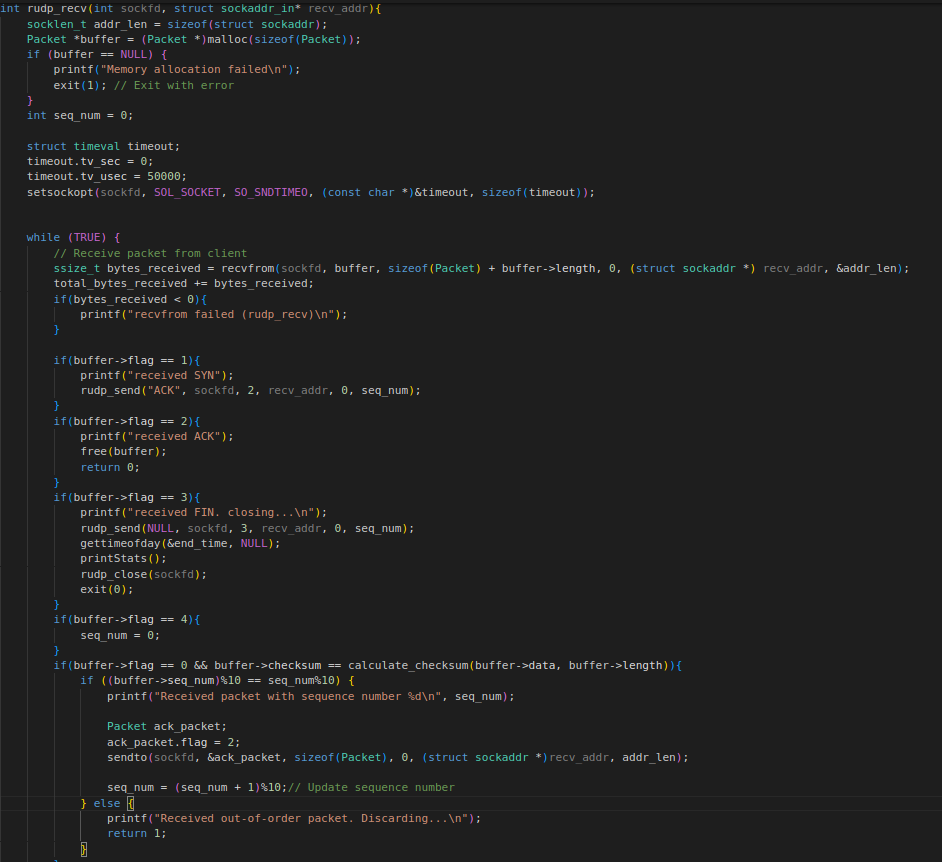
פעולות פתיחת וסגירת הסוקט

A screen shot of a computer program

Description automatically generated

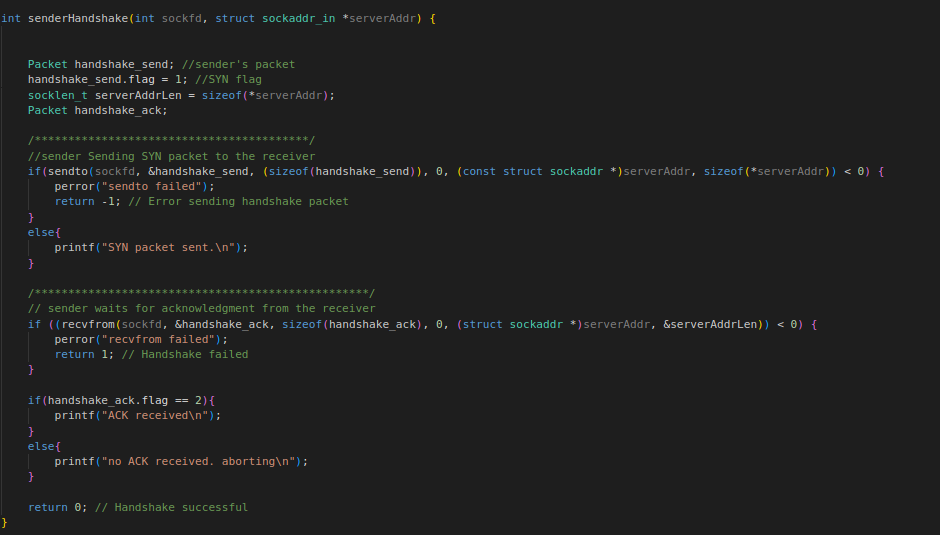
פעולת השליחה. מגדירים את הזמנים ואת מספר הפקטות שנצטרך ביחס לאורך הקובץ. לאחר מכן שלח פקטות כמספר הפעמים שצריך בלולאה כשבכל איטרציה שולחים פאקטה עם הדגל המתאים, כשכל פאקטה מקבלת את הדאטה המתאימה לה. נוסף על כך מחשבים צ'קסאם.

לכל פקטה ששולחים מחכים ל-ACK, ואם אינו מגיע לאחר מספר מסויים של ניסיונות אנו ממשיכים הלאה.



שוב פעם מגדירים זמנים ומקבלים את הפאקטות. בודקים את הדגלים שהוגדרו לפאקטה ופועלים בהתאם.

אם הדגל הוא 0 זה אומר שהפקטה היא פקטת מידע רגילה ומקבלים אותו (בתנאי שצ'קסאם ו-seqnum מתאימים), ושולחים עבורה ACK, מעדכנים את seqnum בהתאם (הערכים שלו הם בין 0-9 למען הנוחות) וממשיכים הלאה.



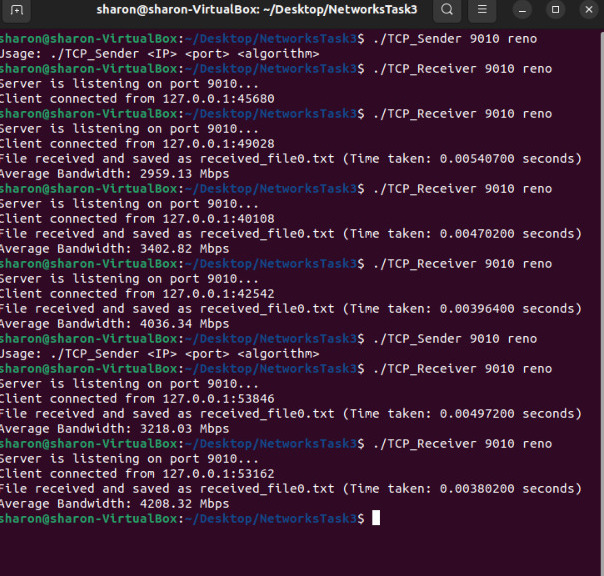
A screen shot of a computer program

Description automatically generated

חלק 3 הקלטות :

הקלטות של חלק א :

רנו לרנו



קוביק לקוביק :

תמונה שמכילה טקסט, חשמל, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

קוביק לרנו

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

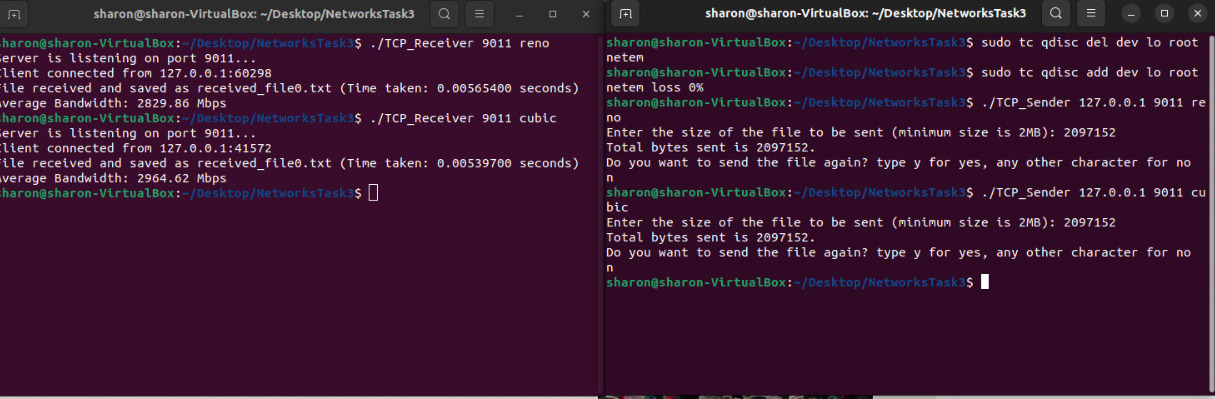
רנו לקוביק

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

מעכשיו צילומי מסך של רנו לרנו וקוביק לקוביק עם פאקט לוס שונה :

רנו וקוביק עם 0 פאקט לוס



רנו וקוביק עם 2 אחוז פאקט לוס

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

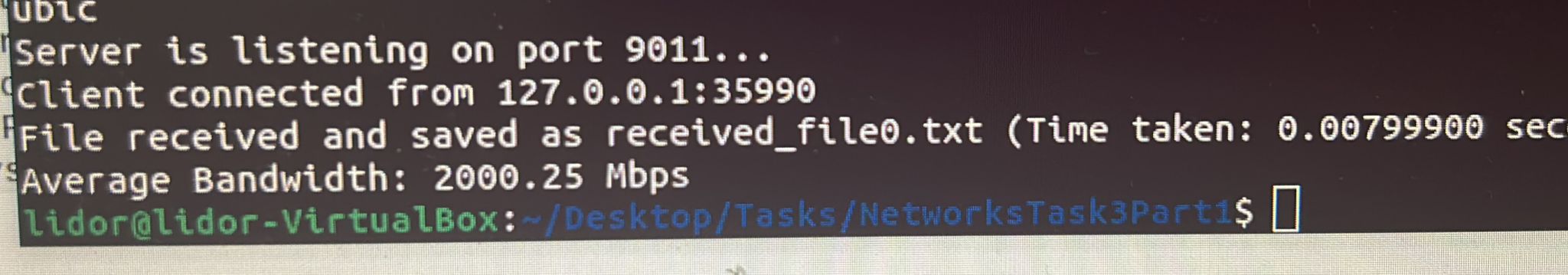
התיאור נוצר באופן אוטומטי

רנו וקוביק עם 5 אחוז פאקט לוס

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

התיאור נוצר באופן אוטומטי

רנו עם 10 אחוז פאקט לוס



קוביק עם 10 אחוז פאקט לוס

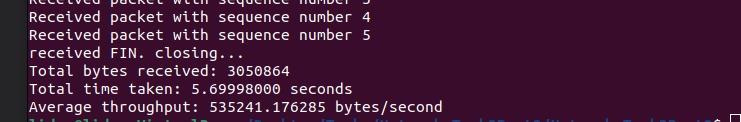
תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

עד לפה כל ההקלטות של חלק א .

ההקלטות של חלק ב :

האלגוריתם שלנו לא הצליח להתמודד עם פאקט לוס פה אז רק הקלטות של 0 אחוז פאקט לוס

 תמונה שמכילה טקסט, צילום מסך, גופן

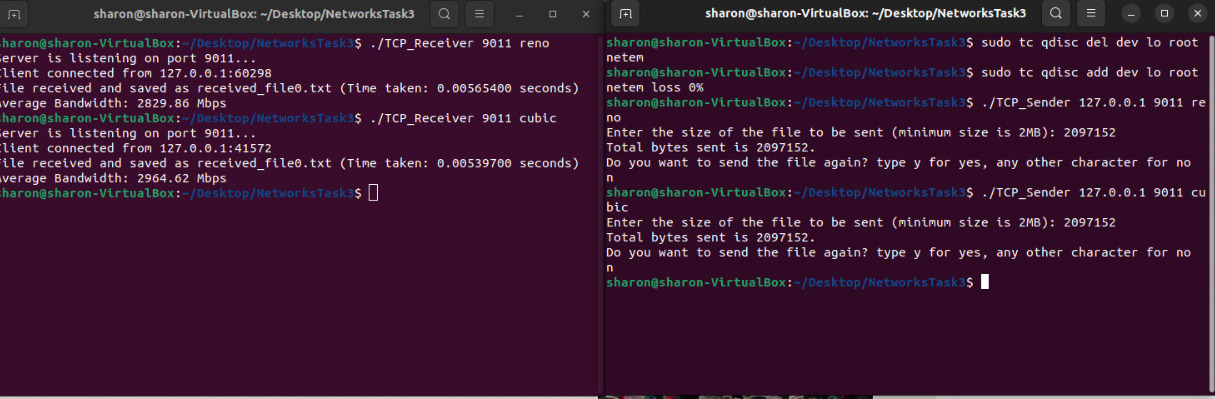
התיאור נוצר באופן אוטומטי תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי תמונה שמכילה טקסט, צילום מסך, גופן

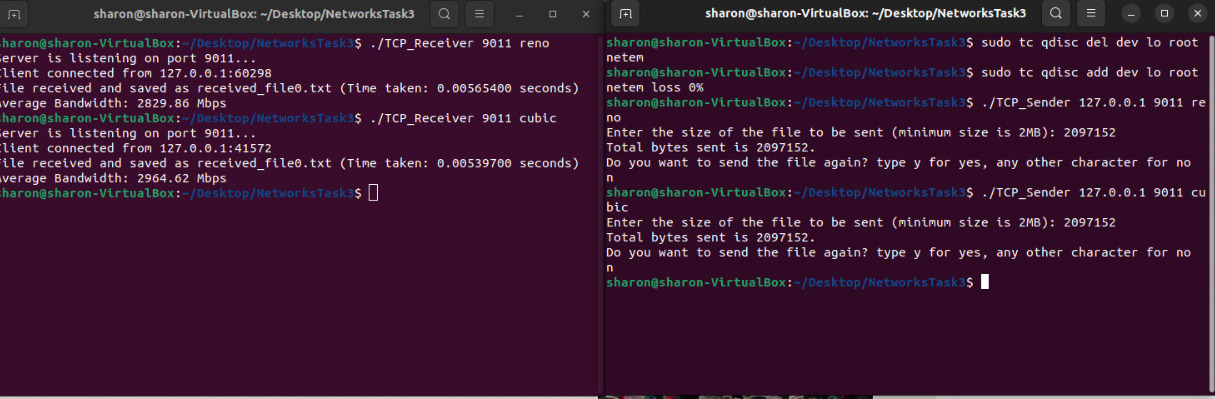
התיאור נוצר באופן אוטומטי

שאלות חלק 3 :

1.ב0 אחוז פאקט לוס ניתן לראות שבקשר TCP קיוביק היו תוצאות יותר טובות ביחס לשל רנו,

ניתן לראות שקיוביק יותר מהיר( לקח לו פחות זמן ) ועם avarege bandwith יותר גבוה

בפאקט לוס שהוא יחסית גבוה ניקח לדוגמא את 5 רנו עם average bandwidth גבוה יותר ולקח לו פחות זמן



2.לא עבד לנו RUDP עם פאקט לוס גבוה אז קראתי באינטרנט מה אמור לקרות עם השיטה שלנו (stoppen weight) עם פאקט לוס

לפי התוצאת בפאקט לוס 0 ניתן לראות שRUDP יותר מהיר ועובד טוב מTCP

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

ניתן לראות שלקח פחות זמן (RUDP זה במילי שניות) ושה ממוצע bandwith של RUDP יותר גבוה משמע יותר מהיר

לכן בפאקט לוס 0 אעדיף לקחת RUDP

לעומת זאת לפי מה שקראתי באינטרנט כלפי השיטה שלנו RUDP יתקשה עם פאקט לוס גבוה ולכן אנחנו נעדיף במקרה הזה להשתמש בקשר TCP

3.לפי התשובה שלי לשאלה 2 בתרחיש שבו יש לי רשת אמינה כלומר יש לי אחוז מאוד נומך של פאקטלוס אני אעדיף להשתמש בקשר RUDP לעומת זאת ברשת שהיא לא ככ אמינה ויכול להיות איבוד פאקטות יחסית גדול ולכן אעדיף להשתמש בקשר TCP

תשובות לשאלות אמריקאיות :

1.ברגע שאנחנו מגדילים את SSTHRESH אנחנו בעצם מגדילים את כמות הפקטות שאנחנו נשלח לפני שנחכה לACK .

במצב כזה יש יותר אי וודאות על האם הפקטות הגיעו בבטחה וזה יכול לגרום לעומס רב על השרת , לכן נרצה להגדיל אותו מתי שהרשת תהיה אמינה כלומר יהיה פחות איבוד פקטות וכשהRTT יהיה גדול כלומר כל החזרה של ACK תיקח יחסית המון זמן ואנחנו נרצה שיהיה כמה שפחות החזרות של ACK (שנוכל לשלוח המון פקטות מבלי לחכות לזה ) . בנוסף לכך בקשר ארוך הגדלת הSSTHRESH תהיה האידיאלית ביחס לקשר קצר מכיוון שהגדלת גודל החלון (כמות הפקטות שאנחנו שולחים בבת אחת) תהיה יעילה יותר כשיש המון מידע לשלוח , (גם יכול לקרות מצב שבו הSSTHRESH שלנו גדול מכמות הפקטות שאנחנו צריכים לשלוח ולכן זה לא אותה יעילות כמו בקשר ארוך).

לכן התשובה שהכי מתאימה למה שהסברתי זו תשובה הראשונה כלומר קשר ארוך ברשת אמינה עם RTT גבוה .

2.הקשר בין A לB יהיה ככה:

מתחילים בתהליך slow start כלומר 1MSS ואז נחכה לAck ולאחר מכן נכפיל את גודל החלון פי 2 עד שנגיע לSSTHRESH וזה במקרה שלנו S\*MSS .

בנוסף rwnd>s\*mss ולכן B מסוגל לקבל יותר פקטות ממה שA יכול לשלוח לו.

לכן B יקבל את כל הפקטות בבטחה והחלון שדרכו נשלח את הפקטות יגדל כל הזמן פי 2 . לכן הכמות הפעמים שיקרה ההגדלה פי 2 הזאת היא לוג 2 של s

התפוקה תהיה התפוקה המירבית שזה בעצם S2\*MSS לחלק בRTT כפול הכמות הכפלות שנצטרך להכפיל עד שנגיע לmss\*S2

ולכן התפוקה בביטים לזמן בערך תהיה א. בערך 

3.האיקס שלנו הוא 4

4=X

הRTT שלנו יהיה מורכב מזמן השידור וזמן העיכוב ביחד

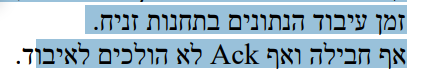
וקצב התקשורת שלנו הוא Gbps8

לכן : זמן העיכוב הוא :המרחק בין התחנות שהוא 1000 מטר וקצב ההתפשטות הוא 2 כפול 10 בחזקת 8 ולכן זמן העיכוב יהיה

(1000 חלקי( 2כפול 10 בחזקת 8)) שזה יוצא 1 חלקי 20000

זמן השידור יהיה 4 חלקי 10 בחזקת 6

ולכן הrtt יהיה : 0.000019(חיבור שלהם והכפלה ב2(הלוך חזור ))

בגלל הנתונים של השאלה : 

אנחנו נגדיל את החלון להיות בגודל המקסימלי והנוסחא אליו תהיה

קצב התקשורת כפול RTT חלקי גודל הפקטה

לכן

Window=18/5=3.6