

TCP – CONGESTION CONTROL ALGORITHMS

NETWORK PROGRAMMING IN C

הנחיות להגשת המטלה:

את מטלה זו יש להגיש בזוגות כקובץ ZIP עם מספרי ת"ז של הסטודנטים/ות. לתיבת ההגשה במודל. הגשות באיחור יתאפשרו עד 4 ימים כאשר לכל יום איחור ירדו 5 נקודות.

- (1) את המטלה יש להגיש עד התאריך המצוין בתיבת ההגשה
- (2) כל הקבצי המטלה (קוד, פלט תעבורה, הסבר) כולל הסברים שלכם והקלטות Wireshark דחוסים לקובץ ZIP ששמו הוא מס' ת.ז. של המגישים עם קו תחתון ביניהם ID_ID. בין הקבצים שאתם מגישים שימו לב, יש להגיש קובץ pdf המכיל צילומי מסך (בכל מקום שאתם עושים משהו תצלמו מסך ותסבירו איך הגעתם למסקנה). אין להגיש צילומי מסך של מרוכזים בקובץ אחד עם הסברים, קובץ ללא הסברים או צילומי מסך בלבד לא יבדקו
- (3) הגשה בזוגות זהו הסטנדרט וזו חובה. לאישורים חריגים שלחו מייל למרצה שלכם ותכתבו את אחראי המתרגלים, אלמוג שור.
- (4) מותר לכם להשתמש בכל החומר שנמצא במודל כולל קוד בתרגולים. חומרים אחרים אין אפשרות. כמובן שאפשר להיעזר באינטרנט להבנה של תהליכים וקוד אבל בשום פנים ואופן לא להעתיק קוד.
- (5) אין איחורים ללא אישור מיוחד של רכז הקורס (אלמוג, במייל בלבד), איחור ללא אישור יגרור אפס אוטומטי.
- (6) הגשת העבודות תתבצע דרך מערכת ה Moodle של הקורס (לא דרך האימייל).
- (7) יש להקפיד על כללי עיצוב הקוד שנלמדו בתואר (נא להקפיד על פלט ברור, הערות קוד במידה ושמות משתנים בעלי משמעות). קוד רץ בלבד יכול לקבל לכל היותר ציון 60, שאר 40 הנקודות זה הסברים שלכם, ידע, קוד קריא וכו'.
- (8) ניתן להגיש תרגילים למערכת מספר בלתי מוגבל של פעמים כאשר כל הגשה דורסת את הקודמת.
- (9) העבודה הינה אישית של הזוג ואסור לקבל עזרה מאנשים מחוץ לאוניברסיטה או בתוכה לה. אנשים המתקשים ורוצים עזרה יכולים לפנות לצוות הקורס בשעות הקבלה או להעלות שאלה לאתר הקורס.
- (10) אסור להעביר קטעי קוד בין סטודנטים, להעלות פתרונות או חלקי פתרונות לאתרים ברשת האינטרנט, פורומים או בקבוצות תקשורת שונות.
- (11) סטודנטים שיעתיקו פתרון, יקבלו 0 בכל המטלות בקורס ונעלה דיווח לוועדת המשמעת המוסדית.
- (12) קבצי הקוד המטלה נדרשים להתקמפל ולרוץ על מערכת הפעלה Ubuntu 22.04LTS

TCP CC

In this Ex you'll write two program files: Sender.c and Receiver.c. The Sender will send a file and the Receiver will receive it and measure the time it took for his program to receive the file.

The file will be sent in two parts (first half and second half [Each half should be 50% of the file]) each half will be sent according to one of the CC algorithms you learned in the lectures.

You are obligated to send the file a few times (at least 5 times in each run) for creating data set (What is a data set and what it's the purpose when you developing a program or an Algorithm? Great question! We believe in you. You can find the answer by yourself).

You are obligated to run the program with a packet lost tool in the following levels:

0% lost, 10% lost, 15% lost, 20% lost.

Important notes:

- ❖ The main part of this Ex is to get familiar with TCP concept of sending msg between two End Points (For instance - Server-Clients Architecture).
- ❖ In this Ex you will have the sender and the receiver of the messages. The message in this Ex will be a file, with at least 1MB size of a file. Create or use any text file you want.
- ❖ For approaching this assignment, please be sure that you are familiar with the CC algorithms and how to change them using the correct method in C. (reminder – the method is getsockopt)
- ❖ The receiver doesn't really care about saving the file itself (or its content). He just cares about the Data-Frame that he gets.
- ❖ In this Ex you'll learn how to use packet lost tools on Linux which is known as TC. To see how to download and work see the attachment below.
- ❖ **Do not forget to use Wireshark and attach the pcap files for full submission!**

The program steps:

Sender.c

1. At first, you'll read the file you've created.
2. Create a TCP Connection between the sender and receiver.
3. Send the first part of the file.
4. Check for authentication (explaining below).
5. Change the CC Algorithm.
6. Send the second part.
7. User decision:
 - a. Send the file again? (For data gathering)
 - i. If yes:
 - ii. Change back the CC Algorithm.
 - iii. Go back to step 3.
 - b. Exit?
 - i. If yes:
 - ii. Send an exit message to the receiver.
 - iii. Close the TCP connection.

Receiver.c:

1. Create a TCP Connection between the sender and receiver. As known, it doesn't create a whole new socket with TCP connection. Its just should establish connection with the socket that the sender created.
2. Get a connection from the sender.
3. Receive the first part of the file.
4. Measure the time, it took to receive the first part.
5. Save the time.
6. Send back the authentication to the sender.
 - i. Change the CC Algorithm.
7. Receive the second part of the file.
8. Measure the time it took to receive the second part.
9. Save the time.
10. If you get an exit message from the sender:
 - i. Print out the times.
 - ii. Calculate the average time for each part of the received files.
 - iii. Print the average file.

נספח א' – עבודה עם כלי לאיבוד פאקטות בלינוקס

כדי לדמות איבוד פקטות, אנו נשתמש בכלי של לינוקס שנקרא tc. אם הכלי אינו קיים אצלכם תוכלו להוריד אותו בעזרת הפקודה הבאה: (אם הפקודה לא עובדת תנסו אותו דבר עם iproute2)

```
sudo apt install iproute
```

לאחר מכן נייצר איבוד פאקטות רנדומלי:

```
sudo tc qdisc add dev lo root netem loss 10%
```

אנו נרצה לשנות את אחוז איבוד הפאקטות בכל סיבוב דגימות:

```
sudo tc qdisc change dev lo root netem loss XX%
```

כאשר נרצה למדוד 10,15,20,25,30 אחוזי איבוד.

כשנסיים את התרגיל, נריץ את הפקודה הבאה על מנת לבטל את איבוד הפקטות:

```
sudo tc qdisc del dev lo root netem
```

המטלה להתקמפל ולרוץ על מערכת ההפעלה Ubuntu 22.04LTD - זוהי תזכורת נוספת שעל
(אפשר להריץ את הקוד רק בסביבת לינוקס מלאה ולא דרך ווינדוס או WSL)

נספח ב' – אותנטיקציה בין שני הצדדים.

על מנת שהשולח ידע שהמקבל קיבל את מה שהשולח רצה לשלוח, נהוג לשלוח איזשהו סימן מוסכם מראש שהדבר המבוקש התקבל כראוי (במיוחד כשיש חשש לאיבוד פקטות).

הסימן שאנחנו נשתמש בו הוא פעולת xor בין 4 הספרות האחרונות של כל אחד מהמגישים.

למשל:

ת.ז מגיש 1: 123456789

ת.ז מגיש 2: 987654321

$XOR(0001\ 1010\ 1000\ 0101, 0001\ 0000\ 1110\ 0001) = 0000\ 1010\ 0110\ 0100$