

## README

### פרויקט 1 תקשורת מחשבים

### דניאלה שבת 316415645

### מור ברקמן 208202325

הוראות שימוש: זהות לדרישות התרגיל.

#### תיאור מבנה הקוד:

מבנה הקוד דומה עבור השולח, ערוץ והמקבל. בתחילת הקוד ישנה פתיחת של socket מתאים ופונק bind עבור הערוץ והמקבל. לאחר מכן ישנה מכונת מצבים שכמעט זהה עבור כל אחד מהרכיבים כך שכל אחד מהרכיבים מאותחל למצב שונה בהתחלה. המצבים במכונת המצבים:

EXIT – יציאה מהתוכנית במקרה של תקלה או כאשר התוכנית הסתיימה בהצלחה.

SEND – שליחה של הקובץ אל הreceiver שמוגדר לרכיב

RECEIVE – ממתין לקבלת הודעה מהsender.

DECODE – מצב זה קיים רק בשרת, במצב זה השרת מפענח את ההודעה שהתקבלה.

ADD\_NOISE – מצב זה קיים רק בערוץ, מצב זה מוסיף רעש להודעה שנתקבלה.

לאחר EXIT התוכנית משחררת את כל המשאבים ומסיימת את הריצה.

#### **קידוד hamming :**

- פונקציית קידוד hamming נעשית בתוכנית הלקוח. תחילה מתבצעת קריאה לפונקציה generate\_bits\_string\_from\_file שקוראת את הבתים מהקובץ קלט ומעבירה את הבתים למחרוזת של ביטים, אותה מחרוזת ביטים (אורכה מתחלק ב11) מועברת לפונקציית קידוד hamming.
- פונקציית הקידוד מחלקת את המחרוזת לframes באורך של 11 ביטים ועבור כל frame כזה מקודדת לפי חוקי hamming כך שכל מחרוזת של 11 ביטים אחרי קידוד בעלת 15 ביטים. לאחר הקידוד יש המרה מביטים לבתים – מחרוזת זו היא המחרוזת שבסופו של דבר תישלח על ידי הלקוח- וזאת על מנת לחסוך בכמות הבתים שמועברים בתקשורת.
- במידה והמחרוזת ביטים המקודדת לא מתחלקת ב8, מתבצע ריפוד אפסים בסופה של המחרוזת.

#### **הוספת רעש בערוץ:**

- פונקציית הוספת הרעש create\_noise נעשית בתוכנית הערוץ. תחילה מתבצעת קריאה לפונקציה encoder\_srting שמעבירה את הבתים במחרוזת שמתקבלת בערוץ למחרוזת של ביטים.
- פונקציית הוספת הרעש מקבלת את הארגומנטים של ההסתברות והגרעין, ומגרילה מספר בין 0 ל1 ובודקת האם הוגרל מספר קטן מההסתברות שהתקבלה כארגומנט נעשה flip לביט, הפעולה נעשית עבור כל ביט בנפרד.
- לאחר הוספת הרעש מתבצעת העברה ממחרוזת ביטים למחרוזת בתים על מנת לחסוך במספר הבתים שמועברים.

#### **תרגום hamming:**

- פונקציית תרגום הקוד - `reverse_hamming` מתבצעת בתוכנית השרת. תחילה מתבצעת קריאה לפונקציה `encoder_srtng` שמעבירה את הבתים במחרוזת שמתקבלת בשרת למחרוזת של ביטים.
- פונקציית תרגום הקוד מבצעת תיקון שגיאות לפי `hamming code` עבור כל 15 ביטים בנפרד מחלצת את `data` ל 11 ביטים, לאחר התיקון הנחוץ. במידה ויש למעלה מטעות אחת במחרוזת של 15 ביטים, הפונקציה וכך הקידוד בנוי, לא תזהה את השגיאה ולא תתקן אותה.
- לאחר התרגום והתיקון, אם צלח או לא- נעשית המרה של מחרוזת הביטים למחרוזת בתים, וזו המחרוזת שנכתבת בסופו של דבר לקובץ הפלט.

קרדיטים:

בכתיבת תקשורת הUDP נעזרנו בלינק הבא:

<https://www.geeksforgeeks.org/udp-server-client-implementation-c/>

בפונקציה `convert_bytes_to_bits` נעזרנו בלינק הבא:

<https://stackoverflow.com/questions/1682996/bytes-to-binary-in-c>

באגים ידועים ומגבלות השימוש:

- ניתן לשלוח רק קובץ באורך המתחלק ב-11.
- המפענח יודע לפענח בצורה תקינה רק מקרים בהם יש שגיאה 1 בתוך מסגרת של 15 ביטים.