

# אריזת נתונים ולכידת מנות בעזרת Wireshark

קורס: תקשורת מחשבים  
פרויקט גמר – ניתוח תעבורה בפרוטוקול TCP/IP

## 1. מבוא

בחלק זה של הפרויקט הודגמה אריזת נתונים בשכבות TCP/IP, החל משכבת היישום ועד שכבת הרשת, תוך שימוש בקובץ הודעות, מחברת Jupyter ולכידת תעבורה בפועל באמצעות Wireshark

## 2. יצירת קובץ CSV (שכבת היישום)

### 2.1 תיאור הקובץ

הוכן קובץ CSV המכיל הודעות בשכבת היישום של פרוטוקול צ'אט (CHAT).

שם הקובץ:  
`groupXX_chat_input.csv`

### 2.2 שדות הקובץ

הקובץ כולל את השדות הבאים:

- `msg_id` – מזהה הודעה
- `app_protocol` – שם פרוטוקול היישום
- `src_app` – מקור ההודעה
- `dst_app` – יעד ההודעה
- `message` – תוכן ההודעה
- `timestamp` – חותמת זמן יחסית

## 2.3 אופן יצירת הקובץ

קובץ ההודעות נוצר ידנית על בסיס תרחיש תקשורת של מערכת צ'אט, במטרה לייצג הודעות טיפוסיות בשכבת היישום בצורה מבוקרת ופשוטה.

## 3. עיבוד הקובץ במחברת Jupyter

### 3.1 טעינת הקובץ

בשלב הראשון נטען קובץ ה־CSV למחברת Jupyter באמצעות Python, והנתונים עובדו למבנה טבלאי.

### 3.2 הדמיית אריזת נתונים

המחברת מדמה את תהליך אריזת ההודעה בשכבות:

- שכבת יישום – תוכן ההודעה
- שכבת תעבורה – הוספת כותרת TCP
- שכבת רשת – הוספת כותרת IP

### 3.3 יצירת תעבורה

בהמשך המחברת יוצרת מנות רשת בהתאם להדמיה, כך שניתן ללכוד אותן בזמן אמת באמצעות Wireshark.

## 4. לכידת תעבורה ב־Wireshark

### 4.1 תהליך הלכידה

- הופעל Wireshark על ממשק רשת פעיל
- הוגדר פילטר תעבורה מתאים
- המחברת הורצה בזמן הלכידה
- קובץ הלכידה נשמר בפורמט `pcap`.

## 5. ניתוח המנות

### 5.1 מבנה המנה

ב-Wireshark ניתן לראות:

- שכבת Ethernet / Loopback
- שכבת IP – כתובות מקור ויעד
- שכבת TCP – פורטים, flags
- שכבת Data – תוכן ההודעה

### 5.2 הסבר הממצאים

הניתוח מדגים כיצד הודעה משכבת היישום נארזת ומועברת ברשת, וכיצד ניתן לצפות בכל שכבה בנפרד באמצעות Wireshark.

## 6. סיכום

בחלק זה של הפרויקט הודגמה הבנה מעשית של אריזת נתונים בשכבות TCP/IP, עבודה עם קובץ קלט, עיבוד באמצעות Jupyter ולכידת תעבורה בפועל. השילוב בין סימולציה ללכידה אמיתית מאפשר הבנה עמוקה של תהליך התקשורת ברשת.