

Communication Networks – Encrypted Network Traffic Analysis and Classification

רועי ינקו

גל וינטר

עדי מוסקוביץ

אורי שפיץ

בית הספר למדעי המחשב, אוניברסיטת אריאל

מבוא

מטרת הפרויקט היא לחקור ולהבין כיצד ניתן לנתח ולסווג תעבורת רשת של אפליקציות שונות, תוך התמקדות במקרים בהם התעבורה מוצפנת. במסגרת הפרויקט נענה על שאלות כלליות בנושא, נבצע הקלטות תעבורה באמצעות Wireshark, ננתח את המאפיינים הייחודיים של כל אפליקציה ונבחן עד כמה ניתן לזהות פעילות משתמש על סמך מידע חלקי בלבד. בנוסף, נסקור מאמרים מחקריים עדכניים בתחום סיווג תעבורת רשת מוצפנת וניישם טכניקות שונות להשוואת הממצאים.

חלק 1:

בחלק הזה בפרויקט נענה על השאלות הבאות:

Part I: Answer the following Questions:

1. A user reports that their file transfer is slow, and you need to analyze the transport layer to identify the potential reasons. What factors could contribute to the slow transfer, and how would you troubleshoot it?
2. Analyze the effects of TCP's flow control mechanism on data transmission. How would it impact performance when the sender has significantly higher processing power than the receiver?
3. Analyze the role of routing in a network where multiple paths exist between the source and destination. How does the path choice affect network performance, and what factors should be considered in routing decisions?
4. How does MPTCP improve network performance?
5. You are monitoring network traffic and notice high packet loss between two routers. Analyze the potential causes for packet loss at the Network and Transport Layers and recommend steps to resolve the issue.

1. ראשית, מדובר ככל הנראה ב-TCP, שכן אמינות החיבור חשובה כאן, כי אם חלק מהקובץ לא יגיע, כל הקובץ יהיה לא תקין.

הסיבות למהירות איטית שנוגעות לשכבת התעבורה הן:

א. **עומס על השרת:** אם השרת עמוס מדי, ה-BUFFER של הפקטות יכול להתמלא. כתוצאה מכך, השרת יזרוק פקטות אם אין מקום. בנוסף, השרת יצטרך לחלק את זמן העיבוד בין המשתמשים, מה שעלול לגרום לעיכוב בהעברת כל חלק של הקובץ (אם מדובר בהורדה מהשרת) או בתגובה

שהתקבל כל חלק מהקובץ (אם מדובר בהעלאה לשרת). עומס על השרת עשוי להאט את מהירות ההעברה וגם לגרום לאיבוד פאקטות ושליחתן מחדש.

ב. **חיבור לא יציב:** יכול לגרום לאיבוד פקטות. כאשר פאקטות הולכות לאיבוד, הצד ששולח אותן יצטרך לשלוח אותן מחדש (וגם את הפאקטות שלא הגיעו אחריהן, לפי סוג ה-Flow Control של ה-TCP).

איך נאבחן:

נוכל לבדוק ב-Wireshark האם חבילות נשלחות מחדש. אם אנו רואים שחבילות נשלחות שוב ושוב, זה יכול להצביע על כך שהשרת עמוס למשל, מה שגורם ל-timeout ולשליחת פקטות מחדש.

לאחר הבדיקה ב-Wireshark נריץ פקודת traceroute בין המשתמש לשרת ונראה באיזו תחנה הזמן גבוה. אם הזמן הארוך הוא בתחנות הראשונות, אז הבעיה בצד המשתמש. אם הזמן הארוך באחרונות, הבעיה בשרת, ואם הזמן הארוך נמצא בתחנות הביניים, הבעיה היא בתשתית הרשת בין המשתמש לשרת.

אם גילינו שנאבדות פאקטות ונשלחות מחדש, וגם שלשרת לוקח יותר זמן לענות ב-traceroute אז כנראה שיש עומס על השרת שגורם לאיבוד פאקטות, ואם נראה שב-traceroute הזמנים הארוכים הם בתחנות הביניים אז נסיק שהאיבוד נגרם כתוצאה מחיבור לא יציב, ואם בתחנות הראשונות אז נסיק שהאיבוד נגרם בגלל בעיה בחיבור של המשתמש.

אם נגלה שאין איבוד פאקטות: אם ב-traceroute לוקח הרבה זמן בשרת זה כנראה מעיד על עומס שגורם למשך זמן ממושך יותר עד שהשרת ישיב, אבל זמן לא מספיק ארוך בשביל לשלוח שוב פאקטות (או שב UDP ואז אין שליחה של פאקטות מחדש), אם לוקח הרבה זמן בתחנות הביניים אז אותה סיבה בגלל חיבור לא יציב, ואם בתחנות הראשונות אז אותו הדבר בגלל חיבור בעייתי אצל המשתמש.

2. **Flow control** - מנגנון בקרת הזרימה של TCP, מטרתו למנוע מהשולח להציף את המקבל על ידי שליחת נתונים מהר יותר ממה שהמקבל יכול לעבד.

דרך אחת לפתור את הבעיה תהיה באמצעות פרוטוקול **receive-window**, שבו המקבל מודיע לשולח כמה זיכרון פנוי יש לו כרגע בבאפר על ידי שליחת ACK עם נתון זה, והשולח משנה את קצב השליחה בהתאם למקום הפנוי (אם יש הרבה מקום אפשר לשלוח בקצב מהיר ולהיפך)

ההשפעות של מנגנון בקרת הזרימה של TCP על העברת נתונים הם תלויות receive-windows של השולח לעומת ה-receive-windows של המקבל. לכל אחד מהצדדים יש גודל buffer מקסימלי. גודל ה-buffer המקסימלי של השולח לא בהכרח זהה למקבל.

כתוצאה מכך, מנגנון בקרת הזרימה יכול לגרום ל:

ניצול לא מלא של משאבי השולח (וזוה עלול לגרום הצטברות של הודעות שלא יכולות להישלח- תור של הודעות),

והשהייה גבוהה: הזמן שהשולח צריך להמתין להודעות אישור והזמן שלוקח לו לעבד את הנתונים, מוסיפים להשהיות ומפחיתים את קצב ההעברה הכולל של הנתונים.

הפתרון יהיה להגדיל את נפח ה-buffer של המקבל או לשדרג את החומרה אצל המקבל.

3. בחירת הנתביב בין המקור ליעד משפיעה ישירות על מהירות, אמינות וזמני השהייה של החיבור. נתיב לא אופטימלי עלול לגרום לעיכובים בהגעת הנתונים וגם לאובדן חבילות מידע, מה שיביא להאטה משמעותית. אם נתיב מסוים עמוס מדי, החבילות עשויות להידחות או ללכת לאיבוד, מה שיגרום לשידור חוזר ולהורדת ביצועים ברשת.

הגורמים המרכזיים שצריך להתייחס אליהם בבחירת הנתביב האופטימלי:

Latency (עיכוב): דרך ארוכה יותר תגרום לעיכוב בהגעת החבילות.

רוחב פס: רוחב פס נמוך יכול ליצור צוואר בקבוק ולעכב את העברת הנתונים.

עומס: נתיבים עמוסים עשויים להוביל לאובדן חבילות ולעיכובים.

אמינות: נתיבים לא אמינים עשויים לגרום לאובדן חבילות ולשליחה מחדש של נתונים, דבר שמפחית את יעילות הרשת.

עלות: אף שהיא לא משפיעה ישירות על המהירות, כל אלגוריתם ניתוב מתחשב בעלות הנתביבים, ויש לקחת זאת בחשבון בבחירת הנתביב האופטימלי.

האלגוריתמים המודרניים לניתוב משלבים את כל הגורמים הללו בבחירת הנתביב האופטימלי ליעד.

4. MPTCP מאפשר שימוש בכמה חיבורים לרשת במקביל. במקום להסתמך על נתיב אחד, MPTCP בוחר את הנתביב היעיל ביותר עבור כל שלב בהעברת הנתונים, תוך התחשבות בפרמטרים מהשאלה הקודמת. בנוסף, MPTCP יכול להשתמש בכלל הנתביבים במקביל, מה שמגדיל את קיבולת הרשת הכוללת ומפחית את הצורך בשידור מחדש של נתונים במקרה של אובדן חבילות. ובכך משפר את ביצועי הרשת על ידי פיצול העברת הנתונים בין כמה נתיבים שונים. התוצאה היא חיבור יציב, מהיר ויעיל יותר, במיוחד בסביבות עם רשתות מרובות או לא יציבות.

5. אובדן חבילות בין שני ראוטרים יכול להיגרם ממספר סיבות בשכבות הרשת והתעבורה. ניתן סיבות מרכזיות לכל שכבה:

שכבת הרשת:

1. עומס ברשת:

בעיה: כשיש הרבה תעבורה ברשת, הראוטרים או הציוד בדרך יכולים להיות עמוסים מדי ולאבד חבילות.

פתרון: אפשר להשתמש ב QoS כדי לתת עדיפות לחבילות חשובות. אם הבעיה נובעת מעומס חמור, לשדרג את הרשת או להוסיף עוד ראוטרים כדי להפחית את העומס.

2. בעיות בניתוב:

בעיה: מקרה בו הראוטרים לא מצליחים למצוא את הדרך הנכונה לחבילות בגלל בעיות בהגדרת הניתוב.

פתרון: יש לבדוק את הגדרות הניתוב ולוודא שהן תקינות.

3. בעיות פיזיות בחיבור:

בעיה: תקלות בכבלים או בחיבורים פיזיים עשויות לגרום לאובדן חבילות.

פתרון: יש לבדוק את הציוד הפיזי, כולל כבלים ויציאות ואם צריך להחליף את הציוד.

שכבת התעבורה:

1. עומס על תחנת היעד:

בעיה: אם התחנה שאליה החבילות מגיעות עמוסה מדי, היא עלולה לזרוק חבילות.

פתרון: להחליף את תחנות היעד בתחנות עם זיכרון והמעבד משופרים, או להשתמש בשיטות שמפחיתות את העומס.

2. חיבור לא יציב:

בעיה: אם החיבור בין הראוטרים לא יציב, החבילות עשויות להיאבד בגלל ניתוקים או חיבורים לא טובים.

פתרון: יש לבדוק אם יש בעיות בחיבור עצמו בעזרת פקודות ping ו-traceroute.

3. שגיאות ב-TCP:

בעיה: אם יש אובדן חבילות ב-TCP, המערכת תנסה לשלוח אותן מחדש, וזה יכול להוביל לעיכובים נוספים.

פתרון: אפשר להשתמש ב Wireshark כדי להבין היכן הבעיה.

חלק 2:

בחלק הזה בפרויקט נענה על השאלות הבאות לכל אחד מהמאמרים:

For each paper, write:

- What is the main contribution of the paper?
- What traffic features does the paper use, and which are novel?
- What are the main results (you may copy the figures from the paper), and what are the insights from their results?

FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition:

1) What is the main contribution of the paper?

The paper introduces FlowPic, a machine learning method to classify encrypted Internet traffic. It uses packet size and arrival time data and transforms them into images and uses a Convolutional Neural Network (CNN) for classification. This approach achieves high accuracy, including 99.7% for application identification, outperforming existing methods.

2) What traffic features does the paper use, and which are novel?

FlowPic uses packet size, arrival time, and flow patterns. The novelty is converting these features into two-dimensional images and using a CNN and machine learning to detect patterns on the images created.

3) What are the main results and what are the insights from their results?

Class	Accuracy (%)			
<i>VoIP</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	99.6	99.4	48.2
	VPN	95.8	99.9	58.1
	Tor	52.1	35.8	93.3
<i>Video</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	99.9	98.8	83.8
	VPN	54.0	99.9	57.8
	Tor	55.3	86.1	99.9
<i>File Transfer</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	98.8	79.9	60.6
	VPN	65.1	99.9	54.5
	Tor	63.1	35.8	55.8
<i>Chat</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	96.2	78.9	70.3
	VPN	71.7	99.2	69.4
	Tor	85.8	93.1	89.0
<i>Browsing</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	90.6	-	57.2
	VPN	-	-	-
	Tor	76.1	-	90.6

Flowpic achieves 85.0% accuracy for non-VPN traffic, 98.4% for VPN, and 67.8% for Tor. Application identification reaches 99.7% accuracy. The model generalizes well, classifying unseen applications with up to 99.9% accuracy. The results show that machine learning is a great way to bypass usual types of encryption, however Tor is harder to classify due to stronger encryption altering flow patterns.

Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application:

1) What is the main contribution of the paper?

The paper introduces a method for identifying a user's operating system, browser, and application of HTTPS encrypted traffic by exploiting traffic patterns with the help of machine learning tools with 96.06% accuracy.

2) What traffic features does the paper use, and which are novel?

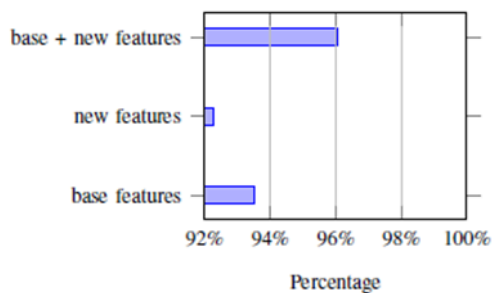
The paper uses two sets of features: base features that are commonly used in traffic classification such as:

Number of packets sent and received, packet size, forward and backward arrival times, etc.

new features introduced in this study are:

SSL features (SSL compression methods, extension count , SSL Version etc.), TCP features (TCP window size, window scaling factor, Maximum Segment Size etc.) ,and the bursty behaviour of the browser which is referred to as “Peak” behaviour.

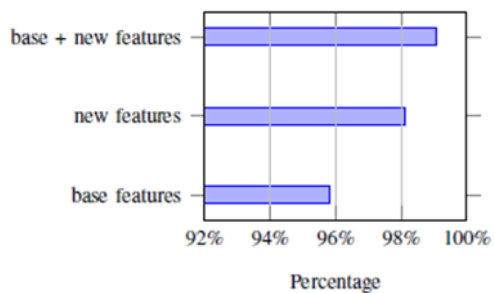
3) What are the main results and what are the insights from their results?



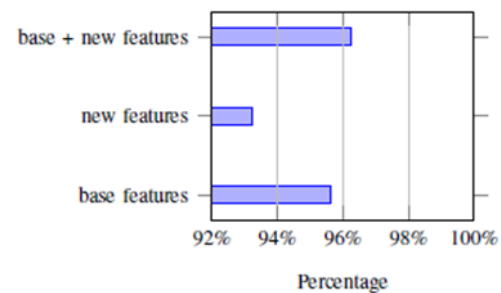
(a) Tuple Accuracy Results



(b) OS Accuracy Results



(c) Browser Accuracy Results



(d) Application Accuracy Results

		Predicted labels																																
		Windows Explorer Twitter	Ubuntu Firefox Google-Background	Windows Non-Browser Microsoft-Background	Windows Chrome Twitter	Windows Firefox Twitter	OSX Safari Google-Background	OSX Safari Youtube	Ubuntu Chrome Unknown	Windows Chrome Google-Background	Ubuntu Firefox Twitter	OSX Safari Unknown	Ubuntu Firefox Unknown	Ubuntu Chrome Google-Background	Ubuntu Chrome Twitter	Windows Firefox Google-Background	OSX Safari Twitter	Ubuntu Firefox Youtube	Windows Non-Browser Teamviewer	Ubuntu Chrome Youtube	Windows Non-Browser Dropbox	Windows Chrome Unknown	Ubuntu Chrome Facebook	Windows Firefox Unknown	Ubuntu Firefox Facebook	OSX Chrome Twitter	Windows Explorer Unknown	Ubuntu Non-Browser Microsoft-Background	Windows Explorer Google-Background	OSX Chrome Google-Background	OSX Chrome Unknown			
Real labels	Windows Explorer Twitter	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	Ubuntu Firefox Google-Background	0	.97	0	0	0	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	Windows Non-Browser Microsoft-Background	0	0	.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	Windows Chrome Twitter	0	0	0	.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0		
	Windows Firefox Twitter	0	0	0	0	.98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.02	0	0	0	0	0	0	0	0		
	OSX Safari Google-Background	0	0	0	0	0	.92	.04	0	0	0	.02	0	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	OSX Safari Youtube	0	0	0	0	0	.02	.97	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	Ubuntu Chrome Unknown	0	0	0	0	0	0	0	.84	0	0	0	0	0	.07	.04	0	0	0	0	.01	0	0	.03	0	0	0	0	0	0	0	0	0	
	Windows Chrome Google-Background	0	0	.01	.03	0	0	0	0	.94	0	0	0	0	0	0	.02	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	
	Ubuntu Firefox Twitter	0	0	0	0	0	0	0	0	0	.95	0	.03	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	OSX Safari Unknown	0	0	0	0	0	.06	.01	0	0	0	.91	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Ubuntu Firefox Unknown	0	.02	0	0	0	0	0	0	0	.08	0	.87	0	0	0	0	0	.01	0	0	0	0	0	0	0	.03	0	0	0	0	0	0	
	Ubuntu Chrome Google-Background	0	.07	0	0	0	0	0	.18	0	0	0	0	.73	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Ubuntu Chrome Twitter	0	.02	0	0	0	0	0	.08	0	0	0	0	.03	.84	0	0	0	0	.01	0	0	.01	0	0	0	0	0	0	0	0	0	0	
	Windows Firefox Google-Background	0	0	0	.01	0	0	0	0	0	.01	0	0	0	0	0	.97	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0
	OSX Safari Twitter	0	0	0	0	0	0	.06	0	0	0	0	.03	0	0	0	0	0	.91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Ubuntu Firefox Youtube	0	.02	0	0	0	0	0	0	0	0	.02	0	.02	0	0	0	0	0	.93	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Windows Non-Browser Teamviewer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
	Ubuntu Chrome Youtube	0	0	0	0	0	0	0	0	.07	0	0	0	.13	.04	0	0	0	0	0	.74	0	0	.02	0	0	0	0	0	0	0	0	0	
	Windows Non-Browser Dropbox	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Windows Chrome Unknown	0	0	.02	.09	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0	0	.86	0	0	0	0	0	0	0	0	0	0	0
	Ubuntu Chrome Facebook	0	0	0	0	0	0	0	0	.73	0	0	0	.04	0	0	0	0	0	0	0	0	0	.67	0	0	0	0	0	0	0	0	0	0
	Windows Firefox Unknown	0	0	.06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.94	0	0	0	0	0	0	0	0	0	0
	Ubuntu Firefox Facebook	0	.06	0	0	0	0	0	0	0	.11	0	.28	0	0	0	0	0	0	0	0	0	0	0	.56	0	0	0	0	0	0	0	0	0
	OSX Chrome Twitter	0	0	0	0	0	0	0	.13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.75	0	0	.06	.06	0	
	Windows Explorer Unknown	.71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.29	0	0	0	0	0	
	Ubuntu Non-Browser Microsoft-Background	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Windows Explorer Google-Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	OSX Chrome Google-Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	OSX Chrome Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) Tuple Confusion Matrix

Even encrypted HTTPS traffic leaks information that can be exploited for OS, browser, and application identification. Machine learning proved to be highly effective for this classification task. Bursty behaviour and SSL/TLS patterns serve as strong indicators of browser and application usage. This leads to potential privacy risks, where attackers could track users or optimize targeted attacks using information inferred from exposed traffic data.

Early Traffic Classification With Encrypted ClientHello: A Multi-Country Study:

1) What is the main contribution of the paper?

It introduces a new traffic classification Algorithm, hRFTC, which combines unencrypted TLS metadata, flow-based time series, and packet size statistics to enhance classification accuracy, achieving an F-score of 94.6%, significantly outperforming state-of-the-art classifiers.

2) What traffic features does the paper use, and which are novel?

The paper uses a combination of flow and packet based data such as geographical information, unencrypted ClientHello data such as Server Name Indication (SNI), packet sizes, packet inter-arrival times. The analysis of the combination of this diversity of encrypted and unencrypted data types with the use of the hybrid Random Forest Traffic Classifier (hRFTC) is not novel in the Internet traffic classification field. However, the paper presented a new approach that takes into account the geographic factor's impact on the ability of the algorithm to accurately classify data flow.

3) What are the main results, and what are the insights from their results?

TABLE 11. Full dataset per class F-score for different classifiers.

Class	F-score [%]						
	Hybrid Classifiers			Flow-based Classifier	Packet-based Classifiers		
	hRFTC [proposed]	UW [35]	hC4.5 [34]	CESNET [63]	RB-RF [24]	MATEC [33]	BGRUA [32]
BA-AppleMusic	92.1	89.5	80.2	89.2	25.5	13.1	14.5
BA-SoundCloud	99.6	98.9	97.8	98.7	84.4	81.8	82.0
BA-Spotify	93.6	90.8	89.0	88.5	16.3	0.0	3.6
BA-VkMusic	95.7	89.7	88.5	91.8	2.6	2.1	3.2
BA-YandexMusic	98.5	93.2	93.7	92.5	1.8	0.2	0.1
LV-Facebook	100.0	99.7	99.8	99.8	100.0	100.0	100.0
LV-YouTube	100.0	100.0	99.9	100.0	100.0	99.0	98.4
SBV-Instagram	89.7	74.7	76.5	78.8	10.0	6.3	6.4
SBV-TikTok	93.3	81.8	81.8	76.3	38.3	34.3	34.5
SBV-VkClips	95.7	94.0	91.3	92.4	53.2	37.7	46.0
SBV-YouTube	98.2	96.6	94.7	96.4	1.1	0.2	0.2
BV-Facebook	87.7	78.2	79.7	77.6	5.6	3.2	3.8
BV-Kinopoisk	94.1	84.1	85.8	89.8	5.4	4.0	4.1
BV-Netflix	98.5	97.2	95.2	93.7	50.7	52.3	56.1
BV-PrimeVideo	91.3	86.7	84.1	84.7	32.5	24.7	26.8
BV-Vimeo	94.8	90.5	90.2	81.4	72.0	19.5	68.6
BV-VkVideo	88.6	80.5	80.4	79.7	10.5	0.0	0.1
BV-YouTube	85.9	84.3	77.0	78.5	22.3	19.6	20.2
Web (known)	99.7	99.5	99.4	99.4	98.0	98.0	98.0
Macro-F-score (average)	94.6	89.9	88.7	88.9	38.4	31.4	35.1

LV is Live Video, (S)BV is (Short) Buffered Video, and BA is Buffered Audio.

The writers of the paper achieved great results proving the ability of their hRFTC algorithm to accurately classify a large and diverse data set and outperform all the known state-of-the-art hybrid and non-hybrid classification algorithms that were tested.

Moreover, they discovered that Since TLS metadata is heavily encrypted, relying on it alone is no longer viable, especially with the introduction of the encrypted ClientHello. Hybrid models that incorporate flow-based statistics are the most viable for encrypted traffic classification

Inorder to achieve high quality identification, it is required to train the algorithm in a geographical environment that matches the one it will later need to operate on since classifiers are effective within known geographic regions but struggle with locations they were yet to train on.

חלק 3:

החלק הזה של הפרויקט עוסק בניתוח תעבורת רשת של אפליקציות שונות וזיהוי מאפייני התעבורה שלהן, גם כאשר התעבורה מוצפנת.

לינק לקבצי pcap שלנו:

<https://drive.google.com/drive/folders/1wOJY19L-psRVSeUh0zKVgGVwcX2T9ucx?dmr=1&e=c=wgc-drive-hero-goto>

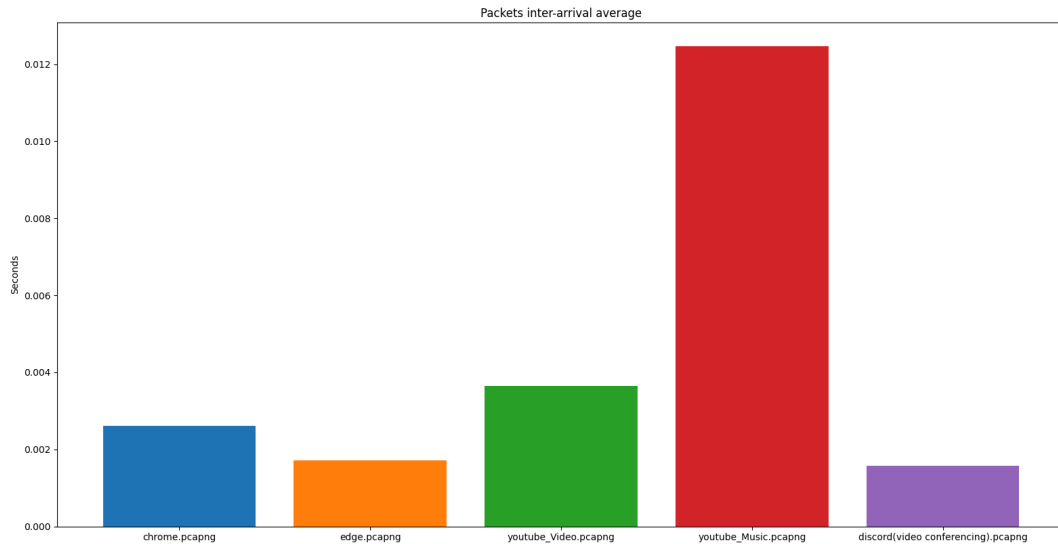
השתמשנו בחבילה Pyshark לצורך הפעלת פונקציות לניתוח תעבורת הרשת, ועשינו בה שימוש ליצירת הגרפים המוצגים. ביצענו הקלטת תעבורת רשת באמצעות Wireshark עבור שימושים שונים של המשתמש ודגמנו 5 שניות של הקלטה עבור כל פעילות. הפעילויות שנדגמו היו:

- גלישה באינטרנט דרך דפדפן Chrome
- גלישה באינטרנט דרך דפדפן Edge
- הזרמת וידאו דרך YouTube באמצעות Chrome
- הזרמת שמע דרך YouTube Music באמצעות Chrome
- שיחת ועידה עם וידאו באמצעות אפליקציית Discord



תרשים א' - גודל חבילה ממוצע בבייטים ל-0.05 שניות בפרק זמן של 5 שניות.

ניתן לראות באופן בולט בגרף שבניגוד לאפליקציות האחרות, אפליקציית דיסקורד משתמשת בעקביות בגודל פאקטות גדול יחסית. הזרמת וידאו ביוטיוב וגלישה ב-אדג' יוצרות התפרצויות של גודל פאקטות גבוה. לעומת זאת גלישה בכרום וב-"YouTube Music" מראה שגודל הפאקטות הממוצע הוא נמוך יחסית באופן עקבי מלבד התפרצויות בודדות.

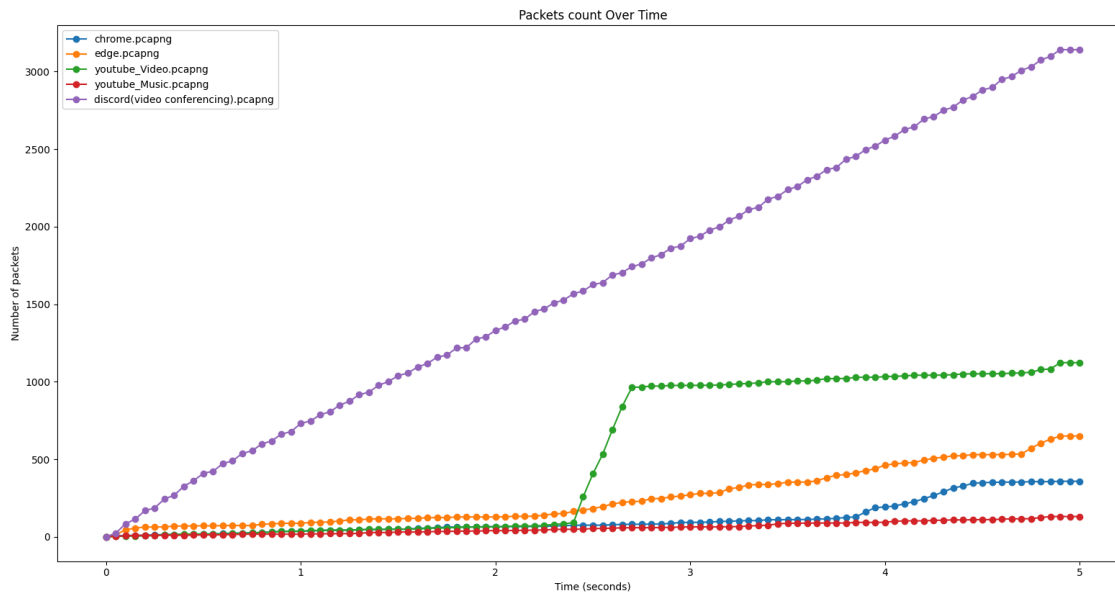


תרשים ב' - זמן ממוצע בין הגעה של זוג חבילות (פקטות)

הגרף מצביע על כך שזמן ההגעה בין פאקטות בהזרמת אודיו נוטה להיות גבוה, מה שמעיד על כך שהזרמת אודיו דורשת תדירות נמוכה במיוחד של הגעת פאקטות. בנוסף, גם פאקטות של וידאו נוטות להגיע עם מרווחי זמן גדולים ביחס לשאר הפעולות שנמדדו.

עם זאת, באופן לא מפתיע, שיחת ועידה עם וידאו בדיסקורד הציגה את זמני ההגעה בין פאקטות הנמוכים ביותר. נתון זה מתיישב עם הצורך של שיחות וידאו להעברת קול ותמונה במהירות, במטרה לצמצם עיכובים בין השולח למקבל ולאפשר שיחה רציפה וללא השהיות.

בשיחת ועידה, אין צורך לטעון ולשלוח חבילה גדולה של מידע, שכן פעולה כזו הייתה מובילה לעיכובים. עם זאת, יש צורך בשליחה מהירה ורציפה של כמות גדולה של מידע, ולכן תדירות השליחה של חבילות במצב זה נמצאה כגבוהה ביותר.

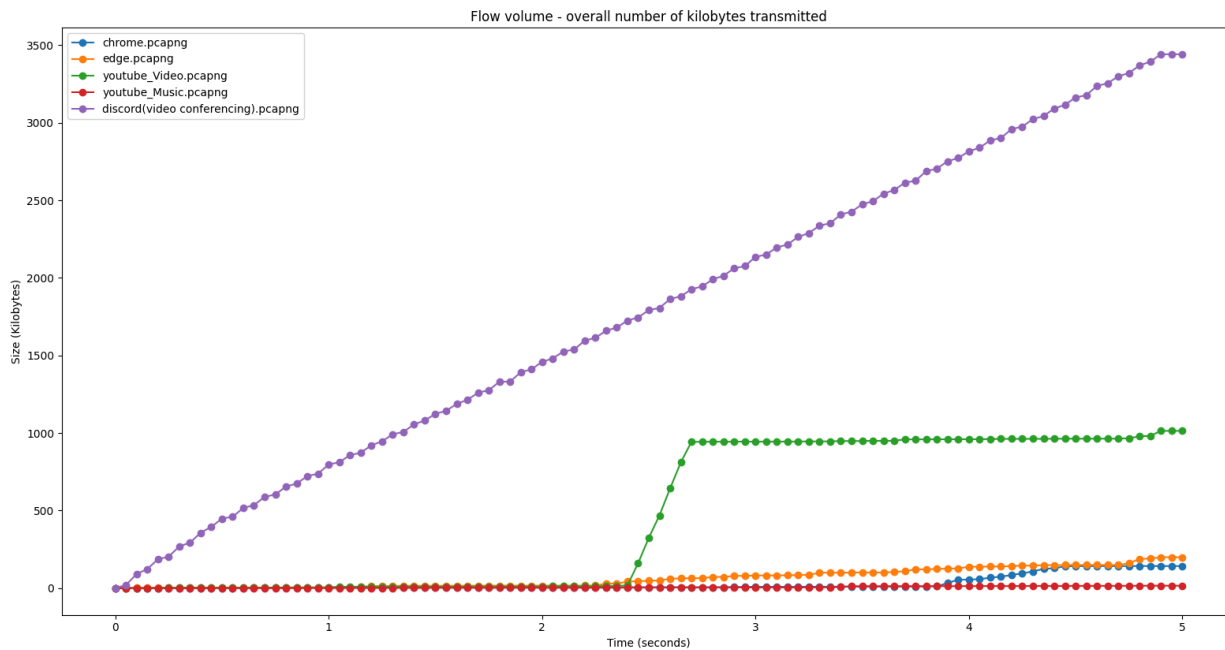


תרשים ג' - מספר חבילות כולל בפרק זמן של 5 שניות

אפשר לראות בתרשים שדיסקורד שולח מספר גבוה של חבילות בקצב עקבי בזמן שיחת ועידה עם וידאו. בצפייה בוידאו ביוטיוב נצפתה קפיצה חדה של העברה של חבילות בין שניות 2 ו-3 ובשאר הזמן היה מעבר מתון של חבילות. עיון בדפי אינטרנט בדפדפנים כרום ואדג' סיפקו עקומות דומות אחת לשניה. ההסבר לכך הוא שבעוד ש-STREAMING ניתן לאגור DATA ולהציגו אחר כך, בשיחת וועידה מדובר ב-DATA בזמן אמת ולכן צריך להציג אותו מיד כשהוא מתקבל כך שלא מתאפשרת אגירה

עם זאת נראה שדפדפן הכרום משתמש בפחות חבילות להעברה בזמן הגלישה אולם מכיוון שמדובר ב-2 הקלטות בודדות, המדגם מצומצם מידי מכדי להסיק על הבדלים מהותיים בין שני הדפדפנים.

כמו כן, הופתענו לגלות שהזרמת אודיו דרך YouTube Music השתמשה במספר הנמוך ביותר של חבילות לאורך ההקלטה. היינו מצפים לראות התנהגות דומה לזו של הזרמת וידאו אך בניגוד לגרף של ההקלטה של YouTube לא צפינו בקפיצה החדה של העברת החבילות בהקלטה של YouTube Music.



תרשים ד' - נפח הגלישה, סך כל נפח המידע שעבר לאורך זמן בקילובייטים.

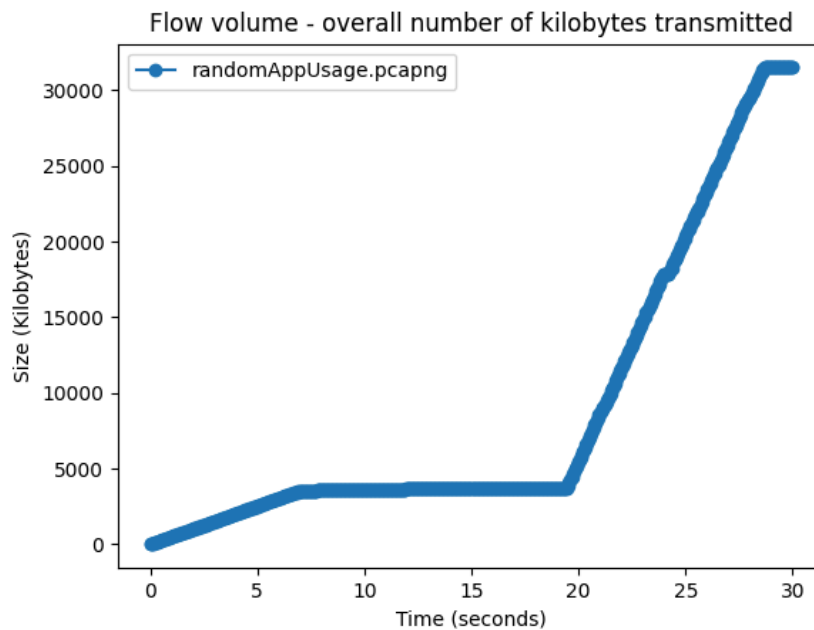
כצפוי, אנחנו רואים התאמה כמעט מושלמת בין גרפים ג' ו-ד' שכן זה מתיישב עם ההיגיון שמספר החבילות שעברו לפרק זמן יעיד גם על נפח המידע שעבר.

עם זאת מעניין לראות שבפרק זמן 4-4.5 שניות של ההקלטה הגלישה בגוגל כרום העבירה מידע בנפח זהה לזה שבגלישה דרך Microsoft Edge אך מספר החבילות שעברו באותו זמן היה נמוך בכ-50%. משמעות הדבר היא שבפרק הזמן המדובר דפדפן Chromen העביר חבילות בנפח גבוה. המסקנה הזאת באה לידי ביטוי גם בגרף א' המתאר גודל חבילות ממוצע גבוה עבור Google Chrome בפרק הזמן הזה.

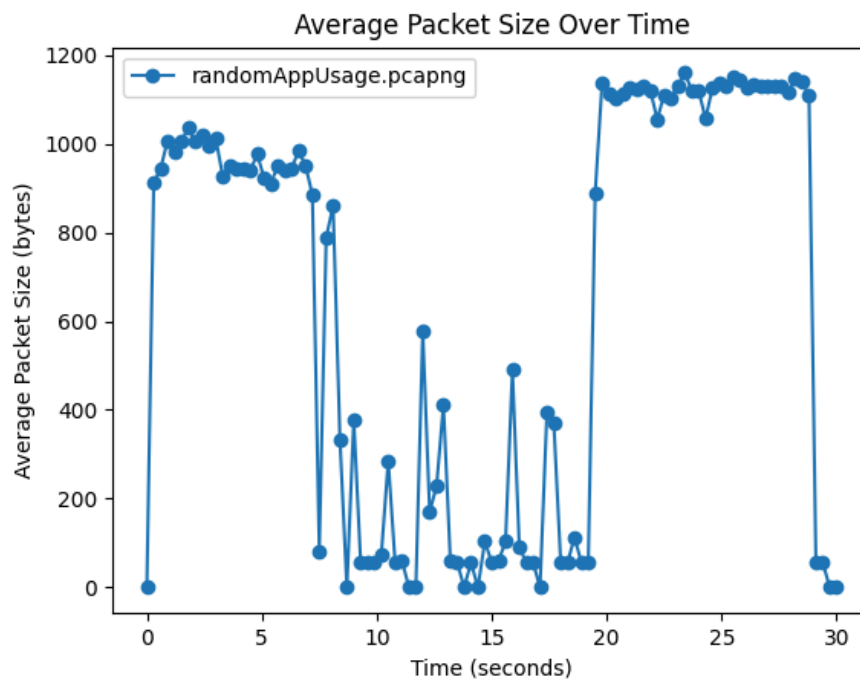
תרחישי תקיפה:

בחלק הזה של הפרויקט נניח שאנחנו תוקף שמנסה לגלות אילו אפליקציות המשתמש הפעיל, בשני תרחישים שונים. הראשון, כאשר מידע כמו גודל חבילות, זמני הגעה, כתובות ה-IP ומספרי פורטים של התעבורה חשופים לתוקף. ובחלק השני נתאר תרחיש שבו לתוקף יש גישה רק לגודל זמן ההגעה של כל חבילה.

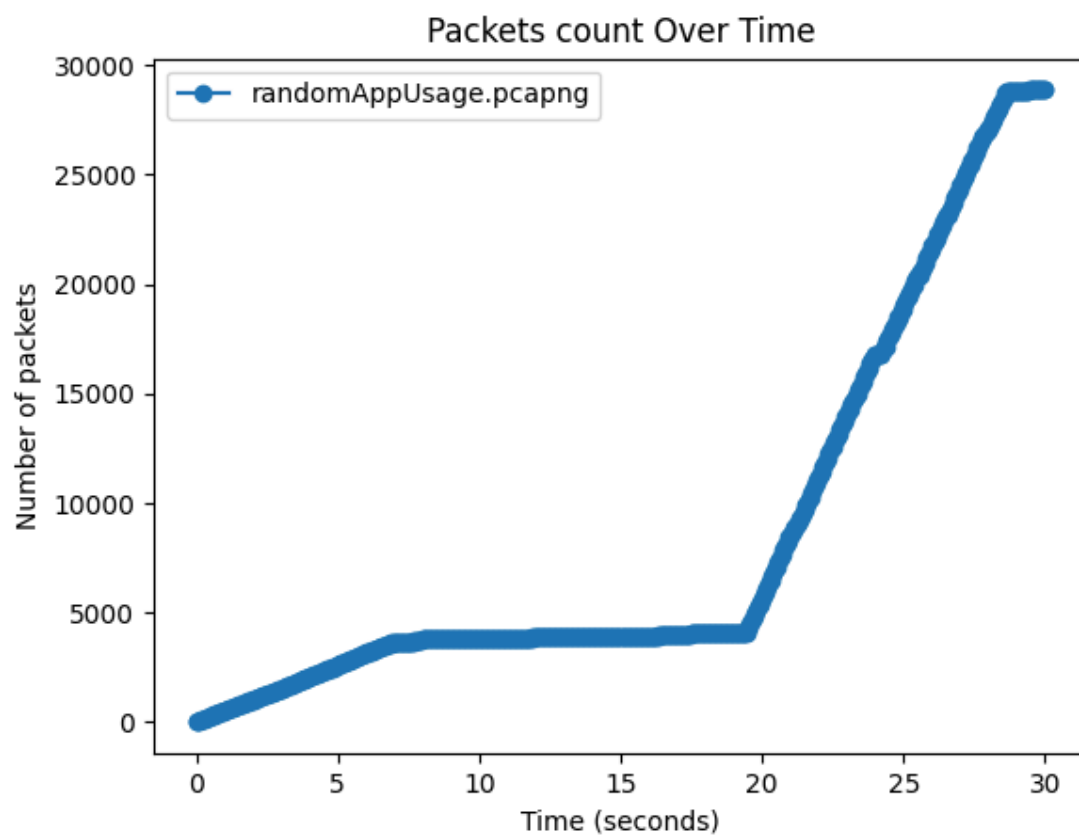
בחרנו להקליט קבצי Video Conferencing ושירות לאחר מכן להקליט קבצי וידאו. להלן התוצאות



תרשים ה' - סך כל נפח הגלישה



תרשים ו' - גודל חבילה ממוצעת



תרשים ז' - סך כל החבילות שעברו

תרחיש ראשון: לתוקף יש מידע על גודל החבילה, חותמת הזמן ו-Hash של מזהה הזרימה (Flow ID)

בתרחיש הזה ניתן לזהות בקלות את האפליקציות שהשתמשו בהן, מכיוון ש-כתובות ה-IP של היעד מספקות מידע ישיר על השרתים אליהם המשתמש מתחבר. כאשר לתוקף יש הרבה מידע נגיש על התעבורה. שדות כמו כתובות IP יכולים להפוך את הפענוח של הפעילות לפשוט מאוד במידה והכתובות מוכרת לתוקף.

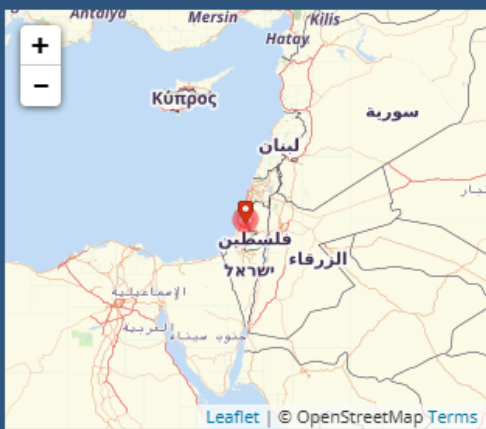
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.12.22.78	34.0.65.103	UDP	1100	58236 → 50001 Len=1058
2	0.004284	34.0.65.103	10.12.22.78	UDP	1079	50001 → 58236 Len=1037
3	0.004284	34.0.65.103	10.12.22.78	UDP	1079	50001 → 58236 Len=1037
4	0.005250	10.12.22.78	34.0.65.103	UDP	1100	58236 → 50001 Len=1058
5	0.006003	10.12.22.78	34.0.65.103	UDP	268	58236 → 50001 Len=226
6	0.006085	10.12.22.78	34.0.65.103	UDP	1100	58236 → 50001 Len=1058

תצלום מסך 1 - 5 חבילות ראשונות מתוך הקלטת wireshark

אפשר לראות שהמשתמש מקבל חבילות תחת פרוטוקול UDP מהכתובת 34.0.65.103. מתוך האתר whatismyipaddress.com/ip-lookup ניתן לראות שמדובר בשרת של שירותי ענן של גוגל הממוקם בישראל.

IP Details For: 34.0.65.103

Decimal:	570442087
Hostname:	103.65.0.34.bc.googleusercontent.com
ASN:	19527
ISP:	Google LLC
Services:	Datacenter
Country:	Israel
State/Region:	HaMerkaz
City:	Petah Tikva
Latitude:	32.0917 (32° 5' 30.26" N)
Longitude:	34.8850 (34° 53' 6.10" E)



CLICK TO CHECK BLACKLIST STATUS

תצלום מסך 2 - פרטי IP של 34.0.65.103

מבדיקה קצרה בעזרת כלי AI גילינו שיכול להיות שהשרת המדובר מספק שירותי ענן לגורמים שונים ולכן בניגוד לאינטואיציה הראשונית, כתובת ה-IP בלבד לא תמיד תספיק כדי לסווג באופן וודאי את סוג התעבורה.

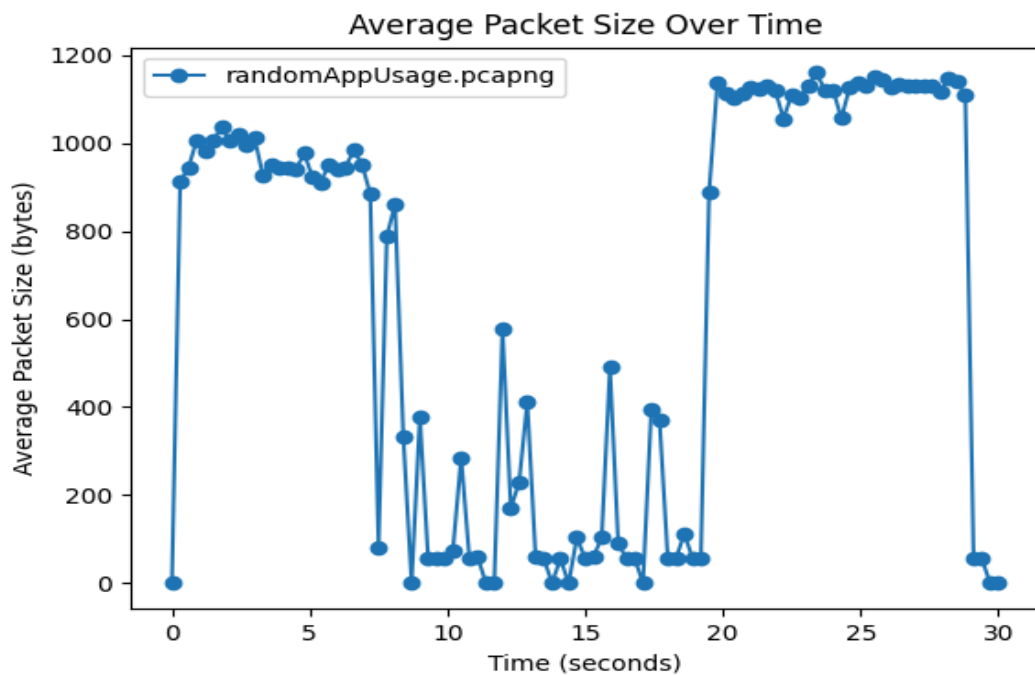
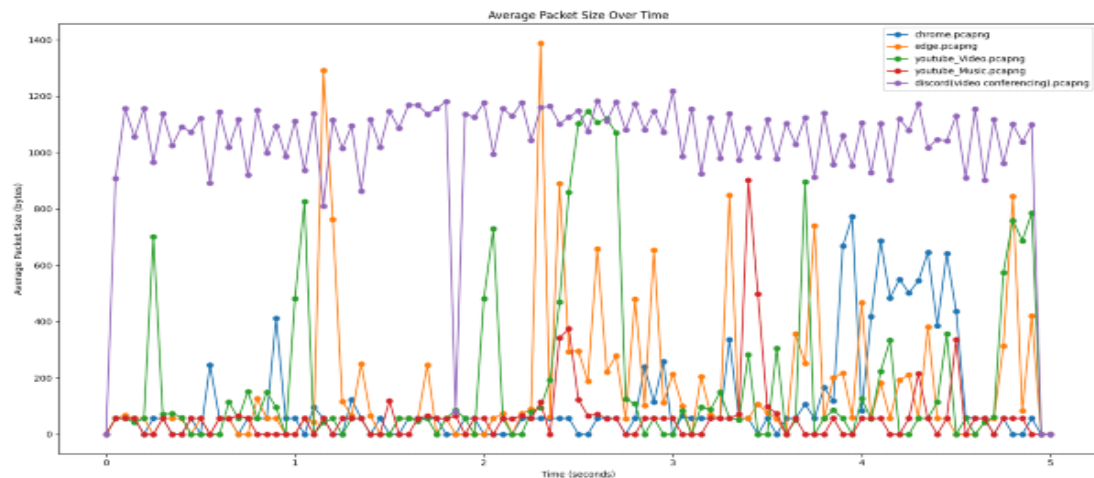
כמו כן, מספרי הפורטים לא נותנים מידע קונקרטי על סוג התעבורה מכיוון שהם יכולים לשמש למגוון של סוגי התקשורות ואפליקציות.

תרחיש שני: לתוקף יש מידע רק על גודל החבילה וחומת הזמן

בתרחיש הזה הזיהוי מורכב יותר אך עדיין אפשרי. ללא כתובת ה-IP, התוקף יכול להשוות את דפוס הגודל והתדירות לדפוסים ידועים של אפליקציות מסוימות. אם לתוקף היה את הגרפים שהוצגו בסעיף הקודם, הוא יהיה יכול להצליב את הדפוסים שכרגע מקבל לבין הדפוסים של הגרפים, ולהניח שמדובר באפליקציה דומה.

לדוגמה: מקטעים גדולים לסירוגין יכולים להעיד על הזרמת וידאו (YouTube). חבילות קטנות בתדירות גבוהה עשויות להתאים לגלישה בדפדפן ואילו חבילות גדולות המגיעות בתדירות גבוהה היו יכולות לרמז על השתתפות בשיחת ועידה.

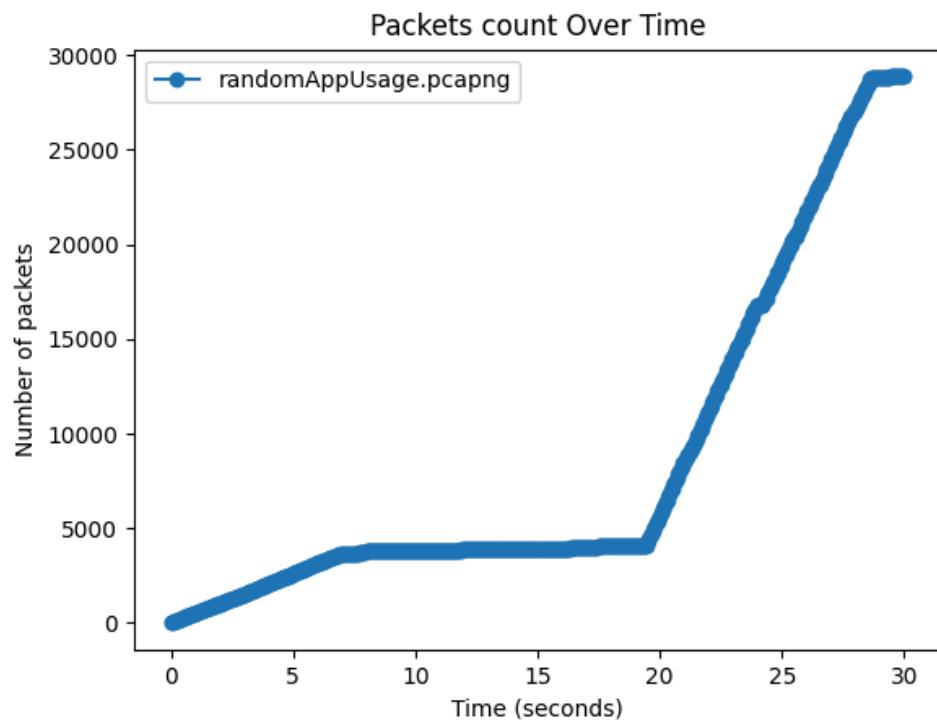
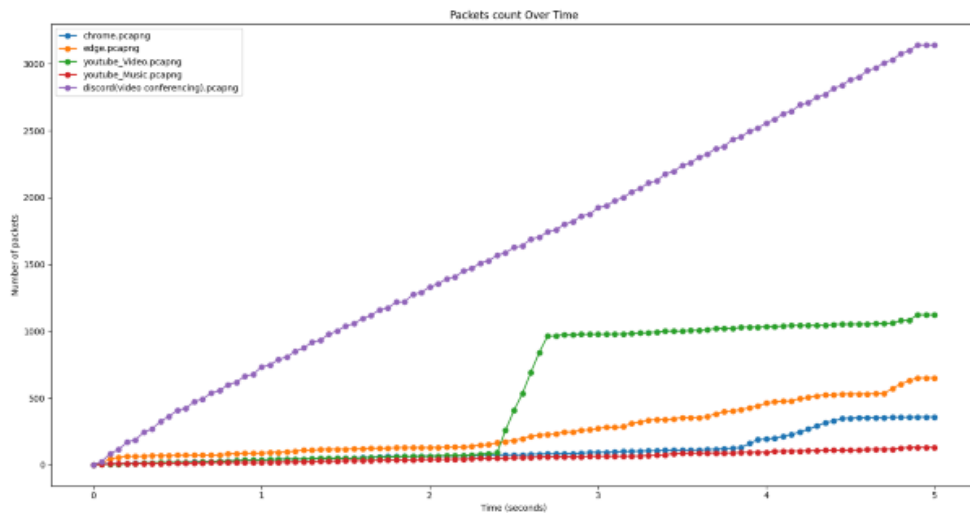
נשווה בין הגרף של הגודל החבילות הממוצע ה"לא ידועה" שברשות התוקף לאותו גרף שהפקנו בחלק 3 ונחפש דפוסים דומים.



תרשים ח' - השוואה בין גרפי גודל חבילות ממוצע (מעל: גודל חבילות ממוצע לפעילות ידועה ברשת, מתחת: גודל חבילות ממוצע של פעילות רשת לא ידועה)

אפשר לראות מהגרף שבתחום ה-8 שניות הראשונות של התעבורה שנקלטה גודל החבילות עמד על בין 800 ל-1100 בייטים באופן עקבי. ניתן להסיק מכך שככל הנראה המשתמש השתתף בשיחת וידאו.

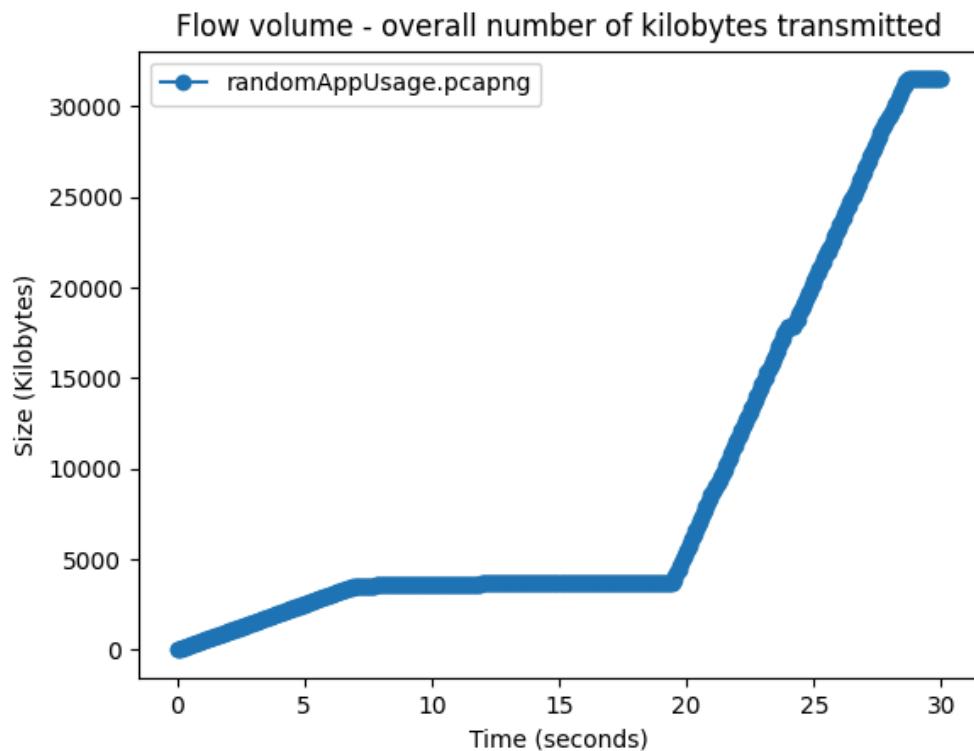
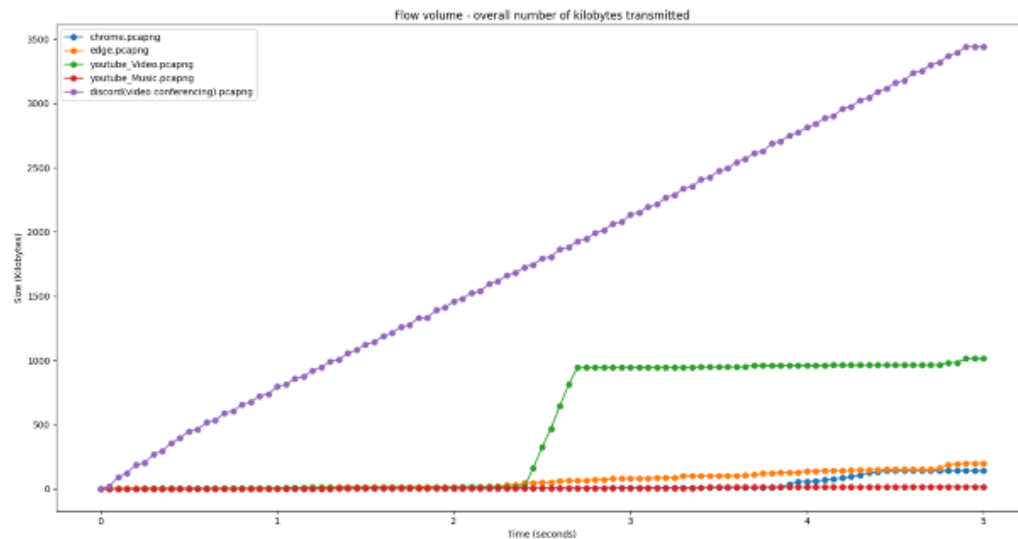
לאחר מכן הגרף צונח מטה ומראה קפיצות חדות בתחום של 0-600 בייטים עד השניה ה-20. בתחום הזה קשה לומר בוודאות אך הוא ככל הנראה גלש בדפדפן אך קיימת אפשרות שהפעילות שנצפתה בתחום הזמן הזה הייתה "רעש". בשנייה ה-20 נצפתה קפיצה חדה לתחום ה-1000-1200 בייטים שאופיינית לשיחות וידאו או שימוש בהזרמת וידאו.



תרשים ט' - השוואה של מספר החבילות שעברו

ניתן לראות מתרשים ו' שבכ-5 שניות של הקלטה עברו כ-3000 חבילות. כעת ניתן מאוד בקלות לזהות שהתוצאות הללו זהות לאלו של משתמש הנמצא בשיחת וידאו בדיסקורד.

עם זאת מהשנייה ה-8 להקלטה אנחנו צופים התמתנות משמעותית במעבר החבילות עד שניה 20. בדומה לגלישה בדפדפן או האזנה להזרמת שמע. מהשנייה ה-20 נצפתה עליה חדה אף יותר במספר החבילות שעברו עם התיישרות העקומה מהשנייה ה-28. בפרק הזמן שבין השנייה ה-20 ל-28 עברו כ-25000 חבילות. משמע, כ-3000 חבילות לשנייה. משמעות הדבר שהמשתמש ככל הנראה מהשנייה ה-20 התחבר מחדש למדיום של העברת וידאו, יוטיוב או שיחת וידאו.



תרשים י' - השוואה של סך כל נפח התעבורה

הדמיון בין התרשימים ו' ו-ז' אינו מקרי ובדומה לחלק 3, כצפוי, נראה שקיים קשר ישיר והגיוני בין מספר החבילות שעברו לבין נפח המידע שעבר. לכן, לא נוסף מידע משמעותי.

לסיכום, אם נצפו בתעבורה כתובות IP מוכרות או שאפליקציה מסויימת השתמשה במספר פורטים שידועים שאלו הם הפורטים שהיא משתמשת בהם לתהליכים שונים הדבר עשוי להקל על הזיהוי של האפליקציה שבה המשתמש עושה שימוש, אך זה לא תמיד מספיק כדי לקבוע חד משמעית באיזה אפליקציה הוא משתמש.

בחלק 4 בפרויקט ידענו שב-10 השניות הראשונות להקלטה המשתמש היה בשיחת וידאו בדיסקורד וה-IP שנצפה היה של שרת שירותי ענן של גוגל, לכן כתובת ה-IP לא הייתה מספיקה כדי לקבוע מה קורה והתוקף הנאיבי היה עלול לחשוב שמדובר בצפייה ב-Youtube בגלל שזה אתר בבעלות Google ומכאן מצטיירת תמונה מורכבת יותר של האופן שבו מועברים נתונים ברשת. לכן לתוקף שחשוף בפניו יותר מידע במקרה הראשון יהיה יותר קל לזהות במה מדובר עם דיוק גבוה יותר אבל לא תמיד.

מידע לא מוצפן כמו חתימות זמן של הגעת חבילות, גודל חבילות ונפח גלישה יכולים לתת תמונה כללית על אופי הפעילות מבוצעת אבל לא באופן חד משמעי מה היא בדיוק. המידע הנתון לא תמיד מספיק לקבוע באיזה אתר בדיוק המשתמש גולש או באיזה דפדפן. יכול להיות שהמשתמש יצפה בכתבה באתר חדשותי או סרטוני חתולים ביוטיוב והנתונים הלא מוצפנים יהיו דומים. לכן התוקף במקרה השני יכול להעריך בקירוב טוב מה אופי התעבורה אבל לא בדיוק באיזה אתר או אפליקציה ספציפיים המשתמש עושה שימוש.

כיצד ניתן להקשות על התוקף?

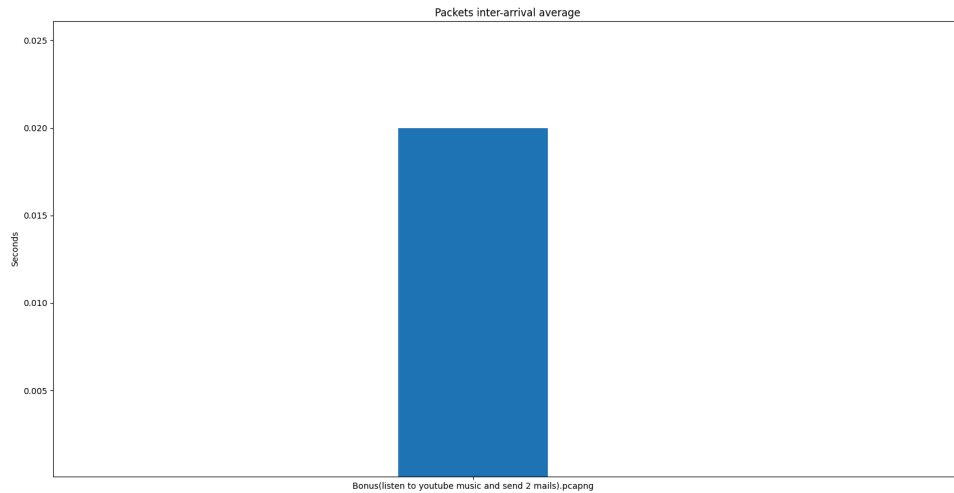
כדי להקשות על תוקף בזיהוי האפליקציות וסוגי הפעילות של המשתמש, ניתן ליישם מספר שיטות. אחת מהן היא תעבורה אקראית (Noise), שבה נשלחות חבילות דמה באופן אקראי כדי לשבור דפוסים קבועים ולהקשות על ניתוח התעבורה.

שיטה נוספת היא שימוש בהצפנה מתקדמת, כגון רשת Tor או חיבור VPN, אשר מסייעות בהסתרת כתובות ה-IP ובטשטוש המידע על הזרימות.

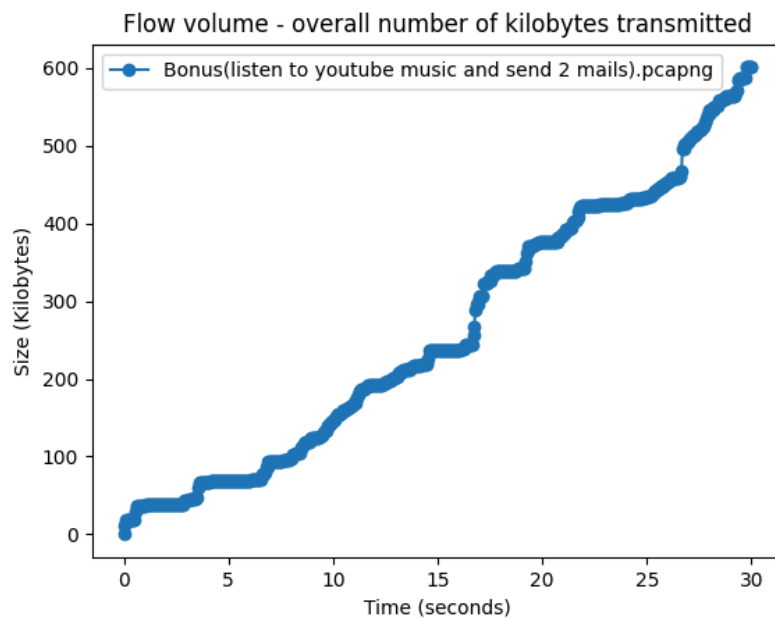
בנוסף, ניתן להשתמש בטכניקת איחוד זרימות (Multiplexing), המשלבת מספר זרימות שונות לחיבור מוצפן יחיד, דבר שמסבך את יכולת התוקף להבחין בין הפעילויות השונות.

בונוס – ניתוח כאשר אפליקציה משנית פעילה ברקע

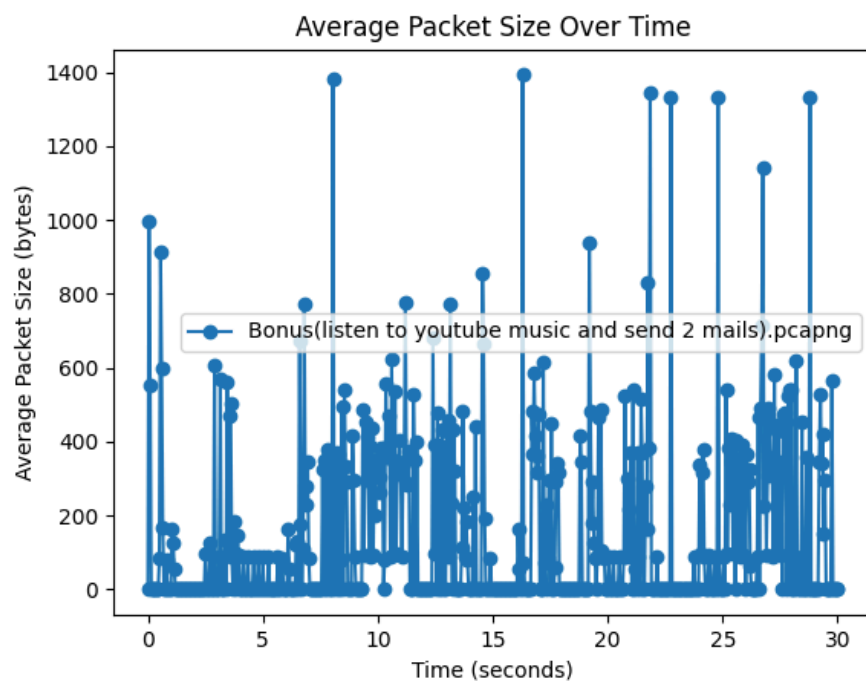
צילמנו תעבורת רשת ב-Wireshark בזמן האזנה ל-Spotify ושליחת אימיילים. להלן התוצאות המוצגות בגרפים הבאים:



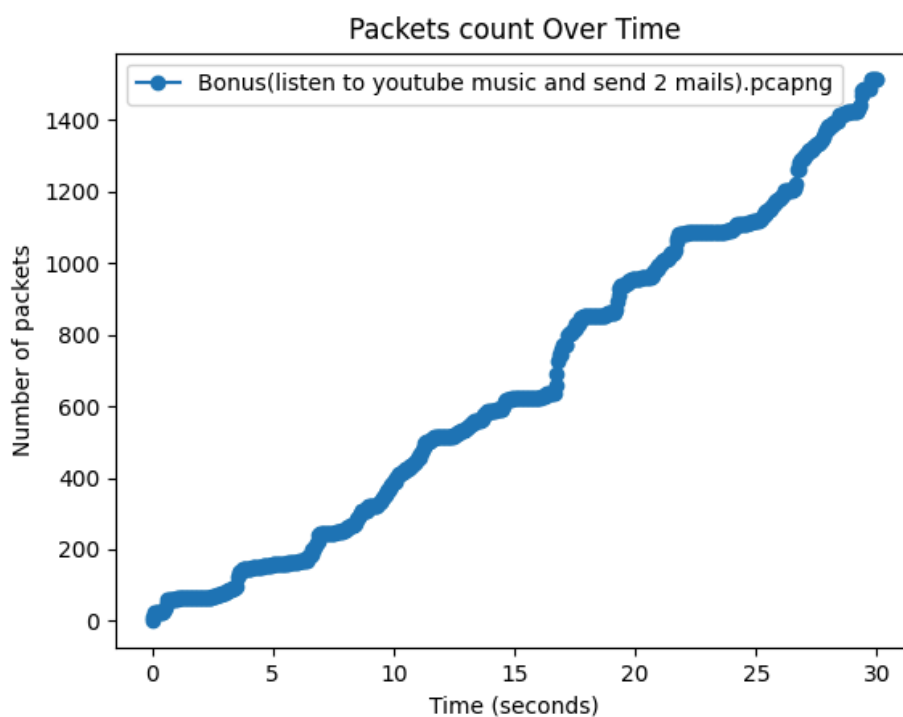
תרשים י"א - זמן בין הגעת חבילות ממוצע



תרשים י"ב - גרף סך הנפח שהועבר במשך 30 שניות



תרשים י"ג - גודל חבילה ממוצעת



תרשים י"ד - סך כל החבילות שעברו

מהגרפים ניתן לראות כי זיהוי הדפוסים הפעם מאתגר יותר. כאשר Spotify פועל ברקע והמשתמש שולח מיילים, התוקף עשוי להבחין בשינויים פתאומיים בדפוס התעבורה. דפוס סטרימינג אחידים יכולים להשתנות עקב פעולות קצרות וממוקדות, כמו שליחת מיילים, מה שמאפשר לתוקף להבחין בין הפעילויות, בדומה לסעיף הקודם.

כדי להקשות על התוקף בזיהוי הפעילות, ניתן ליישם את השיטות שהוזכרו קודם, כגון הוספת תעבורה אקראית, שימוש בפרוטוקולי הצפנה מתקדמים ואיחוד זרימות.

שימוש בכלי בינה מלאכותית (AI):

הprompts שהשתמשנו בהם:

1. "כיצד משפיע בחירת המסלול על ביצועי הרשת, אילו גורמים משפיעים על בחירת הנתבי המסוים"
2. "מהו פרוטוקול Multipath TCP"
3. "כיצד ניתן להקשות על תוקף בזיהוי אפליקציות ופעילויות משתמש באמצעות ניתוח תעבורת רשת, גם כאשר התעבורה מוצפנת"
4. "How to get packets from a pcapng files with pyshark"
5. "How to get only the first and the last packets in pyshark"
6. "In pyshark how to get the packet time and the packet size"
7. "How to set random different colors in plt.bar"
8. "How can I identify who an ip address belongs to?"
9. "What is 103.65.0.34.bc.googleusercontent.com?"

<https://www.geeksforgeeks.org/numpy-tril-python/>