בס"ד

Final project – communication network

323834309_329808554_325445997_322241654

Part 1 –

שאלה 1

גורמים אפשריים להאטה:

- עומס ברשת - TCP משתמש במנגנון בקרת עומס (Congestion Control) כדי להקטין את קצב השידור כאשר הרשת עמוסה.
- בקרת זרימה - TCP משתמש במנגנון חלון הקבלה כדי לוודא שהשולח לא ישלח יותר נתונים ממה שהמקבל יכול לעבד, אם ה Receive Window קטן, העברת הנתונים תהיה איטית.
- אובדן חבילות - אובדן חבילות יכול לגרום להאטה משמעותית בגלל מנגנון ההעברה מחדש של TCP (Retransmission).
- עיכוב חבילה - פרוטוקול TCP כולל מדידה של Round Trip Time (RTT), ואם הוא גדול, זה עלול להאט את ההעברה.

דרכי פתרון:

- בדיקת עומסים ברשת עם Wireshark כדי לזהות עומס.
- הגדלת חלון הקבלה של TCP כדי לשפר את קצב השידור.
- הקטנה של גודל החבילות הנשלחות כך שיתאימו ליכולת הרשת להעביר אותן בשלמותן, ובכך למנוע עיכובים ואובדן מידע.
- מעבר לפרוטוקול UDP במקרים שבהם אמינות לא חשובה (למשל סטרימינג).

שאלה 2

השפעת בקרת הזרימה של TCP :

TCP משתמש בחלון קבלה (Receive Window) שמציין לשולח כמה נתונים ניתן לשלוח מבלי להציף את המקבל. אם השולח חזק מאוד (כגון שרת מהיר) והמקבל איטי יותר (למשל מחשב בעל משאבים מוגבלים), קצב העברת הנתונים יהיה מוגבל בהתאם לחלון הקבלה.

השפעה כאשר השולח חזק מהמקבל:

אם המקבל לא מצליח לעבד את הנתונים במהירות מספקת, הוא ישלח חבילות עם Receive Window קטן, מה שיגרום לשולח להאט את ההעברה.

שאלה 3

תפקיד הניתוב:

ניתוב ברשת מאפשר לחבילות מידע לעבור מנתיב המקור אל היעד על פי אלגוריתמים שונים שמטרתם למצוא את המסלול היעיל ביותר.

השפעת בחירת הנתיב על ביצועים:

- עיכוב - מסלול קצר יותר מפחית עיכובים.
- עומס ברשת - מסלול עמוס גורם להאטת העברת הנתונים.
- אמינות - בחירת נתיב אמין יותר מונעת אובדן חבילות ועיכובים.

גורמים בהחלטות ניתוב:

- שימוש בפרוטוקול OSPF לחישוב מסלולים ברשתות מקומיות.

בס"ד

- שימוש בפרוטוקול BGP לניהול תעבורה בין ספקיות אינטרנט.

שאלה 4

MPTCP מאפשר שימוש במספר נתיבים במקביל במקום להשתמש רק בחיבור אחד, מה שמאפשר:

- שיפור ביצועים - הזרמת נתונים ביותר מנתיב אחד מגדילה את מהירות ההעברה.
- עמידות בפני תקלות - אם אחד הנתיבים נופל, החיבור ממשיך לעבוד דרך נתיב אחר.
- איזון עומסים - עומס הרשת מתחלק בין מספר חיבורים.

שאלה 5

גורמים אפשריים לאובדן חבילות:

שכבת הרשת -

- עומס ברשת - כאשר נתבים מקבלים יותר חבילות ממה שהם יכולים לטפל, חלק מהחבילות נזרקות.
- בעיות חומרה – כשל בנתב או בעיות בתשתית הרשת עלולות לגרום לאובדן חבילות.

שכבת התעבורה -

- מנגנון בקרת עומס של TCP TCP עשוי להקטין את קצב השידור במקרה של עומס ולגרום לעיכובים.
- פרגמנטציה של חבילות – אם חבילות מחולקות למקטעים קטנים יותר, ייתכן שחלקן ילכו לאיבוד.

פתרונות אפשריים:

1. בדיקה עם Wireshark כדי לנתח חבילות ולזהות היכן הן אובדות.
2. שיפור תשתית הרשת והגדרת Quality of Service (QoS) לניהול עדיפויות של חבילות.
3. שימוש בפרוטוקול MPTCP כדי לאזן את העומס ולמנוע הצטברות נתונים על נתיב אחד.

Part 2 –

FlowPic_Encrypted_Internet_Traffic_Classification_is_as_Easy_as_Image Recognition

התרומה העיקרית של המאמר:

המאמר מציג **שיטה חדשה לזיהוי וסיווג תעבורת אינטרנט מוצפנת** .הרעיון הוא **לקחת רק את זמני ההגעה וגודל החבילות, להפוך אותם לתמונה (FlowPic) ולהשתמש ברשתות נוירונים קונבולוציוניות (CNN) כדי לנתח את הדפוסים**.
השיטה מאפשרת **לזהות את קטגוריית תעבורה, כמו - גלישה, וידאו, או צ'אט. וגם את היישום הספציפי כמו למשל – Skype או YouTube, Facebook Video** , גם כשנעשה שימוש **בהצפנה ע"י VPN או Tor** , שיטות שבדרך כלל מקשות מאוד על סיווג תעבורה.
בניגוד לשיטות קודמות ,**אין צורך לבחור ידנית מאפיינים סטטיסטיים** ,מה שמקל על היישום שלה ומשפר את הדיוק.
בנוסף, FlowPic משתמש בחלק קטן מאוד מהנתונים הנשלחים ולא צריך לחכות לכל התקשורת בין שני הצדדים כדי לסווג את התעבורה. זה אומר שהשיטה יכולה לסווג תעבורה מהר יותר ובצורה יעילה יותר משיטות אחרות שדורשות את כל המידע מההתחלה עד הסוף.

יתרון נוסף הוא שהשיטה **לא משתמשת בתוכן החבילות** , מה שאומר שהיא **שומרת על פרטיות המשתמשים** , וגם **דורשת פחות משאבי אחסון וחישוב** לעומת שיטות קודמות.

זאת הפעם הראשונה שנעשה שימוש בטכניקות זיהוי תמונה לסיווג תעבורת רשת מוצפנת.

בס"ד

ברוב המקרים, השיטה מצליחה להשיג תוצאות טובות יותר משיטות קודמות.

<u>מאפייני תעבורה שנעשה בהם שימוש (שגרתיים וחדשים)</u>:

המאמר משתמש רק בשני מאפיינים מתוך זרימות הרשת -גודל החבילות זמן ההגעה שלהן.

החידוש במאמר הזה הוא איך שהנתונים האלה מעובדים - **הנתונים הופכים לגרף שבו ציר ה X-מראה את זמני ההגעה של החבילות וציר ה Y-מראה את גודל החבילות .הגרף הזה הופך לתמונה שנקראת FlowPic. התמונה הזאת מוכנסת לרשת נוירונים (CNN) שמזהה את סוג התעבורה בצורה מאוד מדויקת.**

היתרון של השיטה הזאת הוא שהיא עובדת באופן אוטומטי לגמרי. בשיטות הישנות, החוקרים היו צריכים לבחור בעצמם ובאופן ידני איזה נתונים חשובים לזיהוי - למשל כמה חבילות נשלחו, מה הגודל הכולל שלהן, או מידע מהכותרות של הפרוטוקולים. לעומת זאת, השיטה החדשה פשוט לוקחת את התמונה שנוצרה – FlowPic , ומאפשרת לרשת הנוירונים לגלות בעצמה אילו דפוסים חשובים לזיהוי. זה עובד טוב יותר כי התמונה מראה דפוסים שקשה לתאר במספרים פשוטים, והשיטה עובדת על כל סוגי התעבורה בלי קשר לפרוטוקול שבשימוש.

<u>ממצאים עיקריים ותובנות</u>:

<u>התוצאות העיקריות</u>:

1. דיוק בסיווג תעבורה -

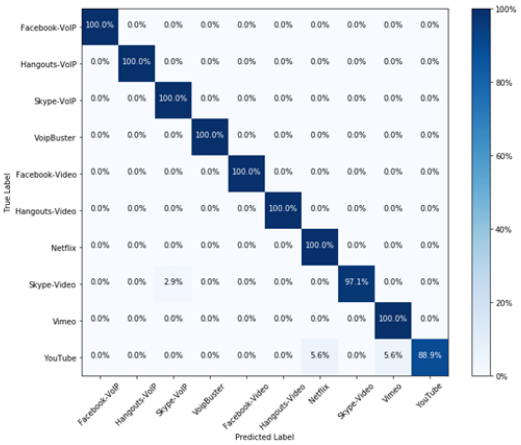
Problem	FlowPic Acc. (%)	Best Previous Result	Remark
Non-VPN Traffic Categorization	85.0	84.0 % Pr., Gil *et al.* [15]	Different categories. [15] used unbalanced dataset
VPN Traffic Categorization	98.4	98.6 % Acc., Wang *et al.* [7]	[7] Classify raw packets data. Not including browsing category
Tor Traffic Categorization	67.8	84.3 % Pr., Gil *et al.* [15]	Different categories. [15] used unbalanced dataset

2. סיווג בינארי של קטגוריה מול שאר הקטגוריות -

Class	Accuracy (%)			
VoIP	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**
Video	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**
File Transfer	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**
Chat	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**
Browsing	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**90.6**	-	57.2
	VPN	-	**-**	-
	Tor	76.1	-	**90.6**

3. זיהוי אפליקציות ספציפיות –

VoIP דיוק של 99.7% בזיהוי בין 10 אפליקציות ווידאו שונות



4. זיהוי סוג ההצפנה –

דיוק כולל של 88.4% בזיהוי בין Tor, VPN , Non-VPN.

זיהוי גבוה של תעבורת Tor – דיוק של 97.7%

Problem	FlowPic Acc. (%)	Best Previous Result	Remark
Non-VPN Traffic Categorization	85.0	84.0 % Pr., Gil *et al.* [15]	Different categories. [15] used unbalanced dataset
VPN Traffic Categorization	98.4	98.6 % Acc., Wang *et al.* [7]	[7] Classify raw packets data. Not including browsing category
Tor Traffic Categorization	67.8	84.3 % Pr., Gil *et al.* [15]	Different categories. [15] used unbalanced dataset
Non-VPN Class vs. All	97.0 (Average)	No previous results	
VPN Class vs. All	99.7 (Average)	No previous results	
Tor Class vs. All	85.7 (Average)	No previous results	
Encryption Techniques	88.4	99. % Acc., Wang *et al.* [7]	[7] Classify raw packets data, not including Tor category
Applications Identification	99.7	93.9 % Acc., Yamansavascilar *et al.* [10]	Different classes

<u>התובנות העיקריות</u>:

1. למידת דפוסים כללים- השיטה לומדת את המאפיינים של קטגוריות תעבורה כמו וידאו או צ'אט, ולא רק של יישומים ספציפיים. זה מאפשר זיהוי מוצלח של יישומים חדשים שלא נראו במהלך אימון השיטה.

2. VPN משפיע על דפוסי התעבורה אבל עדיין ניתן לזהות בדיוק גבוה למרות ההצפנה שלו. ואילו Tor מקשה יותר על הזיהוי ומפחית את הדיוק של השיטה.

3. יעילות – השיטה לא משתמשת בתוכן החבילות עצמן ולכן פרטיות המשתמשים גדולה יותר וגם יש פחות צורך בשימוש בזיכרון. בנוסף השיטה משתמשת בחלק קטן מאוד מהתעבורה , ובכיוון אחד בלבד , וזה מאפשר לסווג את התעבורה בצורה מהירה ויעילה יותר.

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

<u>התרומה העיקרית של המאמר</u>:

המאמר מציע שיטה חדשה לזיהוי וסיווג מוקדם של תעבורה מוצפנת שמתמודדת עם האתגרים שנובעים מהמעבר לשימוש בפרוטוקול TLS1.3 והרחבתו עם ECO. בעבר, ניתן היה לזהות את סוג השירותים שאליהם משתמשים מתחברים בעזרת SNI, אבל עם הצפנתו במסגרת במסגרת ECO, המידע הזה כבר לא זמין, מה שגרם לכך ששיטות סיווג מסורתית הפכו לפחות יעילות. כדי להתמודד עם האתגר, המאמר מציג אלגוריתם חדש בשם hRFTC אשר מסוגל לסווג תעבורה מוצפנת גם כאשר חלק מהמידע הקריטי מוסתר. האלגוריתם משלב בין שני סוגים של מידע: נתונים גלויים מתוך שלב יצירת החיבור ב- Handshake TLS , כמו רשימת הצפנים הנתמכים ואורכי הודעות, יחד עם מאפיינים סטטיסטיים של זרימת הנתונים, כגון גודל חבילות, זמני ההגעה בין החבילות ודפוסים סטטיסטיים אחרים. אחת מההתאמות החשובות במודל היא ביכולת לנתח גם בתעבורה בפרוטוקול QUIC מה שמאפשר לזהות תעבורה באופן מדויק יותר בהשוואה לשיטות הישנות שהתמקדו בעיקר בתעבורת TLS-Over-TCP. הניסויים שבוצעו במסגרת במחקר הראו כי השיטות המסותיות, שהתבססו רק על מידע מתוך TLS, הצליחו להגיע לדיוק של 38.4% בלבד, משום שהן הסתמכו על מאפיינים שכיום מוצפנים על ידי ECH. אבל לעומת זאת, האלגוריתם החדש hRFTC השיג דיוק של 94.6% בזכות השילוב בין נתוני הTLS לבין ניתוח סטטיסטי של זרימת התעבורה עצמה. נוסף על כך, נמצא כי מודלים שעובדים היטב במדינה אחת עשויים לא להיות מדויקים במדינה אחרת, בגלל שמבנה הרשת משתנה מאזור גאוגרפי לאזור גאוגרפי, ולכן כדי להבטיח דיוק גבוה, יש צורך להתאים את האלגוריתם לסביבה ברשתית שבה הוא פועל . בסופו של דבר המאמר מציג פתרון חדשני ויעיל לסיווג מוקדם של תעבורה מוצפנת, גם כאשר מידע חיוני מוסתר בעקבות TLS1.3 ו- QUIK. השיטה החדשה מאפשרת ניתוח מהיר ומדויק, המותאם לסביבת הרשת המודרנית ובכך משפרת משמעותית את היכולות של מערכות ניהול רשתות האבטחה.

בס"ד

מאפייני תעבורה שנעשה בהם שימוש (שגרתיים וחדשים):

המאמר משתמש בשילוב של מאפיינים מבוססי חבילות מתוך תהליך לחיצת היד של TLS ומאפיינים סטטיסטיים מבוססי זרימה כדי לזהות את סוג התעבורה בשלב מוקדם.

המאפיינים המבוססים על חבילות כוללים מידע שמופיע בתוך הודעות ClientHello ו- ServerHello של TLS, כמו:

- גרסת TLS
- רשימת הצפנים (Cipher Suites)
- הרחבות TLS
- אלגוריתמים לחתימה דיגיטלית
- קבוצות קריפטוגרפיות נתמכות

המאפיינים הסטטיסטיים מבוססי זרימה כוללים:

- התפלגות גודל חבילות (כמה חבילות יש מכל גודל, מה הגודל הכי נפוץ ועוד)
- זמן הגעה בין חבילות (זמן ממוצע בין חבילה לחבילה, שונות בין הזמנים ועוד)
- כיוון החבילות – האם החבילה נשלחת מהמכשיר לרשת או מתקבלת מהרשת
- עשרת החבילות הראשונות בזרימה
- כמה פעמים חבילות נשלחו מחדש

בנוסף, המאמר מציג כמה חידושים שמייעלים את הזיהוי המוקדם.

הוא מציע אלגוריתם חדש בשם hRFTC, שמשלב בין מאפייני TLS למאפייני זרימה כדי לשפר את הדיוק בזיהוי התעבורה. בנוסף, במקום להסתמך על מספר קבוע של חבילות, המאמר מציע קריטריון חדש לבחירת חבילות, שבו נבדקות כל החבילות עד שמגיעה חבילת הנתונים הראשונה בתגובה (Downlink).

עוד חידוש הוא תמיכה משופרת בפרוטוקול QUIC, שמאפשרת סיווג גם כשאין לחיצת יד של TLS רגילה. במקום להסתמך על שדות TLS סטנדרטיים, האלגוריתם מתאים את עצמו לשדות הייחודיים של QUIC, מה שעוזר לסווג תעבורה גם כשהיא מוצפנת בשיטות חדשות.

החידושים האלו עוזרים לשפר את מהירות ודיוק הזיהוי, תוך כדי הפחתת התלות במספר גדול של חבילות כדי להגיע לתוצאה טובה.

ממצאים עיקריים ותובנות:

1. השפעת ההצפנה (ECH) על סיווג תעבורה

הוספת Encrypted ClientHello (ECH) בפרוטוקול TLS 1.3 מובילה לירידה דרמטית בדיוק הסיווג של אלגוריתמים מבוססי חבילות.

TABLE 10. Comparison of the packet-based classifiers on the subsets of the dataset.

Selected Traffic Subset	Avg TLS Distinction of the Traffic Subset	Macro F-score [%]		
		RB-RF	BGRUA	MATEC
Live Video	1	100.0	100.0	100.0
Short Buffered Video	0.99	99.2	81.5	82.3
Buffered Video	0.87	87.4	75.7	75.9
Buffered Audio	0.71	71.8	59.9	63.0
Vk	0.65	66.2	57.1	61.1
Google	0.58	55.5	55.0	54.7
Facebook	0.53	52.6	51.4	51.2
Yandex	0.49	40.2	39.8	38.1

בס"ד

הטבלה מראה כי כאשר ההבחנה בין שירותי TLS פוחתת כלומר, כאשר תצורת ה-TLS דומה בין שירותים שונים, דיוק הסיווג. יורד לכ-40% בלבד עבור שירותים המשתמשים בהגדרות TLS דומות.

2. השוואה בין אלגוריתמים מבוססי חבילות, זרימה והיברידיים

האלגוריתם hRFTC שהוצג במאמר מתעלה על כל האלגוריתמים הקיימים מבחינת דיוק סיווג, במיוחד על פני אלגוריתמים מבוססי חבילות.

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.

הטבלה מציגה את שיפור הביצועים macro F-score של hRFTC לעומת האלגוריתמים האחרים:

שיפור של 56.3% לעומת RB-RF

שיפור של 63.3% לעומת MATEC

שיפור של 59.6% לעומת BGRUA

3. חשיבות מאפייני זרימה לעומת מאפייניTLS

TABLE 12. hRFTC: the Gini-impurity-based feature importance normalized by the maximal observed value.

Rank	Feature	Impurity-based Feature Importance
1	CH Cipher Suites length	1.00
2	DL PSs Cumulative Sum	0.62
3	DL PSs Sorted Unique #1	0.50
4	UL PSs Std	0.46
5	UL PSs Cumulative Sum	0.45
6	UL PSs #2	0.44
7	DL IPTs Sum	0.43
8	DL PSs 25th percentile	0.43
9	DL PSs average	0.42
10	UL PSs 75th percentile	0.41
11	SH Cipher Suite	0.40
12	DL PSs Std	0.40
13	UL PSs Sorted Unique #2	0.40
14	UL PSs max	0.38
15	DL PSs 50th percentile	0.37
16	UL PSs average	0.36
17	DL PSs 75th percentile	0.33
18	UL IPTs Sum	0.28
19	UL PS #1 (CH length)	0.27
20	DL PS #2	0.26
21	UL IPTs min	0.26
22	UL PS 750-1000B freq	0.25
23	UL PSs Sorted Unique #3	0.25
24	CH Extensions Length	0.24
25	SH Extension Type #2	0.23
26	UL IPTs 25th percentile	0.23
27	UL IPTs 50th percentile	0.22
28	SH Extension Type #1	0.22
29	DL PS #3	0.21
30	UL IPTs max	0.2

הטבלה מציגה את 30 המאפיינים החשובים ביותר על פי מדד Gini-Impurity.

מאפייני זרימה כגון גדלי חבילות מצטברים וזמני הגעה בין חבילות נמצאו קריטיים, במיוחד עם השימוש ב-ECH, והם תורמים למעלה מ-50% לדיוק הסיווג.

מאפיינים מבוססי TLS עדיין מועילים, אך חשיבותם פוחתת משמעותית כאשר נעשה שימוש ב-ECH, שכן רובם הופכים ללא נגישים.

4. יכולת הכללה של האלגוריתם ושפעת הגיאוגרפיה

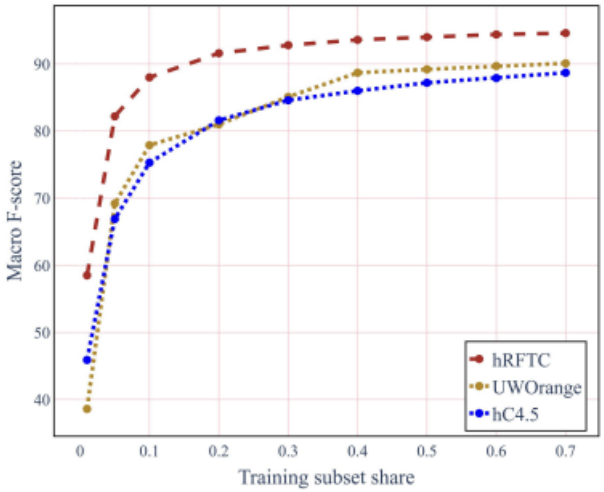


FIGURE 4. F-score depending on the training subset share.

האיור מראה כי כאשר גודל מערך האימון מופחת מ-70% ל-10% מהנתונים, הירידה בדיוק הסיווג היא רק 7%, מה שמעיד על יציבות גבוהה של המודל.

TABLE 14. TC quality depending on training locations.

Test Country	Share in Dataset	Training Country	Classifier Macro F-score [%]		
			hRFTC	hC4.5	UW
Germany	18.8%	Others	38.4	26.9	19.5
Kazakhstan	3.0%	Others	57.3	32.3	27.5
Russia	29.2%	Others	49.8	35.6	20.9
Spain	16.3%	Others	38.5	34.4	12.6
Turkey	25.2%	Others	35.1	26.0	16.4
USA	7.5%	Others	49.2	41.4	21.3

הטבלה מציגה השפעה גיאוגרפית משמעותית – כאשר האלגוריתם אומן במדינה אחת והופעל על תעבורה ממדינה אחרת, הביצועים ירדו משמעותית.

הסיבה האפשרית לכך: שינוי ברשתות ההפצה, שגורם להבדלים במאפייני הזרימה של החבילות (כגון גודל חבילה ממוצע ומרווחי זמן בין חבילות) בהתאם למיקום גיאוגרפי.

תובנות המחקר:

שיטות מבוססות TLS בלבד אינן מספיקות – כאשר נעשה שימוש ב-ECH, האלגוריתמים הישנים מאבדים את רוב יכולת הסיווג שלהם.

מאפייני זרימה הם קריטיים – שימוש בנתונים כמו גדלי חבילות וזמני הגעה משפר משמעותית את הסיווג גם כאשר TLS מוצפן.

האלגוריתם hRFTC משיג תוצאות יוצאות דופן – מציג דיוק גבוה משמעותית מכל שיטה אחרת, ומשפר את הסיווג ב-50% ומעלה בהשוואה לאלגוריתמים קיימים.

יש לאמן אלגוריתמים על פי אזור גיאוגרפי ספציפי – הבדלים בין מדינות משנים את מאפייני הזרימה ומובילים לירידה בדיוק אם המודל לא הותאם לאזור החדש.

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

התרומה העיקרית של המאמר:

המאמר מציג שיטה חדשנית ומדויקת (96.06% התאמה) לזיהוי מערכת ההפעלה, הדפדפן והאפליקציה של המשתמש עבור תקיפה *פסיבית , בהתבסס על ניתוח תעבורת HTTPS מוצפנת, תוך התייחסות למאפיינים נוספים שלא נשקלו במחקרים קודמים כחלק מהניתוח.

השיטה כוללת איסוף נתונים, ניתוח המאפיינים (נעשה סיווג למאפיינים שנלקחו בעבר בחשבון ומאפיינים נוספים בכדי לבדוק את השפעת 2 הקבוצות ופרט הקבוצה המחדשת במחקר), ולבסוף סיווג הנתונים באמצעות אלגוריתם למידת מכונה ושימוש בדפוסים חוזרים למערכת ההפעלה, הדפדפן והאפליקציה המתאימים.

*תקיפה פסיבית = תקיפה שאינה ישירה אל מול המותקף, אלא דרך תעבורת הרשת.

מאפייני תעבורה שנעשה בהם שימוש (שגרתיים וחדשים):

*פירוט המאפיינים מתואר בנספח א מטה.

מאפיינים בסיסיים שכבר נעשה בהם שימוש במחקרים קודמים:

1. מספר חבילות שנשלחו ונקלטו.
2. גודל של חבילות.
3. זמני הגעה בין חבילות.

מאפייני תעבורה חדשים שנלקחים בחשבון:

בס"ד

1. מאפייני TCP.
2. מאפייני SSL.
3. קצב העברת הנתונים.
4. Bursts = התפרצויות של חבילות (לדוג' בדפדפנים שנדרשת טעינה של הרבה מרכיבים יחד).
5. זמנים בין הגעת חבילות עבור שיאי התעבורה.

<u>ממצאים עיקריים ותובנות:</u>

1. תרומת המאפיינים המחדשים:

ניתן לראות שהמאפיינים שאין בהם חידוש לבדם, סייעו במחקר זה לאיתור מערכת ההפעלה, הדפדפן והאפליקציה יחד בדיוק של 93.52% . יחד עם הנתונים החדשים האלגוריתם הגיע לדיוק של 96.06%. בנוסף לכך עבור סיווג מערכת ההפעלה האלגוריתם הגיע לדיוק של 100%, אפילו בהתבסס על המאפיינים החדשים בלבד. בסיווג הדפדפן והאפליקציה – לנתונים החדשים יש תרומה בסיווגם יחד עם מאפייני הבסיס וכן הקושי העיקרי הוא בזיהוי האפליקציה.

(a) Tuple Accuracy Results (b) OS Accuracy Results (c) Browser Accuracy Results (d) Application Accuracy Results

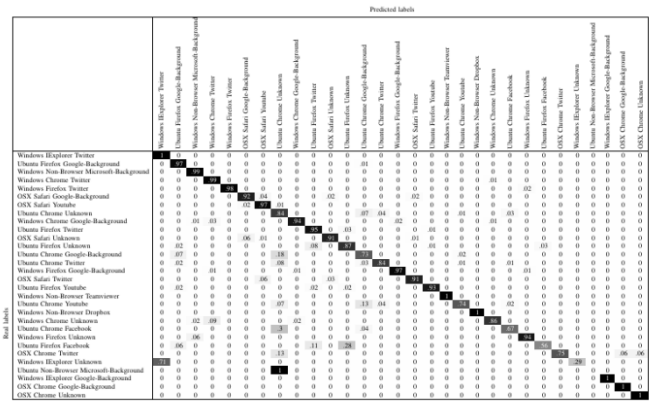
2. דיוק האלגוריתם:

דיוק הסיווג עולה מתוצאות של מעל 20,000 ניסויים ומוצג במאמר במטריצות בלבול כך שהשורות מייצגות את הערכים האמיתיים והעמודות מייצגות את הערכים הצפויים. במטריצה זו ערכים גבוהים באלכסון מעידים על דיוק גבוה ואילו ערכים גבוהים מחוץ לאלכסון מעידים על זיהוי שגוי של קטגוריה מסוימת.

בס"ד

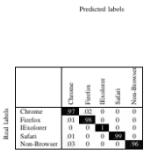
בהתאם לשיטה שהמטריצות מוצגות, ניתן לראות בהסתכל על כל המרכיבים יחדיו: מערכת ההפעלה, הדפדפן והאפליקציה שהסיווג מדויק באופן יחסי אך לא תמיד.
בנוסף לכך, מערכת ההפעלה מזוהה בדיוק מוחלט, הדפדפן והאפליקציה פחות. באפליקציה רוב הבלבול נובע בשל זיהוי שגוי של אפליקציית Facebook.

חשוב לציין שחוסר התאמה בסיווג לא בהכרח נובע מזיהוי שגוי אלא יכול לנבוע בשל חוסר וודאות לגבי הסיווג (2 פלטפורמות קרובות, היעדר פלטפורמה ממאגר הנתונים ועוד).

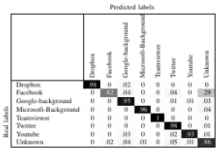


(a) Tuple Confusion Matrix

(b) OS Confusion Matrix



(c) Browser Confusion Matrix



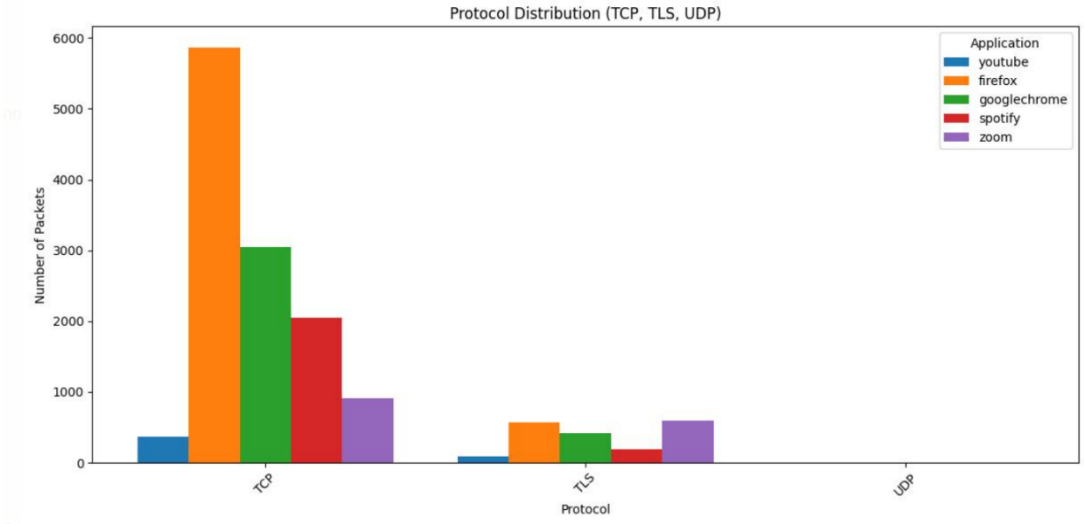
(d) Application Confusion Matrix

ובשורה התחתונה – הרגלי השימוש של המשתמש בפלטפורמות השונות אכן משפיעים על מאפייני התעבורה המוצפנת כך שניתן לזהות אותם.

– Part 3

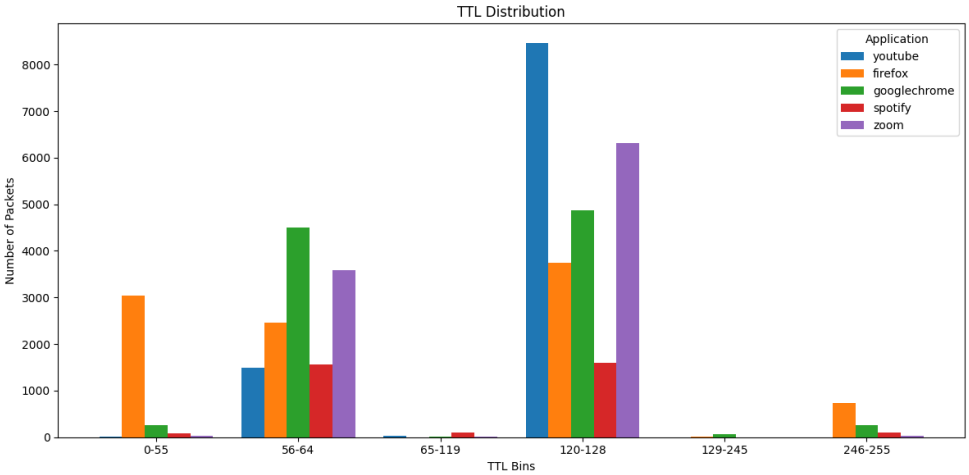
ניתוח גרפים והשוואות:

A) Type of protocol



ניתן לראות עבור ההקלטות שביצענו שכלל האפליקציות משתמשות בפרוטוקול TCP כפרוטוקול עיקרי וכן עבור כולם במידה כזו או אחרת יש שימוש בפרוטוקול TLS להצפנת המידע.
*הערה – יכול להיות שהאפליקציות כלל לא עשו שימוש בפרוטוקול UDP או לחילופין קבצי CSV אינם כוללים את המידע עבור פרוטוקול זה.

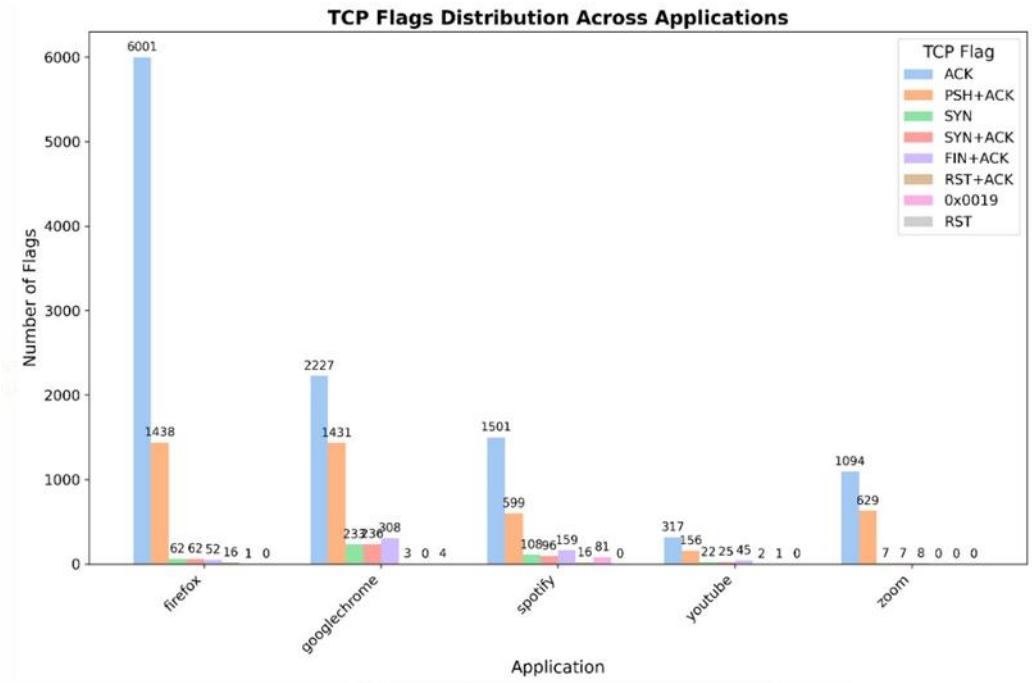
B) IP header fields



בגרף רואים טווחים של ערכי TTL סופיים (אשר קבענו עם השערה שערכי ה-TTL ההתחלתיים היו 64,128 או 255.) כלומר, כמה היה ה־ TTL כשהחבילה הגיעה ליעד. כאשר ה- TTL גבוה (למשל 246–255), זה עשוי להעיד שלא היו הרבה קפיצות בדרך והערך ההתחלתי היה כנראה 255, ניתן לראות כמות נמוכה של חבילות סביב טווח זה . כאשר ה־ TTL נמוך (למשל 0–55), ייתכן שהיו יותר קפיצות בדרך (לפחות 10 עבור ערך התחלתי של 64 או אפילו גדול יותר), וכן ניתן לראות מעט מאוד חבילות בטווחים אלה עבור כלל האפליקציות חוץ מfirefox. נשים לב כי רוב החבילות נמצאות סביב הטווחים של TTL 56-64 ו- 120-128 המעיד על כך שרוב החבילות ביצעו קפיצות של פחות מ10 ברשת בהנחה שהערכים ההתחלתיים הם 64 ו128 בהתאמה. עבור שאר הטווחים ניתן לראות כי בוצעו כנראה יותר מ10 קפיצות ברשת עם ערכים התחלתיים של 128 ו255 אך מופיעות כמות מעטה מאוד של חבילות.

C) TCP header fields

בחרנו לבחון את שכיחות סוגי הדגלים השונים עבור כל אפליקציה.



1. ACK – ניתן לראות שבכל האפליקציות יש שכיחות של דגל זה בעיקר, ובמיוחד בדפדפנים. השכיחות בדפדפנים מעידה על כך שהם מתבססים על אישור קבלת החבילות כחלק מהמנגנון המהימן שמבטיח את הגעת החבילות בסדר הנדרש וניתן לראות ש- firefox מתבסס על מנגנון זה אפילו יותר. לעומת זאת ביתר האפליקציות השכיחות נמוכה יותר כיוון שבסטרימינג יש פחות תלות באישור מידי.
2. PSH+ACK – גם פה, רוב מופעי הדגל הם בדפדפנים וזה מעיד על שליחה ישירה של המידע ללא המתנה. לעומת זאת, עבור YouTube ו – Spotify יש פחות שימוש בדגל זה כיוון שסטרימינג לא דורש תגובתיות מיידית ונעשה שימוש ב- buffering.
3. דגלי ACK ו-PSH+ACK עבור אפליקציית ה- zoom נמצאים בשכיחות בינונית בשל עיקרון השירות של האפליקציה ורצון לשמור על איזון בין מהירות ליציבות. האישור הפחות תלותי באישורים מעיד על כך שניתן לבצע שידור חי תקין ולצד כך גם להימנע מאובדן חבילות.
4. SYN&SYN+ACK -

מופיע בשכיחות הכי גבוהה ב- chrome ויותר מ- firefox, מה שיכול להצביע על פתיחת מספר חיבורים במקביל לטעינת דפים עם הרבה תוכן. מנגד ניתן לראות שכמעט ואין שימוש בדגלים אלה באפליקציות YouTube ו - zoom, מה שיכול לסמן על חיבור אחד ארוך ומתמשך כך שהשידור רציף.

בנוסף לכך ניתן לבחון שלדגלים אלו יש שכיחות בינונית עבור Spotify וניתן ללמוד מכך ששירותי אודיו דורשים יותר בתכיפות תחזוקה של פתיחה וסגירת חיבורים.

5. FIN+ACK –

מופיע בשכיחות הכי גבוהה ב- chrome ובהתאמה לפתיחת חיבורים מרובה (4), ב- firefox השכיחות נמוכה יותר וזה יכול להעיד על כך שאפליקציה זו נוטה להשאיר חיבורים לזמן רב יותר. השכיחות עבור chrome מעידה על החשיבות בסגירה מסודרת של החיבור כחלק ממנגנון האפליקציה.

מנגד ניתן לראות שהשכיחות עבור יתר האפליקציות תואמת לשכיחות פתיחת החיבורים (4).

- RST+ACK

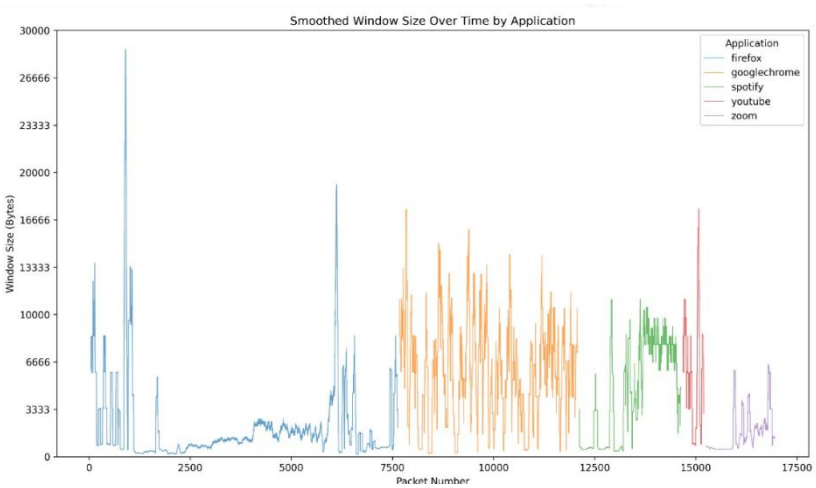
ניתן לראות שהוא שכיח באפליקציית Spotify (16) ואינו שכיח כלל ביתר האפליקציות. שימושו באפליקציית Spotify מלמד שכחלק ממנגנון האפליקציה מתבצע איפוס של חיבורים בעייתיים ומאפשר קבלת אישור על סיום החיבור.

6. 0X0019 –

ניתן לראות שימוש בדגל מותאם אישית באופן ניכר מהיתר עבור Spotify וזה מעיד שהוא משתמש בפרוטוקול מותאם אישית לעומת שאר האפליקציות. היעדר הדגל עשוי להעיד על שימוש בפרוטוקולים שגרתיים בלבד.

7. RST הנמוך או האפסי מעיד על כך שבכלל האפליקציות הנבדקות החיבור יציב.

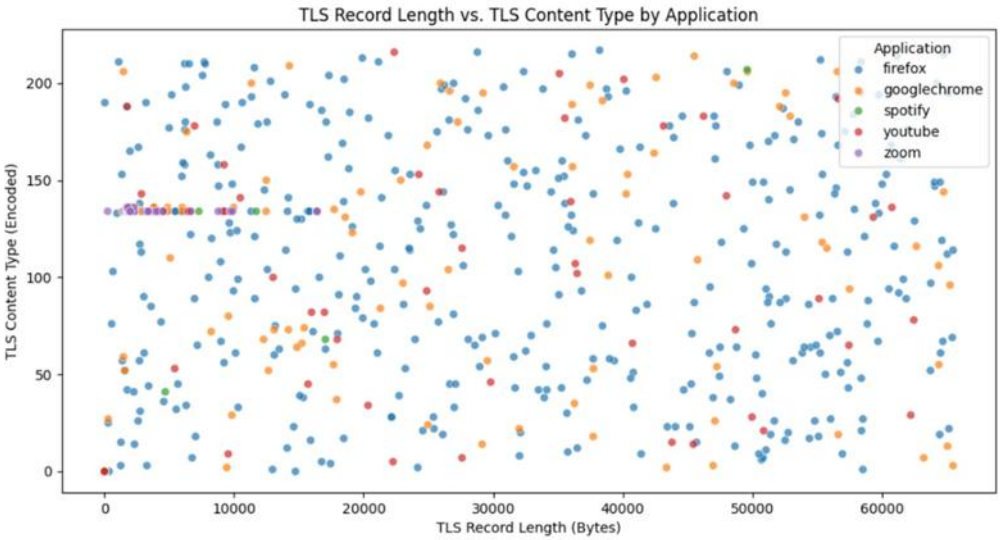
בנוסף לכך, בחרנו לבחון את גודל החלון כחלק ממנגנון מהימנות TCP תוך הצגת מגמת השינוי של גודל החלון לאורך זמן (ההקלטה).



1. ניתן לראות שהמגמה עבור firefox משתנה בתנודתיות גבוהה יותר לעומת chrome שהמגמה יותר מתונה. הבדל זה יכול להעיד שהתאמת גודל החלון עבור firebox נעשתה באמצעות שינויים קיצוניים של הגדלת החלון מה שהוביל ל"כישלון" ואילו ההתאמה עבור chrome נעשית בצורה יותר מבוקרת.
2. ניתן לראות שהמגמה עבור Spotify ו- YouTube יציבה יחסית לאורך ההקלטה ובנוסף לכך גודל החלון נשאר בערכים הנמוכים. לעומת זאת עבור zoom המגמה משתנה בצורה יותר קיצונית וזה בא בהלימה עם כך שאפליקציה זו תלויה ברוחב פס מספק בכדי לאפשר וידאו בשידור חי.
היציבות ב- Spotify תואמת להעברת הנתונים באופן רציף וכן חלון הזזה קטן עבור YouTube תואם לכך שלאפליקציה יש buffering המאפשר את השידור ברציפות.

D) TLS header fields

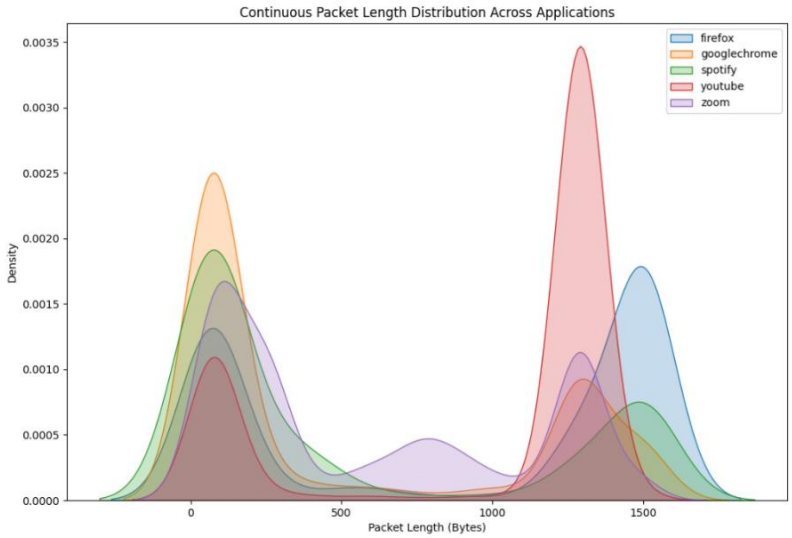
בחנו את שדות הכותרת בפרוטוקול TLS המוצפן שניתחנו בעזרת קבצי המפתחות, תוך ניתוח אורך המידע שמועבר בפרוטוקול TLS ביחס לסוג הדאטה המועבר.



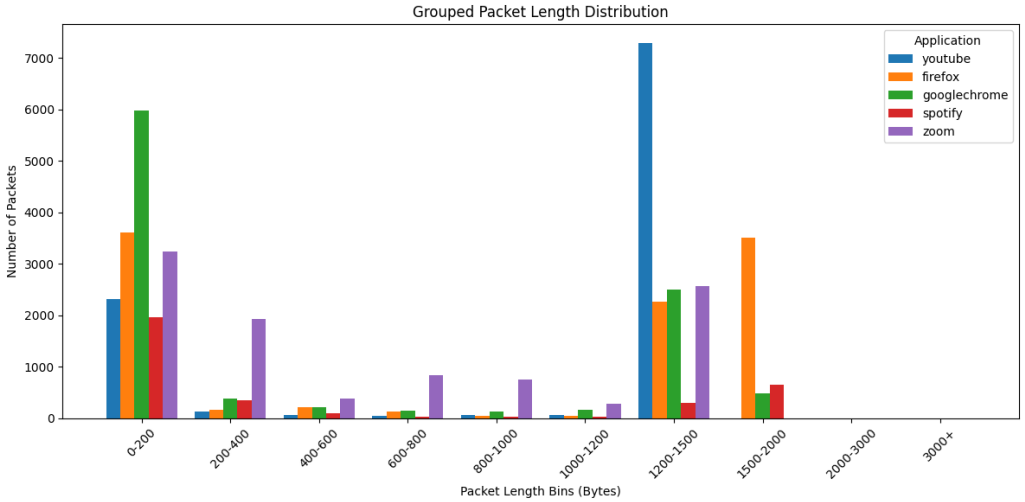
1. בהתאם לפיזור הנקודות ניתן לראות שהדפדפנים מציגים פיזור הכי רחב וזה מעיד על גיוון בדאטה המועבר בתעבורה (בקשות HTTP, טעינת דפים וכו'). מנגד פיזור האפליקציות YouTube כמעט ולא ניכר וזה תואם לכך ששירותי אפליקציות אלה מעבירות מידע במקטעים קטנים וקצרים.
2. בהתאם לפיזור שתיארנו, דרך הדפדפנים מועברת כמות הנתונים הגדולה ביותר כחלק מפרוטוקול TLS ובפרט בfirebox יש העברת כמות דאטה גדולה משל אפליקציית ה – chrome.

E) Packet sizes

כדי לבחון את גדלי החבילות באפליקציות השונות ראינו לנכון להציג את ההתפלגות גודלי החבילות לאורך זמן (ההקלטה) כך שיהיה ניתן לנתח גם את המגמה, שיאים ועוד.



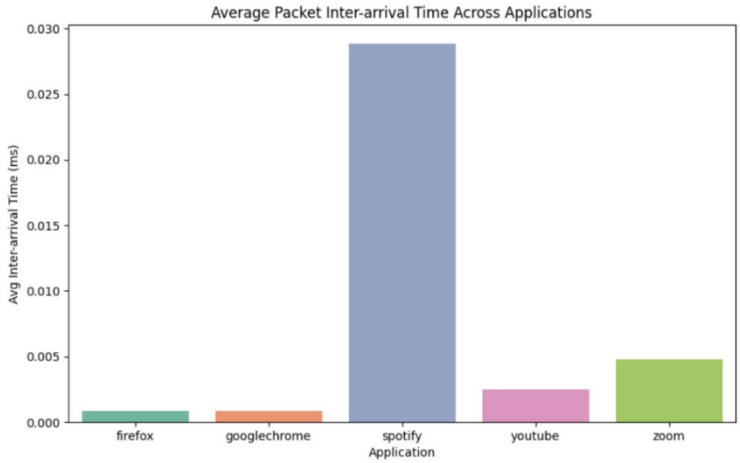
ניתן לראות שבהקלטה של zoom בשונה משאר האפליקציות יש מגמת שינוי בגודל החבילות לאורך כל ההקלטה, כלומר שונות גבוהה יותר בגודל החבילות. הסיבה לכך נובעת מהעובדה שאפליקציה זו כוללת מספר סוגי תעבורה (וידאו, אודיו ושיתוף מסך).
לעומת זאת עבור YouTube ניתן לראות שגודל החבילות עקבי וממוקד סביב טווח גדלים מסוים ובהתאמה לכך שאפליקציה זו מבוססת על העברת נתונים בקצב אחיד. בנוסף לכך עבור הדפדפנים המגמה מעידה על שליחת חבילות קטנות.
כמו כן, עבור Spotify ניתן לראות פחות עקביות, ככל הנראה כיוון שהאודיו מועבר בדינאמיות ובהתאם לאיכות ורוחב הפס הזמין.



הגרף מציג את התפלגות גדלי החבילות (בבייטים) בטווחים שונים של ההקלטות השונות.
ניתן לראות ש־YouTube בולטת בכמות גבוהה של חבילות בטווח 1200–1000בייט, מה שמעיד על הזרמת וידאו בחתיכות (chunks) יחסית גדולות, זאת מכיוון שיוטיוב שולחת ומקבלת תוכן וידאו ברזולוציות שונות כגון HD, 4K . Chrome ו־Firefox מראות התפלגות רחבה יותר בין הטווחים השונים, זאת מפני הגיוון הרב בסוגי התכנים כגון HTML , תמונות, בקשות HTTP או HTTPS וכו'.
Spotify נוטה למספר קטן יותר של חבילות גדולות, כנראה מפני שהזרמת אודיו דורשת פחות נתונים כבדים לעומת וידאו. Zoom מציגה גם היא התפלגות רחבה בין הטווחים, אך לא באותה עוצמה של YouTube .
לסיכום, אפשר ללמוד שהאפליקציות שמזרימות וידאו כבד (YouTube) יוצרות יותר חבילות גדולות, ואפליקציות מבוססות טקסט או אודיו כמו Firefox, Chrome, Spotify מציגות פיזור רחב או חבילות קטנות יותר, בהתאם לאופי התעבורה.

F) Packet inter-arrivals

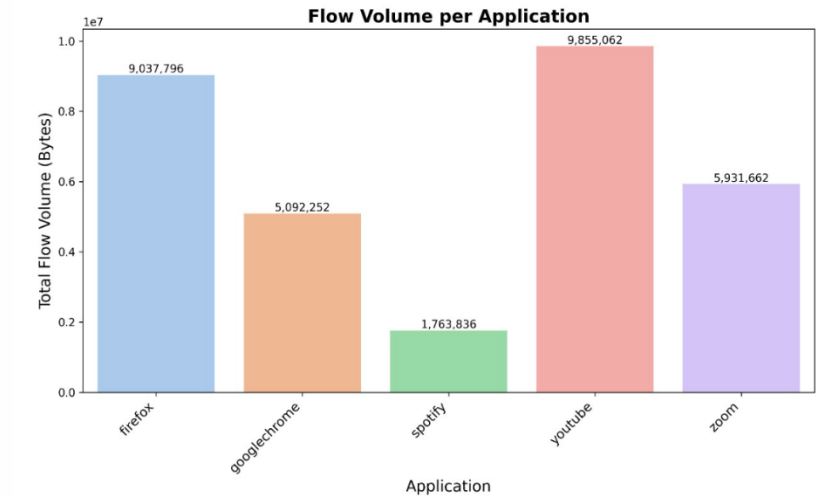
בהתאם לזמני הגעת החבילות וחישוב הזמנים בין כל 2 חבילות עבור כל אפליקציה, בחרנו להציג את ממוצע זמן ההמתנה בכל ערוץ



1. ניתן לראות ש- Spotify בעלת ממוצע הגבוה ביותר של זמני המתנה בין הגעת החבילות באופן משמעותי, כלומר התעבורה פחות רציפה מיתר האפליקציות והסיבה לכך שבאפליקציה זו יש שמירה בbuffering ע"מ לאפשר השמעה רציפה של האודיו. לעומת זאת, האפליקציות YouTube ו- zoom מספקות זרימה רציפה של חבילות (Streaming) כדי לשמור על איכות השירות. זו עשויה להיות הסיבה לכך שאפליקציות אלה מספקות זמני המתנה קצרים יותר ובאמצעות כך יכולים לספק העברה מהירה של הנתונים.
2. שירותי הדפדפנים – chrome ו – firefox משתמשים בפרוטוקול TCP אשר מעביר את החבילות ברציפות תוך שמירה על מנגנון הסדר ואישור ההגעה שלמדנו. לאור מנגנון מהימן זה העיכובים בהגעת החבילות מצטמצמים עבור הדפדפנים ביחס ליתר האפליקציות ובנוסף לכך הדפדפנים בשונה מיתר האפליקציות לא נדרשים בהעברת אודיו/וידאו.

G) Flow volume

בחינת ממוצע נפח התעבורה עבור האפליקציות השונות באמצעות שימוש בנתון של flow hash (יכול להיות לאפליקציה אחת מספר מזהים עבור זרימות שונות).



1. ניתן לראות שאפליקציות youtube ו- firefox משתמשות בנפח תעבורה הגבוה ביותר מבין האפליקציות, כנראה בשל הצורך בהעברת קבצי וידאו כבדים וכן עמודי אינטרנט מורכבים. לעומת firefox ניתן להבחין שהדפדפן chrome צורך נפח תעבורה קטן יותר באופן משמעותי וזה עשוי להעיד על מנגנון מתקדם יותר באפליקציה זו להעברת נתונים "כבדים".
2. ניתן לראות שגם אפליקציית zoom צריכה נפח תעבורה גדול יחסית בהתאם לצורך לשדר שידור חי ובו זמנית ביצוע הצפנה.
3. מנגד לכל האפליקציות שפירטנו לעיל, Spotify צורכת נפח הכי קטן בהתאם לשימוש ב – buffering והעברת נתונים מסוג אודיו שצורכים פחות מווידאו לדוגמא.

הדמיית התוקף:

כאשר התוקף מנסה לזהות איזה אפליקציות המשתמש הפעיל, הוא יכול להשתמש בניתוח דפוסי התעבורה בהסתמכות על הנתונים שזמינים לו. קיימות 2 אפשרויות:

1. התוקף יודע את גודל החבילות , חותמת הזמן שלהן ואת ה- Hush-tuple-4 : במקרה כזה- לתוקף יש מידע על הכתובות ברמת הזרימה. כלומר, הוא יכול להצליב את כתובות ה- IP עם מסדי נתונים של שירותים ידועים וכך לזהות. בנוסף, התוקף יכול לנתח את הפורטים כדי לשייך את החיבור לפרוטוקול מסוים. לדוגמה, שימוש בפורט 443 מעיד על תעבורה מוצפנת, כי הוא בעצם שימוש ב TLS. בעוד ששימוש בפורטים ייחודיים יכול לרמוז על שירותים מסוימים אחרים.

לדוגמא , אנחנו הקלטנו הפעלה של google meet ולאחר השוואה והתבוננות בגרפים הקיימים לנו מהקלטות קודמות ניתן לזהות בדפוסים דומים, ולשייך את התעבורה כתעבורה של שיחת וידאו. למשל, זמני ההגעה בין החבילות קצרים מאוד (בערך 0.0015 שניות), מה שמעיד על זרימת נתונים רציפה , שזה סימן לשיחת וידאו . בנוסף , התפלגות גודל החבילות מציגה שילוב של חבילות קטנות (בקרת חיבור) וחבילות גדולות (וידאו), תבנית שמופיעה גם בזום, ומכאן ניתן ללמוד שבדומה לזום , גם פה זה תעבורה רשת שקשורה לשיחת וידאו. כלומר בעצם , בהשוואה לנתונים שכבר ניתחנו על Zoom, תוקף יכול לזהות דפוס תעבורה דומה מאוד ולהסיק שמדובר בשירות שיחות וידאו.

2. התוקף יודע רק את גודל החבילה ואת חותמת הזמן שלה:

במקרה כזה , היכולת של התוקף לזהות את האפליקציה היא פחות מדויקת ,אבל הוא עדיין יכול להסיק מסקנות לגבי סוג השירות באופן הבא:

אם זמני ההגעה בין החבילות נמוכים מאוד והחבילות מגיעות ברציפות, אפשר להניח שמדובר בשיחת וידאו, כיוון ששם החבילות נשלחות בתדירות קבועה כדי לשמור על איכות השידור. בנוסף , אם קיימות חבילות קטנות לצד חבילות גדולות לסירוגין , ניתן להסיק שיש שימוש בשיחת וידאו כיוון שזהו דפוס אופייני עבור שירות כזה. כלומר למעשה , התוקף לא יכול להגדיר במדויק את סוג השירות , אלא לשער אותו בעזרת המידע הנתון לו.

מסקנה: אם לתוקף יש את כתובות הIP והפורטים , הוא יכול לזהות את השירות בדיוק גבוה. כיוון שכתובות הip יכולות להעיד על השרתים של השירות, ואילו הפורטים יכולים לתת כיוון על הפרוטוקולים שבשימוש.

לעומת זאת , אם התוקף יודע רק את גודל החבילות וזמן ההגעה שלהן, הוא יתקשה לזהות במדויק. הוא עדיין יוכל לזהות את סוג השירות, אבל לא בהכרח את האפליקציה המדויקת.

על מנת לטשטש את הנתונים ולהקשות על זיהוי האפליקציות שבהן נעשה שימוש, ניתן להשתמש בכמה טכניקות להגנה על תעבורת הרשת. כמו למשל שימוש ב VPN או ב- TOR שמסתירים את כתובת הIP של המשתמש , כך שהתוקף לא יוכל לקשר את הזרימה לשרת מסוים או למשתמש ספציפי.

בס"ד

נספח א:

מאפיינים בסיסיים שכבר נעשה בהם שימוש במחקרים קודמים:

**Forward = מהלקוח לשרת, Backward = מהשרת ללקוח. בניתוח השיטה שהוצעה במאמר בחנו את העברת המידע לשני הצדדים.*

4. **מספר חבילות שנשלחו ונקלטו:**
- Forward/Backward packets – מספר החבילות שנשלחו בבדיקה אחת.
- Mean Forward/Backward packets – מספר החבילות הממוצע שנשלח עבור מספר בדיקות.
- STD forward/Backward packets – סטיית התקן של מספר החבילות שנשלחו עבור מספר בדיקות.
- Total packets – סה"כ מספר החבילות שנשלחו בשני הכיוונים (Forward and Backward).

5. **גודל של חבילות:**
- Forward/Backward total Bytes – מספר הבתים הכולל שנשלח (עבור כל החבילות).
- Minimum Forward/Backward packet – הגודל הקטן ביותר של חבילה שנשלחה.
- Maximum Forward/Backward packet – הגודל הגדול ביותר של חבילה שנשלחה.
- Minimum packet size – הגודל הקטן ביותר של חבילה שנשלחה מתוך החבילות שנשלחו בשני הכיוונים.
- Maximum packet size – הגודל הגדול ביותר של חבילה שנשלחה מתוך החבילות שנשלחו בשני הכיוונים.
- Mean packet size – הגודל הממוצע לחבילה שנשלחה מתוך החבילות שנשלחו בשני הכיוונים.
- Packet size variance – שונות של גודל החבילה מתוך החבילות שנשלחו בשני הכיוונים.

6. **זמני הגעה בין חבילות:**
- Min Forward/Backward inter arrival time difference – הזמן הקצר ביותר שעבר בין 2 חבילות רצופות.
- Max Forward/Backward inter arrival time difference – הזמן הארוך ביותר שעבר בין 2 חבילות רצופות.
- Mean Forward/Backward inter arrival time difference – הממוצע של הזמנים בין 2 חבילות רצופות.
- Mean forward TTL value – ערך ה- TTL הממוצע של חבילות שנשלחו Forward.

מאפייני תעבורה חדשים שנלקחים בחשבון:

6. **מאפייני TCP:**
- TCP initial window size – גודל חלון חיבור TCP בהתחלה.
- TCP window scaling factor – מקדם שינוי גודל החלון מעבר.
- Keep alive packets – מספר החבילות שנשלחו לצורך שמירה על חיבור פעיל.
- TCP Maximum Segment Size – גודל החבילה המרבי שניתן לשליחה בחיבור.

7. **מאפייני SSL:**
- SSL compression methods – מספר שיטות הדחיסה הנתמכות בפרוטוקול SSL.
- SSL extension count – מספר ההרחבות המשמשות בפרוטוקול SSL/TLS (יכולות לכלול מאפיינים נוספים בתהליך ההצפנה).
- SSL cipher method – מספר שיטות ההצפנה הנתמכות בפרוטוקול TLS/SSL.
- SSL session ID len – אורך המזהה אשר משמש לניהול חיבורי TLS.
- Forward SSL Version – גרסת TLS/SSL שנעשה בה שימוש בתעבורה Forward.

8. **קצב העברת הנתונים:**
- Forward/Backward peak MAX throughput – קצב העברת הנתונים המקסימלי לכל כיוון.
- Mean throughput of forward/backward peaks – הממוצע של שיאי קצב נתונים בתעבורה לכל כיוון.

- Forward/Backward min peak throughput – קצב העברת הנתונים המינימלי בתעבורה לכל כיוון.
- Forward/Backward STD peak throughput – סטיית התקן של שיאי קצב הנתונים בתעבורה לכל כיוון.

9. Bursts = התפרצויות של חבילות (לדוג' בדפדפנים שנדרשת טעינה של הרבה מרכיבים יחד):

- Forward/Backward number of bursts – מספר התפרצויות של חבילות לכל כיוון.
- Forward min peak throughput – קצב העברת הנתונים המינימלי שנמדד בהתפרצויות של חבילות בתעבורה Forward.

10. זמנים בין הגעת חבילות עבור שיאי התעבורה:

- Minimum backward/forward peak inter arrival time diff – הזמן המינימלי בתעבורה לכל כיוון.
- Maximum backward/forward peak inter arrival time diff – הזמן המקסימלי בתעבורה לכל כיוון.
- Mean backward/forward peak inter arrival time diff – ממוצע בין הגעת שיאי התעבורה לכל כיוון.
- STD backward/forward peak inter arrival time diff – סטיית התקן של הזמנים בתעבורה לכל כיוון.