חלק 1 - שאלות פתוחות:

שאלה 1

כאשר משתמש מדווח על איטיות בהעברת קבצים, ניתוח שכבת התעבורה יכול לסייע בזיהוי הגורמים האפשריים לבעיה.

הגורמים האפשריים הם:

1. **רוחב פס לא מנוצל באופן מלא –** ייתכן שחיבור הרשת לא מספק בפועל את המהירות הצפויה עקב מגבלות תשתית, כרטיסי רשת לא תואמים או עומסים על הרשת.

   פתרון: יש לבדוק את מהירות החיבור בפועל באמצעות כלי iPerf3 ולוודא שכרטיס הרשת תומך במהירות הנדרשת. אם הכרטיס מוגבל למהירות נמוכה מהמצופה, יש צורך בשדרוג.

   בנוסף, יש להשתמש בכבלים תואמים ולהתאים את גודל החלון בהתאם לשהיית רוחב הפס (כמות הנתונים שיכולה להיות "בתנועה" ברשת בכל רגע נתון, לפני שהמקבל מאשר את הקבלה (ACK)).

   לבסוף, מומלץ לבדוק עומסים ברשת והגבלות בנתב או בחומת האש שעלולות להשפיע על קצב ההעברה.

2. **אלגוריתם בקרת עומס של TCP –** פרוטוקול TCP משתמש באלגוריתמים שונים (כגון Cubic, Reno, BBR), שמשפיעים על קצב ההעברה, במיוחד ברשתות עם עיכובים גבוהים או אובדן חבילות.

   פתרון:

   יש לבדוק באיזה אלגוריתם בקרת עומס משתמשת מערכת ההפעלה ולבחור באלגוריתם המתאים.

3. **שימוש בחיבור TCP יחיד לעומת חיבורים מרובים –** TCP לרוב משתמש בחיבור אחד להעברת קובץ גדול, מה שעלול להגביל את המהירות בשל מגבלות חלון ה-TCP או עומסים ברשת.

פתרון: לבדוק אם ניתן להשתמש במספר חיבורים במקביל או שימוש בתוכנות התומכות בחיבורים מרובים.

4. **אובדן חבילות -** אובדן חבילות ושגיאות ברשת עלולים לגרום להאטה בהעברת קבצים, כיוון שהן גורמות לשידורים חוזרים ולהגדלת זמני המתנה.

פתרון: ניתן להשתמש בכלים כמו Wireshark לניתוח תעבורת הרשת וזיהוי בעיות כגון שידורים חוזרים, ACK כפולים וזמני המתנה גבוהים.

לסיכום, כדי לפתור בעיית איטיות בהעברת קבצים, יש לבדוק את רוחב הפס הזמין, תשתית החומרה, התאמת חלון ה-TCP, אלגוריתם בקרת העומס, ושיטת ההעברה. כלים כמו Wireshark לניתוח חבילות ו-iPerf3 למדידת קצב התעבורה יכולים לסייע בזיהוי צווארי בקבוק ולשפר את ביצועי ההעברה.[1]

[1] https://www.networkdatapedia.com/post/2019/03/19/three-reasons-your-file-transfers-are-slow

שאלה 2

מנגנון בקרת זרימה בTCP נועד להבטיח שהיעד המקבל יקבל כמות חבילות שהוא מסוגל לעבד ולא מעבר לכך. במקרה שיש עומס אצל המקבל שהוא היעד, הוא יעדכן את השולח שיש כרגע עומס ומודיע לשולח כמה מקום נשאר, כלומר יעדכן את הrecieve window.

מנגנון זה מבוסס על שימוש בsliding window שגודלו מוגדר על ידי היעד אשר מקבל את החבילות מהשולח.

כאשר השולח חזק ומהיר ואילו המקבל חלש ואיטי יותר לעומתו, בקרת הזרימה של TCP תשפיע על הביצועים באופן הבא:

1. **האטה בקצב שליחת החבילות מצד השולח:**
   כאשר השולח מעביר נתונים מהר יותר ממה שהמקבל יכול לעבד, ה- buffer של המקבל מתמלא, והוא חייב להאט או לעצור את השידור עד שיתפנה מקום. בעקבות זאת המקבל יקבע את ערך ה-recieve window להיות קטן, מה שיגרום לשולח להאט את קצב ההעברה.
   דבר זה יגרום לכך שהשולח ייאלץ לחכות לאישורים (ACKs) לפני שיוכל לשלוח עוד נתונים, מה שיגרום לירידה בביצועים.
   בפועל, השולח לא ינצל את מלוא יכולתו לשלוח נתונים במהירות גבוהה.
2. **עיכובים הנגרמים עקב המתנות לACK-ים:**
   כאשר הגענו לרף window size - בצד המקבל, נאמר שה-buffer התמלא. דבר זה ייגרם עקב כך שהמקבל (שהוא איטי יותר) לא יצליח לקבל את כל החבילות שנשלחות אליו בקצב של השולח (שהינו מהיר יותר).
   דבר זה יגרום לכך שהמקבל לא יקבל יותר את החבילות עד להגעת ACK חדש.
   עקב כך השולח נאלץ להמתין לקבלת ACK לפני שיוכל לשלוח מידע נוסף.
3. **סינדרום חלון טיפשי:**
   כאשר המקבל מגדיר את גודל החלון לערכים קטנים מאוד, השולח שולח חבילות קטנות ולא יעילות, מה שמעמיס על הרשת. הפתרון הוא שימוש באלגוריתם Nagle's Algorithm בצד השולח וב-Delayed ACKs בצד המקבל על מנת להפחית את העומס.[2]
4. **מנגנון בקרת עומס:** TCP משתמש במנגנון בקרת עומס כדי למנוע עומס יתר ברשת, כאשר השולח מהיר יותר מהמקבל, יגרום להפחתת קצב ההעברה. נשתמש באלגוריתמים BBR ו CUBIC שיכולים לשפר את ביצועי הרשת במצבים כאלה.

---

[2] https://en.wikipedia.org/wiki/Silly_window_syndrome

האלגוריתמים הנ"ל משתמשים במדידת רוחב הפס הזמין ובפונקציה בחזקה שלישית בהתאמה על מנת לנהל באופן יעיל את קצב שליחת נתונים בהתאם לתנאי הרשת.[3]

[3] https://arxiv.org/abs/2312.11790

שאלה 3

הניתוב (Routing) ממלא תפקיד קריטי ברשתות התקשורת, במיוחד כאשר קיימים מספר מסלולים אפשריים בין המקור ליעד. שכבת הרשת אחראית על קביעת הנתיב שבו תעבור חבילת המידע, תוך שימוש באלגוריתמים שונים כמו Link-State (המבוסס על אלגוריתם דייקסטרה) ו-Distance Vector (המבוסס על אלגוריתם בלמן-פורד).

ניתוב נכון משפיע ישירות על ביצועי הרשת בכך שהוא משפיע על מהירות התעבורה, זמן השהיה, אמינות וניצול משאבים. נתיב עמוס או ארוך יותר עלול לגרום להאטה ולהגדלת זמן ההשהיה, בעוד שנתיב בעל רוחב פס גבוה וללא עומסים יספק חוויית שימוש טובה יותר. בנוסף, גורמים כמו אובדן חבילות, רוחב פס, יציבות הקישור ומדיניות הניתוב משפיעים על הבחירה במסלול האופטימלי.

פרוטוקול OSPF משמש להעברת נתונים בין ראוטרים שונים הנמצאים באותה מערכת אוטונומית, בעוד ש- BGP אחראי על ניתוב בין ארגונים וספקי אינטרנט, ומאפשר ניתוב דינמי בהתאם למדיניות עסקית וטכנולוגית.
בחירה לא נכונה של נתיב עלולה להוביל לעומסים, לאובדן מידע ולניצול לא יעיל של הרשת, ולכן יש להפעיל אלגוריתמים חכמים המסוגלים להתאים את עצמם לשינויים בתנועה ברשת.

<u>גורמים שיש לקחת בחשבון בבחירת נתיב:</u>

-<u>זמן השהיה</u> - מסלול עם פחות נתבי ביניים יהיה מהיר יותר.
- <u>רוחב פס זמין</u> - נתיבים רחבים יותר עדיפים עבור תעבורה אינטנסיבית.
- <u>עמידות</u> - בחירה במסלול יציב יותר תמנע תקלות חוזרות.
-מדיניות - ב-BGP, ספקי אינטרנט יכולים לבחור נתיבים בהתאם למדיניות עסקית ולא רק לביצועים.
- <u>יתירות</u> - קיום מספר נתיבים בין המקור ליעד מאפשר יתירות, כך שבמקרה של כשל באחד הנתיבים, התעבורה יכולה לעבור לנתיב חלופי, מה שמגביר את אמינות הרשת.
- <u>אבטחת מידע</u> - בחירת נתיב יכולה להשפיע על רמת האבטחה של התעבורה. נתיבים מסוימים עשויים להיות פגיעים יותר למתקפות, ולכן יש לשקול את ההיבט הזה בעת קביעת מסלול התעבורה.
- <u>עלות</u> - פרוטוקולי ניתוב מסוימים מתחשבים בעלות השימוש בנתיב, כמו עלויות כספיות, שימוש באנרגיה או משאבים אחרים. בחירת נתיב בעלות נמוכה יכולה להיות עדיפה במצבים מסוימים.

שאלה 4

MPTCP (Multipath TCP) הוא הרחבה של פרוטוקול TCP המאפשר שימוש במספר נתיבים במקביל.
בעוד ש-TCP רגיל משתמש בנתיב יחיד להעברת נתונים, MPTCP מאפשר פיצול של החיבור לכמה ערוצים במקביל, למשל ניצול בו-זמני של Wi-Fi ורשת סלולרית. היתרונות המרכזיים של MPTCP כוללים שיפור מהירות באמצעות שילוב רוחב הפס של מספר נתיבים, אמינות גבוהה הודות ליכולת להעביר נתונים לנתיב חלופי במקרה של כשל, ואיזון עומסים בין רשתות שונות.
הרחבה זו מספקת מספר יתרונות עיקריים:

1. **הגדלת קצב העברת הנתונים** – על ידי שילוב רוחב הפס ממספר ממשקים, כגון Wi-Fi, 4G ו-5G, MPTCP מאפשר שליחת נתונים דרך מספר נתיבים בו-זמנית. כך ניתן לנצל את רוחב הפס הכולל בצורה יעילה יותר ולשפר את קצב ההעברה בהשוואה ל-TCP המוכר.[4]

2. **אמינות ורב גוניות גבוהות יותר** – MPTCP מספק יתירות בכך שהוא מחלק את הנתונים בין מספר נתיבים. במקרה שאחד מהם חווה תקלה, הפרוטוקול מנתב את התנועה לנתיבים הנותרים, ומבטיח שהחיבור לא ינותק. תכונה זו מועילה במיוחד בסביבות ניידות שבהן תנאי הרשת משתנים במהירות.[5]

3. **תמיכה במעבר חלק בין רשתות** – במקרים שבהם מכשיר עובר בין רשתות, למשל מ-Wi-Fi לרשת סלולרית, MPTCP מאפשר מעבר חלק מבלי לקטוע את החיבור. הדבר מבטיח ביצועים יציבים של האפליקציות גם כאשר מתבצעת החלפה של ממשק הרשת.

4. **ניצול משאבים אופטימלי** – MPTCP מתאים את עצמו לתנאי הרשת המשתנים, על ידי מעקב אחר ביצועי כל נתיב ושימוש דינמי בהם בהתאם לצורך. כך ניתן לבצע איזון עומסים בצורה מיטבית ולהימנע מגודש ברשת.

לסיכום, MPTCP משפר את ביצועי הרשת בכך שמשלב יתרונות של מספר נתיבי תקשורת, מה שמוביל להגדלת מהירות ההעברה, אמינות גבוהה יותר, מעבר חלק בין רשתות וניצול משאבים מיטבי.

---

[4] https://www.researchgate.net/publication/364297669_Performance_Evaluation_of_MPTCP_on_Simultaneous_Use_of_5G_and_4G_Networks

[5] https://www.redhat.com/en/blog/understanding-multipath-tcp-networking-highway-future

שאלה 5

אובדן חבילות גבוה בין שני נתבים יכול לפגוע בביצועי הרשת באופן משמעותי.
ניתוח הגורמים הפוטנציאלים בשכבות הרשת (Network Layer) והתעבורה (Transport Layer) חיוני לפתרון הבעיה.

גורמים פוטנציאלים בשכבת הרשת (Network Layer):

1. עומס ברשת – תנועה מרובה ברשת יכולה להעמיס על קיבולת הנתבים ולגרום לאובדן חבילות. מצב זה קורה בדרך כלל כאשר רוחב הפס אינו מספיק או כאשר יש עליות פתאומיות ומשמעותיות בכמות הבקשות הנכנסות או תעבורה לאתר או לאפליקציה בפרק זמן קצר.

2. בעיות חומרה – רכיבים תקולים או ישנים, כמו נתבים או כרטיסי רשת, עלולים לגרום לאובדן חבילות. תחזוקה שוטפת ושדרוגים בזמן יכולים לסייע במניעת הבעיה.

3. שגיאות בהגדרות הרשת – הגדרות שגויות של נתבים, כגון חוסר התאמה במצבי דופלקס או גודל MTU לא תקין, עלולות לגרום לאובדן חבילות. חשוב לוודא שההגדרות אחידות בין כל רכיבי הרשת.

4. בעיות בשכבת התקשורת הפיזית – כבלים פגומים או חיבורים רופפים עלולים לגרום לשגיאות שיובילו לאובדן חבילות במהלך השידור. בדיקה ותחזוקה שוטפת של החיבורים הפיזיים היא נחוצה.

גורמים פוטנציאלים בשכבת התעבורה (Transport Layer):

1. הגדרות שגויות בפרוטוקול TCP – כמו גודל חלון או ערכי timeout לא תקינים עלולים להשפיע על בקרת זרימת הנתונים ולהוביל לאובדן חבילות.

2. מנגנוני בקרת עומסים – אלגוריתמים של בקרת העומסים של TCP עשויים להפחית את קצב השידור בתגובה לעומס שנתפס ברשת, מה שעלול לגרום לאובדן חבילות.

צעדים מומלצים לפתרון אובדן חבילות:

ניטור תעבורת הרשת: נשתמש בכלי ניטור רשת לזיהוי נקודות גודש ולניתוח דפוסי תעבורה. דבר זה יעזור לאתר את מקורות העומס ברשת.[6]

[6] https://obkio.com/blog/how-to-troubleshoot-packet-loss/

בדיקת רכיבי החומרה: בדיקת נתבים, מתגים וכבלים באופן קבוע, לאיתור סימני בלאי או נזק. החלפה או תיקון רכיבים תקולים יכולים להפחית אובדן חבילות.[7]

וידוא תקינות ההגדרות: מומלץ לוודא שכל רכיבי הרשת מוגדרים בצורה עקבית ואופטימלית, תוך התמקדות במצבי דופלקס, גדלי MTU ופרוטוקולי ניתוב.[8]

ניהול עומסי רשת: תעדוף תעבורה קריטית לניהול רוחב הפס בצורה יעילה להפחתת אובדן חבילות שנגרם עקב עומסים.

---

[7] https://www.dnsstuff.com/reduce-packet-loss

[8] https://documentation.meraki.com/General_Administration/Tools_and_Troubleshooting/Troubleshooting_Packet_Loss_Between_Devices

חלק 2 - מאמרים

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

שאלה 1:

התרומה המרכזית של המאמר היא הצגת שיטה חדשנית לסיווג תעבורת רשת מוצפנת על ידי המרת זרמי רשת לתמונות ("FlowPic") ושימוש ברשתות נוירונים קונבולוציוניות (CNNs) כדי לבצע את הסיווג.

היתרונות המרכזיים של השיטה:

אין צורך בהנדסת תכונות ידנית לסיווג התעבורה – בניגוד לשיטות מסורתיות שמתבססות על בחירה ועיבוד ידני של תכונות סטטיסטיות מתוך זרמי הנתונים, FlowPic מאפשר לרשת הנוירונים ללמוד באופן עצמאי מתוך ייצוג תמונתי של הזרם.

התאמה לתעבורה מוצפנת – שיטות מסורתיות רבות מתקשות בסיווג תעבורה מוצפנת. בפרט, שיטות מבוססות Deep Packet Inspection (DPI) מסווגות תעבורה על סמך תוכן החבילות (Payload-based classification). המאמר מציין כי שיטות אלו אינן מסוגלות להתמודד עם רוב תעבורת הרשת המודרנית בשל השימוש הנרחב בהצפנה. FlowPic עוקף את הבעיה בכך שהוא מתבסס על מאפיינים סטטיסטיים של הזרם, כמו גודל חבילות וזמני הגעה, מה שמאפשר לו לסווג תעבורה מוצפנת ללא תלות בתוכן הנתונים.

ייצוג תעבורה כשכבת תמונה – במקום לעבוד ישירות עם מספרים וסטטיסטיקות, FlowPic ממיר את הנתונים לייצוג חזותי. השיטה מחלקת כל זרם לחלונות זמן, וממירה כל חלון כזה למטריצה דו-ממדית של גודל חבילה מול זמן הגעתה, ושומרת אותה כתמונה לניתוח באמצעות רשת CNN.

שיפור דיוק הסיווג – המאמר מציין כי FlowPic מציג ביצועים גבוהים יותר משיטות מסורתיות בסיווג תעבורה מוצפנת, על ידי שימוש ברשת נוירונים קונבולוציונית לניתוח ייצוגים חזותיים של הזרם.

באמצעות המעבר מניתוח מבוסס תכונות מספריות לניתוח תמונתי, המאמר מציע גישה חדשה ואפקטיבית לסיווג תעבורת רשת מוצפנת.

שאלה 2

השיטה "FlowPic" שמוצגת במאמר מבוססת על שני מאפיינים עיקריים של תעבורת הרשת:

1. גודל החבילה – גודל כל חבילה בבייטים.
2. זמן הגעת החבילה – הזמן שבו החבילה מתקבלת, מנורמל בתוך חלון זמן מסוים.

מאפיינים אלו אינם חדשים – הם שימשו בעבר בשיטות סיווג מבוססות זרם. מה שחדשני בשיטה הזו הוא האופן בו הנתונים מוצגים ומעובדים.

<u>מה הופך את FlowPic לחדשני:</u>

ייצוג תעבורה כמטריצה דו-מימדית (FlowPic)

- במקום להתייחס לגודל החבילה וזמן ההגעה כמדדים סטטיסטיים נפרדים, FlowPic יוצר מהם מטריצה דו-מימדית:
  - ציר X = זמן הגעת החבילה (מנורמל לפי חלון זמן).
  - ציר Y = גודל החבילה (עד 1500 בייט).
  - כל תא בטבלה מייצג את כמות החבילות שהתקבלו בגודל מסוים בטווח זמן מסוים.

המרת המטריצה לתמונה

- המטריצה נשמרת כתמונה בגודל 1500×1500 פיקסלים בגווני אפור, שבה:
  - פיקסלים כהים יותר מציינים כמות חבילות גדולה יותר בגודל וזמן מסוים.
  - פיקסלים בהירים יותר מציינים כמות קטנה יותר של חבילות.
- הייצוג הוויזואלי מאפשר לרשת הנוירונים לזהות דפוסים בעצמה, במקום להשתמש בתכונות שנבחרות מראש.

ביטול הצורך בהנדסת תכונות ידנית

- שיטות מסורתיות לסיווג תעבורה דורשות בחירה ידנית של מאפיינים סטטיסטיים, כמו גודל חבילה ממוצע, זמני הגעה ממוצעים, והתפלגות החבילות.
- FlowPic מבטל את הצורך בכך, משום שהשיטה מאפשרת לרשת ה-CNN ללמוד את הדפוסים ישירות מהתמונה.

עמידות לתעבורה מוצפנת

- כיוון ש-FlowPic לא מסתמך על ניתוח תוכן החבילה, הוא יעיל גם עבור תעבורה מוצפנת.
- בניגוד לשיטות כמו Deep Packet Inspection (DPI), שאינן פועלות היטב תחת הצפנה, FlowPic מתבסס אך ורק על מאפיינים סטטיסטיים של הזרם.

שאלה 3:

תוצאות עיקריות של המאמר:

1. דיוק גבוה בסיווג תעבורה

- סיווג תעבורה לא מוצפנת (Non-VPN): 85.0% דיוק.
- סיווג תעבורה מוצפנת ב-VPN: 98.4% דיוק.
- סיווג תעבורה מוצפנת ב-Tor: 67.8% דיוק.

תובנות:

- FlowPic מציג ביצועים טובים מאוד בזיהוי וסיווג תעבורה, גם תחת הצפנה ב-VPN.
- תעבורת Tor קשה יותר לסיווג עקב טכניקות הסוואה כגון ריפוד נתונים ומולטיפלקסינג, הגורמות לכך שהדפוסים הנלמדים על ידי הרשת פחות ברורים.

2. זיהוי אפליקציות עם דיוק של 99.7%

- FlowPic נבדק על אפליקציות VoIP ווידאו.
- השיג 99.7% דיוק, ועקף מודלים קודמים של למידת מכונה.

תובנות:

- המודל מצליח להבחין בין אפליקציות שונות בהתבסס על גודל החבילות ותזמון הגעתן בלבד.
- בשונה משיטות מסורתיות, הוא אינו נדרש לבדוק את תוכן החבילות, מה שהופך אותו לידידותי לפרטיות.

3. סיווג תעבורה מוצפנת ללא אימון מוקדם על נתונים מוצפנים

- גם כאשר המודל אומן רק על נתוני Non-VPN, הוא הצליח לסווג תעבורה מוצפנת ב-VPN בדיוק של 99.4%.
- בסיווג תעבורת Tor, הדיוק היה עד 89%, למרות המורכבות הגבוהה יותר של המשימה.

תובנות:

- FlowPic עמיד בפני הצפנה ומסוגל לזהות תבניות תעבורה פנימיות שנותרות יציבות גם לאחר ההצפנה.

- הרשת הנוירונית לומדת את מאפייני התעבורה הבסיסיים, מה שמאפשר יכולת הכללה טובה יותר.

4. סיווג אפליקציות לא מוכרות

- המודל לא אומן על Facebook Video אך נבדק עליו לאחר מכן.
- הצליח לזהות אותו נכון כאפליקציית וידאו עם דיוק של 99.9%.

תובנות:

- FlowPic מצליח להכליל ולזהות אפליקציות חדשות, גם ללא אימון ישיר עליהן.
- הוא לומד התנהגות ברמת קטגוריה במקום להתמקד רק באפליקציות ספציפיות.

5. הבחנה בין קטגוריות תעבורה באמצעות CNN

- FlowPic הצליח לסווג VoIP, וידאו, העברת קבצים, צ'אט וגלישה.
- השיג דיוק גבוה גם תחת שיטות הצפנה שונות.

תובנות:

- השיטה מבוססת התמונות מאפשרת ל-CNN לחלץ דפוסים ייחודיים בתעבורה.
- בשונה משיטות סטטיסטיות, אין צורך בהנדסה ידנית של תכונות - הרשת לומדת אותן בעצמה.

תובנות מרכזיות

- FlowPic מציג תוצאות פורצות דרך בסיווג תעבורה מוצפנת ועולה על מודלים קיימים, במיוחד בתנאי הצפנה.
- הוא מצליח לסווג אפליקציות גם ללא אימון ישיר עליהן, מה שמעיד על יכולת הכללה טובה.
- השיטה עמידה לשינויים עתידיים ומתאימה לאבטחת רשתות ושמירה על פרטיות.

ניתוח המאמר :

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

שאלה 1:

התרומה המרכזית של המאמר היא הצגת שיטה חדשנית לניתוח תעבורת רשת מוצפנת (HTTPS) על מנת לזהות את מערכת ההפעלה, הדפדפן והיישום בהם המשתמש עושה שימוש, גם כאשר התוכן עצמו מוצפן באמצעות SSL/TLS.

במאמר זה החוקרים מציגים פיתוח סט של מאפיינים חדש לניתוח תעבורה מוצפנת:

החוקרים פיתחו סט מאפיינים חדשני, מעבר למאפיינים הבסיסיים שנמצאים בשימוש במודלים קודמים של סיווג תעבורה:

מאפייני SSL – סוגי הצפנה, מזהה ההפעלה של SSL, מספר הרחבות SSL.

מאפייני TCP – גודל חלון TCP, קצב שליחת הנתונים.

מאפייני "שיאים" (Peak Features) – ניתוח של קטעי תעבורה עם שליחת נתונים אינטנסיבית ולא אחידה (Bursty Traffic).

שאלה 2:

המאמר משתמש בשני סטים עיקריים של מאפיינים (Features) לניתוח וסיווג תעבורת רשת מוצפנת:

1. מאפיינים בסיסיים (Base Features) – מאפיינים שנמצאים בשימוש נפוץ בסיווג תעבורה.
2. מאפיינים חדשים (New Features) – מאפיינים חדשניים שהוצעו במאמר כדי לשפר את הדיוק.

מאפיינים בסיסיים (Base Features)

המאפיינים הבסיסיים הם מאפיינים סטנדרטיים לניתוח תעבורת רשת, שנמצאים בשימוש במודלים קיימים של סיווג תעבורה.
הם מתמקדים במדדים כלליים של זרימת הנתונים ברשת, כמו מספר החבילות, גודל החבילות, וזמני ההגעה ביניהן.

<u>דוגמאות למאפיינים הבסיסיים של תעבורת רשת בהם נעשה שימוש במאמר:</u>

- מספר החבילות שנשלחו קדימה (Forward Packets) ואחורה (Backward Packets).
- סך כל הנתונים שנשלחו בחבילות קדימה (Forward Total Bytes) ואחורה (Backward Total Bytes).
- הבדלים בזמני הגעה בין חבילות (Inter-Arrival Time Differences) – מינימום, מקסימום, ממוצע וסטיית תקן.
- מספר כולל של חבילות בהפעלה (Total Packets).
- גודל החבילה המינימלי והמקסימלי (Minimum / Maximum Packet Size).
- זמן ה- (Time To Live) TTL הממוצע של חבילות (Mean Forward TTL Value).

מאפיינים חדשים (New Features)

המאמר מציג סט חדש של מאפיינים שהשילוב שלהם לא היה בשימוש נרחב בעבר, אשר מבוססים על ניתוח מעמיק של פרוטוקולי SSL/TLS, TCP, והתנהגות דפדפנים.
המאפיינים החדשים מתמקדים בשלושה היבטים מרכזיים:

- מאפייני SSL (פרוטוקול האבטחה של HTTPS).
- מאפייני TCP (פרוטוקול התקשורת הבסיסי ברשת).
- התנהגות "מתפרצת" של דפדפנים (Bursty Behavior).

מאפייני SSL/TLS – פרוטוקולי אבטחה ברשת

הצפנת תעבורה- SSL/TLS הוא הפרוטוקול המשמש להצפנת תעבורת אינטרנט (כגון HTTPS), אך הוא עדיין משאיר מאפיינים גלויים ב-headers של החבילות.
"Handshake"- המאמר משתמש במאפיינים של תהליך ה-"Handshake" של SSL/TLS, שבו דפדפנים ושרתים מחליפים מידע לפני תחילת ההצפנה.

מאפייני רשת שנבדקו המופיעים ב-Headers של חבילות ה-TLS Handshake:

- הרחבות SSL ושיטות הצפנה – כיצד המכשיר בוחר להצפין את החיבור.
- אורך מזהה ההפעלה של SSL – משתנה בין דפדפנים שונים.
- גרסת ה-SSL בשימוש – יכולה לחשוף איזו מערכת הפעלה פועלת.

מאפייני TCP – ניהול חבילות נתונים ברשת

TCP מנהל את שליחת החבילות בין משתמשים לשרתים, ולכן ניתן להפיק ממנו מידע על סוג מערכת ההפעלה והדפדפן.

מאפייני רשת שנבדקו:

- גודל חלון ה-TCP – קובע כמה נתונים ניתן לשלוח לפני שמתקבל אישור.
- שינוי גודל החלון – מאפשר להרחיב את גודל החלון על פי הצורך.
- גודל מקטע מרבי (Maximum Segment Size – MSS) – יכול להיות שונה בין מערכות הפעלה.

מאפייני דפוסי תעבורה – ניתוח "התנהגות מתפרצת" של דפדפנים (Bursty Behavior)

דפדפנים לא שולחים נתונים בצורה רציפה, אלא ב"קפיצות" – יש שקט ברשת ואז שליחה אינטנסיבית של חבילות בבת אחת.
המאמר מנתח את הדפוסים האלו כדי להבין באיזה דפדפן מדובר.

מאפייני רשת שנבדקו:

- מספר "התפרצויות" של נתונים קדימה ואחורה (כלומר, כמה פעמים הדפדפן שולח נתונים בקצב גבוה ואז עוצר).
- מהירות השיא של הנתונים בתעבורה קדימה ואחורה – מודד כמה מהר נשלחו הנתונים במהלך ההתפרצות.
- הבדלים בזמני ההגעה בין התפרצויות – כמה זמן עבר בין שליחה אחת לשנייה.

מה הופך את השיטה שהוצעה במאמר לחדשנית:

- שילוב ראשון מסוגו של שלושה סוגי מאפיינים לזיהוי מערכת הפעלה, דפדפן ויישום

המאמר משלב שלושה סטים של מאפיינים שלא שולבו יחד בעבר לצורך סיווג תעבורה מוצפנת:

מאפייני SSL/TLS – ניתוח פרמטרים מתוך ה-Handshake של TLS, כמו מספר הרחבות ושיטות הצפנה.
מאפייני TCP – ניתוח מאפיינים כמו גודל חלון ה-TCP ושינוי הגודל שלו, שהיו בשימוש לצורכי ביצועים אך לא לזיהוי דפדפנים.
התנהגות מתפרצת (Bursty Behavior) של דפדפנים – זיהוי כיצד דפדפן שולח נתונים בקפיצות ולא בצורה אחידה.

החידוש:
- מחקרים קודמים חקרו חלק מהמאפיינים הללו, אך אף מחקר לא שילב את שלושתם יחד כדי לזהות דפדפן, מערכת הפעלה ויישום בתעבורה מוצפנת.
- השילוב הזה הוא שגורם לשיפור הדיוק ל-96.06%, בניגוד לשיטות קודמות.

- יישום חדש של ניתוח ההתנהגות המתפרצת (Bursty Behavior) של דפדפנים

לראשונה, המאמר משתמש בדפוסי ההתפרצות של שליחת הנתונים לזיהוי הדפדפן ומערכת ההפעלה.
המאמר מראה שכל דפדפן שולח נתונים בצורה ייחודית, וניתן להשתמש בכך לזיהוי.

החידוש:
בעבר נותחו הבדלים בזמני ההגעה בין חבילות (Inter-Arrival Time Differences), אך לא נותחו התפרצויות שלמות (Peaks) של תעבורה.

- שימוש ראשון במאפייני SSL/TLS לצורך זיהוי מערכת הפעלה ודפדפן

SSL/TLS משמשים בדרך כלל לאבטחת תעבורה, אך המאמר משתמש בהם לזיהוי משתמשים.

החידוש:
המאמר מראה שכל דפדפן ומערכת הפעלה תומכים במספר הרחבות SSL, שיטות הצפנה וגרסאות TLS שונות.

מחקרים קודמים ניתחו מאפייני SSL לצורכי אבטחה בלבד, ולא השתמשו בהם לסיווג דפדפנים ומערכות הפעלה.

- שיפור משמעותי בדיוק על ידי שילוב המאפיינים החדשים

השיטה החדשה משפרת את הדיוק בהשוואה לשימוש רק במאפיינים בסיסיים.

<u>החידוש:</u>

דיוק המודל עם מאפיינים בסיסיים בלבד = 93.51%.
דיוק המודל עם המאפיינים החדשים + בסיסיים = 96.06%.
שיפור של 3% בדיוק מראה שהמאפיינים החדשים מספקים מידע נוסף שלא היה נגיש בעבר.

- ההשפעות האפשריות של השיטה – פרטיות ואבטחת מידע

המאמר מראה שניתן לזהות מידע רגיש על משתמשים גם כאשר התעבורה מוצפנת.
כלומר, תוקפים יכולים להשתמש בזה כדי לעקוב אחר משתמשים ולבנות פרופילים אישיים, אפילו אם הם גולשים בצורה אנונימית.

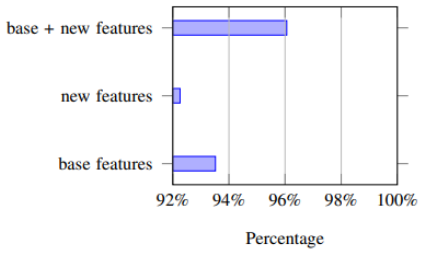
<u>החידוש:</u>
מחקרים קודמים הציעו ניתוחי תעבורה מוצפנת, אך המאמר מראה שניתן להשתמש בשיטה זו לזיהוי זהות משתמשים ולא רק לאפיון סוג התעבורה.

שאלה 3:

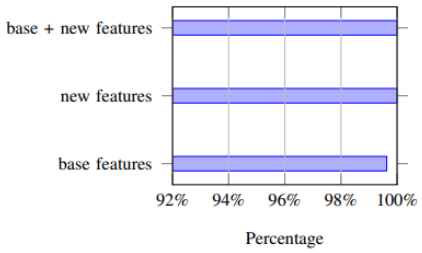
תוצאות עיקריות של המאמר:

תוצאה 1:

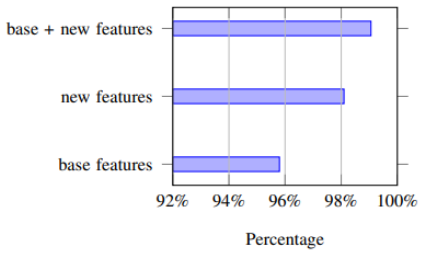
המאמר מציג את התוצאות בגרפים הבאים:



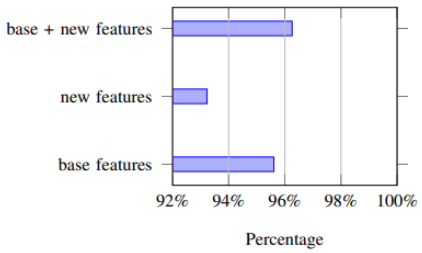
(a) Tuple Accuracy Results



(b) OS Accuracy Results



(c) Browser Accuracy Results



(d) Application Accuracy Results

*הגרפים נלקחו מתוך המאמר הנ"ל.

<u>תובנה 1 בהתאם לגרפים לעיל:</u>

ניתן לראות שבכל הגרפים המוצגים סיווג התעבורה המוצפנת על ידי שילוב בין המאפיינים הבסיסיים והמאפיינים החדשים נתן את רמת הדיוק הגבוהה ביותר.

בכך החוקרים מוכיחים את יעילות השיטה שהציעו.

כלומר בגרפים ניתן לראות ש:

-השימוש במאפיינים החדשים בלבד (New Features) לא שיפר משמעותית את הדיוק לעומת המאפיינים הבסיסיים ונתן תוצאות דומות.

-השימוש בשילוב של כל המאפיינים (Base + New Features) נתן את התוצאות הטובות ביותר.

תוצאה 2:

השיטה שבה ביצעו את דגימות התעבורה המוצפנת שנתפסה התנהלה באופן הבא:

החוקרים אספו כ-20,000 דגימות שונות במאגר נתונים ששימש למחקר ולאחר מכן סיווגו אותן על פי שלושה פרמטרים- מערכת הפעלה, דפדפן ויישום.

במאגר הנתונים שנאסף ישנם 30 שילובים אפשריים לשלושת הפרמטרים האלה (שלישייה זו נקראת תווית והיא הייתה מוצמדת לכל דגימה של תעבורה).

במטריצה מטה (אשר נלקחה מתוך המאמר) ניתן לראות את התוצאות של רמת הדיוק של סיווג התעבורה על ידי שימוש בשילוב של המאפיינים הבסיסיים והישנים ומראה עד כמה המודל מצליח לסווג נכון את הנתונים. כלומר, עד כמה רמת הדיוק של סיווג התעבורה של המודל תואם לדגימה שנלקחה בפועל(עם התייחסות לכל התוויות האפשריות).

Predicted labels

Real labels	Windows IExplorer 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 IExplorer Unknown	Ubuntu Non-Browser Microsoft-Background	Windows IExplorer Google-Background	OSX Chrome Google-Background	OSX Chrome Unknown
Windows IExplorer 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
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
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
OSX Safari Google-Background	0	0	0	0	0	.92	.04	0	0	0	.02	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
Ubuntu Chrome Unknown	0	0	0	0	0	0	0	.84	0	0	0	0	.07	.04	0	0	0	0	.01	0	0	.03	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	.02	0	0	0	0	0	.01	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	.01	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	.01	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	.01	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	0	.02	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
Windows Firefox Google-Background	0	0	0	.01	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
OSX Safari Twitter	0	0	0	0	0	0	.06	0	0	0	.03	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	.02	0	.02	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	1	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Youtube	0	0	0	0	0	0	0	.07	0	0	0	0	.13	.04	0	0	0	0	.74	0	0	.02	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	1	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	.86	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Facebook	0	0	0	0	0	0	0	.3	0	0	0	0	.04	0	0	0	0	0	0	0	0	.67	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
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
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	.75	0	0	0	.06	.06
Windows IExplorer 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	.29	0	0	0	0
Ubuntu Non-Browser Microsoft-Background	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
Windows IExplorer 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	1	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	1	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	1

(a) Tuple Confusion Matrix

*המטריצה נלקחה מתוך המאמר הנ"ל.

<u>תובנה 2 בהתאם למטריצה שלמעלה:</u>

השורות שבמטריצה מייצגות את הערכים האמיתיים (הפרמטרים הנכונים), ו-העמודות מייצגות את התחזיות של המודל, כלומר הערכים החזויים שלו.

ערכים גבוהים על האלכסון (בצבע שחור) מעידים על דיוק גבוה של המודל, כי זה אומר שהמודל חזה נכון את התווית.

ערכים מחוץ לאלכסון מייצגים טעויות, כלומר – מקרים שבהם המודל זיהה פרמטר לא נכון.
המודל טעה בעיקר בין פרמטרים דומים או פרמטרים שערכם הוא "Unknown", שלא ניתן היה לוודא אם הסיווג שלהם שגוי.

התוצאות מראות כי ניתן לזהות מערכת הפעלה, דפדפן ויישום בדיוק גבוה, גם כאשר התוכן מוצפן. הדבר מוכיח כי ניתוח תעבורה מוצפנת יכול לשמש לזיהוי משתמשים ללא גישה ישירה לתוכן הנתונים.

על פי הנתונים במאמר-
בזיהוי התווית (מערכת הפעלה, דפדפן, יישום), הוספת המאפיינים החדשים שיפרה את הדיוק מ-93.52% ל-96.06%.

ניתוח המאמר:

Early Traffic Classification With Encrypted ClientHello A Multi-Country Study

שאלה 1:

התרומה העיקרית של המאמר הינה פיתוח אלגוריתם סיווג תעבורה חדש בשם hRFTC (Hybrid Random Forest Traffic Classifier) המסוגל לזהות בצורה מדויקת סוגים שונים של תעבורה מוצפנת, אפילו כאשר נעשה שימוש בהצפנה מסוג Encrypted ClientHello (ECH). המאמר מדגים כי לבצע סיווג מוקדם של תעבורה (eTC) בדיוק גבוה, למרות השימוש הגובר בהצפנה בפרוטוקול TLS והסתרת מטא-דאטא חשובים כמו ה-Server Name (SNI) Indication).

התרומות המרכזיות של המאמר:

- <u>פיתוח אלגוריתם חדש (hRFTC):</u> האלגוריתם משלב מאפיינים מבוססי חבילות (מהמידע הלא מוצפן ב-TLS) עם מאפיינים מבוססי זרימה (כמו גודל חבילות וזמני ביניים בין חבילות). גישה זו מאפשרת לאלגוריתם להיות אפקטיבי גם כשמידע TLS מרכזי מוסתר על ידי ECH. האלגוריתם שומר על דיוק גבוה, בהשוואה לאלגוריתמים מובילים אחרים.

- <u>דיוק גבוה בהשוואה לאלגוריתמים קיימים:</u> האלגוריתם מציג תוצאות טובות יותר ביחס לאלגוריתמים מובילים אחרים לסיווג תעבורה מוצפנת, תוך שמירה על יכולת הכללה גבוהה גם במצבים שבהם האלגוריתם אומן על מערך נתונים קטן יחסית.

- <u>שימוש בגישה היברידית לסיווג תעבורה מוצפנת:</u> הגישה הייחודית של שילוב מאפיינים מבוססי חבילות וזרימה, וההתמקדות בניתוח חבילת הנתונים הראשונה היורדת, מאפשרת סיווג מהיר ומדויק יותר גם בתנאי הצפנה מתקדמים כמו ECH.

- <u>איסוף מערך נתונים רחב:</u> החוקרים אספו יותר מ-600,000 זרימות TLS ממדינות שונות כדי להבטיח גיוון גיאוגרפי. מערך הנתונים כולל מגוון רחב של סוגי תעבורה, כגון וידאו מאוחסן, וידאו קצר, וידאו חי ואודיו מאוחסן.

שאלה 2

המאמר משתמש בשילוב של מאפיינים מבוססי חבילות ומאפיינים מבוססי זרימה. המאפיינים האלו חשובים כדי לזהות סוגי תעבורה למרות ההצפנה שמספקים TLS ו-ECH.

- **מאפיינים מבוססי חבילות (Packet-Based):**

  - מטא-דאטא של TLS: מאפיינים שנשלפים מהחלקים הלא מוצפנים של ה-Handshake בפרוטוקול TLS, כגון:
    - הצפנים הנתמכים (Cipher Suites).
    - קבוצות שיתוף מפתח (Key Share Groups).
    - גרסת TLS.

- **מאפיינים מבוססי זרימה (Flow-Based):**

  - סטטיסטיקות של גודל חבילות: מאפיינים שנגזרים מגודל החבילות שעוברות בתקשורת, כגון:
    - גודל חבילה מינימלי.
    - גודל חבילה מקסימלי.
    - ממוצע גדלי החבילות.

  - סטטיסטיקות זמני ביניים בין חבילות: מאפיינים שנגזרים מזמני המעבר בין חבילות עוקבות, כגון:
    - הפרש זמנים מינימלי בין חבילות.
    - הפרש זמנים מקסימלי בין חבילות.
    - ממוצע זמנים בין חבילות.

- **מאפיינים חדשים שהוצגו במאמר:**

  - <u>שילוב מאפיינים היברידי:</u> השילוב של מאפיינים מבוססי חבילות ומאפיינים מבוססי זרימה הוא החידוש המרכזי במאמר. האלגוריתם hRFTC מצליח להשתמש בשניהם במקביל כדי להשיג דיוק סיווג גבוה גם כאשר רוב המטא-דאטא מוצפן.
  - <u>שימוש בחבילת הנתונים הראשונה היורדת (First Downlink Packet):</u> האלגוריתם נוקט בגישה ייחודית לסיווג מוקדם של תעבורה (eTC). בניגוד

למחקרים קודמים שהסתמכו על ניתוח מספר קבוע של חבילות, האלגוריתם מנתח את כל החבילות עד להגעת חבילת הנתונים הראשונה היורדת מהשרת, אשר מכילה נתוני אפליקציה. גישה זו מאפשרת לאלגוריתם לקבל את המידע הקריטי הנדרש לסיווג כבר בשלבים המוקדמים של התקשורת, ובכך להפחית עיכובים בתהליך הסיווג. השיטה מתמקדת במידע שמתקבל מהשרת בתחילת התקשורת, מה שמאפשר סיווג מדויק ומהיר יותר.

שאלה 3:

תוצאות עיקריות:

- דיוק גבוה: האלגוריתם השיג ציון F-מקרו של 94.6%, תוצאה טובה יותר בהשוואה לאלגוריתמים מובילים אחרים, כגון:
  - אלגוריתמים מבוססי חבילות כמו RB-RF ו-MATEC.
  - אלגוריתמים מבוססי זרימה (Flow-Based) כמו CESNET.
  - אלגוריתמים היברידיים אחרים כמו hC4.5 ו-UW.
- השפעת גודל החבילות וזמני ביניים: המחקר מראה שסטטיסטיקות של גודל החבילות היו קבוצת המאפיינים החשובה ביותר, ותרמו יותר מ-50% לדיוק הסיווג. דבר זה מראה שהמאפיינים מבוססי הזרימה הם קריטיים לזיהוי תעבורה מוצפנת.
- השפעת מיקום גיאוגרפי: המחקר מצא כי אלגוריתמי סיווג שאומנו על נתוני תעבורה מאזור גיאוגרפי אחד הציגו ביצועים ירודים כאשר הם חלו על תעבורה מאזורים גיאוגרפיים שונים. ממצא זה מדגיש את החשיבות של אימון האלגוריתמים על נתוני תעבורה מהאזור הגיאוגרפי בו הם ייושמו.

תובנות מהתוצאות:

- <u>חשיבות הגישה ההיברידית:</u> השילוב בין מאפיינים מבוססי חבילות ומאפיינים מבוססי זרימה היה קריטי להשגת דיוק גבוה, במיוחד בתעבורה מוצפנת.
- <u>חשיבות מאפיינים מבוססי זרימה:</u> התרומה הגבוהה של מאפיינים כמו גודל חבילות וזמני ביניים מראה שגישה מבוססת זרימה היא קריטית כאשר מאפייני חבילות רגילים מוסתרים בעקבות הצפנה.
- <u>יכולת הכללה גבוהה:</u> האלגוריתם הראה יכולת הכללה טובה מאוד, ושמר על דיוק גבוה גם כאשר הוא אומן על מערך נתונים קטן יחסית.

- <u>האתגר בזיהוי תעבורה ממדינות שונות:</u> המאמר מצביע על כך שחשוב לאמן את האלגוריתם עם נתונים מאזור גיאוגרפי מסוים על נתונים מהאיזור עליו הוא עתיד לפעול, על מנת לשפר את הדיוק.

חלק 3 - ניתוח הבדלים בתעבורות הרשת בין אפליקציות שונות

במסגרת חלק זה, השתמשנו בכלי Wireshark כדי לנתח את תעבורות הרשת של האפליקציות: YouTube, Spotify, Google Chrome browsing, Microsoft Edge browsing, Zoom.

ניתוח תעבורת הרשת מאפשר לזהות אפליקציות שונות, לאתר איומים פוטנציאליים, לייעל רשתות ולשפר את אבטחת המידע. כל גרף נותן מבט שונה על דפוסי התעבורה ועוזר בהבנת אופן פעולת האפליקציות. מטרת הניתוח היא לזהות דפוסים ייחודיים לכל אפליקציה, להבין אילו מאפיינים מבדילים אותן, ולבצע השוואה בין סוגי התעבורה השונים.

לשם כך, הפקנו סדרת גרפים המייצגים פרמטרים מרכזיים בתעבורת הרשת, כמו:

- גודל חבילות לאורך זמן – להבנת דפוסי העברת הנתונים.
- התפלגות TTL – כדי לזהות מבנה תעבורה אופייני לכל אפליקציה.
- נפח תעבורה מול מספר הזרמים – להערכת דפוסי תקשורת שונים.
- התפלגות דגלי TCP – להבנת סוגי ההתקשרות (חיבורים חדשים, סגירות וכו').
- שימוש בגרסאות TLS – לזיהוי רמות אבטחה והצפנה.
- גודל חלון TCP ממוצע – לבדיקת השפעת בקרת עומסים.
- התפלגות זמני הגעה בין חבילות (Inter-Arrival Time) – לזיהוי הבדלים בין יישומים בזמן אמת ליישומים מבוססי קובץ.
- גודל חבילה ממוצע – להשוואת רמות הדחיסה והפרוטוקולים.
- משך החיים של זרם חבילות – כדי לראות כיצד האפליקציות מתחזקות חיבורים.
- התפלגות פרוטוקולים – לזיהוי סוגי הפרוטוקולים הנפוצים בכל אפליקציה.

באמצעות הגרפים הללו, ניתן להבחין בהבדלים משמעותיים בין האפליקציות:

- YouTube ו-Spotify מציגות חבילות גדולות לאורך זמן, מאחר והן מבוססות זרימה.
- Chrome ו-Microsoft Edge מציגות מספר גבוה של זרמים קטנים עקב הגלישה באתרים שונים.
- Zoom משתמש בחבילות קטנות עם מספר רב של זרמים בשל תקשורת בזמן אמת.

כעת, נציג ונפרט על כל אחד מהגרפים- מטרתו, מבנהו וכיצד הוא מסייע בסיווג התעבורה בין האפליקציות.

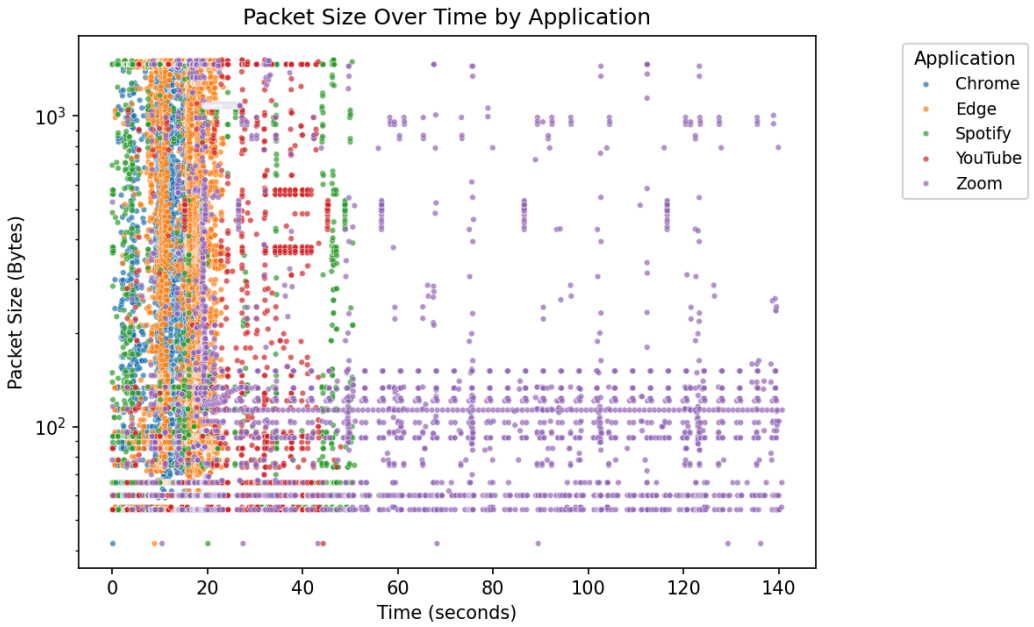
גרף מספר 1: גודל חבילות לאורך הזמן לכל אפליקציה

<u>מבנה הגרף:</u>

- ציר X – זמן (בשניות).

- ציר Y – גודל החבילה (בבתים), בלוגריתם לשיפור הנראות.

- כל נקודה מייצגת חבילה שנשלחה ברשת, כאשר הצבעים השונים מייצגים אפליקציות שונות.



<u>מטרה:</u>

- להמחיש כיצד משתנה גודל החבילות לאורך זמן עבור כל אפליקציה.
- לזהות מגמות בדפוסי שליחת הנתונים – אילו אפליקציות שולחות חבילות גדולות או קטנות, באיזו תדירות, ובאיזו אחידות.
- להבין הבדלים בין סוגי השירותים (גלישה, סטרימינג, תקשורת בזמן אמת).

<u>תיאור הנתונים מהגרף:</u>

שירותי סטרימינג (YouTube, Spotify)

- שולחים חבילות גדולות באופן עקבי, מה שמעיד על שימוש בפרוטוקולים כמו HTTP Adaptive Streaming.
- חבילות הווידאו/אודיו משתנות עם הזמן, בהתאם לאיכות הזרימה ולשינויים ברוחב הפס.
- YouTube מציג שינויים בגודל החבילה, מה שעשוי להעיד על מעבר בין איכויות וידאו או טעינת מקטעים שונים של הסרטון.

דפדפנים (Chrome, Edge)

- שולחים הרבה חבילות קטנות - תוצאה של טעינת עמודים המכילים רכיבים מרובים (תמונות, סקריפטים, CSS וכודומה).
- גודל החבילה משתנה בהתאם למבנה האתר, סוגי הקבצים, והעדפות המטמון של הדפדפן.
- נראים זמנים קצרים של פעילות אינטנסיבית, ולאחר מכן ירידה עם סיום טעינת הדף.

Zoom – תקשורת בזמן אמת

- משתמש בחבילות קטנות ואחידות לאורך זמן, כיוון שהוא עובד עם UDP, ולכן אינו מצריך אישור קבלה לכל חבילה.
- פיזור החבילות מצביע על זרימה רציפה של נתונים עם השהיה נמוכה, דבר שמשמעותי לשיחות וידאו וקול.
- שינויי דפוסי שליחה יכולים להעיד על שינויי איכות וידאו בהתאם לרוחב הפס.

<u>תובנות מרכזיות:</u>

- שירותי סטרימינג (YouTube, Spotify) - שולחים חבילות גדולות בצורה יציבה, מה שמעיד על זרימה מתמשכת של נתונים עם שינויים בהתאם לאיכות התוכן.
- דפדפנים (Chrome, Edge) - שולחים הרבה חבילות קטנות בפרצים קצרים, מה שמעיד על טעינת דפי אינטרנט עם משאבים מרובים.
- Zoom - שולח חבילות קטנות ואחידות לאורך זמן, מה שמעיד על תקשורת דו-כיוונית יציבה בזמן אמת עם השהיה נמוכה.
- שינויים פתאומיים בגודל החבילה - יכולים להצביע על טעינת דף מחדש, מעבר איכות וידאו, או שיחה שנקטעה והתחברה מחדש.

מסקנה:

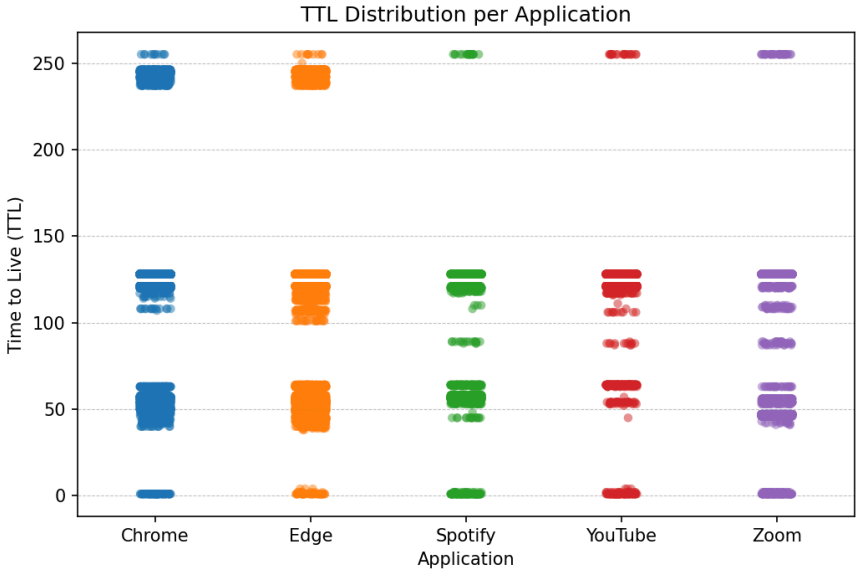
ניתוח גודל חבילות לאורך זמן מאפשר לסווג בקלות את סוג השירות:

- סטרימינג (YouTube, Spotify) שולח חבילות גדולות ומתמשכות.
- דפדפנים שולחים הרבה חבילות קטנות בפרצים קצרים.
- Zoom שולח חבילות קטנות ורציפות, אופייני לתקשורת בזמן אמת.

גרף מספר 2: התפלגות TTL (Time to Live) לכל אפליקציה

מבנה הגרף:

- ציר X– שם האפליקציה.
- ציר Y – ערך TTL (Time to Live), כלומר, מספר התחנות (נתבים) שהחבילה יכולה לעבור לפני שתושלך.
- כל נקודה מייצגת חבילה שנשלחה עם TTL מסוים, וההתפלגות מאפשרת להבין כיצד כל אפליקציה קובעת TTL לחבילות שלה.



מטרה:

- להציג את ההתפלגות של ערכי TTL בין האפליקציות השונות.
- TTL משקף כמה נתבים חבילה יכולה לעבור לפני שהיא נזרקת, ומשמש למניעת לולאות ניתוב.

- לאפליקציות שונות יש דפוסי TTL אופייניים, בהתאם לאופן ניהול התעבורה שלהן והמרחק לשרתי היעד.

<u>תיאור הנתונים מהגרף:</u>

דפדפנים (Chrome, Edge)

- נוטים להשתמש בערכי TTL קבועים יחסית (למשל, 64, 128 או 255), שמוגדרים כברירת מחדל במערכות הפעלה ובאפליקציות. TTL קבוע מצביע על כך שהבקשות נשלחות לשרתים קרובים יחסית עם מסלולי ניתוב יציבים.
- החבילות נשלחות עם TTL מוגדר על ידי מערכת ההפעלה, ולכן הערכים לרוב אחידים ללא שונות גבוהה.

שירותי סטרימינג ווידאו (YouTube, Zoom)

- לרוב בעלי TTL גבוה, כיוון שהם מסתמכים על רשתות CDN (Content Delivery Network).
- חבילות הווידאו מגיעות ממיקומים גיאוגרפיים שונים, ולכן TTL גבוה מאפשר להן לשרוד מסלולים ארוכים.
- שירותים אלו מתעדפים מהירות והפחתת השהיה, ולכן TTL גבוה עוזר לספק תוכן באופן אופטימלי.

Spotify – TTL משתנה

- מציג שונות גבוהה בערכי TTL, כיוון שהחבילות עשויות להגיע משרתי מדיה שונים.
- בהשוואה לדפדפנים, יש יותר פיזור בערכים, מה שמרמז על שימוש במסלולים דינמיים ותשתיות מרובות.
- Spotify לא מסתמך על רשת CDN קבועה, אלא על מספר מסלולים שונים להזרמת מוזיקה, ולכן ה- TTL משתנה בין חיבורים שונים.

<u>תובנות מרכזיות:</u>

- TTL קבוע יחסית - מאפיין אפליקציות ששולחות חבילות לשרתים קרובים עם ניתוב יציב, כמו דפדפנים.

- TTL גבוה ומרוכז סביב ערך מסוים - מעיד על שירותים המשתמשים ברשתות CDN עם ניתוב יציב, כמו YouTube ו-Zoom.
- TTL משתנה בטווח רחב -מצביע על שימוש ברשתות חיצוניות מרובות או שרתים מבוזרים, כמו Spotify.

<u>מסקנה:</u>

התפלגות TTL מסייעת בזיהוי סוגי שירותים ברשת:

- דפדפנים מציגים TTL יציב, מה שמעיד על חיבורים ישירים לשרתים קרובים.
- שירותי סטרימינג ווידאו משתמשים ב-TTL גבוה, מה שמעיד על שימוש ברשתות CDN גלובליות.
- אפליקציות עם TTL משתנה (כגון Spotify) כנראה תלויות במסלולי ניתוב דינמיים ושרתים מבוזרים.

גרף מספר 3: נפח תעבורה מול מספר הזרמים לכל אפליקציה

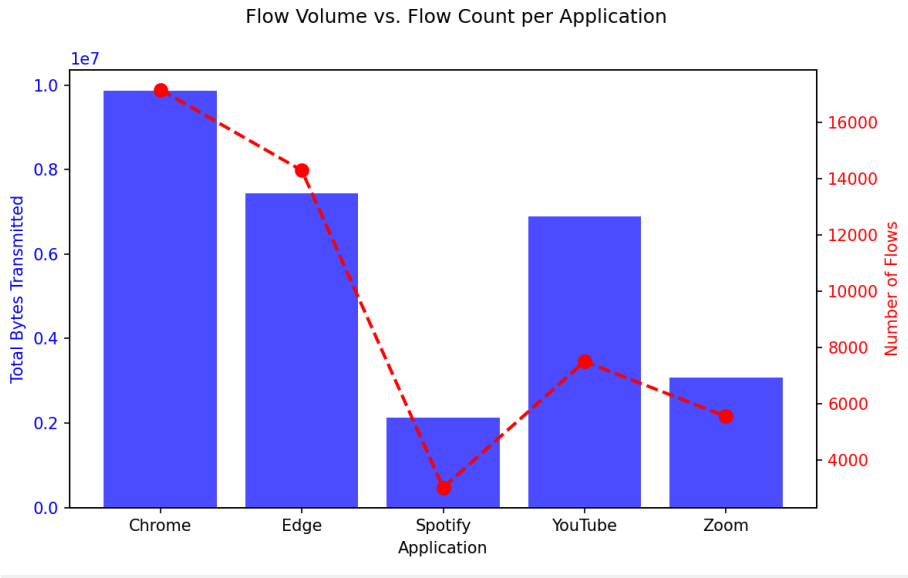
מבנה הגרף:

- ציר X – שם האפליקציה.

- ציר Y שמאלי – סך כל כמות הנתונים שהאפליקציה שלחה בבתים.

- ציר Y ימני – מספר הזרמים.

- שימוש בגרף עמודות (bar) לכמות הנתונים, וגרף קווי (line) למספר הזרמים.



מטרה:

- להציג את כמות הנתונים שנשלחה על ידי כל אפליקציה ביחס למספר הזרמים שנפתחו.
- להבין את דפוסי השימוש ברוחב הפס ובמספר החיבורים של כל אפליקציה.
- להבדיל בין אפליקציות עם זרמי נתונים קצרים ורבים (דפדפנים) לעומת אפליקציות עם זרמים ארוכים ומתמשכים (סטרימינג, Zoom).

תיאור הנתונים מהגרף:

דפדפנים (Chrome, Edge)

- שולחים את נפח התעבורה הגבוה ביותר עם מספר זרמים גדול.

- כמות הזרמים הגבוהה נובעת מבקשות HTTP רבות לכל דף אינטרנט (CSS, JavaScript, תמונות וכו').
- כל זרם אינו בהכרח גדול מאוד, אך יחד הם יוצרים נפח משמעותי של נתונים.

Spotify – שירות סטרימינג אודיו

- מספר זרמים נמוך עם נפח נתונים נמוך יחסית.
- כיוון שמדובר בסטרימינג אודיו, כמות הנתונים נמוכה בהשוואה לווידאו.
- מספר הזרמים הנמוך מצביע על כך שכל זרם מכיל חיבור ממושך עם השרת (הזרמת שירים).

YouTube – שירות סטרימינג וידאו

- שולח נפח נתונים גבוה אך עם מספר זרמים בינוני.
- הזרמת וידאו דורשת תעבורה רבה, אך משתמשת במספר זרמים נמוך יחסית בהשוואה לדפדפנים.
- משתמש במספר זרמים נוספים עבור תמונות ממוזערות, פרסומות, וטעינת וידאו ברקע.

Zoom – תקשורת בזמן אמת

- מייצר זרמים רבים אך בנפח קטן יחסית.
- מתאים לשיחות וידאו וקול בזמן אמת, שבהן נשלחות חבילות קטנות בתדירות גבוהה.
- מספר הזרמים הגדול מצביע על ריבוי חיבורים דו-כיווניים לשיחות וידאו ואודיו (RTC).

<u>תובנות מרכזיות מהגרף:</u>

- דפדפנים (Chrome, Edge) יוצרים זרמים רבים בנפח גבוה, המאפיינים טעינת דפי אינטרנט ושירותים מבוזרים.
- Spotify מציג מספר זרמים נמוך עם תעבורה נמוכה, מה שמעיד על חיבורים ארוכים להזרמת אודיו.
- YouTube שולח נפח גבוה של נתונים עם מספר זרמים בינוני, מה שמעיד על סטרימינג וידאו יציב.
- Zoom יוצר הרבה זרמים קטנים, המאפיינים תקשורת דו-כיוונית בזמן אמת.

מסקנה:

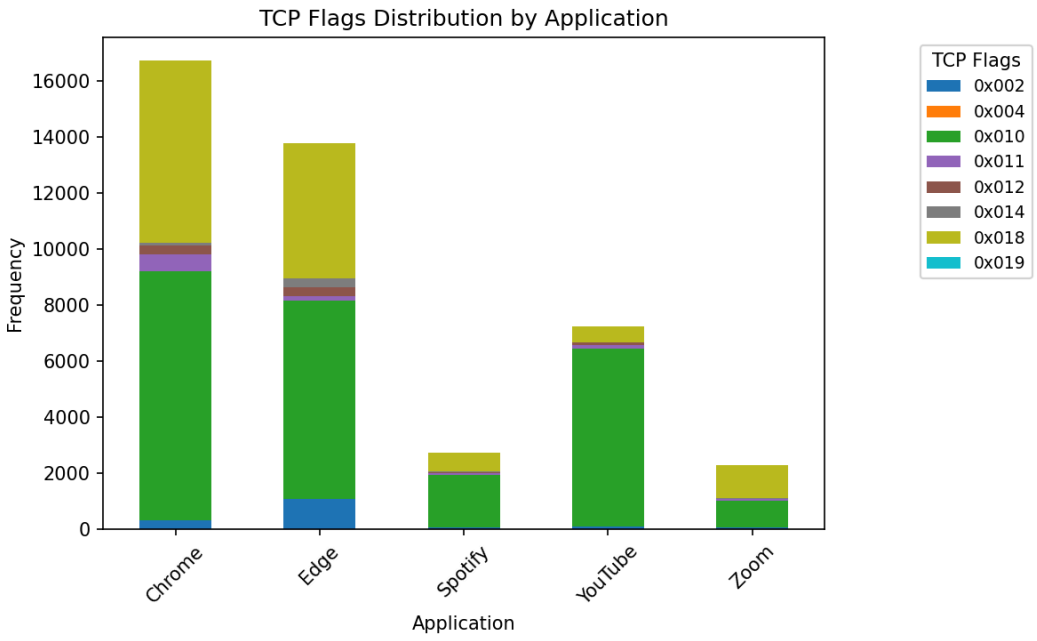
יחס הזרמים מול נפח הנתונים מאפשר לסווג סוגי שירותים ברשת:

- דפדפנים יוצרים הרבה חיבורים קטנים ונפח נתונים גבוה.
- שירותי סטרימינג משתמשים במספר חיבורים נמוך עם נפח נתונים גבוה.
- אפליקציות זמן-אמת כמו Zoom פותחות הרבה זרמים עם מעט נתונים בכל אחד.

גרף מספר 4: התפלגות דגלי TCP לכל אפליקציה

מבנה הגרף:

- ציר X – האפליקציות השונות.
- ציר Y – מספר המופעים של כל דגל TCP.
- כל צבע מייצג דגל אחר, כאשר הדגלים נערמים כדי להציג את ההתפלגות הכוללת לכל אפליקציה.



מטרה:

- לנתח את השימוש בדגלי TCP כדי להבין כיצד אפליקציות שונות מנהלות חיבורים והעברת נתונים.

- להבדיל בין אפליקציות שיוצרות הרבה חיבורים קצרים (דפדפנים) לבין אפליקציות עם זרמים רציפים (סטרימינג, Zoom).
- דגלי TCP משמשים לניהול תקשורת, ולכן ההתפלגות שלהם מספקת מידע על אופי הפעילות של כל אפליקציה.

<u>תיאור הנתונים מהגרף:</u>

דפדפנים (Chrome, Edge)

- הרבה SYN ו-FIN - מעיד על פתיחת וסגירת חיבורים רבים (TCP 3-way handshake).
- יוצרים הרבה חיבורים קצרים לרכיבים שונים בדפי אינטרנט.
- נוכחות גבוהה של ACK לאישור קבלת נתונים מהשרתים, עקב טעינת משאבים מרובים בדפי אינטרנט.

שירותי סטרימינג (YouTube, Spotify)

- מציגים בעיקר דגלי ACK - סטרימינג שולח זרם נתונים רציף, ולכן יש מעט חיבורים אך הרבה אישורי ACK.
- מעט SYN ו-FIN - סטרימינג משתמש בחיבור אחד ארוך במקום לפתוח חיבורים רבים.
- נוכחות PSH-ACK - דוחף נתונים קריטיים במהירות (כגון וידאו ואודיו).

Zoom – תקשורת בזמן אמת

- הרבה PSH-ACK - זום משתמש בדגל זה כדי למנוע השהיות ולשלוח נתונים מיידית.
- ACK גבוה - מאשר קבלת נתונים בתקשורת דו-כיוונית רציפה.
- מעט SYN ו-FIN - זום שומר על חיבור מתמשך ולא פותח וסוגר חיבורים כל הזמן.

<u>תובנות מרכזיות:</u>

- ריבוי SYN ו-FIN מאפיין אפליקציות שיוצרות הרבה חיבורים קצרים, כמו דפדפנים.
- ריבוי ACK מצביע על אפליקציות שמשתמשות בזרם נתונים יציב ומתמשך, כמו סטרימינג ו-Zoom.
- ריבוי PSH-ACK נמצא בעיקר באפליקציות זמן אמת כמו Zoom וסטרימינג חי, כדי למנוע השהיות בתקשורת ולשלוח נתונים מיידית.

מסקנה:

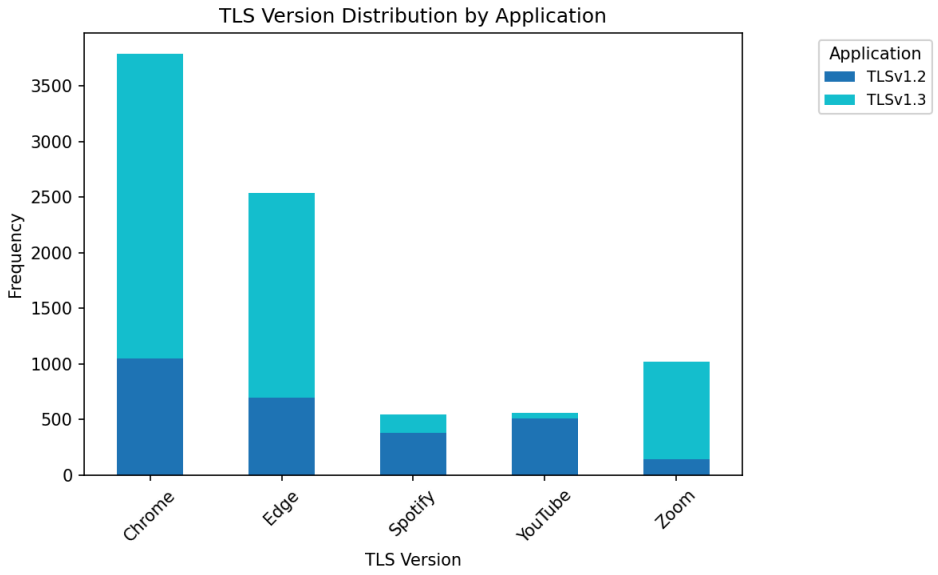
ניתוח דגלי TCP מאפשר זיהוי ברור של סוגי שירותים ברשת:

- דפדפנים משתמשים בהרבה SYN ו-FIN ליצירת חיבורים קצרים ורבים.
- שירותי סטרימינג מסתמכים על חיבור יציב עם הרבה ACK ומעט SYN-FIN.
- Zoom ואפליקציות זמן-אמת משתמשות בריבוי PSH-ACK להבטחת שידור נתונים ללא השהיות.

גרף מספר 5: גרסת TLS בשימוש לכל אפליקציה

מבנה הגרף:

- ציר X – שם האפליקציה.
- ציר Y – מספר החבילות שנשלחו תחת פרוטוקולי TLS שונים.
- צבעים שונים מייצגים גרסאות TLS שונות (למשל, TLS 1.2, TLS 1.3).



מטרה:

- לבדוק באילו גרסאות TLS כל אפליקציה משתמשת במהלך התקשורת שלה.
- להבדיל בין אפליקציות שמסתמכות על חיבורים מאובטחים מודרניים לבין כאלה שעדיין תומכות בפרוטוקולים ישנים.

- TLS (Transport Layer Security) הוא פרוטוקול הצפנה חיוני שמגן על המידע בזמן העברתו באינטרנט, ולכן ניתוח גרסאות TLS יכול לספק מידע על רמת האבטחה והשימוש בטכנולוגיות עדכניות.

<u>תיאור הנתונים מהגרף:</u>

דפדפנים (Chrome, Edge)

- משתמשים בעיקר ב-TLS 1.3, הגרסה החדשה והמאובטחת ביותר, המציעה שיפור ביצועים והפחתת זמני חיבור.
- TLS 1.2 מופיע במקרים מסוימים, כאשר הדפדפן מתחבר לאתרים ישנים או לשרתי מדיה שאינם תומכים בפרוטוקול החדש.
- דפדפנים נוטים לעדכן את מנועי ההצפנה שלהם לעיתים קרובות, ולכן קל יחסית לזהות תעבורת גלישה על בסיס הפרוטוקול.

שירותי סטרימינג (YouTube, Spotify)

- עשויים להשתמש ב-TLS 1.2 או 1.3, בהתאם לשרתי המדיה אליהם הם מתחברים.
- שרתי סטרימינג מעדיפים אבטחה חזקה אך לעיתים תומכים בפרוטוקולים ישנים כדי להבטיח תאימות עם מכשירים ישנים יותר.

Zoom – תקשורת מוצפנת בזמן אמת

- משתמש בעיקר ב-TLS 1.3, המספק זמן שיהוי נמוך יותר, קריטי לשיחות וידאו וקול בזמן אמת.
- רוב ספקי הענן והתשתיות המודרניות תומכים כברירת מחדל ב-TLS 1.3, ולכן Zoom מאמץ אותו באופן גורף.
- חלק מהתעבורה של Zoom עדיין משתמשת ב-TLS 1.2, במיוחד כאשר TLS 1.3 אינו נתמך (למשל, במכשירים ישנים או בסביבות ארגוניות עם מערכות מדור קודם).

<u>תובנות מרכזיות:</u>

- דפדפנים (Chrome, Edge) מסתמכים בעיקר על TLS 1.3, אך תומכים גם ב-TLS 1.2 עבור אתרים ישנים.
- שירותי סטרימינג (YouTube, Spotify) משתמשים ב-TLS 1.2 או 1.3, בהתאם לשרתים ולתשתיות המדיה.

- Zoom מתעדף TLS 1.3, בשל צורך באבטחה גבוהה ושיהוי נמוך, אך עדיין יש שימוש ב-TLS 1.2 עבור חיבורים שאינם תומכים בגרסה החדשה.

<u>מסקנה:</u>

ניתוח גרסאות TLS מאפשר להבין את רמות האבטחה וסוגי השירותים השונים:

- אפליקציות מודרניות נוטות להשתמש בעיקר ב-TLS 1.3 בשל יתרונותיו בשיפור ביצועים והצפנה חזקה יותר.
- אפליקציות עם תמיכה לאחור (תמיכה בגרסאות קודמות, כמו סטרימינג ו-Zoom) שומרות על תאימות ל-TLS 1.2, אך המעבר ל-TLS 1.3 הופך לנפוץ יותר.
- שימוש נרחב ב-TLS 1.2 עשוי להעיד על תשתיות ישנות או על צורך בתמיכה רחבה במכשירים מיושנים.

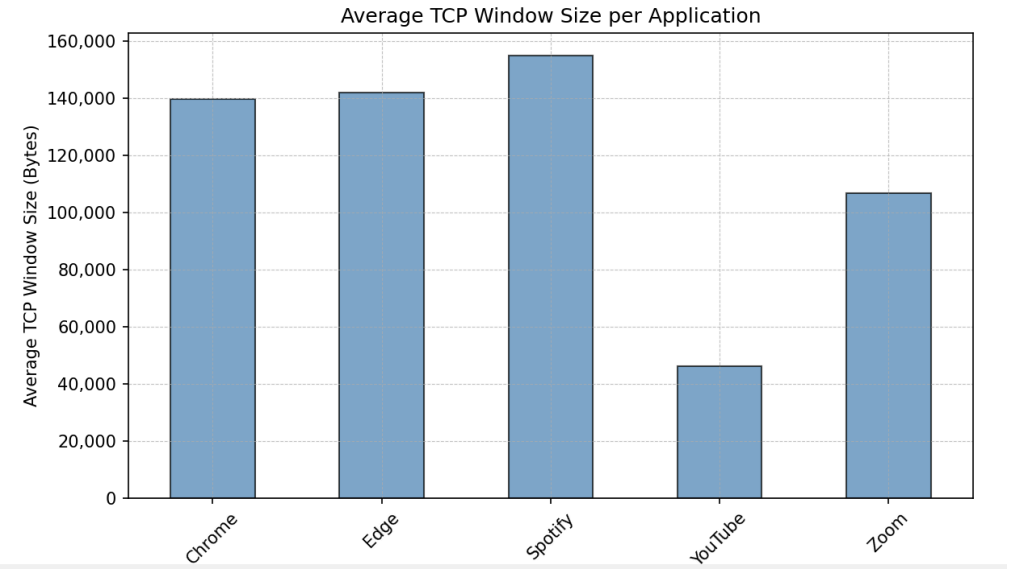
גרף מספר 6: גודל חלון TCP ממוצע לכל אפליקציה

<u>מבנה הגרף:</u>

- ציר X – שם האפליקציה.

- ציר Y – גודל החלון בבתים.

- הגרף מציג את גודל החלון הממוצע לכל אפליקציה.



מטרה:

- לנתח כיצד אפליקציות שונות מנהלות את בקרת הזרימה ב-TCP באמצעות גודל חלון TCP ממוצע.
- להבין את הקשר בין גודל חלון TCP לבין סוג השירות, כמו גלישה, סטרימינג או תקשורת בזמן אמת.

תיאור הנתונים מהגרף:

Spotify, Chrome ו-Edge – גודל חלון TCP גדול

- מאפשר העברת נתונים רציפה בקצב גבוה, חיוני לגלישה מהירה ולהזרמת מדיה.
- דפדפנים (Chrome, Edge) דורשים חלון TCP גדול כדי לאפשר טעינה מהירה של דפים ומשאבים מרובים במקביל.
- Spotify משתמש בחלון גדול במיוחד, מכיוון שסטרימינג אודיו דורש חיבור רציף והורדת מקטעים קדימה.

YouTube – גודל חלון TCP קטן משמעותית

- עשוי להעיד כי השירות מסתמך יותר על buffering חכם (מנגנון שבו הווידאו נטען מראש ונשמר באופן זמני בזיכרון לפני שהמשתמש צופה בו), ולכן אינו דורש חלון TCP גדול.
- אפשרות נוספת היא שמנגנוני הזרמת הווידאו מותאמים כך שאין צורך באחסון כמויות גדולות של נתונים בזיכרון TCP.

Zoom – גודל חלון TCP בינוני

- מאזן בין שיהוי נמוך (חבילות קטנות) לבין צורך בהעברת נתונים רציפה עבור וידאו חי.
- מאפשר תקשורת אינטראקטיבית, אך לא דורש נפח חלון גדול כמו סטרימינג, שכן הוא שולח חבילות קטנות בתדירות גבוהה.

תובנות מרכזיות:

- חלון TCP גדול (Spotify, דפדפנים) - מעיד על תעבורה אינטנסיבית, הורדות גדולות וטעינה מהירה של נתונים.

- חלון TCP קטן יותר (YouTube) - מצביע על תעבורה מבוססת buffering חכם והזרמת וידאו ללא צורך בזיכרון TCP גדול.
- Zoom – חלון בינוני - מאפשר שיהוי נמוך עם איזון בין נפח הנתונים לרציפות התקשורת.
- ניתן להבדיל בין סוגי שירותים לפי אופן ניהול חלון ה-TCP שלהם, מה שמסייע בסיווג תעבורה ברשת.

<u>מסקנה:</u>

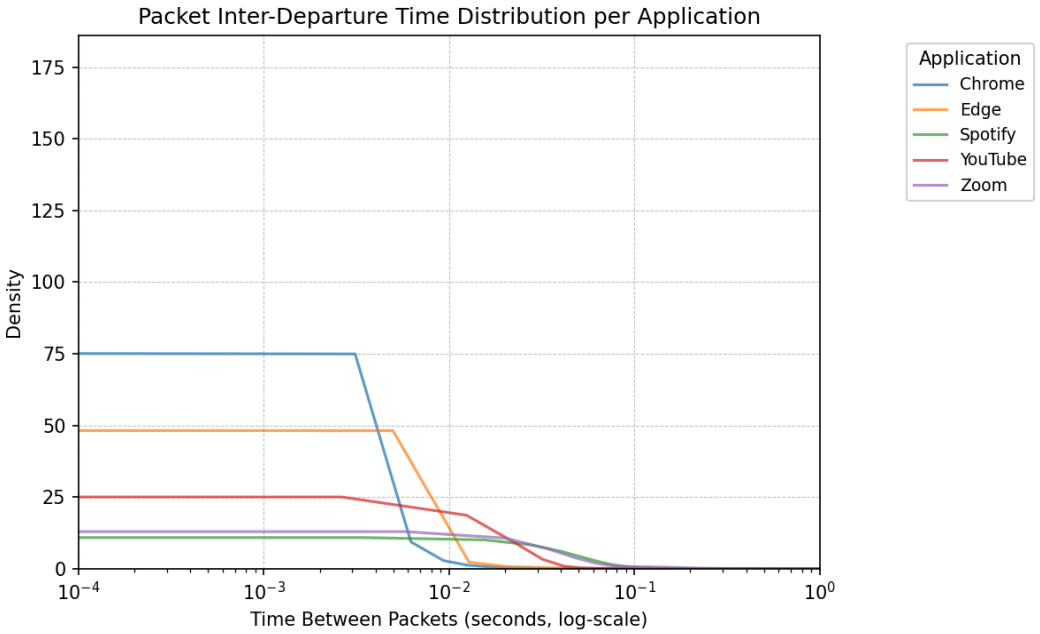
- שירותים כמו סטרימינג וגלישה דורשים חלונות גדולים כדי למקסם מהירות הורדת נתונים.
- שירותי וידאו כמו YouTube משתמשים בחלון קטן יחסית, כנראה בגלל מנגנוני buffering מתקדמים.
- אפליקציות זמן אמת כמו Zoom משתמשות בגודל חלון בינוני, המאפשר שמירה על תקשורת רציפה עם שיהוי נמוך.

גרף מספר 7: התפלגות זמני הגעה בין חבילות לכל אפליקציה

<u>מבנה הגרף:</u>

- ציר X – זמן בין חבילות (שניות, לוגריתם).

- ציר Y – צפיפות.



מטרה:

- לנתח את ההפרשים בין חבילות (Packet Inter-Departure Time) על מנת לזהות מאפייני תקשורת של אפליקציות שונות.
- להבין כיצד תדירות שליחת החבילות משתנה בהתאם לסוג האפליקציה והפרוטוקול שבו היא משתמשת.

תיאור הנתונים מהגרף:

שירותי סטרימינג (YouTube, Spotify) – זמני הגעה קצרים ויציבים

- שולחים נתונים באופן רציף - נדרשת תדירות חבילות אחידה כדי לשמור על איכות הזרימה.
- זמני הגעה קצרים יחסית וללא שינויים גדולים, כיוון שהאפליקציה דואגת להעביר את התוכן בצורה עקבית וללא הפסקות.

אפליקציות גלישה (Chrome, Edge) – פערים משתנים בין חבילות

- דפדפנים תלויים באינטראקציה של המשתמש ולכן קצב שליחת החבילות אינו אחיד.
- בעת טעינת דף אינטרנט - נשלחות הרבה חבילות ברצף, אך לאחר מכן נוצר פער עד הבקשה הבאה.
- צפיפות גבוהה בתחילת הגרף - מציינת בקשות רבות תוך זמן קצר, ולאחר מכן דעיכה כאשר הדף נטען במלואו.

Zoom – זמני הגעה קבועים יחסית

- Zoom משתמש ב-UDP- לכן אין בקרת עומסים כמו ב-TCP, והחבילות נשלחות בתדירות אחידה כדי להבטיח שידור וידאו וקול ללא השהיות.
- שמירה על פערים קצרים ויציבים בין חבילות - הכרחי לתקשורת בזמן אמת, כדי למנוע השהיות בשיחה.

תובנות מרכזיות:

- סטרימינג (YouTube, Spotify) – זמני הגעה אחידים וקצרים - מאפשרים זרימה חלקה וללא עצירות.
- דפדפנים (Chrome, Edge) – זמני הגעה משתנים - תלויים באינטראקציה של המשתמש ובמבנה האתר.

- Zoom – קצב חבילות קבוע יחסית - מאפיין חיבורים מבוססי UDP, המאפשרים תקשורת בזמן אמת עם מינימום השהיות.
- TCP מול UDP - חיבור מבוסס TCP מווסת את קצב הנתונים לפי עומס הרשת, ולכן דפדפנים מציגים פערים משתנים בין חבילות, בעוד Zoom משתמש ב-UDP ושומר על יציבות.

<u>מסקנה:</u>

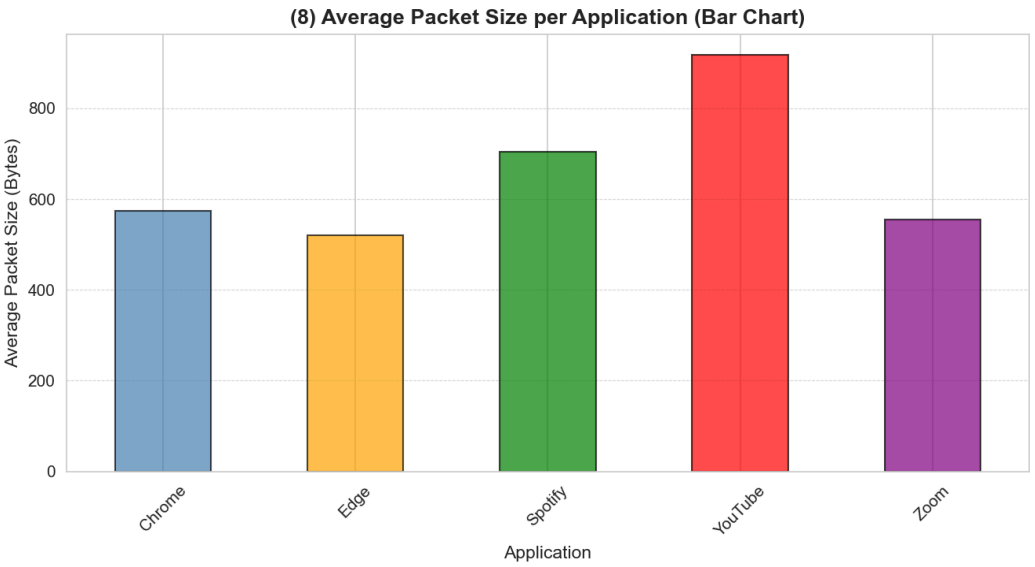
- אפליקציות שונות מציגות דפוסים ייחודיים של זמני הגעה בין חבילות, בהתאם לשיטת ההעברה ולסוג השירות.
- שירותי סטרימינג שומרים על זמנים יציבים וקצרים, גלישה מבוססת TCP מציגה פערים משתנים, ואילו Zoom מבוסס UDP שומר על קצב אחיד.
- הבדלים אלו מאפשרים סיווג של סוגי התעבורה, על סמך תדירות שליחת החבילות והפרוטוקול שבשימוש.

גרף מספר 8: גודל חבילה ממוצע לכל אפליקציה

<u>מבנה הגרף:</u>

- ציר X – שם האפליקציה.

- ציר Y – גודל חבילה ממוצע.



מטרה:

- להציג את גודל החבילה הממוצע עבור כל אפליקציה, כדי להבין כיצד כל שירות מעביר נתונים ברשת.
- לזהות דפוסי תעבורה המבדילים בין שירותי סטרימינג, גלישה, וידאו בזמן אמת ועוד.

תיאור הנתונים מהגרף:

YouTube – גודל החבילה הממוצע הגבוה ביותר

- שירותי סטרימינג של וידאו שולחים כמות נתונים גדולה במקטעים גדולים, ולכן גודל החבילה הממוצע הוא הגדול ביותר.
- חבילות גדולות מאפשרות הזרמת תוכן בקצב גבוה, עם פחות תקורה לכל חבילה.
- השימוש בפרוטוקול DASH (Dynamic Adaptive Streaming over HTTP) מאפשר שליחת נתונים במקטעים גדולים לפי רוחב הפס הזמין.

Spotify – גודל חבילה ממוצע גבוה

- שירותי סטרימינג אודיו שולחים חבילות גדולות יחסית כדי לשמור על רציפות השמעת המוזיקה.
- גודל חבילה ממוצע נמוך יותר מ-YouTube, כי קובצי אודיו קטנים יותר, אבל עדיין דורשים זרימה חלקה.

דפדפנים (Chrome, Edge) – גודל חבילה ממוצע בינוני

- דפדפנים שולחים בקשות HTTP רבות בעת טעינת דפים, אך כל בקשה מכילה כמות מידע קטנה יחסית.
- טעינת דף מורכבת מהרבה חבילות קטנות, לעומת סטרימינג שבו יש פחות חבילות אבל הן גדולות.
- ההבדל בין Chrome ו-Edge עשוי לנבוע מהאופן שבו כל דפדפן מבצע בקשות והעדפות שימוש ב-TLS ודחיסת נתונים.

Zoom – גודל חבילה ממוצע הנמוך ביותר

- Zoom משתמש ב-UDP, מה שמאפשר שליחת חבילות קטנות בתדירות גבוהה. חבילות קטנות מפחיתות שהיה ומבטיחות שידור רציף של וידאו ושמע בזמן אמת.

- Zoom מתעדף שליחה רציפה של חבילות קטנות במקום שליחה מרוכזת של חבילות גדולות, כדי לשמור על איכות שיחה גבוהה עם מינימום איבודי מידע.

<u>תובנות מרכזיות:</u>

- שירותי סטרימינג (YouTube, Spotify) שולחים חבילות גדולות כדי לשמור על קצב זרימה יציב של מדיה.
- דפדפנים (Chrome, Edge) שולחים חבילות בגודל בינוני אך בקצב משתנה, בגלל אופן טעינת דפי אינטרנט.
- Zoom משתמש בחבילות קטנות מאוד כדי להבטיח שיחות וידאו עם שיהוי מינימלי.
- גודל החבילה משפיע על ביצועי התעבורה – סטרימינג משתמש בחבילות גדולות למקסום יעילות, בעוד וידאו-צ'אט מתעדף חבילות קטנות לשליטה טובה יותר בזמן אמת.

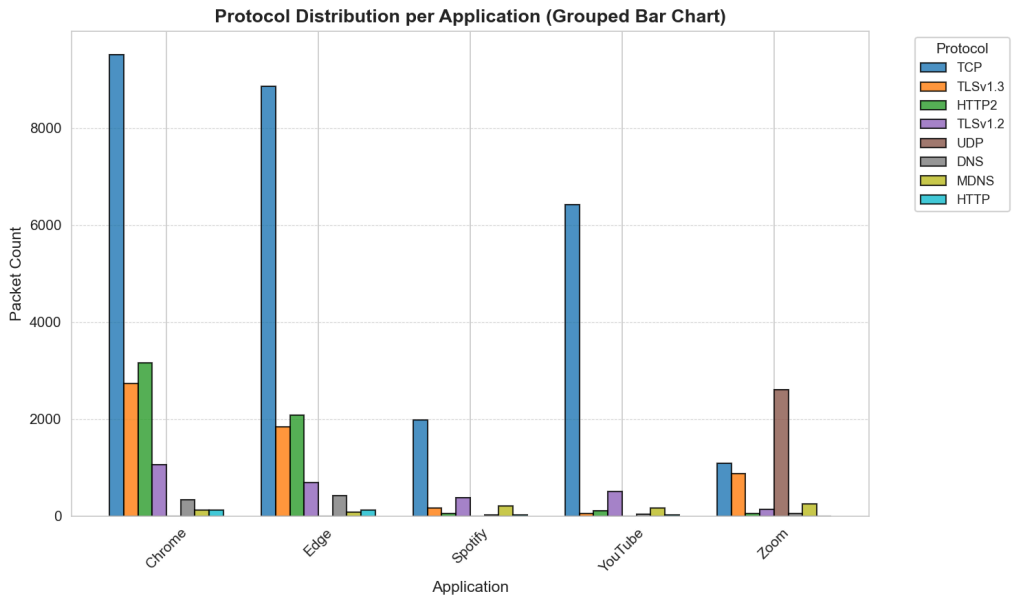
<u>מסקנה</u>:

- אפליקציות מדיה שולחות חבילות גדולות כדי למקסם ניצול רוחב פס ולמנוע קטיעות בהזרמה.
- אפליקציות מבוססות אינטראקציה (דפדפנים, Zoom) משתמשות בחבילות קטנות יותר, כדי להתאים את השליחה לצרכי המשתמש או למנוע השהיות בתקשורת.

גרף מספר 9: התפלגות פרוטוקולים לכל אפליקציה

מבנה הגרף:

- ציר X – שם האפליקציה.
- ציר Y – מספר החבילות.
- כל צבע מייצג פרוטוקול רשת אחר (TCP, UDP, TLS, HTTP, DNS וכדומה).



מטרה:

- לנתח את הפרוטוקולים השונים בהם משתמשת כל אפליקציה כדי להבין את דפוסי התקשורת שלהן.
- להבדיל בין אפליקציות מבוססות TCP (מהימנות גבוהה) ל-UDP (מהירות גבוהה) ולזהות פרוטוקולי אבטחה כמו TLS 1.2/1.3.
- להבין את השימוש ב-DNS ו-MDNS, המסייעים באיתור שרתים ופתרון שמות דומיין.

תיאור הנתונים מהגרף:

דפדפנים (Chrome, Edge) – נשענים בעיקר על TCP, TLS ו-HTTP/2

- TCP דומיננטי – גלישה מבוססת חיבורים אמינים ומהימנים.
- TLS 1.3 ו-TLS 1.2 – הצפנה מאובטחת בתקשורת עם אתרי אינטרנט.

- HTTP/2 נפוץ מאוד – דפדפנים מודרניים משתמשים בו לטעינת דפים בצורה מהירה ויעילה.
- שימוש ב-DNS – פתרון שמות דומיין לטעינת אתרים.

שירותי סטרימינג (Spotify, YouTube) – תלויים ב-TCP וב-TLS, עם שימוש נמוך ב-UDP

- TCP דומיננטי – סטרימינג דורש אמינות כדי למנוע איבוד נתונים.
- TLS נפוץ – הצפנת המדיה בזמן העברתה.
- YouTube ו-Spotify מסתמכים בעיקר על TCP ולא על UDP.

Zoom – מסתמך על UDP ו-TCP

- שימוש גבוה ב-UDP – שיחות וידאו דורשות זמני השהיה נמוכים ומהירות תגובה גבוהה.
- TLS ו-TCP קיימים אך פחות דומיננטיים – משמשים בעיקר לאימות והתחברות לשירותים מאובטחים.
- שימוש נרחב ב-DNS ו-MDNS – זום מחפש שרתים זמינים ברשת ומנהל חיבורים דינמיים.

<u>תובנות מרכזיות:</u>

- דפדפנים (Chrome, Edge) מסתמכים על TCP, TLS ו-HTTP/2 – גלישה דורשת חיבורים אמינים ומאובטחים עם שרתים.
- שירותי סטרימינג (YouTube, Spotify) משתמשים ב-TCP ו-TLS – כדי להבטיח זרימה תקינה ואמינות גבוהה של התוכן המוזרם.
- Zoom מתבסס בעיקר על UDP – שיחות וידאו דורשות מהירות תגובה גבוהה, ולכן הפרוטוקול הזה עדיף על פני TCP.
- השימוש ב-DNS גבוה ביישומים שדורשים פתרון שמות שרתים – דפדפנים ו-Zoom מציגים שימוש גבוה יותר ב-DNS ו-MDNS.

<u>מסקנה:</u>

סוג הפרוטוקול שבו אפליקציה משתמשת מאפשר לסווג את סוג הפעילות שלה:

- TCP ו-TLS דומיננטיים בגלישה ובסטרימינג, בשל מהימנות גבוהה.

- UDP נפוץ בשירותי תקשורת בזמן אמת כמו Zoom, שבהם המהירות חשובה יותר מאמינות מוחלטת.
- DNS ו-MDNS משחקים תפקיד חשוב באיתור וטעינת שרתים, בעיקר בדפדפנים ובאפליקציות תקשורתיות.

שאלה 4 + בונוס

<u>חלק 1:</u>

כעת ננתח תעבורת רשת חדשה בתור תוקפים. מטרת ניתוח זה היא להשוות בין תעבורת הרשת של האפליקציות השונות אשר ניתחנו בשאלה 3, לבין קבוצת הנתונים החדשה (chrome_spotify_attacker.csv).

לצורך השוואת דפוסי התעבורה בין אפליקציות שונות לבין קבוצת הנתונים החדשה, אשר משלבת חיפוש בChrome במקביל להקשבה לשירים מאפליקציית spotify, בוצע ניתוח מבוסס ארבעה גרפים עיקריים:

1. **גודל חבילה לאורך זמן – חמש אפליקציות נפוצות**
   - גרף זה מציג את התנהגות התעבורה של Chrome, Microsoft Edge, Spotify, YouTube ו-Zoom לאורך ציר הזמן, תוך התמקדות בגודל החבילות הנשלחות.
   - הניתוח מאפשר לזהות האם קיימים הבדלים מובהקים בין אופי התעבורה של האפליקציות השונות.
2. **גודל חבילה לאורך זמן – חמש אפליקציות לעומת קבוצת הנתונים החדשה**
   - גרף זה משווה את דפוסי התעבורה של חמש האפליקציות מול קבוצת הנתונים החדשה (chrome_spotify_attacker.csv).
   - ההשוואה מאפשרת לזהות הבדלים וחריגות בין דפוסי התעבורה הרגילים לבין דפוסי התעבורה החדשה, תוך בחינת גודל החבילות ושכיחות השליחה לאורך זמן.
3. **התפלגות גודל הזרימה (Flow Size Distribution)**
   - גרף זה מציג את מספר החבילות בכל זרימה של כל אפליקציה.
   - ניתוח זה מאפשר לזהות אילו אפליקציות שולחות כמויות גדולות של נתונים לאורך זמן, ומסייע בזיהוי התנהגויות ממושכות ולא שגרתיות.
4. **דירוג דמיון בין האפליקציות לתעבורה החדשה**
   - לצורך מדידת רמת הדמיון בין קבוצת הנתונים החדשה לבין האפליקציות הנבדקות, חושב מרחק אוקלידי בין המדדים הסטטיסטיים שלה לבין כל אפליקציה.
   - תוצאות דירוג הדמיון מציגות איזו אפליקציה הכי קרובה אליה, מבחינת דפוסי התעבורה שלה, ומהי מידת החריגה של שאר האפליקציות.

באמצעות מדדים אלה, ניתן לזהות האם קיים דמיון משמעותי בין תעבורת הרשת החדשה לבין האפליקציות האחרות, ובכך להסיק באילו אפליקציות השתמשו בזמן הקלטת התעבורה החדשה.

כדי לנתח את קבוצת הנתונים החדשה ולהבין את ההבדלים בינה לבין תעבורת האפליקציות הרגילות, נבצע את ההשוואה בשני שלבים:

שלב ראשון (סעיף א): השוואת התפלגות גודל הזרם וגודל החבילות לאורך הזמן

- **גרף 1: גודל חבילה לאורך זמן – השוואת חמש האפליקציות לבין קבוצת הנתונים החדשה**
  - נבחן כיצד גודל החבילות משתנה לאורך זמן עבור Chrome, Edge, Spotify, YouTube ו-Zoom בהשוואה לתעבורה החדשה.
  - נבדוק כיצד נוכחות ה-Tuple 4 משפיעה על דפוסי השליחה של החבילות והאם ניתן לראות חריגות בתדירות השליחה או בגודל החבילה.
- **גרף 2: התפלגות גודל הזרם – השוואת חמש האפליקציות מול קבוצת הנתונים החדשה**
  - ננתח **כיצד מספר החבילות בכל זרימה משתנה בין האפליקציות** ונשווה זאת לקבוצת הנתונים החדשה.
  - נראה **האם קיימים הבדלים משמעותיים בגודל הזרימה** בין האפליקציות לבין התוקף, וכיצד ה-Tuple 4 משפיע על גודל הזרימה.

שלב שני (סעיף ב): חישוב הדמיון הסטטיסטי בין האפליקציות לתעבורה החדשה

- **גרף 3: התפלגות זמן ההגעה בין חבילות (Inter-Arrival Time Distribution)**
  - גרף זה מציג את התפלגות זמני ההגעה בין חבילות (IAT) עבור כל אפליקציה והתבורה החדשה.
  - השוואה בין האפליקציות השונות תאפשר לזהות האם התעבורה החדשה פועלת בדפוס תעבורה שונה באופן משמעותי מהתנהגות רגילה.
- **גרף 4: התפלגות גודל החבילות (Packet Size Distribution)**
  - גרף זה מציג התפלגות גודל החבילות עבור כל אפליקציה, כולל קבוצת הנתונים החדשה.
  - המטרה היא להשוות האם דפוס גודל החבילות של התעבורה החדשה דומה לאפליקציות רגילות או שונה באופן משמעותי.

חלק 1:

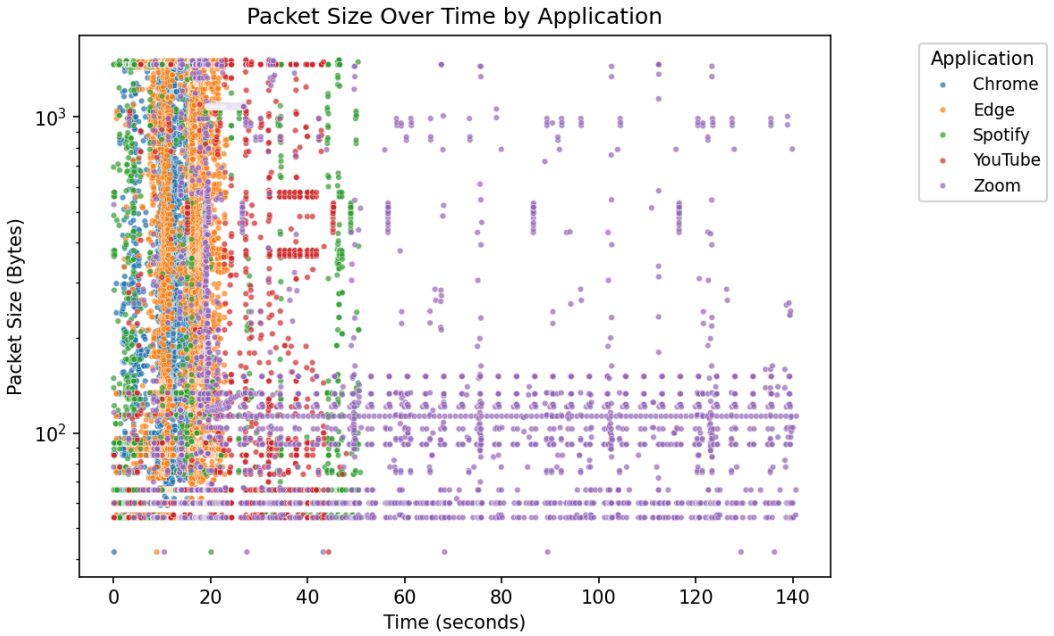
ניתוח גרף 1: גודל חבילה לאורך זמן (Packet Size Over Time)

לצורך ניתוח דפוסי התעבורה, יוצגו שני גרפים המשווים את גודל החבילות לאורך זמן.
הגרף הראשון מציג את התנהגות חמש האפליקציות הנבדקות (Chrome, Edge, Spotify, YouTube, Zoom) ללא קבוצת הנתונים החדשה (נלקח מתשובתינו לשאלה 3), ובכך מאפשר לזהות את המאפיינים הטבעיים של כל אפליקציה.
הגרף השני כולל גם את קבוצת הנתונים החדשה (chrome_spotify_attacker.csv), כדי לבחון כיצד נוכחותה משפיעה על דפוסי התעבורה.

באמצעות השוואת שני הגרפים, ניתן לנתח האם התעבורה החדשה משתלבת באופן טבעי בתוך האפליקציות הקיימות או מציגה מאפיינים המעידים על פעילות חריגה.

לצורך ניתוח דפוסי התעבורה, יוצגו שני גרפים המשווים את גודל החבילות לאורך זמן.
הגרף הראשון מציג את התנהגות חמש האפליקציות הנבדקות (Chrome, Edge, Spotify, YouTube, Zoom) ללא קבוצת הנתונים החדשה (נלקח מתשובתינו לשאלה 3), ובכך מאפשר לזהות את המאפיינים הטבעיים של כל אפליקציה.
הגרף השני כולל גם את קבוצת הנתונים החדשה (chrome_spotify_attacker.csv), כדי לבחון כיצד נוכחותה משפיעה על דפוסי התעבורה.

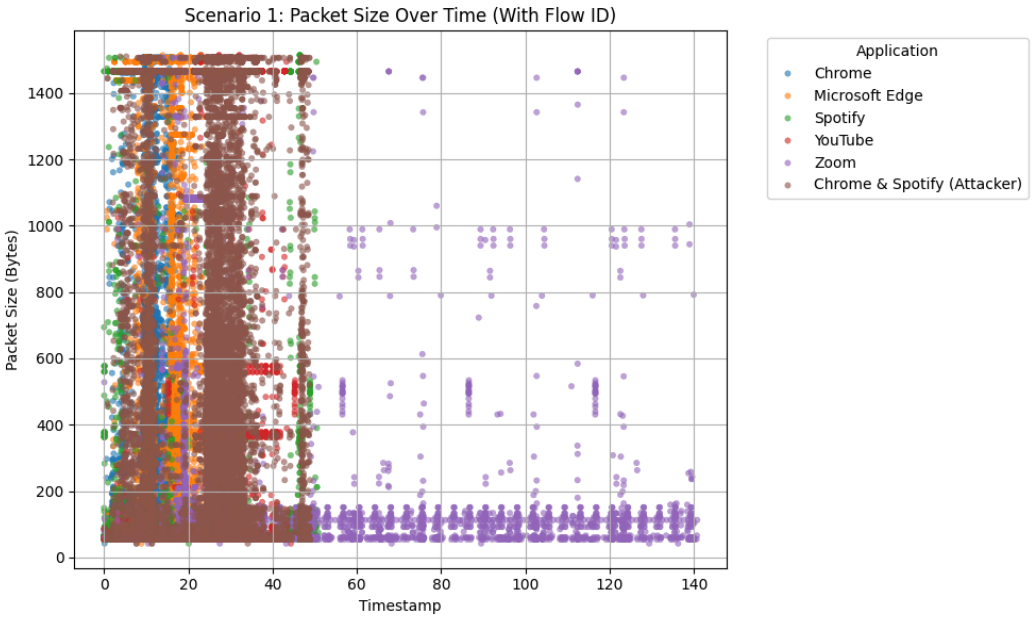
באמצעות השוואת שני הגרפים, ניתן לנתח האם התעבורה החדשה משתלבת באופן טבעי בתוך האפליקציות הקיימות או מציגה מאפיינים חריגים המעידים על פעילות



גרף זה מציג את גודל החבילות (בבתים) לאורך זמן עבור חמש האפליקציות:
Chrome, Edge, Spotify, YouTube ו-Zoom.
המטרה היא לזהות דפוסי תעבורה רגילים ולבחון כיצד כל אפליקציה שולחת חבילות לאורך ציר הזמן.

ניתוח ראשוני מהגרף:

- Chrome ו-Edge מציגים שליחת חבילות בתדירות גבוהה, עם פיזור אחיד של גדלים שונים.
- Spotify ו-YouTube שולחות חבילות גדולות יותר (לרוב מעל 1000 בתים), מה שמעיד על סטרימינג של מדיה.
- Zoom מציג חבילות קטנות באופן עקבי, עם מקבצים של חבילות קטנות לאורך כל זמן התעבורה, כנראה עקב תקשורת רציפה בווידאו או קול.



גרף זה מציג את אותן חמש אפליקציות יחד עם התעבורה החדשה (chrome_spotify_attacker.csv).
המטרה היא לזהות שינויים בדפוסי התעבורה כאשר מוסיפים את קבוצת הנתונים החדשה, ולראות האם קיימת חריגה משמעותית.

השוואה בין הגרפים וממצאים:

1. גודל החבילה הממוצע של התוקף (609.74 בתים) גבוה יותר מזה של Chrome (574.58 בתים), אך עדיין בטווח הדומה לדפדפן רגיל.
2. ברבעון העליון (75%) של גודל החבילות, התוקף מגיע ל-1465 בתים, בדומה לאפליקציות אחרות, מה שעשוי להעיד על שימוש במגבלת MTU סטנדרטית.
3. התוקף שולח יותר חבילות גדולות בתדירות גבוהה בהשוואה ל-Chrome הרגיל, דבר שעשוי להעיד על תקשורת מתמשכת עם נפח גבוה יותר.
4. האם התוקף מחקה אפליקציה מסוימת?
   - מבחינת גודל חבילה – התוקף קרוב ל-Chrome, אך עם סטייה קטנה למעלה.
   - מבחינת תדירות השליחה – נראה שהתוקף שולח יותר חבילות בצפיפות גבוהה מאשר Chrome רגיל, מה שעשוי להוות אינדיקציה לפעילות לא טבעית.
5. Zoom ממשיך להציג חבילות קטנות ויציבות, ללא שינוי משמעותי בנוכחות התוקף, מה שמרמז כי אין חפיפה בין תעבורת התוקף לזו של Zoom.

<u>מסקנות כלליות מההשוואה</u>

- התוקף מחקה את דפוסי הזרימה של Chrome, אך עם חבילות מעט גדולות יותר ותדירות שליחה גבוהה יותר.
- ייתכן שהנתונים הנוספים בחבילות התוקף קשורים להעברת מידע מוצפן או שליטה מרחוק.
- לא נראית התאמה ברורה ל-Zoom או YouTube, מה שמחזק את ההשערה שהתוקף מנסה להשתלב בתעבורת דפדפן ולא בתעבורת סטרימינג.
- בכדי לבדוק אם יש באמת פעילות חשודה, ניתן לנתח את זמני ההגעה בין חבילות (Inter-Arrival Time) ולראות האם קיימים פערים לא טבעיים.

בכדי להבין את דפוסי התעבורה הרגילים, הופק הניתוח הסטטיסטי הבא לכל אחת מהאפליקציות, כולל התעבורה החדשה:

```
=== Packet Size Statistics by Application ===
                              count        mean  ...     75%     max
Application                                      ...
Chrome                      17175.0  574.579854  ...  1465.0  1506.0
Chrome & Spotify (Attacker) 57650.0  609.739410  ...  1465.0  1510.0
Microsoft Edge              14301.0  520.356339  ...  1274.0  1514.0
Spotify                      3024.0  704.629630  ...  1465.0  1514.0
YouTube                      7508.0  917.263719  ...  1466.0  1514.0
Zoom                         5555.0  554.656526  ...  1078.0  1506.0

[6 rows x 8 columns]
```

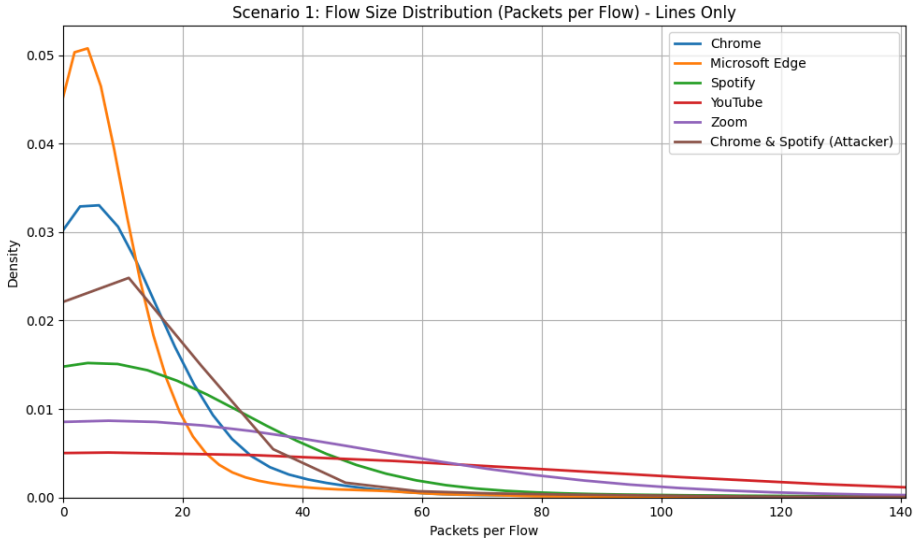
הטבלה מציגה את הסטטיסטיקות של גודל החבילות לפי אפליקציה, כולל ממוצע, רבעון עליון (75%) וגודל החבילה המקסימלי.

<u>מסקנות עיקריות</u>

- התעבורה החדשה דומה בהתנהגותה ל- Chrome אך שולחת מספר גבוה יותר של חבילות ובממוצע מעט גדול יותר.
- Edge מציג מאפייני תעבורה שונים מהתעבורה החדשה, כך שנראה שאינה משתמשת באפליקציה זו.
- הופעת חבילות גדולות בתדירות גבוהה יותר בתעבורה החדשה עשויה להעיד על שליחת נתונים מבוקרת או הצפנה.

ניתוח גרף 2: התפלגות גודל הזרימה (Flow Size Distribution)

גרף זה מציג את מספר החבילות בכל זרימה עבור חמש האפליקציות הנבדקות יחד עם קבוצת הנתונים החדשה (Chrome & Spotify – Attacker).
המטרה היא להבין כיצד גודל הזרימה מתפלג ולבחון האם הזרימות של התעבורה החדשה תואמות תואמות את דפוסי האפליקציות הרגילות או מציגות מאפיינים שונים.



<u>מסקנות כלליות מההשוואה</u>

- ההתפלגות הכללית מציגה מגמה שבה רוב הזרימות מכילות מספר קטן של חבילות, עם ירידה הדרגתית ככל שמספר החבילות בזרימה עולה.
- דפדפנים (Chrome ו-Microsoft Edge) מציגים זרימות קצרות יותר, עם ריכוז גבוה של חיבורים המכילים מספר נמוך של חבילות.
- אפליקציות מדיה (Spotify ו-YouTube) מציגות זרימות ארוכות יותר, דבר המתאים למאפייני ההזרמה הרציפה שלהן.
- Zoom מציג התפלגות שטוחה יותר, המעידה על תקשורת רציפה ומתמשכת לאורך זמן.
- התעבורה החדשה מציגה התפלגות דומה לזו של Chrome, אך עם זנב ארוך יותר, כלומר יש לה יותר זרימות ארוכות מהממוצע של הדפדפן הרגיל.
- בהשוואה ל-Edge, התעבורה החדשה מציגה זרימות ארוכות יותר, מה שמעיד על תעבורה מתמשכת יותר מאשר דפדפן רגיל.
- ההתפלגות של התעבורה החדשה קרובה יחסית ל-Spotify, אך עם נטייה גבוהה יותר לחיבורים בינוניים-ארוכים.

בכדי להבין את דפוסי התעבורה הרגילים, הופק הניתוח הסטטיסטי הבא לכל אחת מהאפליקציות, כולל התעבורה החדשה:

```
=== Flow Size (Packets per Flow) Statistics by Application ===
                              count       mean         std  ...  50%   75%     max
Application                                                ...
Chrome                       1370.0  12.536496   37.783810  ...  3.0  14.0   579.0
Chrome & Spotify (Attacker)  3858.0  14.942976   66.592635  ...  7.0  14.0  2326.0
Microsoft Edge               1572.0   9.097328   24.227204  ...  3.0  10.0   405.0
Spotify                       185.0  16.345946   67.901057  ...  6.0   7.0   850.0
YouTube                       331.0  22.682779  247.150560  ...  5.0   6.0  4291.0
Zoom                          208.0  26.706731  123.994411  ...  4.0  15.0  1320.0

[6 rows x 8 columns]
```

הטבלה מציגה את הסטטיסטיקות המרכזיות של מספר החבילות בכל זרימה עבור חמש האפליקציות הרגילות ולצד התעבורה החדשה (Chrome & Spotify – Attacker).

הנתונים כוללים את הממוצע, החציון (50%), הרבעון העליון (75%) והערך המקסימלי, מה שמאפשר להשוות את דפוסי הזרימה בין האפליקציות השונות ולבחון את השתלבות התעבורה החדשה ביחס אליהן.

<u>מסקנות עיקריות</u>

- התעבורה החדשה דומה לדפדפן Chrome מבחינת הרבעון העליון (75%), אך בעלת חציון גבוה יותר, מה שמעיד על זרימות ארוכות יותר מהממוצע של דפדפן רגיל.
- ההבדלים המרכזיים בין התעבורה החדשה ל-Chrome הרגיל באים לידי ביטוי באורך הזרימה – היא מציגה יותר זרימות בינוניות וארוכות.

<u>מסקנה סופית: באילו אפליקציות התעבורה החדשה משתמשת?</u>

בהתבסס על גודל החבילות לאורך זמן ומבנה הזרימות, ניתן לקבוע שהתעבורה החדשה דומה בעיקר לדפדפן Chrome ול-Spotify, אך עם מאפיינים ייחודיים.

דמיון ל-Chrome:

- הרבעון העליון של גודל החבילה (1465 בתים) והממוצע קרובים ל-Chrome, מה שמעיד על דפוס תעבורה דומה.
- הרבעון העליון של מספר החבילות לזרימה (14) זהה ל-Chrome, אך החציון גבוה יותר (7 לעומת 3), מה שמעיד על זרימות מעט ארוכות יותר.

- הערך המקסימלי של 2326 חבילות לזרימה גבוה משמעותית משל Chrome, דבר שעשוי להעיד על תעבורה אינטנסיבית יותר.

דמיון ל-Spotify:

- הממוצע של מספר החבילות לזרימה (14.94) קרוב מאוד ל-(16.34) Spotify, מה שמעיד על דפוס זרימה דומה.
- החציון (7) דומה ל -6) Spotify), אך המקסימום גבוה בהרבה (2326 לעומת 850), מה שמעיד על זרימות חריגות בארכן.

אי התאמה ל-Edge, Zoom ו-YouTube:

- Edge מציג מספר חבילות נמוך יותר, כך שהתעבורה החדשה אינה תואמת את דפוסיו.
- Zoom מציג ממוצע חבילות גבוה יותר (26.7), מה שמעיד על תקשורת רציפה יותר.
- YouTube כולל זרימות קצרות מאוד או ארוכות במיוחד (עד 4291 חבילות), מה שאינו תואם לתעבורה החדשה.

מסקנה: התעבורה החדשה משתמשת בעיקר ב-Chrome עם מאפיינים נוספים של Spotify, אך אינה תואמת ל-Edge, Zoom או YouTube.

חלק 2

בחלק זה נבחנים שני גרפים נוספים:

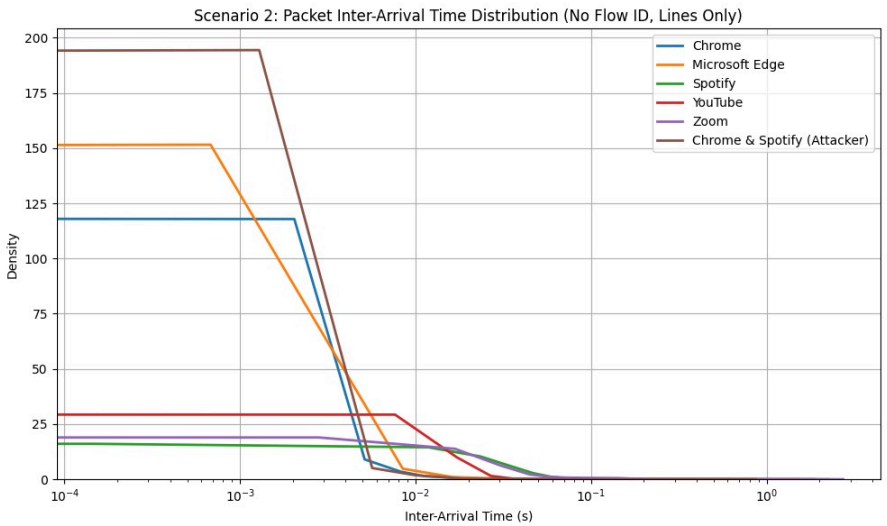
1. **התפלגות זמני ההגעה בין חבילות (Inter-Arrival Time Distribution)**
2. **התפלגות גודל הזרימה (Packet Size Distribution)**

מטרת הניתוח היא להעמיק את ההבנה לגבי מאפייני התעבורה החדשה, לבחון את הדמיון בינה לבין האפליקציות שנבדקו, ולהעריך עד כמה ניתן לזהות את אופי השימוש בהתבסס על מאפייני התעבורה הסטטיסטיים בלבד.

ניתוח גרף 3: התפלגות זמני ההגעה בין חבילות (Inter-Arrival Time Distribution) ללא Flow ID

גרף זה מתאר את פיזור זמני ההגעה בין חבילות לכל אפליקציה.
לכל סוג יישום קיימים דפוסים אופייניים של זמני הגעה בין חבילות:

- דפדפנים (Chrome, Edge) מאופיינים בפרצי שליחה עם זמני השהיה משתנים.
- שירותי מדיה (Spotify, YouTube) מייצרים זרימות רציפות עם זמני הגעה אחידים יחסית.
- אפליקציות תקשורת בזמן אמת (Zoom) שולחות חבילות בקצב קבוע עם השהיות מינימליות.



מסקנות כלליות מההשוואה

- ההתפלגות הכללית מציגה מגמה שבה רוב החבילות מגיעות בהפרשי זמן קצרים, עם ירידה חדה ככל שההפרש בין החבילות גדל.
- דפדפנים (Chrome, Microsoft Edge) מציגים זמני הגעה קצרים יותר, מה שמעיד על שליחת חבילות בצפיפות גבוהה מאוד.
- שירותי סטרימינג (Spotify, YouTube) מציגים זמני הגעה ממוצעים עם שונות גבוהה, דבר המתאים לאופיים של שירותי מדיה.
- Zoom מציג זמני הגעה גדולים יותר, דבר המעיד על שליחת חבילות בקצב יציב אך לא מהיר כמו דפדפן או סטרימינג.
- התעבורה החדשה (Chrome & Spotify – Attacker) מציגה זמני הגעה דומים ל-Chrome אך עם פיזור שונה, דבר שיכול לרמוז על תעבורה לא סטנדרטית או ניסיון חיקוי של זרמים שונים.
- זמני ההגעה בתעבורה החדשה קצרים יותר מהממוצע של Chrome, אך ארוכים משל Zoom, מה שמעיד על קצב שליחה גבוה יחסית אך לא עקבי כמו זה של אפליקציות תקשורת בזמן אמת.

בכדי להבין את דפוסי התעבורה הרגילים, הופק הניתוח הסטטיסטי הבא לכל אחת מהאפליקציות, כולל התעבורה החדשה:

```
=== Inter-Arrival Time Statistics by Application ===
                              count      mean  ...       75%       max
Application                                    ...
Chrome                      17174.0  0.001039  ...  0.000276  0.606544
Chrome & Spotify (Attacker) 57649.0  0.000849  ...  0.000149  0.863896
Microsoft Edge              14300.0  0.001643  ...  0.000349  1.533355
Spotify                      3023.0  0.016823  ...  0.000835  2.181174
YouTube                      7507.0  0.006460  ...  0.000040  1.849140
Zoom                         5554.0  0.025338  ...  0.007148  2.672106

[6 rows x 8 columns]
```

הטבלה מציגה את הסטטיסטיקות המרכזיות של זמני ההגעה בין חבילות עבור חמש האפליקציות הרגילות ולצד התעבורה החדשה, תוך התמקדות בערכים מרכזיים כגון מספר החבילות, הממוצע, החציון, הרבעון העליון (75%) והערך המקסימלי. נתונים אלו מאפשרים להשוות את דפוסי שליחת החבילות בין האפליקציות השונות ולבחון האם התעבורה החדשה משתלבת באופן טבעי בדפוסי האפליקציות או מציגה מאפיינים ייחודיים שעשויים להעיד על חריגות.

מסקנות עיקריות:

- התעבורה החדשה דומה ביותר ל-Chrome מבחינת זמני ההגעה, שכן הממוצע והרבעון העליון שלה קרובים מאוד לערכים של Chrome. יחד עם זאת, ניכרת צפיפות גבוהה יותר של חבילות בפרצי השליחה, כפי שניתן לראות מהערך הנמוך של הרבעון העליון (0.000149 שניות לעומת 0.000276 ב-Chrome). נתון זה מעיד שהתעבורה החדשה מתאפיינת בקצב שליחה מהיר יותר בהשוואה לדפדפן רגיל.
- התעבורה החדשה אינה מתאימה לשירותי מדיה כמו Spotify ו-YouTube, שכן זמני ההגעה הממוצעים בה נמוכים משמעותית (0.000849 שניות) בהשוואה ל-YouTube (0.006460) ול-Spotify (0.016823). שירותי סטרימינג נוטים לשלוח חבילות בקצב קבוע עם מרווחי זמן ארוכים יותר, בעוד שהתעבורה החדשה מציגה שליחה מהירה מאוד של חבילות, שאינה תואמת למאפייני הזרמת מדיה.
- Zoom אינו רלוונטי, מאחר והתעבורה החדשה אינה מציגה קצב שליחה אחיד, כפי שמאפיין אפליקציות תקשורת בזמן אמת. בעוד Zoom מתאפיין במרווחי זמן יציבים יחסית בין חבילות (עם ממוצע של 0.025338 שניות ורבעון עליון של 0.007148), התעבורה החדשה מציגה פערים קטנים יותר וזמני הגעה בלתי סדירים, דבר השונה לחלוטין מהאופן שבו מתנהלת תעבורת וידאו.

באילו אפליקציות התעבורה החדשה משתמשת?

- הניתוח מצביע על כך שהתעבורה החדשה משתמשת בדפדפן Chrome, אך בפרצי שליחה מעט צפופים יותר מהממוצע של דפדפן רגיל.
- לעומת זאת, אין עדות לשימוש בשירותי מדיה כמו Spotify ו-YouTube, וכן לא באפליקציות תקשורת כמו Zoom, מאחר שהתעבורה אינה מציגה את דפוסי השליחה הקבועים האופייניים להן.

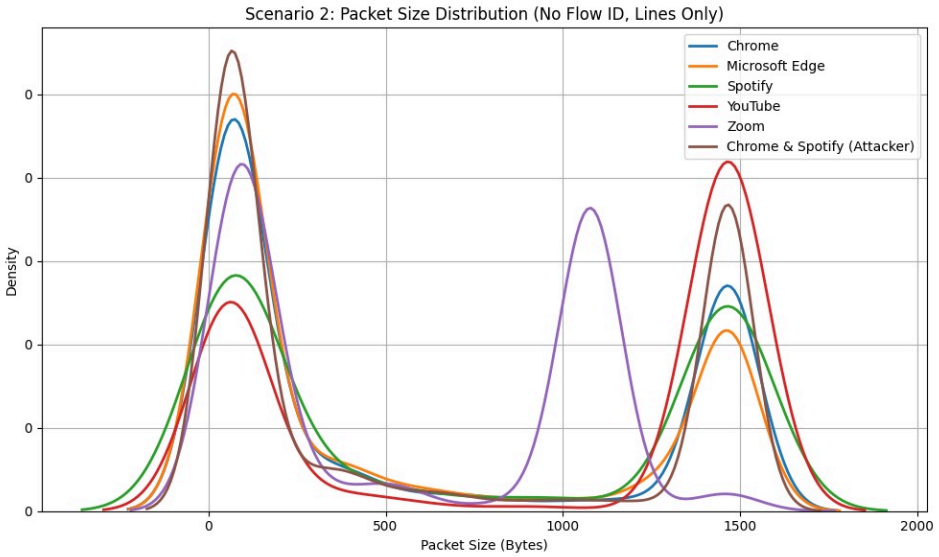
אי התאמה ל-Edge, Zoom ו-YouTube

- Microsoft Edge מציג זמני הגעה דומים ל-Chrome אך עם פחות שונות, ולכן התעבורה החדשה אינה תואמת לחלוטין לדפוסיו.
- Zoom מציג זמני הגעה גדולים ויציבים, ולכן אינו מתאים לדפוסי התעבורה החדשה.
- YouTube כולל זמני הגעה נמוכים מאוד או גבוהים במיוחד (כתוצאה מהזרמת וידאו במקטעים), מה שאינו תואם לחלוטין לנתונים החדשים.

מסקנה: התעבורה החדשה מדמה בעיקר את Chrome עם מאפיינים נוספים של Spotify, אך אינה תואמת ל-Edge, Zoom או YouTube.

ניתוח גרף 4: התפלגות גודל החבילות (Packet Size Distribution)

גרף זה מציג את התפלגות גדלי החבילות של כל אפליקציה, כולל התעבורה החדשה. מטרת הניתוח היא לזהות דפוסים ייחודיים בגודל החבילות ולבחון האם התעבורה החדשה מתאימה למבנה חבילות של אחת מהאפליקציות שנבדקו.



<u>מסקנות כלליות מההשוואה</u>

דפדפנים (Chrome, Edge) – מציגים שתי פסגות:

- חבילות קטנות (100-200 בתים) לבקרת חיבור (SYN, ACK, HTTP Requests).
- חבילות גדולות (1400-1500 בתים) להעברת תוכן (טעינת דפים, קבצים).
- Edge דומה מאוד ל-Chrome, עם דפוס תעבורה כמעט זהה.

שירותי סטרימינג (Spotify, YouTube) – משתמשים בעיקר בחבילות גדולות:

- Spotify: חבילות בינוניות (500-700 בתים) לצד חבילות גדולות, מצביע על דחיסה דינמית.
- YouTube: כמעט כל החבילות הן 1500 בתים, משקפות סטרימינג רציף וללא פיזור חבילות.

Zoom – דפוס ייחודי עם שתי פסגות נפרדות:

- חבילות קטנות (~200 בתים) לבקרת חיבור.
- חבילות בינוניות (~1000 בתים) מותאמות לשיחות וידאו, בניגוד לשירותי סטרימינג גדולים.

התעבורה החדשה (Chrome & Spotify – Attacker)

- דומה מאוד לדפדפנים, עם שתי הפסגות האופייניות (100-200 בתים ו-1400-1500 בתים).
- מציגה שונות רחבה יותר, מה שמעיד על ניסיון לשלב מספר שירותים או לערבב תעבורה.

התעבורה החדשה דומה בעיקר ל-Chrome ו-Edge, אך עשויה להכיל גם מאפיינים של Spotify בשל פיזור חבילות רחב יותר. יתכן שמדובר בניסיון חיקוי או בהסוואת תעבורה.

<u>להלן מספר אמצעים בהם ניתן לנקוט על מנת למזער את יכולת התוקף לנתח תעבורה ולזהות באילו אתרים או אפליקציות המשתמש גולש, גם כאשר התוכן מוצפן או אנונימי:</u>

1. **הוספת ריפוד (Padding) וגודל חבילות אחיד:**
   - מרפדים את התעבורה כך שכל חבילה תגיע לגודל קבוע או לגודל רנדומלי שנמצא בטווח קטן מאוד. כך, גם אם תוקף רואה את גודל החבילה, הוא לא יכול להסיק מהו התוכן האמיתי.
2. **עיצוב תעבורה והסתרת זמני שידור (Traffic Shaping & Timing Obfuscation):**
   - הוספת עיכובים אקראיים או שידור חבילות בקצב קבוע, כך שהתבניות בזמן השידור מיטשטשות ואינן ניתנות לזיהוי בקלות.
3. **מיזוג וריבוב תעבורה:**
   - שילוב תעבורה ממספר מקורות או יישומים יחד לתוך ערוץ יחיד, כך שהדפוסים של כל יישום מעורבבים ולא ניתנים להפרדה בקלות.
4. **כיסוי תעבורה (Cover Traffic):**

- יצירת תעבורה מזויפת (dummy packets) שמטרתה להסוות את התעבורה האמיתית. כך התוקף לא יכול לזהות אילו חבילות הן אמיתיות ואילו לא.

5. **שימוש ברשתות אנונימיות (כמו VPN, Tor):**

- ניתוב התעבורה דרך רשת שמסתירה את כתובת המקור והיעד, וכך מקשה על תוקף לקשר בין הנתונים לבין המשתמש.

באמצעות אמצעים אלו ניתן להסתיר או לעוות את המאפיינים החיצוניים של התעבורה (כגון גודל, זמני שידור, ותבניות זרימה), ובכך להפחית את יכולת הניתוח של תוקף שמשתמש במידע צדדי כדי לזהות את האתרים או היישומים שהמשתמש גולש בהם.