# **פרוייקט סוף – רשתות תקשורת**

**מגישים:**

יפתח דודו הלוי- 20696447

אור ביבי- 207707613

אוראל מזין- 213435902

ישי אישון- 206318776

# תוכן העניינים:

**חלק 1:**



**חלק 2:**



**חלק 3:**

# חלק 1

1. A user reports that their file transfer is slow, and you need to analyze the transport layer to identify the potential reasons. What factors could contribute to the slow transfer, and how would you troubleshoot it?

**1. זיהוי הגורמים האפשריים ,בדיקתם ופתרונם:**

**א.** הגבלת רוחב פס (כמות הנתונים שניתן להעביר דרך ערוץ תקשורת בפרק זמן נתון.)

גורמים: החיבור מוגבל ברוחב הפס על ידי ספק האינטרנט (ISP) או מדיניות רשת מקומית , תוכנות רקע עשויות להשתמש ברוחב פס משמעותי ,אפליקציות כגון סטרימינג, משחקים מקוונים, והורדות יכולות לצרוך רוחב פס משמעותי ולהשפיע על המהירות הכללית.

בדיקה: שימוש בכלים כמו Wireshark, למדידת קצב ההעברה בפועל. בדיקת מהירות הרשת בעזרת speedtest.net.

- פתרון: **1.** הגדרת תעדוף QOS (Quality of Service) - נגדיר עדיפויות ברמת הנתב או שרת הרשת, כך שנתוני ההעברה החשובים יקבלו קדימות על פני שימושים אחרים, **2.** ניתוק מכשירים אחרים מהרשת - אם רשת משותפת מכילה מכשירים רבים שצורכים רוחב פס, ניתוק זמני שלהם יכול לשפר את ביצועי ההעברה, **3.** שימוש בקו (Ethernet) במקום (Wi-Fi) - חיבור חוטי מספק רוחב פס יציב יותר ומפחית בעיות של הפרעות ושינויים בגישה לאינטרנט.

**ב.** חביון גבוה(Latency) או עומס ברשת

גורמים: חביון (זמן בין שליחת נתונים לבין הגעתם ליעד) גבוה משפיע ישירות על מהירות ההעברה, במיוחד בפרוטוקול TCP. עומס ברשת יכול לגרום לאובדן פאקטות, מה שמחייב העברת פאקטות מחדש.

בדיקה: שימוש בפקודה ping לכתובת היעד למדידת ה-(RTT). שימוש בפקודה traceroute (Linux) או tracert (Windows) לזיהוי עיכובים בניתוב.

פתרון: שימוש בפרוטוקול UDP להעברות הרגישות למהירות. צמצום מספר ההתקנים המחוברים לרשת במקרים מסוימים, ניתן להשתמש ב-UDP במקום TCP כדי להימנע מהשהיות כלומר על פי מה שלמדנו בהרצאות הוא אינו דורש אישור קבלה (ACK) ולכן מהיר יותר אך פחות אמין.

**ג.** בעיות בשליחת ACK או מנגנון שליטה בזרימה (בשימוש בפרוטוקול TCP)

גורמים: **1.** TCP Window Size קטן מדי, השולח יאלץ להמתין הרבה זמן לפני שיוכל לשלוח נתונים נוספים, מה שיגרום להאטה בקצב ההעברה, **2.** העברת פאקטות מחדש עקב איבוד נתונים כלומר כאשר חבילות נעלמות במהלך ההעברה, השולח לא מקבל אישור (ACK), ולכן נאלץ לשדר אותן מחדש וזה גורם לעיכוב.

בדיקה: ניתוח חבילות TCP עם Wireshark .

פתרון: כוונון חלון התעבורה (TCP Window Scaling). שימוש באופטימיזציה כמו SACK( Selective Acknowledgement) המאפשר למקבל לדווח בדיוק אילו פאקטות התקבלו ואילו חסרות, כך שהשולח ישלח רק את החלקים החסרים במקום לשדר הכל מחדש.

ד. קצב שגיאות גבוה

גורמים: בעיות בקו התקשורת כמו Wi-Fi חלש או כבל פגום עלולות לגרום לנפילות פאקטות , מה שמכריח את TCP לשלוח אותן מחדש.

בדיקה: ניתן להשתמש בפקודת ping כדי לבדוק אם יש איבוד מנות (Packet Loss) במהלך ההעברה, לכידת מנות עם Wireshark לבדיקת שגיאות TCP.

פתרון: ניתן לאתר את מקור הבעיה ולשפר את היציבות ע"י מעבר לחיבור קווי, החלפת ציוד ישן.

ה. מגבלות לקוח:

גורמים: עומס על המעבד או על הדיסק הקשיח עלול לגרום לעיכובים בעיבוד הנתונים.

בדיקה: בדיקה במנהל המשימות את העומס על המעבד.

פתרון :אם עומס המעבד או זיכרון ה RAM-גבוהים, ניתן לשדרג את החומרה או למזער שימוש במעבד.

2. Analyze the effects of TCP's flow control mechanism on data transmission. How would it impact performance when the sender has significantly higher processing power than the receiver?

**פתרון שאלה 2:**

- בקרת הזרימה מבוססת על ערך הנקרא (**TCP Receive Window (rwnd**, המייצג את כמות הנתונים שהמקבל מסוגל לקלוט ולעבד בכל רגע נתון.
- גודל ה-rwnd מתעדכן במסגרת ה-(ACK), שבו המקבל מאותת לשולח על נתונים שהתקבלו ומודיע כמה מקום פנוי נשאר בזיכרון הקלט שלו. גודל החלון יכול לגדול ולקטון.
- אם הזיכרון של המקבל מתמלא, הוא מקטין את חלון הקבלה, מה שגורם לשולח להאט את קצב השידור. במקרה קיצוני, אם הזיכרון מלא לחלוטין, ה-rwnd יוגדר ל-0, מצב המכונה Zero Window Condition, שבו השולח עוצר לחלוטין את שליחת הנתונים עד שהמקבל ישחרר מקום.

השפעת בקרת הזרימה של TCP כאשר השולח חזק משמעותית מהמקבל - כאשר לשולח יש כוח עיבוד גבוה בהרבה מהמקבל, קיים חוסר איזון בתקשורת, שבו השולח מסוגל לשלוח נתונים במהירות גבוהה מאוד, אך המקבל אינו מסוגל לעבד אותם באותו קצב. במצב זה, מנגנון בקרת הזרימה של( TCP (Flow Control נכנס לפעולה ומגביל את קצב ההעברה כדי למנוע עומס על המקבל. מגבלה זו יכולה להשפיע על ביצועי הרשת בכמה דרכים:

**1.** האטת קצב ההעברה בגלל הקטנת חלון הקבלה (TCP Receive Window Shrinking)

- TCP משתמש במנגנון Receive Window (rwnd) כדי שהמקבל יודיע לשולח כמה נתונים הוא מסוגל לקבל ולעבד, אם המקבל לא מספיק לעבד את הנתונים במהירות שהשולח מסוגל לשלוח, הוא יקטין את גודל החלון כדי למנוע עומס על הזיכרון והמעבד שלו, ככל שהחלון קטן יותר השולח ישלח פחות נתונים בכל מחזור,מה שיכול להאט משמעותית את הביצועים ולגרום לקצב העברה נמוך יותר מזה שהרשת מסוגלת לתמוך בו.

**2.** עיכובים בתקשורת

- כאשר חלון הקבלה קטן, השולח צריך להמתין לעיתים קרובות יותר לאישורים (ACK) לפני שהוא יכול לשלוח מנות נוספות, השהיות אלו מגדילות את זמן התגובה (Latency) ומקטינות את היעילות של השימוש ברוחב הפס, במיוחד בחיבורים בעלי זמן התגובה (Latency) גבוה (כגון רשתות ארוכות טווח)

**3.** עצירת העברה במצב חלון אפס (Zero Window Condition)

- אם המקבל מגיע לנקודת עומס מלאה, הוא עשוי להגדיר את גודל ה- TCP Receive Window ל-0, מצב זה נקרא Zero Window Condition, והוא גורם לכך שהשולח יעצור לחלוטין את שליחת הנתונים עד שהמקבל ישלח עדכון שהוא מוכן לקבל מידע נוסף.

**4.** ניצול לא יעיל של רוחב הפס

- כאשר לשולח יש יכולת עיבוד גבוהה והוא מחובר לרשת מהירה, הוא מסוגל לנצל קצבי העברה גבוהים אבל, אם המקבל מוגבל מבחינת משאבים, בקרת הזרימה של TCP תמנע מהשולח לשלוח נתונים בקצב האפשרי והטוב ביותר עבורו, ולכן המשמעות היא בזבוז של רוחב פס ברשת, כיוון שהשולח נאלץ להאט את קצב ההעברה למרות שהרשת עצמה מסוגלת לתמוך במהירות גבוהה יותר.

לסיכום :נוכל לומר כי מבחינה כללית כאשר לשולח יש עוצמת עיבוד גבוהה משמעותית מהמקבל, בקרת הזרימה של TCP תמנע ממנו לנצל את מלוא הקיבולת של הרשת כדי להגן על המקבל. כתוצאה מכך, קצב ההעברה עלול לרדת, ההשהיה בתקשורת עלולה לעלות, וייתכנו מצבים של עצירת ההעברה לחלוטין במקרים קיצוניים. על מנת לשפר את הביצועים, ניתן לבצע אופטימיזציה של TCP, לשפר את משאבי המקבל או להשתמש בפרוטוקולים מתקדמים שמתמודדים טוב יותר עם הבדלים ביכולת עיבוד בין שני הצדדים.

3. Analyze the role of routing in a network where multiple paths exist between the source and destination. How does the path choice affect network performance, and what factors should be considered in routing decisions?

**תפקיד הנתב ברשת שבה קיימים מסלולים מרובים בין המקור ליעד-** במצב שבו קיימים מסלולים מרובים בין המקור ליעד, תהליך הניתוב ממלא תפקיד חשוב ונחוץ בקביעת אופן העברת המידע דרך הרשת. תהליך זה נועד לבחור את המסלול האופטימלי ביותר עבור חבילות הנתונים, תוך התחשבות בגורמים כמו עלות נתיב, עומס רשת, זמינות קישורים, והשהיות.

**תפקידי הניתוב ברשת עם מסלולים מרובים:**

בחירת הנתיב היעיל ביותר- כאשר קיימים מספר מסלולים אפשריים, הנתב נדרש לקבוע באיזה נתיב להשתמש על מנת להעביר את הנתונים בצורה מיטבית לצורך כך, פרוטוקולי ניתוב משתמשים באלגוריתמים כמו דייקסטרה (Dijkstra) ו-Bellman-Ford אשר למדנו בהרצאות, אשר שמחשבים את הדרך הקצרה ביותר לכל יעד ברשת, כאשר כל נתב מחשב עבור הרשת בה הוא נמצא ולא באופן כללי את כל המסלול.

מניעת כשלים - פרוטוקול BGP מסייע למניעת כשלים בכך שהוא מאפשר ל- subnets לפרסם את קיומם וליידע את הרשת על מסלולים חלופיים במקרה של כשל ובנוסף הוא מספק לכל AS התייחסות ל-(iBGP eBGP, eBGP ) מחליף מידע בין מערכות אוטונומיות (ASes) כדי להבטיח שניתן יהיה לנתב תנועה גם אם נתיב אחד כשל, iBGP מפיץ את המידע בתוך AS כדי שכל הנתבים הפנימיים יכירו את הנתיבים הזמינים.

התמודדות עם השהיות וניתוב דינמי- פרוטוקול OSPF הוא פרוטוקול ניתוב דינמי מבוסס מצב קישור (Link-State), ולכן הוא יעיל בהתמודדות עם השהיות ברשתות גדולות, כל נתב מחזיק טופולוגיה מלאה של האזור שבו הוא נמצא, ומחשב את הנתיב האופטימלי בעזרת אלגוריתם דייקסטרה (Dijkstra), מכיוון ש-OSPF מתעדכן בזמן אמת, הוא מתאים את הניתוב בהתאם לעומסי הרשת ומאפשר בחירת נתיבים עם השהיה מינימלית, כל נתב יודע לגשת רק ליעדים המוגדרים באזור שלו, ולכן OSPF מאפשר ניהול יעיל של מסלולים פנימיים בתוך
AS.

השפעת בחירת הנתיב על ביצועי הרשת :

בחירת הנתיב ברשת היא גורם מכריע שמשפיע על ביצועי הרשת במונחים של מהירות, השהיה (Latency), ניצול רוחב פס, אמינות ואיזון עומסים. נתיב לא אופטימלי עלול לגרום ל- אובדן פאקטות, והאטת קצב ההעברה, בעוד שבחירה חכמה של נתיב יכולה למזער עומסים ולמקסם את היעילות של הרשת.

-רוחב פס (Bandwidth) וניצול משאבים- אם נתיב מסוים עמוס מדי, קצב ההעברה דרך אותו נתיב יהיה נמוך יותר למרות שהחיבור עצמו תומך במהירויות גבוהות, בחירת נתיב עם רוחב פס פנוי גבוה מאפשרת שיפור בביצועים, בעיקר בהעברת נתונים כבדה כמו הורדת קבצים, סטרימינג והעברת מדיה גדולה.

-אובדן פאקטות ושגיאות נתיב- אם נתיב מסוים חווה אובדן מנות גבוה, חבילות נתונים יידרשו לשידור חוזר וזה מוביל לעומס נוסף והאטת התקשורת, רשתות עם עומס גבוה, הפרעות חשמליות או ציוד תקשורת מיושן עשויות לגרום לבעיות אובדן פאקטות ולבחירת נתיבים לא יעילים,(אם הנתיב שנבחר עמוס בכמות גבוהה של תעבורה, הנתבים במסלול עלולים לא להספיק לעבד ולנתב את כל החבילות בזמן וההשפעה של זה היא שכאשר נתב מקבל יותר חבילות ממה שהוא יכול לעבד, הוא נאלץ להפיל חלק מהן, נתיבים שמובילים לשרתים פופולריים עלולים להיות עמוסים בשעות השיא. אם הנתיב שבחר הפרוטוקול עובר דרך אותם שרתים, ייתכן ואובדן הפאקטות יהיה גבוה יותר.)

**מספר גורמים שיש לקחת בחשבון בהחלטות ניתוב:**

1. סוג התעבורה והיישום – יישומים עם דרישות זמן אמת (סטרימינג, משחקים, ועוד..) דורשים נתיבים עם זמן שהיה מינימלית, בעוד שהעברות קבצים דורשות רוחב פס גבוה .
2. מצב עומס הרשת – יש לוודא שהנתיב שנבחר אינו עמוס יתר על המידה, גם אם הוא קצר יותר.
3. יתירות וזמינות נתיבים – נתיבים חלופיים חייבים להיות זמינים למניעת כשלי רשת במקרה של תקלות.
4. אבטחת נתונים – יש לוודא שהנתיב לא עובר דרך קישורים חשופים לפגיעות אבטחה.

## 4. How does MPTCP improve network performance?

**MPTCP (Multipath TCP)** הוא בעצם הרחבה לפרוטוקול **TCP** המאפשר להשתמש במספר חיבורים במקביל עבור אותו חיבור TCP. בעוד ש-TCP רגיל פועל על נתיב אחד בלבד, MPTCP מאפשר למכשירים עם מספר ממשקי רשת להשתמש בכל הנתיבים הזמינים בו זמנית כדי לשפר את מהירות, אמינות וניצול הרשת.

נבין קודם את היתרונות MPTCP על פני TCP רגיל כדי שנבין את השיפור על ביצועי הרשת:

- שיפור קצב ההעברה: ב-TCP רגיל, קצב ההעברה מוגבל לרוחב הפס של נתיב יחיד, MPTCP משלב מספר חיבורים, כך שהתעבורה מנצלת את סך רוחבי הפס הזמינים.

- אמינות גבוהה יותר : אם נתיב אחד נכשל (למשל, Wi-Fi נפל), TCP רגיל מנתק את החיבור כולו עד להתאוששות, ב-MPTCP, התעבורה פשוט מועברת לנתיב השני באופן שקוף למשתמש, מה שמבטיח שהחיבור לא ינותק, תכונה זו קריטית במיוחד עבור אפליקציות כמו שיחות, סטרימינג ומשחקים מקוונים.

- ניצול טוב יותר של משאבי הרשת: מפזר את התעבורה על פני מספר רשתות שונות, כך שהעומס לא נופל על קישור יחיד, תכונה זו מפחיתה עומסים על רשתות צפופות, במיוחד כאשר הרבה משתמשים מחוברים לאותה אנטנה סלולרית.

-מעבר חלק בין רשתות: משתמשים בטלפונים ניידים לעיתים קרובות עוברים בין רשתות Wi-Fi לרשתות סלולריות, TCP רגיל דורש פתיחת חיבור חדש במקרה של מעבר כזה, מה שיוצר
השהיה.
MPTCP מאפשר המשכיות חיבור – אם משתמש עובר מ-Wi-Fi ל-G4, הנתונים פשוט זורמים דרך הנתיב השני וגם בלי שום הפרעה.

לאחר שהבנו את היתרונות של MPTCP על פני TCP רגיל, ניתן להסביר כיצד בדיוק הוא משפר את ביצועי הרשת:

- ניצול מקסימלי של משאבי הרשת MPTCP מאפשר שימוש במספר חיבורים בו-זמנית, כמו Wi-Fi ו-G4, כך שהרשת מנצלת את כל רוחבי הפס הזמינים ולא מוגבלת לקישור יחיד, וכתוצאה מכך, מהירות ההעברה גבוהה יותר ואין צוואר בקבוק שנגרם עקב עומס על נתיב אחד בלבד.
- הפחתת זמן השהיה ב-MPTCP, ניתן לשלוח נתונים בכמה נתיבים במקביל, וכך לבחור את הנתיב המהיר ביותר ולהקטין השהיות, מה שמשפר את הביצועים ביישומים רגישים לזמן אמת כמו סטרימינג, שיחות וידאו ומשחקים מקוונים.
- **איזון עומסים חכם** כאשר יש עומס על נתיב אחד, MPTCP מפצל את הנתונים בין מספר נתיבים חלופיים, מה שמונע מצב שבו קישור אחד עמוס והשני ריק כך, התקשורת הופכת **ליעילה יותר** ואין צורך לחכות לפרק זמן ממושך כדי לקבל אישור (ACK) על כל חבילה.
- **מעבר רציף בין רשתות -** מכשירים ניידים נעים בין רשתות שונות, MPTCP מאפשר מעבר חלק בין הרשתות מבלי שהחיבור ייפסק, מה שתורם לחוויית משתמש רציפה ללא ניתוקים.

**ולכן נאמר לסיכום**: כי MPTCP משפר את ביצועי הרשת על ידי ניצול מספר נתיבים בו-זמנית, הפחתת עומסים, הפחתת השהיות, משפר את המעבר הרציף בין רשתות, כל אלו מבטיחים מהירות גבוהה יותר, יציבות משופרת וניצול יעיל יותר של הרשת בהשוואה ל-TCP רגיל.

5. You are monitoring network traffic and notice high packet loss between two routers. Analyze the potential causes for packet loss at the Network and Transport Layers and recommend steps to resolve the issue.

**גורמים פוטנציאליים לאיבוד מנות בשכבת הרשת (Network Layer)**

עומס יתר ברשת- כאשר הנתבים מתמודדים עם כמות נתונים גבוהה מעבר לקיבולת שלהם, הם עלולים להתחיל להפיל מנות כדי למנוע קריסת רשת, עומסים כאלה נגרמים מהעברת מידע מסיבית, שימוש לא מבוקר בתעבורה, או חוסר איזון בניתוב.

בעיות בניתוב- טבלאות ניתוב לא מעודכנות או שגויות עלולות לגרום למנות לא להגיע ליעדן, תקלות בפרוטוקולי ניתוב דינמיים (כגון RIP, OSPF, BGP שלמדנו בהרצאות) עלולות להוביל לניתוב דרך נתיבים עמוסים או מנותקים.

קישורים לא יציבים או פגומים - כבלי רשת פגומים, ניתוקים פיזיים, או שיבושי תדרים ברשתות אלחוטיות עלולים לגרום להפסד פאקטות באופן די תדיר.

**גורמים פוטנציאליים לאיבוד מנות בשכבת התעבורה (Transport Layer)**

-ניהול עומס- TCP - (TCP Congestion Control) כולל מנגנוני ניהול עומסים כמו Slow Start

1. כאשר מתרחשת איבוד פאקטות , TCP מתייחס לכך כאיתות לעומס ברשת.
2. TCP מפעיל את מנגנון בקרת העומסים (Congestion Control), שגורם להפחתת קצב ההעברה כדי למנוע עומס נוסף.
3. אם הרשת עמוסה מאוד, TCP עלול להפחית את קצב ההעברה בצורה דרסטית, מה שעלול להאט את הביצועים משמעותית. אך נשים לב כי לא תמיד אובדן פאקטות תמיד מפעיל את Congestion Control אלא זה תלוי בסוג אובדן פאקטות : אובדן מנות בגלל עומס כן מפעיל את מנגון ניהול העומס, אך אובדן פאקטות בגלל שגיאות פיזיות בקו התקשורת עדיין עלול להגיב בירידת קצב, אך זה לא נובע מעומס אלא מהפרעות בתקשורת.

# חלק 2

**ניתוח המאמרים לפי השאלות הנדרשות:**

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

1. התרומה העיקרית של המאמר:

זהו מחקר ראשון מסוגו אשר מציג שיטה לזיהוי מערכת ההפעלה של המשתמש, הדפדפן שלו והאפליקציה שלו, וכל זאת מתוך תעבורת HTTPS של המשתמש וזאת למרות שהיא תעבורה מוצפנת. המאמר מציג מאגר נתונים ענק שבו יותר מ-20,000 דוגמאות, ומציע שיטה חדשה לזיהוי מערכת הפעלה, דפדפן ואפליקציות של המשתמש. השיטה מבוססת על מאפיינים ייחודיים של תעבורת HTTPS .

2. תכונות התעבורה בהן משתמש המאמר ומהן התכונות החדשניות:

תכונות בסיסיות:

- מספר החבילות שנשלחו קדימה (מהלקוח לשרת) ושהתקבלו אחורה (מהשרת ללקוח).
- סך כל הבתים שנשלחו קדימה ושהתקבלו אחורה.
- הפרש הזמן המינימלי בין הגעת חבילות קדימה וכן בין חבילות אחורה.
- הפרש הזמן המקסימלי בין הגעת חבילות קדימה וכן בין חבילות אחורה.
- ממוצע הפרש הזמן בין הגעת חבילות קדימה וכן בין חבילות אחורה.
- סטיית תקן של הפרש הזמן בין הגעת חבילות קדימה וכן בין חבילות אחורה.
- ממוצע מספר החבילות שנשלחו קדימה וכן שהתקבלו אחורה, בפרק זמן נתון.
- סטיית תקן של מספר החבילות שנשלחו קדימה וכן של מספר החבילות שהתקבלו אחורה.
- מספר החבילות שהתקבלו אחורה (מהשרת ללקוח)
- ממוצע ערך ה-TTL קדימה – ערך TTL ממוצע של החבילות שנשלחו קדימה ( מספר הצמתים שעוברת חבילה לפני פקיעתה).
- גודל החבילה הקטן ביותר שנשלחה קדימה וכן שהתקבלה אחורה.
- גודל החבילה הגדול ביותר שנשלחה קדימה וכן שהתקבלה אחורה.
- מספר החבילות הכולל – סך כל החבילות שנשלחו והתקבלו יחד.
- גודל החבילה המינימלי.
- גודל החבילה המקסימלי.
- ממוצע גודל החבילות.
- שונות בגודל.

תכונות חדשניות:

- גודל חלון TCP התחלתי
- גורם קנה מידה של חלון

- גודל מקטע מקסימלי של TCP
- מספר חבילותKeep Alive (חבילות שמוודאות שהחיבור נשאר פתוח גם כשאין תעבורה).
- מספר שיטות דחיסת SSL
- מספר הרחבותSSL
- מספר שיטות הצפנה SSL
- אורך מזההSession SSL
- גרסת SSL קדימה
- תפוקה מקסימלית קדימה של שיא
- תפוקה ממוצעת של שיא אחורה
- תפוקה מקסימלית של שיא אחורה
- תפוקה מינימלית של שיא אחורה
- סטיית תקן של שיאים אחוריים
- מספר שיאים קדימה ואחורה
- תפוקה מינימלית של שיא קדימה
- תפוקה ממוצעת של שיאים קדימה
- סטיית תקן של תפוקה בשיאים קדימה
- ממוצע הפרש זמן בין הגעת שיאים אחורה
- הפרשי זמן מקסימליים ומינימליים בין שיאים אחורה
- סטיית תקן של הפרש זמן בין שיאים אחורה
- ממוצע הפרש זמן בין שיאים קדימה
- הפרשי זמן מקסימליים ומינימליים בין שיאים קדימה
- סטיית תקן של הפרש זמן בין שיאים קדימה

## 3. תוצאות עיקריות ותובנות:

המאמר הציג שיטה חדשנית לזיהוי מערכת ההפעלה, הדפדפן והאפליקציה של המשתמש, למרות שהתעבורה מוצפנת ב .HTTPS-התוצאות הראו דיוק של 96.06% בסיווג, באמצעות שילוב של מאפיינים סטטיסטיים של הזרימה ,מאפייני TLS לא מוצפנים ,והתנהגות דפדפנים ברשת .אחד הממצאים המרכזיים הוא שהאלגוריתם יכול להבדיל בין דפדפנים שונים ( Chrome Firefox, Edge ) על סמך תבניות שליחת וקבלת חבילות, גם כאשר אין גישה למידע ישיר על כתובת היעד.

תובנות מרכזיות:

- ניתן לזהות דפדפנים ומערכות הפעלה גם כשהתוכן מוצפן – דפדפנים שונים מייצרים חתימות רשת שונות בגלל מאפייני הבקשות שלהם, כמו מספר ההרחבות ב-TLS, גודל החבילות הראשונות, וזמן ההמתנה בין שליחת נתונים לקבלתם.

- מאפיינים סטטיסטיים כמו התנהגות "מתפרצת" (Bursty Behavior) מסייעים בזיהוי – למשל, גלישה באינטרנט יוצרת "קפיצות" בזרימת הנתונים (בקשות קטנות ולאחריהן הזרמות גדולות), לעומת שיחות וידאו שמשדרות בקצב אחיד.
- שילוב מאפייני TLS ומאפייני זרימה משפר את הדיוק – בעוד ששיטות ישנות הסתמכו רק על ניתוח TLS (שכבר אינו חושף מספיק מידע תחת TLS 1.3), שילוב סטטיסטיקות רשת הקפיץ את הדיוק בכ-20%.
- האלגוריתם יכול לזהות שימוש בפרוטוקולים שונים, כמו QUIC ו-HTTP/2 – למשל, אם שירות משתמש ב-(QUIC) HTTP/3, האלגוריתם מזהה זאת על בסיס גודל החבילות ותזמונן, גם כאשר היעד מוצפן.
- היישום יכול לשמש לזיהוי תעבורה למטרות אבטחה – למשל, אפשר לזהות האם משתמש עובד עם Tor או VPN על בסיס חתימת TLS ייחודית ושימוש נרחב ב-Keep Alive Packets.

FlowPic_Encrypted_Internet_Traffic_Classification_is_as_Easy_as_Imag Recognition e

1. התרומה העיקרית של המאמר:

התרומה העיקרית של המאמר היא הצגת גישה חדשנית לסיווג תעבורת אינטרנט מוצפנת.

במקום להשתמש בניתוח סטטיסטי של מאפייני תעבורה (כמו שעושים רוב המחקרים בתחום), המאמר מציע גישה חדשה לגמרי: הפיכת זרימות רשת לתמונות (FlowPic) וסיווגן באמצעות רשתות נוירונים לעיבוד תמונה (CNNs). הרעיון הוא שכל סוג של אפליקציה מייצר תבנית שונה של שליחת וקבלת נתונים. כלומר במקום לנסות לקרוא את החבילות עצמן אפשר להסתכל על הדפוס הכללי של הזרימה.

2. תכונות התעבורה בהן משתמש המאמר ומהן התכונות החדשניות:

המאמר לא מסתמך על ניתוח מספרי רגיל, אלא ממיר כל זרימת רשת לתמונה דו-ממדית שבה:

ציר X מייצג את הזמן שבו הגיעו החבילות.

ציר Y מייצג את גודל החבילות.

הצפיפות של הנתונים יוצרת דפוס גרפי שאפשר לנתח כמו תמונה.

במקום להסתמך על "חוקי אצבע" וסיווגים סטטיסטיים, המודל משתמש בכוח של רשתות נוירונים קונבולוציוניות (CNNs) כדי לזהות תבניות בתמונה ולסווג את התעבורה.

3. תוצאות עיקריות ותובנות:

המאמר מציג את שיטת FlowPic שהצליחה להגיע לדיוק של 96% בסיווג קטגוריות תעבורה ו 99.7%בזיהוי אפליקציות ספציפיות. במקום לנתח חבילה אחר חבילה, החוקרים המירו זרימת נתונים לתמונה והשתמשו ברשתות נוירונים קונבולוציוניות (CNNs) לזיהוי תבניות. השיטה הצליחה לזהות שירותים גם כאשר נעשה שימוש ב VPN-או Tor שבהם כתובות היעד מוסתרות.

תובנות מרכזיות:

- ניתן לסווג תעבורה מוצפנת על ידי זיהוי דפוסי זרימה גרפיים – בעוד ששיטות מסורתיות מנתחות נתונים כמספרים, FlowPic ממיר אותם לתמונה ומאתר תבניות חוזרות בכל שירות (למשל, סטרימינג מייצר רצף חבילות קבועות, בעוד שגלישה יוצרת מקטעים מתפרצים).
- השיטה מצליחה לזהות שירותים גם כאשר הנתונים עצמם מוצפנים לחלוטין – ב-TLS 1.3 או QUIC, אין אפשרות לקרוא את תוכן ההודעות, אך הצורה שבה הנתונים זורמים נשארת ייחודית לכל שירות.
- CNNs הם כלי עוצמתי לזיהוי זרימות רשת – האלגוריתם הצליח להכליל מידע ולזהות שירותים גם אם הם לא הופיעו ישירות באימון, למשל, זיהה Vimeo כקטגוריית סטרימינג גם כשהוא לא נכלל בסט האימון.
- VPN ו-Tor לא מסתירים את חתימת התעבורה לגמרי – השימוש ב-VPN יוצר מעט שינויים בפרופיל הזרימה, אך FlowPic הצליח להתגבר על כך באמצעות למידה עמוקה שמזהה מאפיינים עדינים יותר כמו קצב שליחת הנתונים.
- ניתן לשלב את השיטה עם ניתוח מסורתי לשיפור דיוק – FlowPic מספק זיהוי מעולה של קטגוריות, אך שילובו עם נתונים מספריים (כמו גודל חבילות וזמן הגעה) יכול לשפר דיוק במקרים קשים לזיהוי.

Early_Traffic_Classification_With_Encrypted_ClientHello_A_

Multi-Country Study

1. התרומה העיקרית של המאמר:

המאמר מתמודד עם אחת הבעיות הקשות ביותר כיום בזיהוי תעבורה – השימוש בפרוטוקול TLS 1.3 עם הרחבת Encrypted ClientHello (ECH), שמסתירה מידע קריטי כמו ה- SNI (Server Name Indication). כדי לעקוף את הבעיה, החוקרים פיתחו אלגוריתם חדש בשם Hybrid Random Forest Traffic Classifier (hRFTC), שמצליח לבצע סיווג גם כאשר כמעט כל המידע על התעבורה מוסתר.

2. תכונות התעבורה בהן משתמש המאמר ומהן התכונות החדשניות:

מטען לא מוצפן של לחיצת יד TLS: גרסת פרוטוקול TLS , חבילות הצפנה נתמכות, אורך כולל של הרחבות, הרחבות GREASE .

סדרות זמן מבוססות זרימה: גודל החבילות הנשלחות, זמן הגעה בין חבילות עוקבות, קצב התעבורה לאורך זמן.

סטטיסטיקות של גדול חבילות: התפלגות של גדלי החבילות בתעבורה.

תכונות חדשניות:

במקום להשתמש רק בנתוני TLS לא מוצפנים או רק בסטטיסטיקות זרימה האלגוריתם משלב ביניהם: הוא מנצל את מה שנותר גלוי מההצפנה, ומנתח את דפוסי הזרימה כדי לקבל הקשרים נוספים.

3. תוצאות עיקריות ותובנות:

- הפרוטוקול TLS 1.3, ובמיוחד הרחבת Encrypted ClientHello (ECH), מסתיר מידע רגיש כמו Server Name Indication (SNI), מה שמונע משיטות ניתוח מסורתיות לסווג תעבורה. עם זאת, האלגוריתם hRFTC מצליח לזהות שירותים כמו YouTube, Netflix ו-WhatsApp על ידי שימוש במאפיינים שאינם מוצפנים, כגון גרסת TLS, הצפנות נתמכות ומידע מתוך תהליך ה-Handshake. כלומר, השיטה מאפשרת סיווג מדויק גם כשהתוכן והכתובת של השרת מוסתרים.
- כאשר SNI מוסתר באמצעות ECH, שיטות קודמות שהתבססו רק על ניתוח TLS צונחות לדיוק של-38.4% בלבד. האלגוריתם hRFTC, שמשלב בין מאפייני TLS וסטטיסטיקות זרימה, השיג דיוק של 94.6%, מה שמדגים את החשיבות של שילוב נתונים. ניתוח נוסף של גודל חבילות, קצב נתונים ודפוסי שליחה/קבלה מאפשר סיווג גם כשהתוכן מוצפן לחלוטין.
- HTTP/3 מבוסס על QUIC במקום TCP, כך שאין מספרי רצף או מנגנוני שליטה בחלון TCP, מה שמקשה על ניתוח מסורתי. כל ה-Handshake של QUIC מוצפן מהחבילה הראשונה, מה שמונע גישה למידע ישיר על החיבור. למרות זאת, האלגוריתם hRFTC הצליח לזהות שירותים כמו YouTube, Facebook ו-Google באמצעות ניתוח דפוסי זרימה, כולל קצב שליחת חבילות, פערי זמן והתנהגות בתחילת החיבור. ומכאן שניתן לזהות שירותים גם ללא נתונים גלויים ממסגרת ה-TCP.
- האלגוריתם הצליח לזהות שירותים חדשים מאותה קטגוריה (למשל, Vimeo כקטגוריית סטרימינג, למרות שלא נכלל באימון). עם זאת, הוא התקשה במדינות שונות עקב שינויים ברשתות הפצת תוכן (CDN) ומאפייני רשת מקומיים.

# חלק 3

בחלק זה הקלטנו 14 הקלטות לניתוח התעבורה ברשת.

3.1 הקלטנו 5 הקלטות לפי הדרישות ואלו שמותיהם וסוגיהם:

הקלטה של דפדוף ברשת בדפדפן chrome - chromewikipedia1.pcapng

הקלטה של דפדוף ברשת בדפדפן edge - edgepages.pcapng

הקלטה של צפייה בוידאו באתר youtube - youtube1.pcapng

הקלטה של שיחת לייב ב-discord - discord1.pcapng

הקלטה של שמיעת אודיו באתר spotify - spotify1.pcapng

3.2 + 3.3 יצרנו 2 קודים בתיקיית src לניתוח התעבורה בשביל ההשוואה בין ההקלטות השונות:

הקוד הראשון, analyzing_network.py :

מנתח את הנתונים הבסיסים של ההקלטות ומשווה בינהם ויוצר 9 גרפים בהתאם לסוגי הניתוח, אלו הסוגים:

מספר סך החבילות, כמות הדאטה הכוללת בבייטים, גודל חבילה ממוצע, זמן ממוצע בין הגעת חבילות, TTL ממוצע, אחוזי חבילות TCP/UDP , אחוז חבילות TLS , הפורט הכי נפוץ, 2 כתובות IP הכי נפוצות (לא כולל IP עצמי).

הקוד לוקח רק את הדקה הראשונה של כל הקלטה כדי שההשוואה תהיה כמה שיותר סטרילית, יש לשים לב שאם מכניסים הקלטה קצרה מדקה זה ישפיע על תוצאות ההשוואה. (אנחנו הקלטנו בכל מקרה בסביבות הדקה) ובמידה ותוכנס הקלטה קצרה מדקה או גדולה מדקה יודפס הודעה בהתאם.

הערות והסברים על שורות הקוד מופיעים בקוד עצמו.

<u>הוראות הפעלה לקוד הראשון:</u>

יש להכניס את ההקלטות אשר רוצים להשוות לתיקיית "records/records_comparing", והגרפים שיווצרו ישמרו בתיקיית "res/Graphs". בפורמט: '.pcapng'.

הקוד השני, analyzing_network_flowpic.py :

מנתח את זמני הגעת החבילות, את גודל החבילות ואת כמות החבילות שהגיעו באותה נקודת זמן (בשנייה) ויוצר לנו גרף flowpic בסיסי עבור כל הקלטה בהתבסס על המאמר:

ציר ה-x מסמל את הזמן בשניות, ציר ה-y מסמל את גודל החבילה בבייטים, סקלת הצבעים מסמלת את כמות החבילות.

גם פה הקוד לוקח רק את הדקה הראשונה של כל הקלטה כדי שההשואה תהיה כמה שיותר סטרילית, יש לשים לב שאם מכניסים הקלטה קצרה מדקה זה ישפיע על תוצאות ההשוואה – יווצר שטח ריק בתמונה. (אנחנו הקלטנו בכל מקרה בסביבות הדקה) ובמידה ותוכנס הקלטה קצרה מדקה או גדולה מדקה יודפס הודעה בהתאם. הערות והסברים על שורות הקוד מופיעים בקוד עצמו.

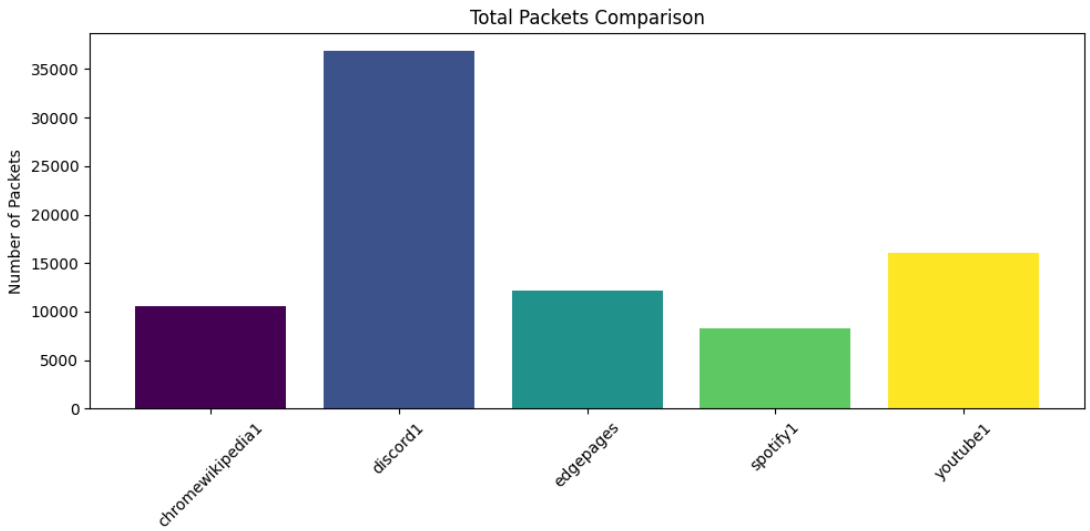
<u>הוראות הפעלה לקוד השני:</u>

יש להכניס את ההקלטות אשר רוצים ליצור להם גרף מתאים לתיקיית "records/all_records", בפורמט: '.pcapng' . בהפעלה של הקוד המשתמש ישאל אם הוא רוצה חבילות מסוננות לפי כתובת IP הכי נפוצה (לא IP עצמי) (כדי לראות את התעבורה בצורה סטרילית עבור אפליקצייה מסוימת) אם כן צריך להקיש 1 ואז הגרפים ישמרו בתיקיית "res/FlowPicsFilter".ואם לא יש להקיש 0 והגרפים שיווצרו ישמרו בתיקיית "res/FlowPics".

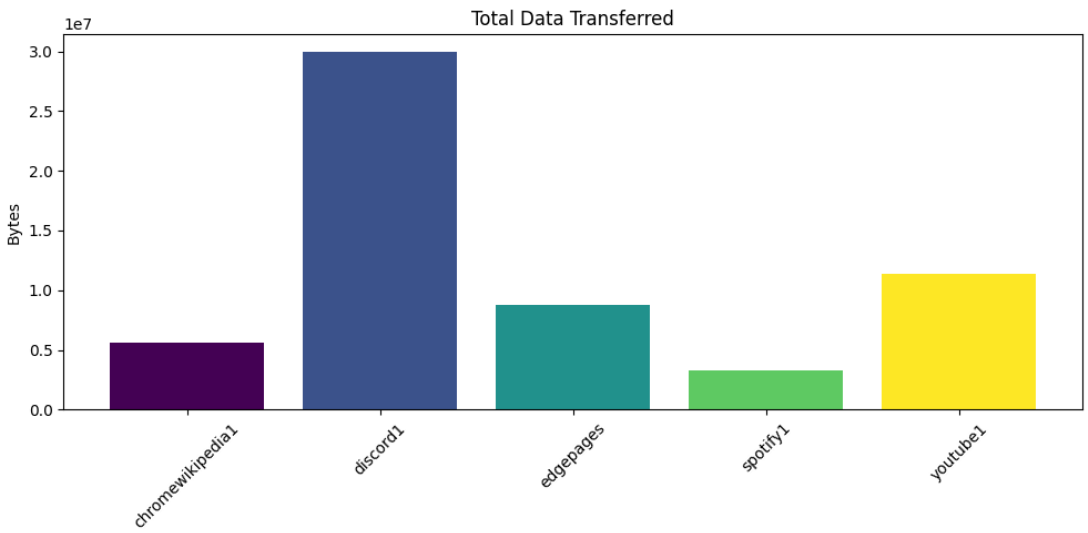
כל הניתוחים אשר מוצגים הם מהקלטות באורך של דקה (הבאנו את ההדפסה במהלך הריצה):

```
Processing file: records_comparing\chromewikipedia1.pcapng
The capture records_comparing\chromewikipedia1.pcapng is more than a minute long and we take the first minute.

Processing file: records_comparing\discord1.pcapng
The capture records_comparing\discord1.pcapng is less than a minute long and lasted 59.9902708530426 seconds.

Processing file: records_comparing\edgepages.pcapng
The capture records_comparing\edgepages.pcapng is less than a minute long and lasted 59.95236802101135 seconds.

Processing file: records_comparing\spotify1.pcapng
The capture records_comparing\spotify1.pcapng is more than a minute long and we take the first minute.

Processing file: records_comparing\youtube1.pcapng
The capture records_comparing\youtube1.pcapng is more than a minute long and we take the first minute.

Graph saved as Graphs\Total_Packets_Comparison.png
Graph saved as Graphs\Total_Data_Transferred.png
Graph saved as Graphs\Average_Packet_Size.png
Graph saved as Graphs\Average_Inter-Arrival_Time.png
Graph saved as Graphs\Average_TTL.png
Graph saved as Graphs\TLS_Percentage.png
Graph saved as Graphs\Protocol_Usage_Comparison.png
Graph saved as Graphs\Most_Used_Ports.png
Graph saved as Graphs\Most_Used_IPs.png
```

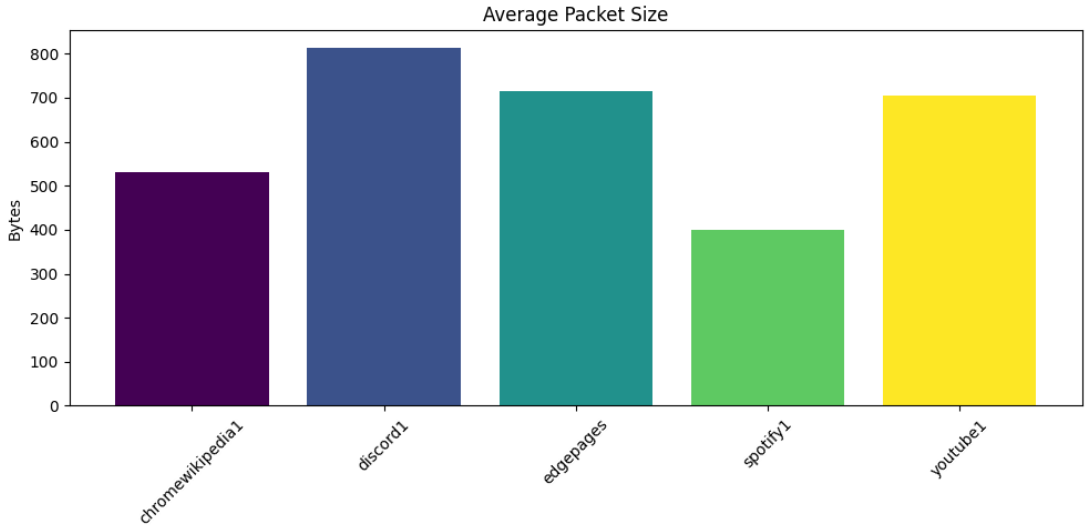
להלן הגרפים והסקת המסקנות בהתאם:



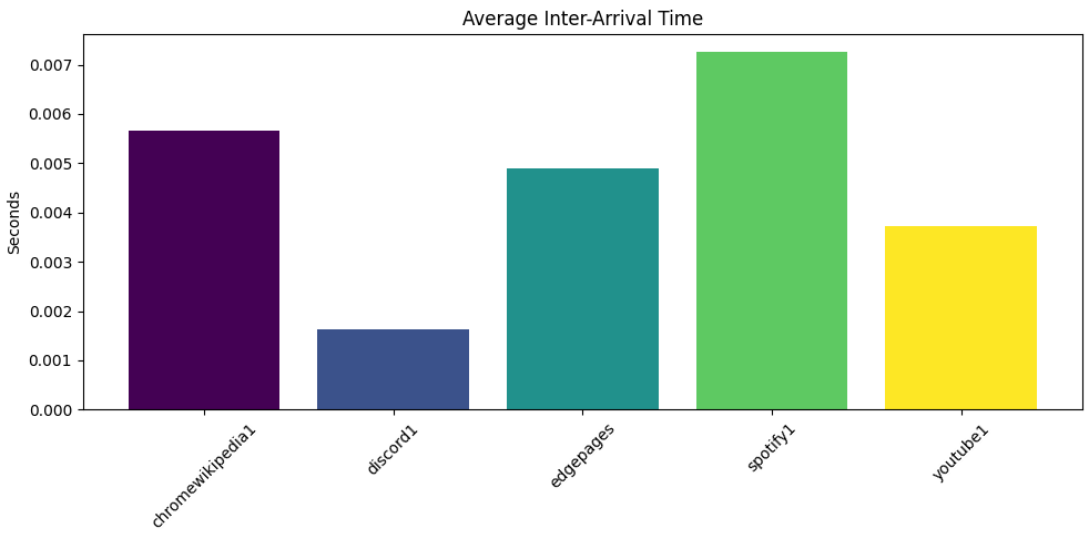

בגרף זה ניתן לראות בבירור ששימוש באפלקציית discord אשר מייצגת ועידת וידאו שולחת הרבה יותר חבילות בצורה משמעותית משאר הדגימות. ניתן להסיק ששימוש באפלקציות של ועידת וידאו מייצרת הרבה יותר חבילות.



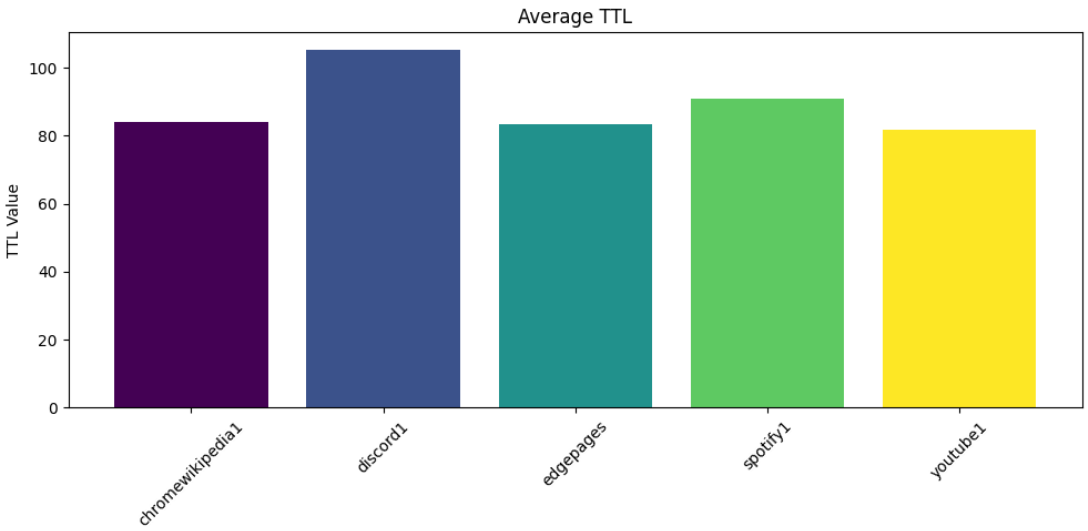

בגרף זה ניתן לראות הלימה עם מה שהסקנו בגרף הקודם, ולהסיק שאפלקציות מסוג ועידת וידאו שולחות כמות דאטה גדולה. בנוסף ניתן לראות את ההבדלים המשמעותיים גם בין youtube ל-spotify (יותר מפי 2 כמות מידע).



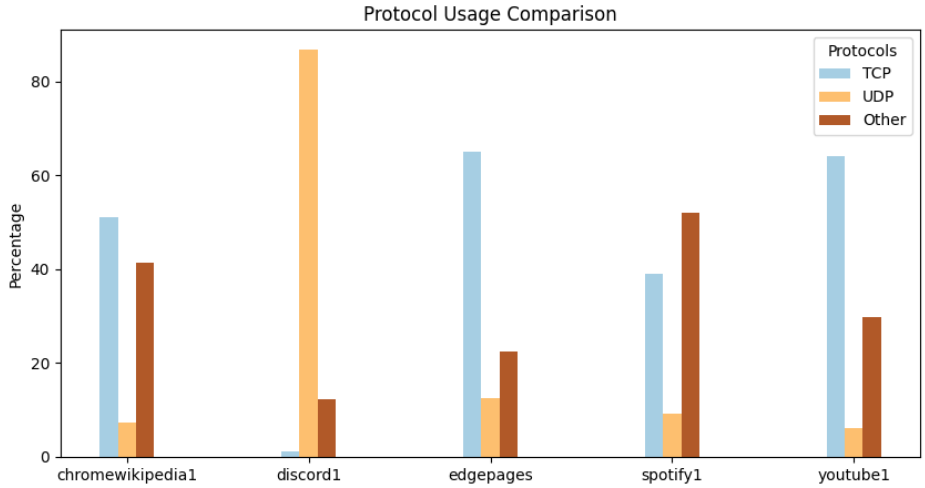

בגרף זה ניתן לראות שבאפלקציית spotify שגודל החבילות הממוצע שנשלחות הם קטנות משמעותית ביחס לשאר האפלקציות. בנוסף ניתן להסיק שגודל החבילה הממוצע בזמן גלישה בדפדפן מושפע מאוד מטעינות דפים שונים ומכמות הטעינות. (בדפדפן edge גלשנו ב LeetCode לעומת דפדוף בchrome- ששם גלשנו בויקפדיה).



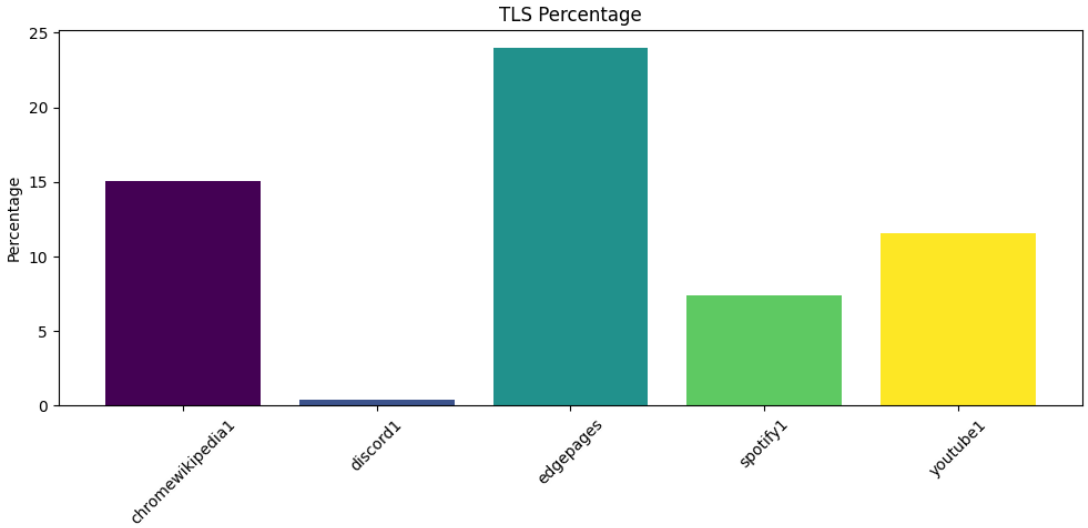

בגרף זה ניתן להסיק כי זמן הגעה בין חבילות באפליקציית ועידת וידאו הוא קצר מאוד ביחס לשאר, וזאת בכדי לייצר וידאו חלק.



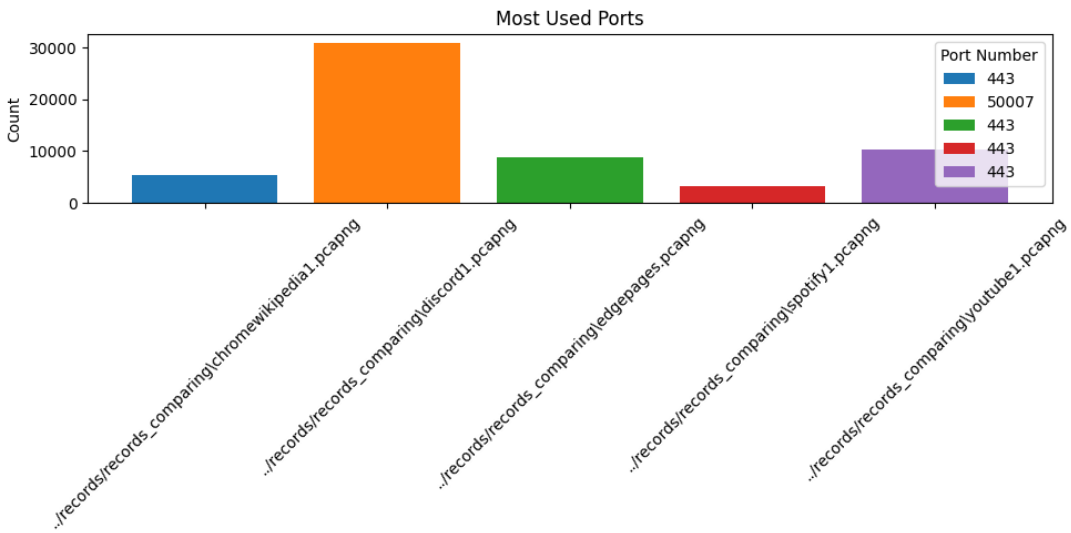

בגרף זה ניתן לראות שה- TTL הממוצע של discord הוא הכי גבוה ומכך ניתן להסיק ששרתי discord הכי קרובים והחבילות עבורו פחות נתבים מאשר ביחס לשימוש באפלקציות האחרות.



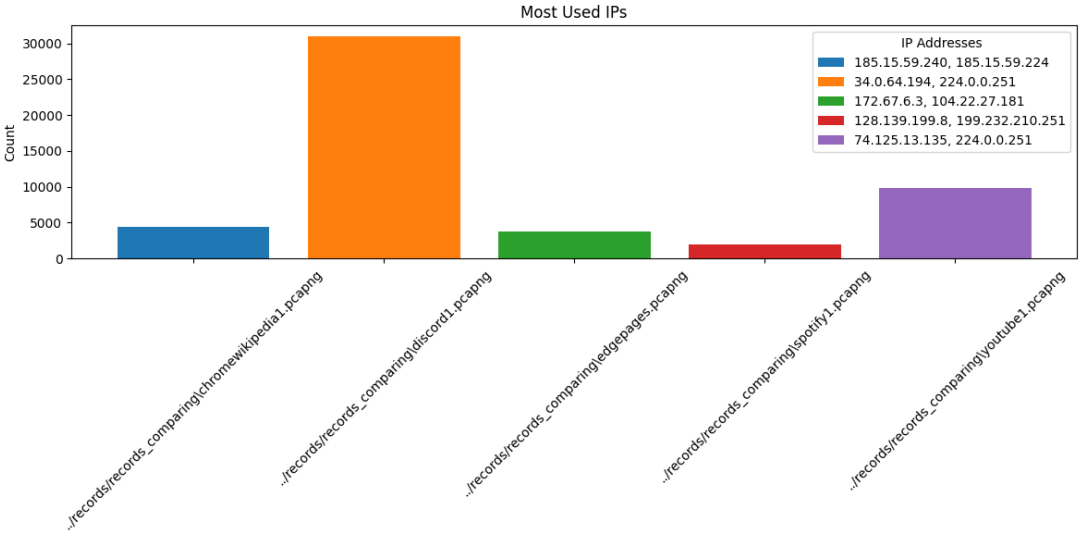

בגרף זה ניתן לראות שחוץ מ discord כלל האפלקציות שדגמנו מעדיפות להשתמש בפרוטוקול TCP על פני UDP וזאת משום שהם מתעדפות תקשורת אמינה על פני מהירות. לעומת זאת באפלקציית מסוג ועידת וידאו יותר חשוב המהירות ולכן השימוש הוא ב UDP.




בגרף זה ניתן לראות הלימה עם הגרף הקודם ולראות ש discord לא משתמש כמעט בכלל בפרוטוקול TLS שהוא פרוטוקול לתעבורה מוצפנת (שם דגש על מהירות ולא אמינות).

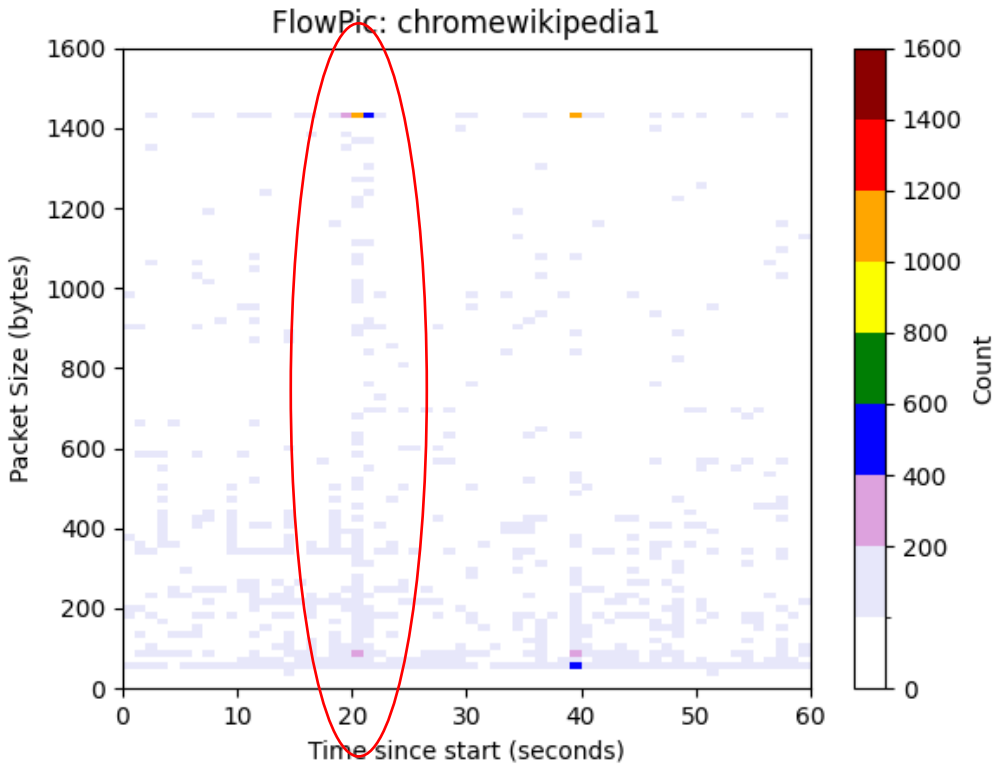


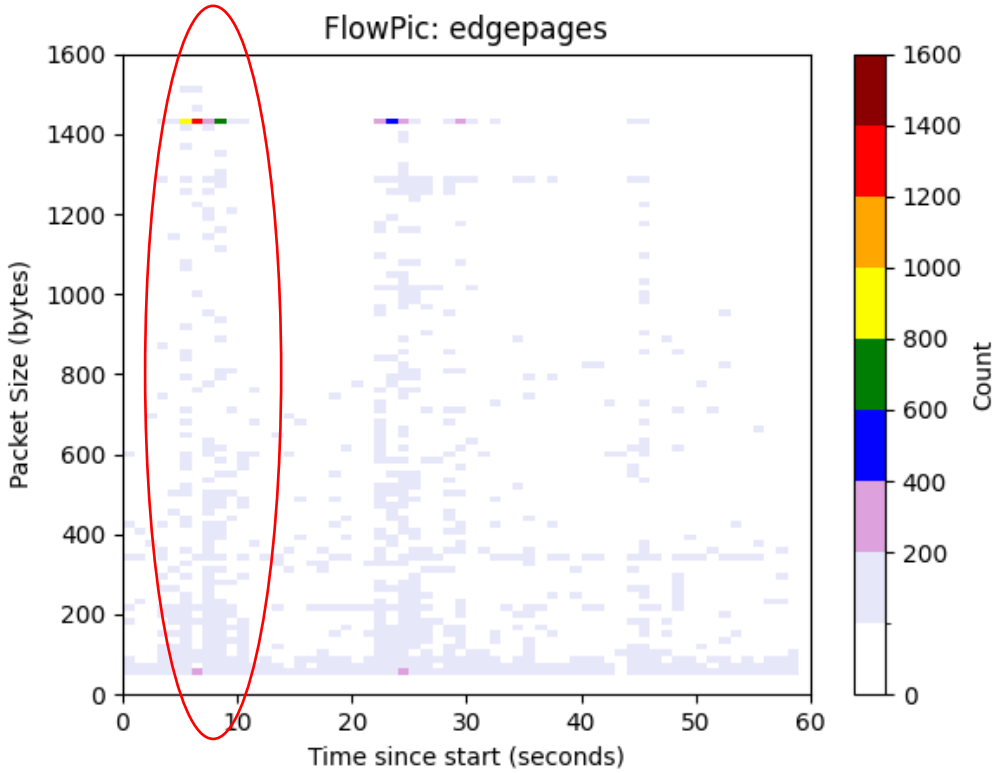

בגרף זה ניתן לראות שפורט 443, המשמש לפרוטוקול HTTPS , הוא הפורט הנפוץ ביותר ברוב היישומים שהוצגו בגרף, מה שמדגים את השימוש הנרחב בתקשורת מאובטחת עבור רוב האפלקציות. גם פה ניתן לראות הלימה בנודע למסקנות הקודמות של שימוש באבטחה באפליצקיית וידאו. ניתן לראות של discord יש פורט יחודי ולא כל כך נפוץ ומוכר.



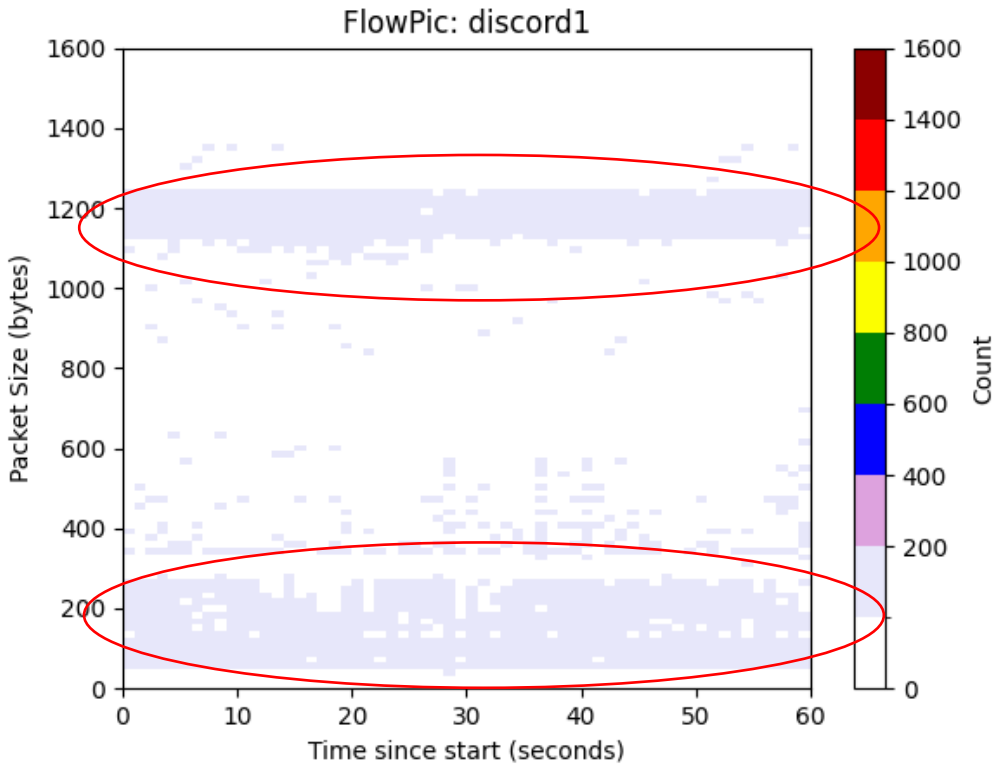

בגרף זה ניתן לראות מה ה IP הכי בשימוש עובר כל אפליקצייה ומכך בעזרת ניתוח יותר מעמיק ניתן להסיק על האפלקציות בהיבט של סינון לפי IP, בחלק 3.4 של המטלה יודגם השימוש בסינון לפי IP .

המשך עבור 3.2 + 3.3 , להלן הגרפים מקוד 2 אשר מציגים נתונים בהיבט שונה אשר מוסיפים לניתוח הכללי עבור סעיפים אלו וההקלטות שדגמנו.

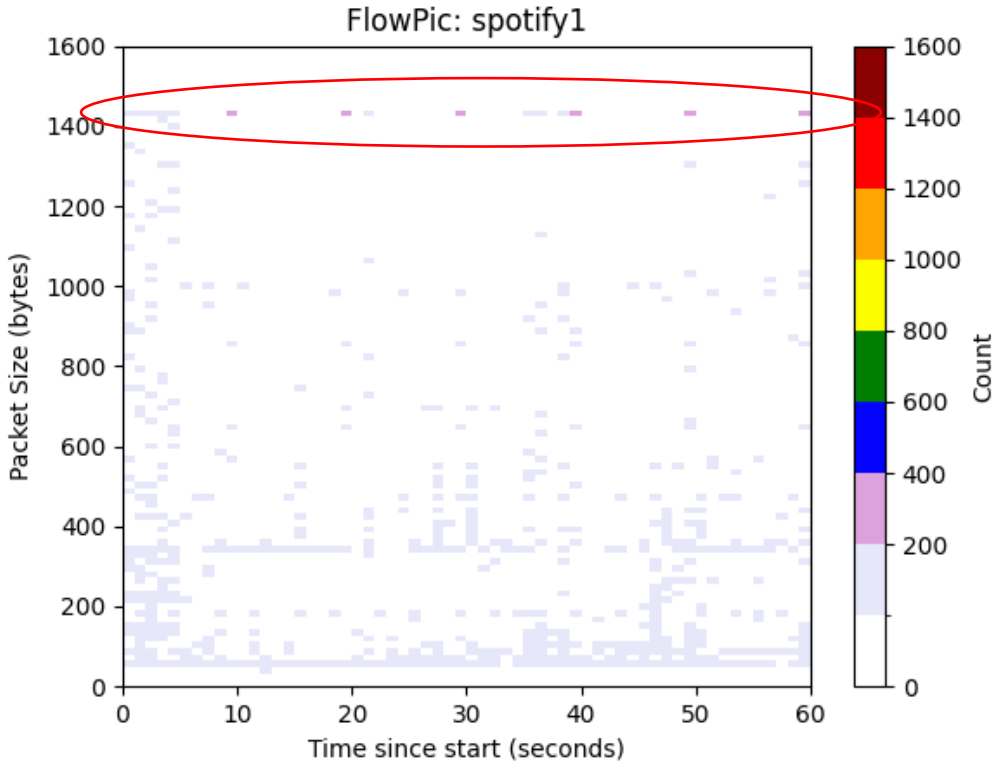


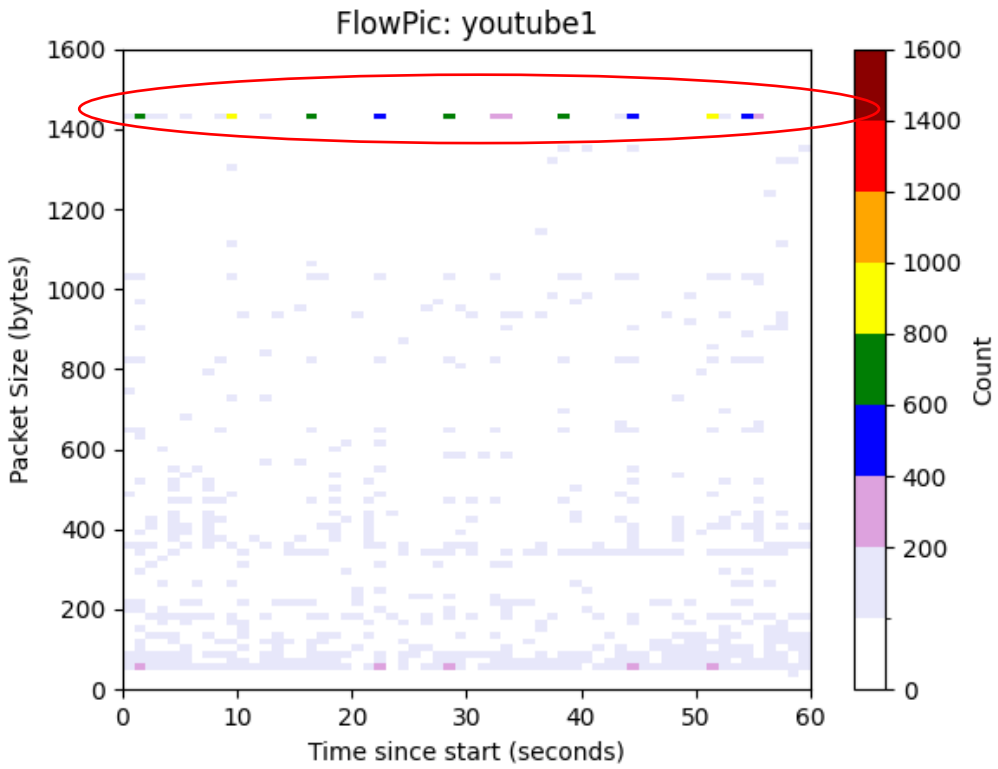

בגרפים אלו ניתן לראות את ההשפעה של טעינת דף חדש על הזרמת החבילות ברשת




בגרף זה ניתן לראות בהמשך לניתוחים הקודמים שהזרמת המידע היא רציפה. בנוסף ניתן לראות שגודל החבילות הוא באזורי גדלי חבילות קבועים.




בגרף זה ניתן לראות שבspotify יש הזרמת חבילות מאסיבית כל 10 שניות , בגדלים קבועים ובכמות זהה, מה שמראה ש spotify לא שולחת את כל המידע של השיר בהתחלה אלא בפעימות, (אנו חושבים שזה בשביל לחסוך בהעברת מידע משום שלפעמים אנשים עוברים שירים תוך כדי).




בגרף זה ניתן לראות שבyoutube יש הזרמת חבילות מאסיבית (מגיע גם ל800-1000 חבילות בשניה) כל כמה שניות , בגדלים קבועים ובכמות משתנה, מה שמראה שyoutube לא שולחת את כל המידע של הסרטון בהתחלה אלא בפעימות.

3.4 בסעיף זה אנו משתמשים בכל ההקלטות שיצרנו (13) ומניחים כי אנו תוקפים שרוצים לזהות את האפליקצייה (בהרצת הקוד לניתוח הflowpics תנותח גם הקלטת הבונוס אשר באותה התיקייה):

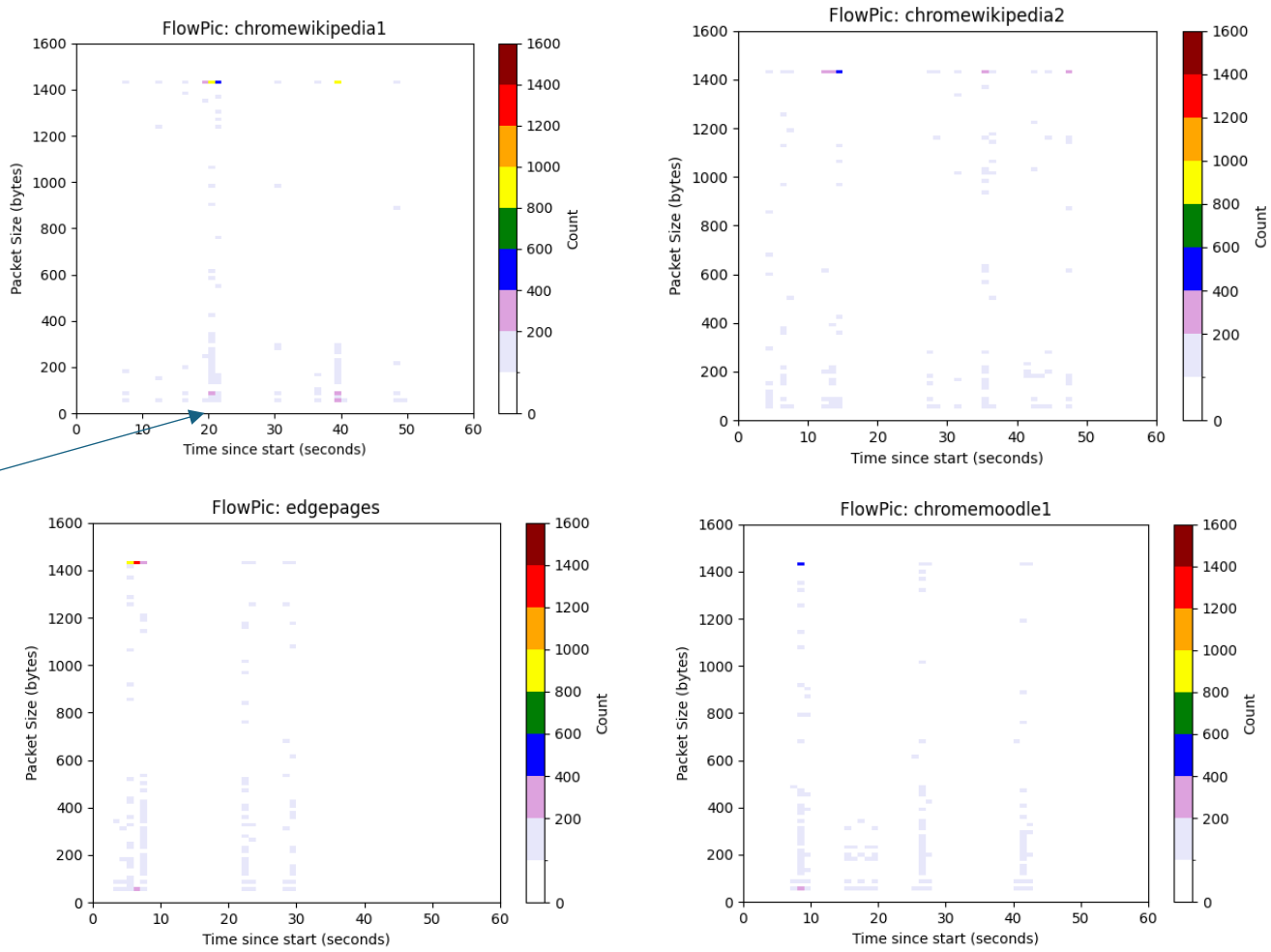
הנחה ראשונה: אנו יודעים את גדלי החבילות , זמני ההגעה ו PORT&IP של החבילות.

ייצרנו תמונות FlowPics מסוננות בעזרת ה IP הכי נפוץ (לא IP עצמי) .

```
Do you want to filter by most common IP in the records? 1 for yes, 0 for no:1
Extracting most used IP addresses in all_records\chromemoodle1.pcapng
34.96.118.58
Processing file: all_records\chromemoodle1.pcapng
The capture all_records\chromemoodle1.pcapng is less than a minute long and lasted 42.35254788398743 seconds.
Generating image for chromemoodle1
Image saved as FlowPicsFilter\chromemoodle1_graph.png
```

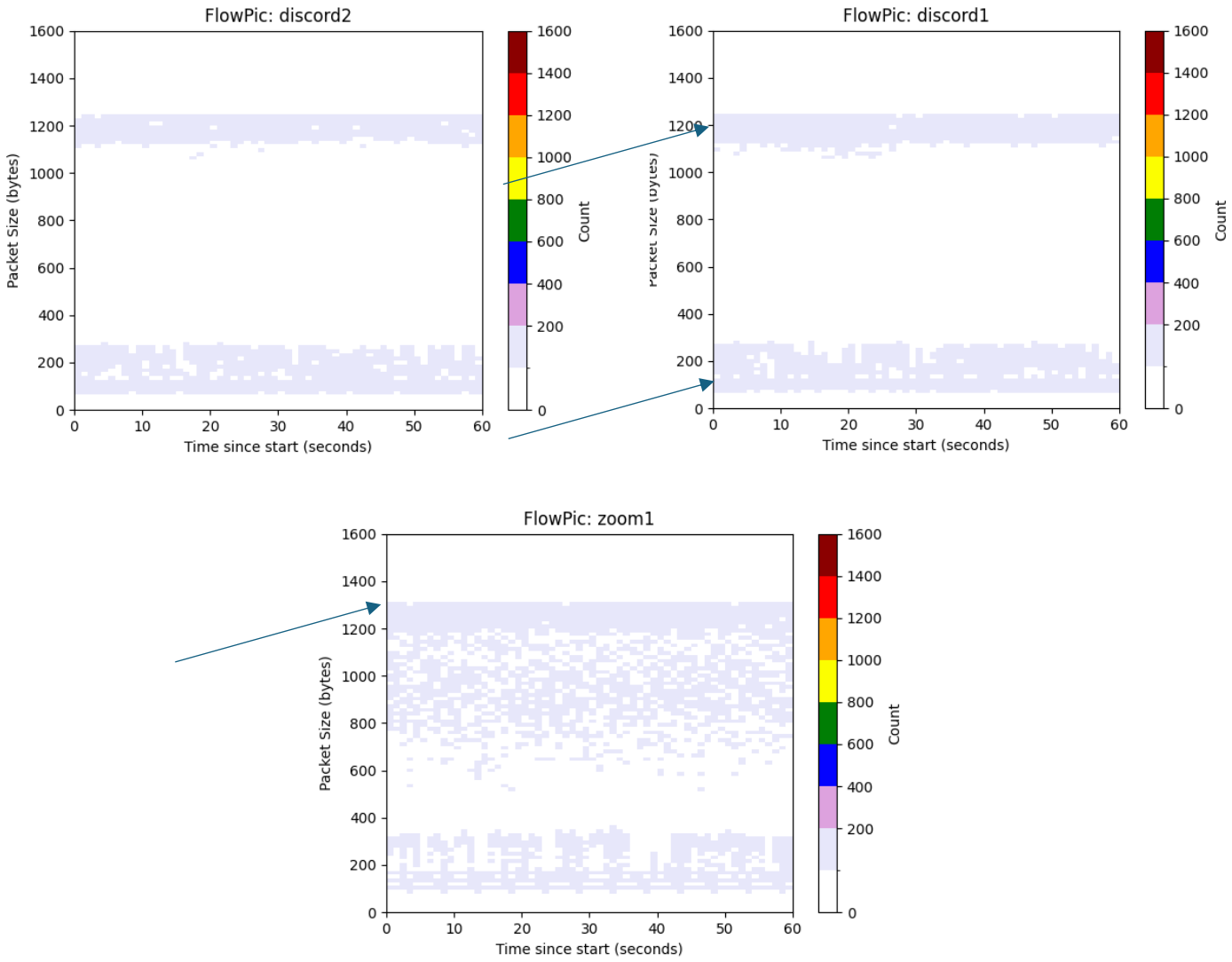
על פי הגרפים שנוצרו אפשר לזהות דפוס דומה בין כל ההקלטות מאותו אפליקצייה (תמונה דומה). הזיהוי אמור להתבצע על ידי מודלים של רשתות נוירוניות אשר יזהו דמיון בין התמונות ויסווגו אותם לאפלקציות. כמובן שבשביל שמודל כזה יהיה בדיוק גבוה צריך הרבה דאטה ודגימות, **פה אנו לא נדגים בעזרת מודל רשת נוירונים אמיתית אלא על פי הסתכלות פשוטה.**

להלן האפלקציות ותמונתיהם : גלישה בדפים בדפדפן



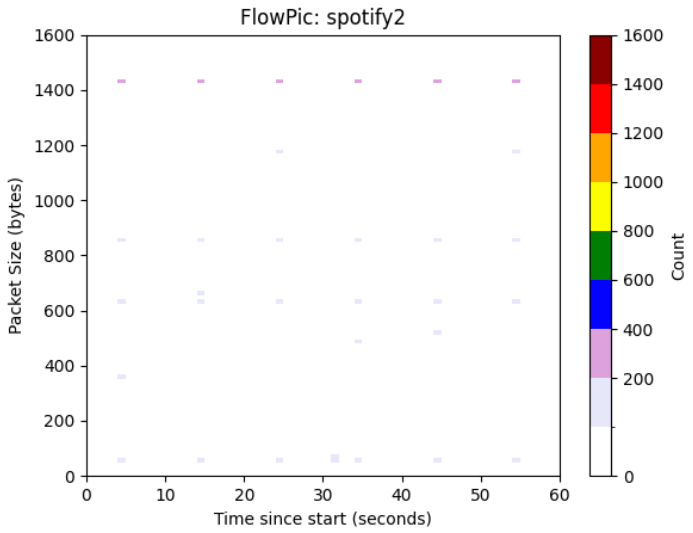

ניתן לראות את הדפוס של גלישה בדפדפן באתרים, את הזרמות המידע כאשר נכנסים לדף חדש, ואת כך שגודל החבילות המקסימלי מגיע לגודל קבוע , קצת יותר מ-1400 וכמות החבילות בגודל זה כאשר טוענים דף הוא לרוב גבוה(בהתאם לדף שנטען), כלומר אם לאחר סינון נזהה תמונה דומה לאלו נידע שזה גלישה בדפים בדפדפן.

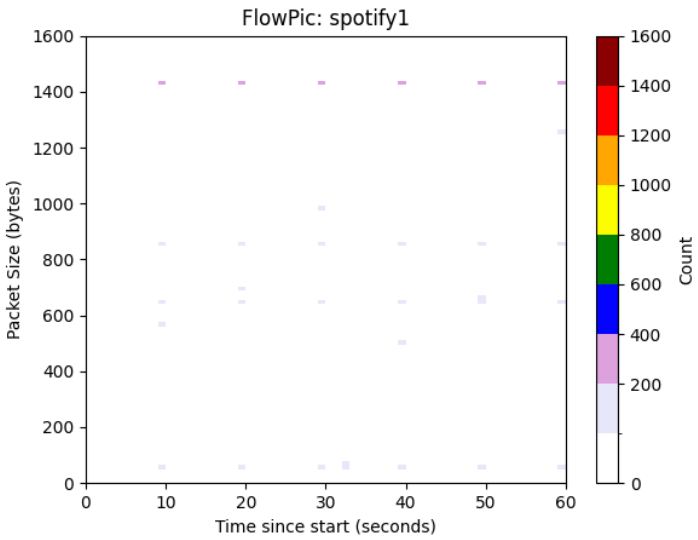
# שימוש בועידת וידאו

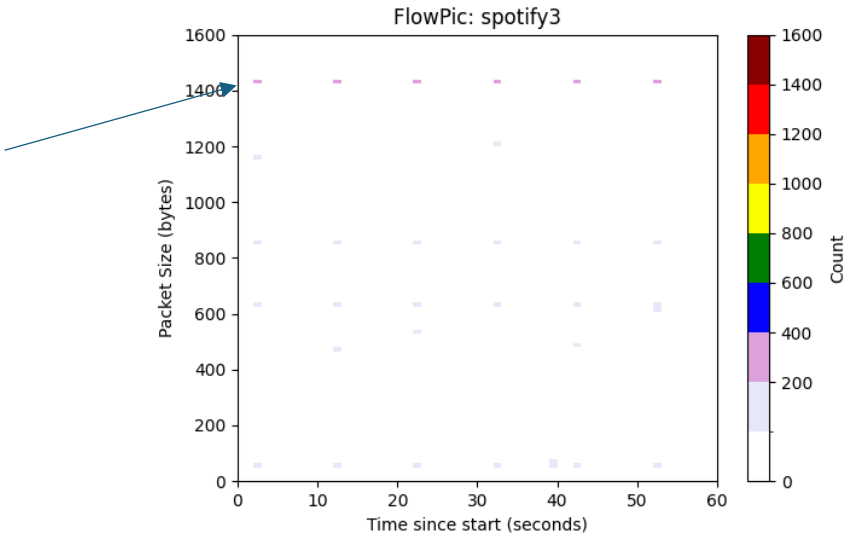



ניתן לראות את הדפוס של שימוש בועידת וידאו, את הזרמות המידע הרציפות באזורי גדלי חבילות קבועים, משתנה לפי סוג האפליקצייה. בdiscord ניתן לראות את הרציפות באזור גודל 1200 וב-200, בzoom ניתן לראות את שגודל החבילות הנשלחות ברציפות הוא יותר גדול מאשר הגודל בdiscord, כלומר לזהות את ההבדל. כלומר אם לאחר סינון נזהה תמונה דומה לאלו נידע שימוש בועידת ודיאו ובנוסף נוכל לזהות את סוג האפלקצייה המדוייק.

# האזנה לאודיו בspotify

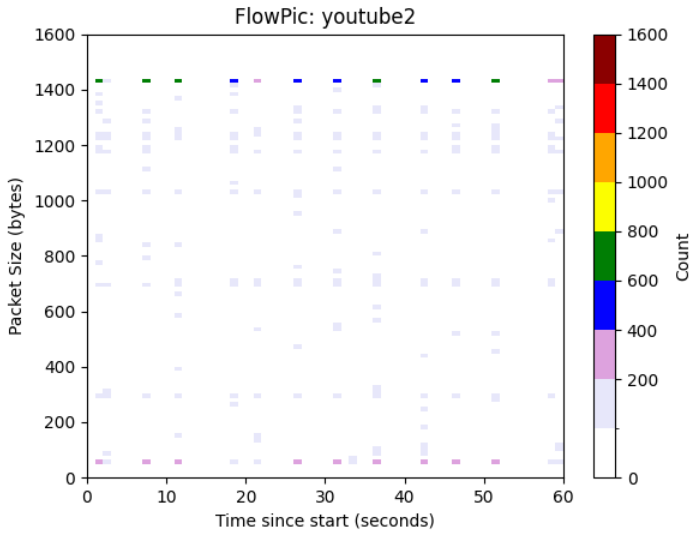


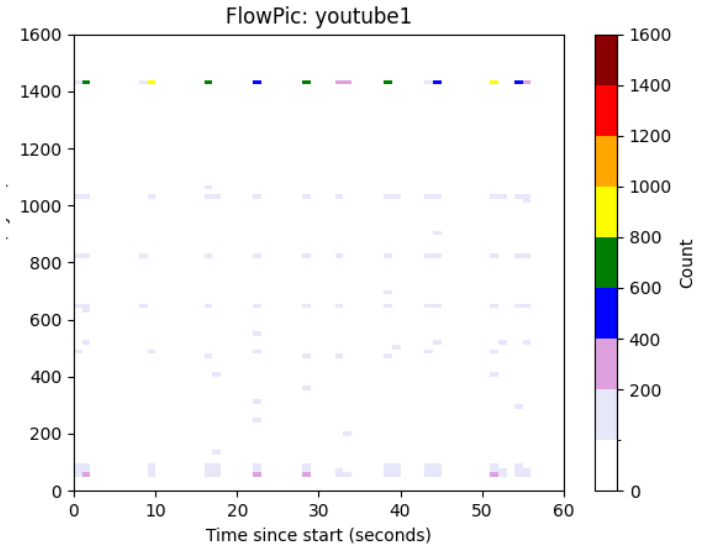



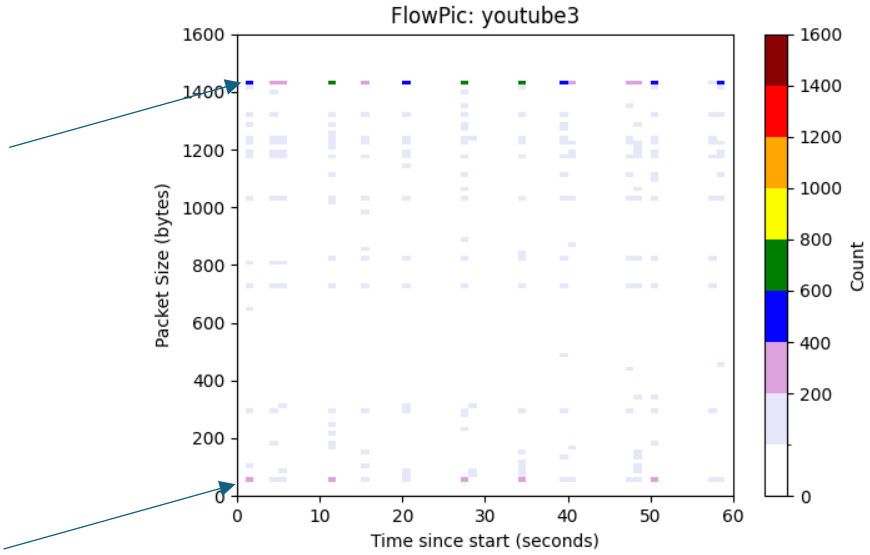




ניתן לראות את הדפוס של ההאזנה לאודיו בspotify , הזרמת מידע כל 10 שניות כר נשלחות כמות חבילות יחסית גדולה בגודל קבוע שהוא קצת יותר מ1400 ומספר החבילות הנשלחות בפעימה הוא בין 200 ל400. כלומר אם לאחר סינון נזהה תמונה דומה לאלו נידע שזה האזנה של אודיו בspotify .

# צפייה ב-youtube




ניתן לראות את הדפוס של צפייה בסרטונים בyoutube , הזרמת מידע כל כמה שניות כאשר נשלחות כמות חבילות יחסית גדולה בגודל קבוע שהוא קצת יותר מ1400 ומספר החבילות הנשלחות בפעימה הוא משתנה , ככל הניראה בהתאם לאיכות הוידאו אשר משפיע על כמות הדאטה שצריך להעביר, בנוסף נוצר גם קו של הזרמת מידע בגודל נמוך וקבוע שהוא באזור ה50 , כלומר אם לאחר סינון נזהה תמונה דומה לאלו נידע שזה צפייה בסרטונים בyoutube.

הנחה שנייה: אנו יודעים את גדלי החבילות ואת זמני ההגעה של החבילות בלבד.

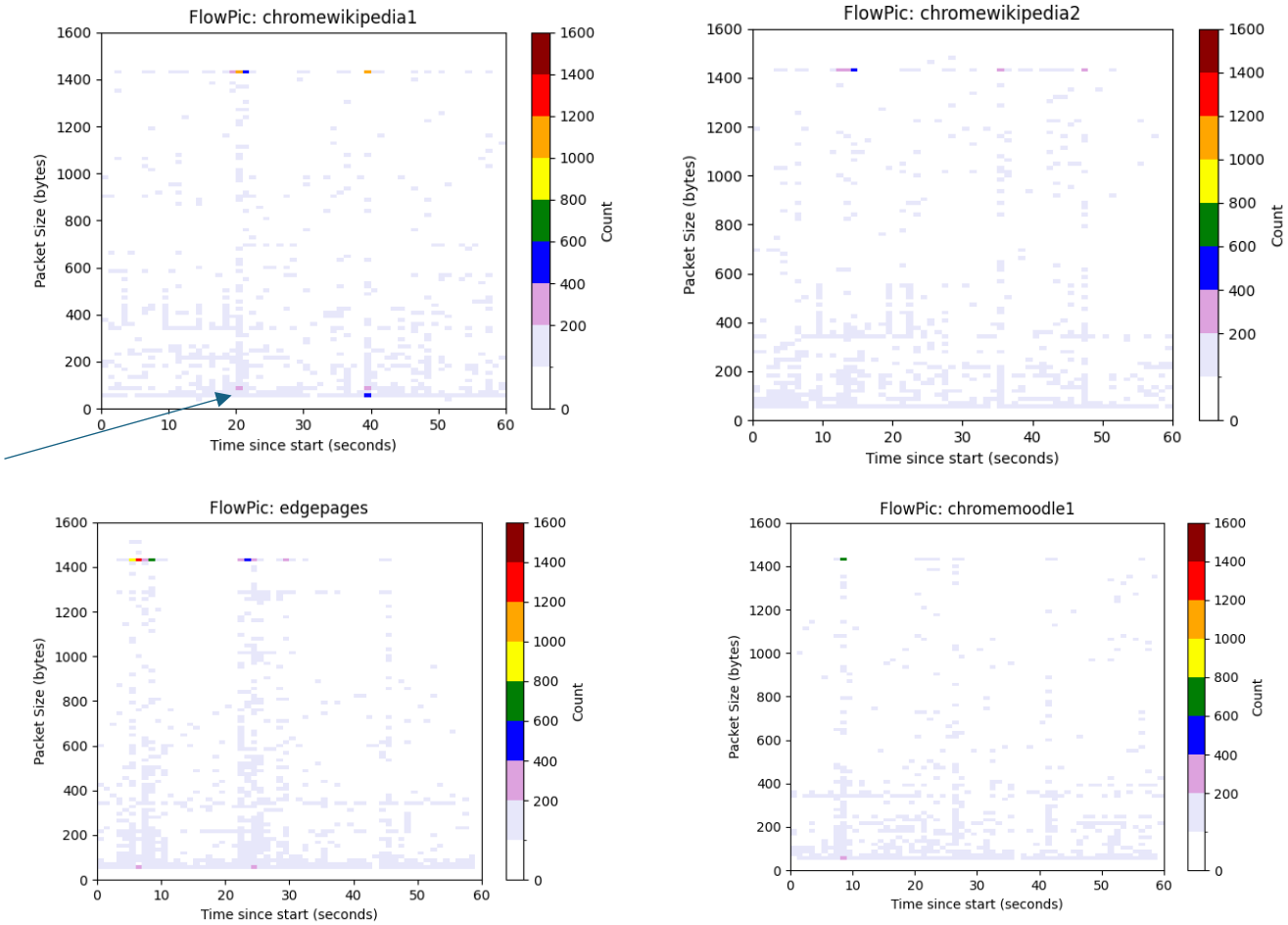
ייצרנו תמונות FlowPics לא מסוננות לפי כתובת IP.

```
Do you want to filter by most common IP in the records? 1 for yes, 0 for no:0
Processing file: all_records\chromemoodle1.pcapng
The capture all_records\chromemoodle1.pcapng is less than a minute long and lasted 59.596181869506836 seconds.
Generating image for chromemoodle1
Image saved as FlowPics\chromemoodle1_graph.png
```

על פי הגרפים שנוצרו אפשר לזהות דפוס דומה בין כל ההקלטות מאותו אפליקצייה (תמונה דומה), רמת הדמיון בין התמונות יותר נמוכה מרמת הדימיון בסעיף הקודם אך עדיין קיים דמיון רב, ולכן אנו משערים שבעזרת רשת נויורנים אמיתית עדיין תהיה הצלחה רבה בזיהוי (**כמובן שבאחוזים יותר נמוכים מאשר ההצלחה בסעיף הקודם**).

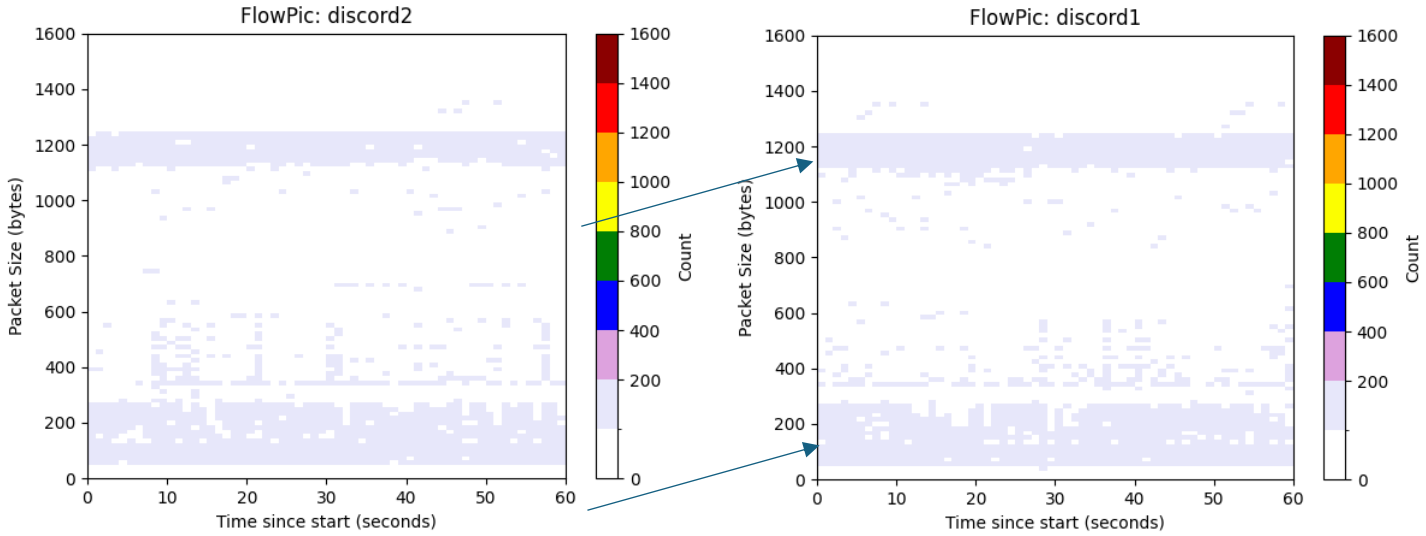
הזיהוי אמור להתבצע על ידי מודלים של רשתות נוירוניות אשר יזהו דמיון בין התמונות ויסווגו אותם לאפלקציות. כמובן שבשביל שמודל כזה יהיה בדיוק גבוה צריך הרבה דאטה ודגימות, **פה אנו לא נדגים בעזרת מודל רשת נויורנים אמיתית אלא על פי הסתכלות פשוטה.**

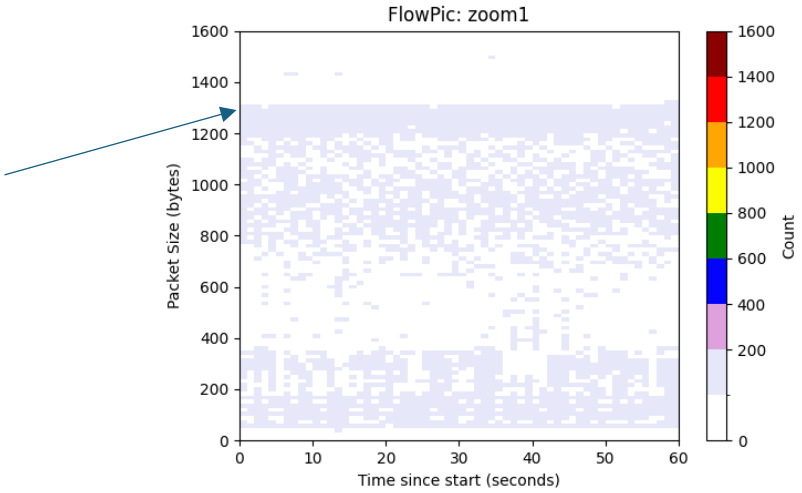
להלן האפלקציות ותמונתיהם : גלישה בדפים בדפדפן




**עדיין** ניתן לזהות את הדפוס של גלישה בדפדפן באתרים, את הזרמות המידע כאשר נכנסים לדף חדש, ואת כך שגודל החבילות המקסימלי מגיע לגודל קבוע , קצת יותר מ-1400 וכמות החבילות בגודל זה כאשר טוענים דף הוא לרוב גבוה(בהתאם לדף שנטען), כלומר אם נזהה תמונה דומה לאלו נידע שזה גלישה בדפים בדפדפן.

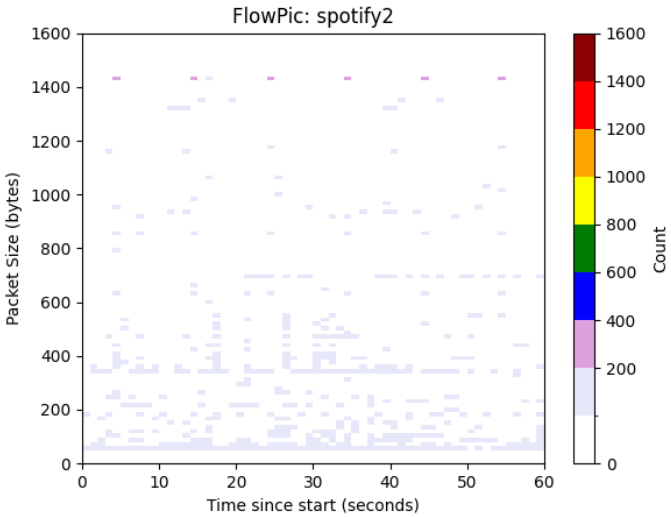
# שימוש בועידת וידאו

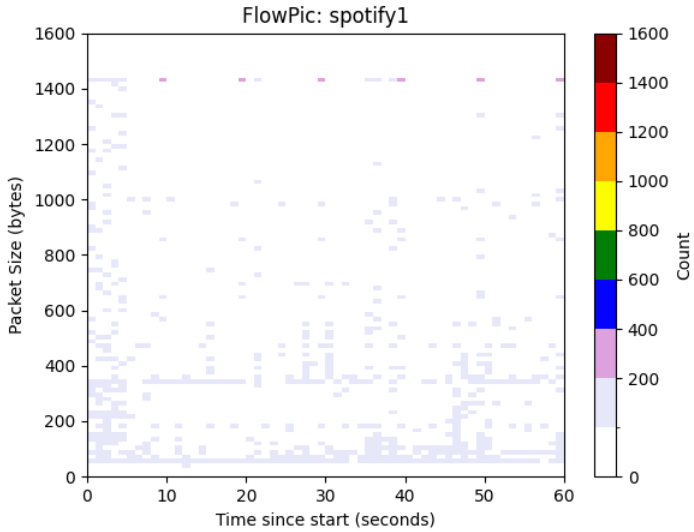


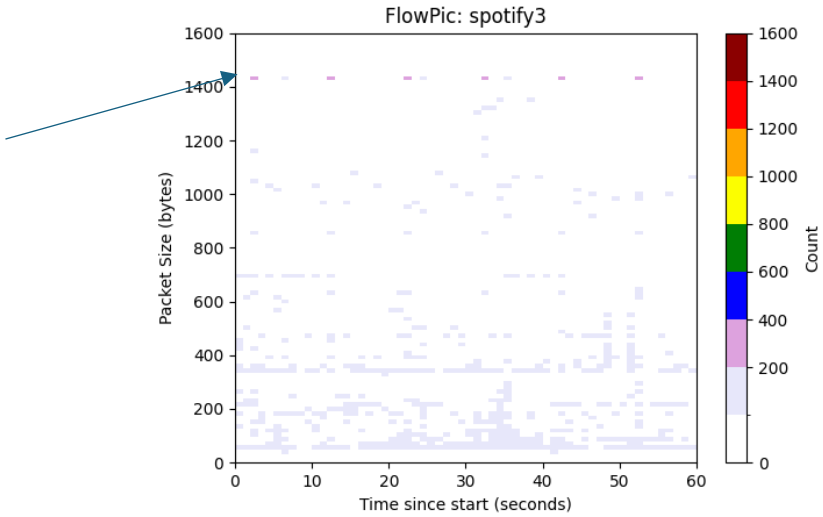




**עדיין** ניתן לזהות את הדפוס של שימוש בועידת וידאו, את הזרמות המידע הרציפות באזורי גדלי חבילות קבועים, משתנה לפי סוג האפליקצייה. בdiscord ניתן לראות את הרציפות באזור גודל 1200 וב-200, בzoom ניתן לראות את שגודל החבילות הנשלחות ברציפות הוא יותר גדול מאשר הגודל בdiscord, כלומר לזהות את ההבדל. כלומר אם נזהה תמונה דומה לאלו נידע שימוש בועידת ודיאו ובנוסף נוכל לזהות את סוג האפלקצייה המדוייק.

# האזנה לאודיו בspotify

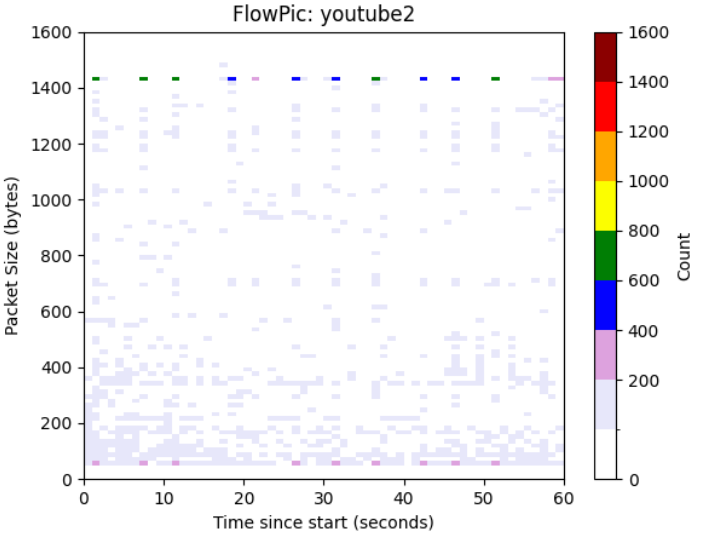


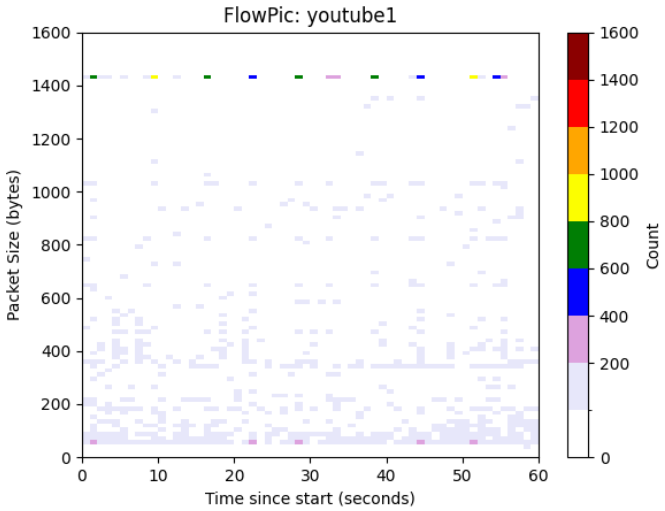



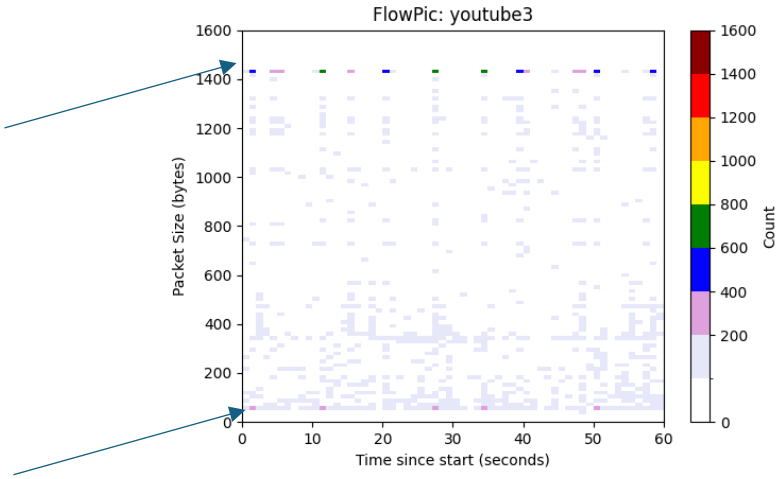




**עדיין** ניתן לזהות את הדפוס של ההאזנה לאודיו בspotify , הזרמת מידע כל 10 שניות כר נשלחות כמות חבילות יחסית גדולה בגודל קבוע שהוא קצת יותר מ1400 ומספר החבילות הנשלחות בפעימה הוא בין 200 ל400. כלומר אם נזהה תמונה דומה לאלו נידע שזה האזנה של אודיו בspotify .

## צפייה ב-youtube




**עדיין** ניתן לזהות את הדפוס של צפייה בסרטונים בyoutube , הזרמת מידע כל כמה שניות כאשר נשלחות כמות חבילות יחסית גדולה בגודל קבוע שהוא קצת יותר מ1400 ומספר החבילות הנשלחות בפעימה הוא משתנה , ככל הניראה בהתאם לאיכות הוידאו אשר משפיע על כמות הדאטה שצריך להעביר, בנוסף נוצר גם קו של הזרמת מידע בגודל נמוך וקבוע שהוא באזור ה50 , כלומר אם נזהה תמונה דומה לאלו נידע שזה צפייה בסרטונים בyoutube.

### לסיכום סעיף 3.4:

עבור 2 ההנחות בתור תוקף נוכל לזהות את סוג האפלקצייה , בהנחה הראשונה אחוז הדיוק גבוה יותר מכיוון שנוכל להשתמש בסינונים לפי כתובות IP , והתמונה שתיווצר תהיה יותר ברורה.

### כיצד ניתן למתן את ההתקפה?

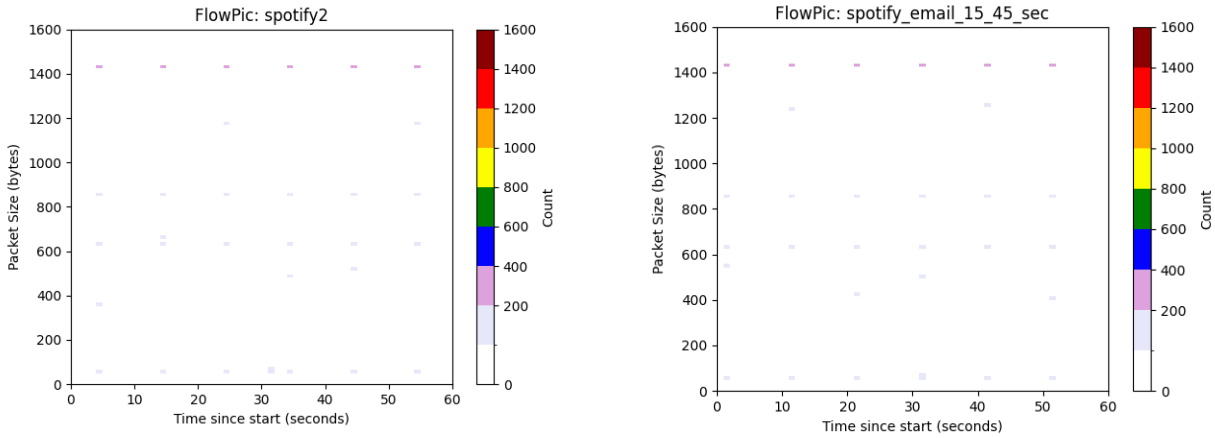
ניתן לגלוש בכמה אתרים ואפליקציות במקביל וכך להפריע לתוקף לזהות את האפליקציות (התמונה תצא מחוברת עבור כל האפליקציות ולא ברורה), בנוסף אנו יכולים להשתמש בvpn שמפזר את החבילות ברשת מה שמקשה על זיהוי וניתוח התעבורה. בנוסף נוכל להשתמש ב- Tor ולרפד את החבילות ובכך לשנות את התמונה.

**בונוס:**

השתמשנו בדוגמא שהבאתם – האזנה למוזיקה באמצעות Spotify ושליחת הודעות דוא"ל מדי פעם (השליחה של הדוא"ל הייתה 15 ו-45 שניות מתחילת ההקלטה).

**הנחה ראשונה: אנו יודעים את גדלי החבילות , זמני ההגעה ו PORT&IP של החבילות.**

ייצרנו תמונות FlowPics מסוננות בעזרת ה IP הכי נפוץ (לא IP עצמי) .

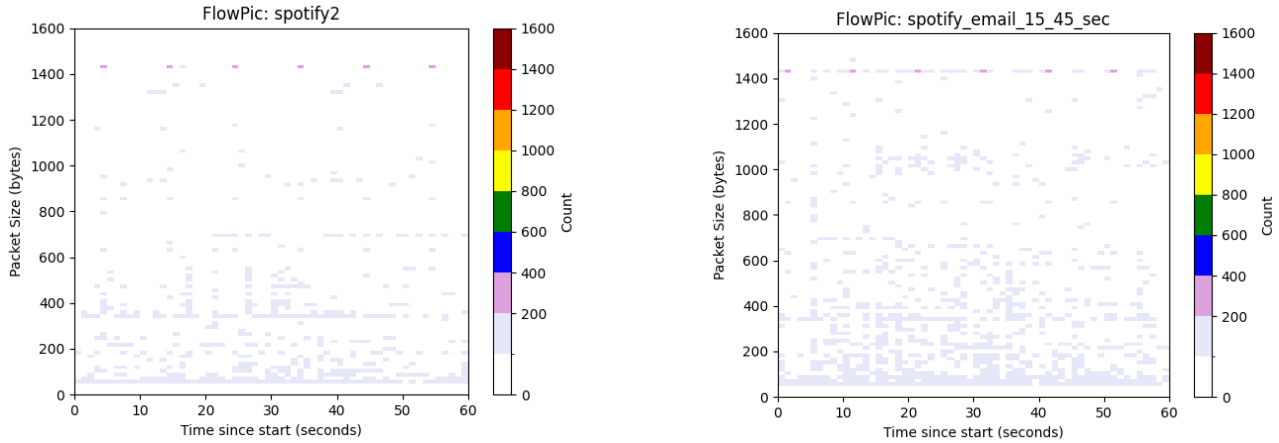



ניתן לראות שבעזרת סינון לפי הipכ הכי נפוץ קיבלתי תמונה שזהה לדפוס של אפליקציית spotify מסונן. כלומר ניתן לזהות את הגלישה בצורה מאוד מדוייקת, בנוסף אם נסנן לפי כתובת הipכ של הדוא"ל גם נקבל תמונה זהה לזרימת מידע של דוא"ל.

**לסיכום:** מסקנתינו היא שבעזרת סינון לפי כתובות ip נוכל להפריד כל אפליקציה לflowpic משלה וכך לזהות את כלל האפליקציות (בקוד שלנו סיננו רק את הipכ הכי נפוץ, אבל כמובן שניתן להפריד עבור כל ip).

**הנחה שנייה: אנו יודעים את גדלי החבילות ואת זמני ההגעה של החבילות בלבד.**

ייצרנו תמונות FlowPics לא מסוננות לפי כתובת IP.




עדיין יש דמיון כלשהו אבל התמונה הרבה יותר קשה לזיהוי – ה"לכלוך" של תעבורת הדוא"ל מאוד משפיע על התמונה, עדיין נישארה התבנית של גלישה בספוטיפיי (לדוגמה שורת הורודים כל 10 שניות קצת מעל קו 1400) אבל נוסף לה כיסוי מסויים.

**לסיכום:** הגלישה ב2 אתרים במקביל **מאוד משפיעה** על הזיהוי, אך עדיין קיים סיכוי כלשהו לזהות (גם אם נמוך יותר) במיוחד כשאחת האפליקציות הרבה יותר דומיננטית.

דוגמה זו **תומכת** בטענותינו על כיצד ניתן למתן את ההתקפה.

קישורים להתכתבויות עם ChatGPT :

https://chatgpt.com/share/67b5aa89-eff0-8013-a5c0-1d8b348a2521 - למידה לגבי חילוץ נתונים על פקטות בעזרת pyshark