

## תקשורת ומחשוב מטלה 4

מגישות:

רז אלבז ת"ז: 207276775

ליאור נגר ת"ז: 209399294

## חלק א' - DoH

### *DNS over HTTPS*

#### **שאלה 1:**

הציגו יתרון אחד לשימוש ב-DoH והסבירו אותו (כמובן, מעבר לעובדה שהוא מאובטח ומוצפן)

#### **פתרון שאלה 1:**

ראשית, נפרט את ראשי התיבות של פרוטוקול *DoH – DNS over HTTPS* פרוטוקול אשר מעביר מידע אודות DNS בצורה מוצפנת על גבי חיבורי *http*. אחד היתרונות הבולטים לשימוש ב-DoH מעבר לכך שהוא פרוטוקול המעביר את המידע בצורה מאובטח ומוצפנת הוא שנוכל לבדוק כיצד פרוטוקול זה ישתלב עם רשתות נוספות וטרם הפיכתו לפרוטוקול ברירת המחדל שלנו, נוכל לדעת מה יהיו הבעיות אשר נקבל בעתיד ולפתור אותן. לאחר התיקון נוכל להשתמש בו ולהימנע מהבעיות אשר היו נגררות אילולא נבדק קודם לכן.

#### **שאלה 2:**

הציגו והסבירו על שני חסרונות לשימוש בשיטת DoH לעומת DNS הרגיל.

#### **פתרון 2:**

##### **חסרון 1:**

אחד היתרונות של שימוש בפרוטוקול DNS הוא שישנה אפשרות לסינון ובכך להבטיח שלא תהיה גישה לאתרים אשר יכולים להוות נזק, אתרים אשר יכולים לגנוב את המידע האישי שלנו- כמו סיסמאות לאתרים, פרטים אישיים וכולי, או לעזור להורים למנוע גישה עבור אתרים בעלי תכנים לא הולמים עבור הילדים שלהם. אנשים וארגונים רבים מסתמכים על היתרון המשמעותי של הפרוטוקול הנ"ל.

החסרון של DoH לעומת ה-DNS הוא שכאשר מפעילים אותו, הוא עוקף את המדיניות של הפעלת מסנן ה-DNS ובכך ה-DoH לא מאפשר סינונים כמו בשירות ה-DNS. בכך, תפתח גישה לכל אתר אפשר מה שעלול להוות סיכון למשתמש. DNS גובר ביתרוננו על השימוש ב-DoH כאשר אין בו את האפשרות של הסינון.

##### **חסרון 2:**

החסרון השני כאשר אנו משתמשים ב-DoH לעומת DNS הוא שזמני התגובה של פרוטוקול DoH גבוהים יותר מאשר בפרוטוקול DNS מכיוון שבפרוטוקול זה אנו משתמשים בחיבור TCP אשר מבטיח אמינות אך מאט את אופן החיבור ולכן זמני התגובה של DoH איטיים יותר לעומת DNS שבו אנו משתמשים בפרוטוקול UDP.

**שאלה 3:**

בחרו אחד מהחסרונות משאלה (2), הציעו דרך למתן\לעקוף\לפתור חיסרון זה והסבירו אותה.

**פתרון 3:**

החיסרון שבחרנו הוא חיסרון 1 חוסר היכולת של הפרוטוקול לבצע סינונים ולמנוע כניסה לאתרים אשר יכולים להוות נזק למשתמש.

אחד הפתרונות הפשוטים הוא פשוט לכבות את פרוטוקול *DoH* בצורה ידנית עבור דפדפנים של חברות או ילדים מהם נרצה למנוע גישה לאתרים מסוימים. אך מכיוון שיש לו יתרונות רבים נרצה למצוא פתרון אשר נוכל מחד גיסא להשתמש בו ומאידך גיסא לפתור את אחד מהחסרונות שלו. הפתרון המוצע הוא להשתמש במסנני אינטרנט, מסנני האינטרנט יגבילו את המידע אשר נגיש למשתמש ללא חשיפה לתכנים לא רצויים כלשהם, הפעלת המסננים הנ"ל יכולה להתבצע בצורה ידנית על ידי שימוש בתוכנות מסוימות (או כיבוי ה-*DoH* כמו שהסברנו בפתרון הפשוט) או באמצעות ספק מסוים אשר מספק את השירותים של סינון הגלישה. הפתרון הנ"ל יוכל לעזור להורים לבצע סינון לאתרים מסוימים ולהגביל את הגישה של ילדיהם וחברות אשר מסתמכות על הבדיקה של תעבורת ה-DNS באינטרנט יוכלו להשתמש בתוכנות אשר יאפשרו להם לדעת באילו דפדפנים אשר משתמשים בפרוטוקול *DoH* העובדים שלהם גולשים ולהגביל אותם בהתאם.

**שאלה 4:**

ישנן 4 דרכים בהן ניתן לשלב את שיטת ה-*DoH* באינטרנט שלנו:

- מימוש *DoH* ברמת האפליקציות (לדוגמא: לעדכן את קוד הדפדפן כך שישלחו שאילתות דרך HTTPS)
  - מימוש *DoH* ברמת שרת proxy\* ברשת (מהמחשב לשרת נשלח לפורט 53 והלאה, כבר 443)
  - מימוש *DoH* ברמת שרת proxy מקומי (על המכונה רץ שרת proxy)
  - התקנת plugin המממש *DoH* ברמת הגדרות המחשב ("מעכשיו, אתה שולח רק *DoH*")
- כתבו השוואה בין כל ארבעת השיטות, בהשוואתכם הראו יתרונות וחסרונות לכל שיטה והציגו מהי, לדעתכם, השיטה המועדפת מבין הארבעה. כלומר, הציגו את השיטה בה, לדעתכם, היתרונות הגדולים ביותר לעומת החסרונות הקטנים ביותר.

**פתרון 4:****מימוש *DoH* ברמת האפליקציות:**

ניתן לממש את פרוטוקול *DoH* ברמת האפליקציה כאשר לדפדפנים יהיה מימוש של הפרוטוקול הנ"ל כברירת מחדל, בכך, הם יעקפו בשכבת האפליקציה את פרוטוקול ה-DNS הרגיל. היתרון בכך הוא שבאופן אוטומטי השאילתות יהיו מוצפנות, המשתמש יזכה להשיג הגנה על הפרטיות שלו והגנה על הנתונים שלו. אחד המימושים הוא על ידי כך שהמשתמש יגדיר בתור תחבורת השאילתות לאפליקציית האינטרנט או לדפדפן מסוים דרך HTTP. חיבור ה-HTTP מספק את אבטחת תחבורה עבור האינטראקציה בין שרתי *DoH* ללקוח. החיסרון הוא אם פרוטוקול *DoH* לא נתמך במחשב או אם הוא מקבלת שאילתות *DoH* שגויות אזי האפליקציה או הדפדפן יבצעו דילוג על אותן שאילתות.

### מימוש DoH ברמת שרת PROXY\* ברשת:

המימוש של DoH ברמת שרת PROXY ברשת יהיה על ידי התקנת PROXY אשר תומך ב-DoH בשרת השמות ברשת המקומית. השאילות יישלחו על ידי פרוטוקול DNS רגיל בפורט מספר 53 והתשובות לאותן שאילות יאספו על ידי השרת PROXY באמצעות פרוטוקול DoH. כל שאילות עוברת בהתחלה לשרת ה-PROXY, ה-PROXY יבצע את בקשות האינטרנט עבורנו ומחזיר לנו את המידע אודות דף האינטרנט בכדי שנוכל לראות אותו בדפדפן של המחשב שלנו. היתרון הראשון הוא שלא צריך להתקין את ה-PROXY בכל מערכת אשר ברצונה להשתמש בפרוטוקול DoH. היתרון השני הוא שכאשר שרת ה-PROXY יבצע את העבודה בשבילנו עם ה-IP שלו, אנחנו נהיה מוסתרים וכן הנתונים שלנו בזמן זה. החיסרון הוא שלשרת ה-PROXY יש תצורות שונות וכמו כן גם לרשת המקומית שלנו, על כן ישנה אפשרות בה שרת ה-PROXY לא יתאים תמיד לרשת. כאשר מצב כזה יקרה, ואם נצטרך להשתמש בשרת ה-PROXY ברשת המקומית שלנו, נצטרך להיות בעלי שרת PROXY אשר מתאים לרשת בה אנחנו משתמשים אחרת השימוש לא יהיה יעיל, והבעיה היא שלא בהכרח נהיה.

### מימוש DoH ברמת שרת PROXY מקומי

המימוש של DoH ברמת שרת PROXY מקומי הוא על ידי כך שבהגדרות של מערכות ההפעלה השאילות הן DNS OVER HTTP והן יבצעו אותם על שרתי PROXY אשר פועלים באופן מקומי. היתרון הוא ששרת ה-PROXY המבצע את השאילות בצורה מוצפנת ומאובטחת יגרום להגנה על נתוני המשתמש ועל הרשת הפנימית מפני אפליקציות ויישומים זדוניים אחרים. החיסרון הוא שיש צורך בהתקנת שרת ה-PROXY בכל אפליקציה או דפדפן אשר ברצונם להשתמש ב-DoH.

### התקנת plugin המממש DoH ברמת הגדרות

כאשר נתקין plugin המממש DoH ברמת הגדרות אנחנו נפעיל את DoH ברמת המערכת כך שמאותו הרגע כל תוכנה אשר מבוססת על שימוש באינטרנט וכל דפדפן יתמכו בפרוטוקול DoH. היתרון הוא שאין צורך להגדיר באופן ידני על כל אחד מהם את הגדרות ה-DoH אלא ההגדרות הנ"ל יחולו על ידי מערכת ההפעלה שלנו על כל דפדפן ועל כל אפליקציה שבה נרצה להשתמש באופן אוטומטי. החיסרון הוא שלא כל מחשב יתמוך ויכול לממש את השיטה, אם מערכת ההפעלה שלנו לא תהיה מעודכנת לגרסאות אשר תומכות אז לא נוכל ליהנות מההתקנה של ה-plugin המממש DoH ברמת הגדרות.

### **שאלה 5:**

נניח שאנו ברשת שקיים בה איבוד פקטות (packet loss) באחוז לא ידוע ואנו רוצים לטעון דף שצריך 25 שאילות כדי לבקש את כל המשאבים שבו. הציגו יתרון ברור שיש ל-DoH לעומת Do53. (רמז: מנגנון הקיים ב-TCP)

### **פתרון 5:**

היתרון הברור של שימוש ב-DoH על פני שימוש ב-Do53 הוא כאשר קיים מצב של איבוד פקטות. DoH משתמש בפרוטוקול TCP, פרוטוקול אשר שומר על אמינות מבחינת העברת מידע ומוודא שכלל הפקטות נשלחו אל יעדן, הוודאו מתבצע על ידי כך שלאחר כל פקטה שהגיעה הוא צריך לקבל אישור, כלומר במצב של איבוד פקטות, לא תשלח הודעת אישור ולאחר זמן מסוים תתבצע שליחה מחדש של הפקטה שנאבדה שוב אל היעד.

לעומת זאת, *Do53* אשר משתמש בפרוטוקול *UDP* אינו שולח בשנית פקטות שנאבדו והוא איננו ממתין לאישור על כל כך שכל פקטה הגיעה ליעדה. ולכן הפקטות שנאבדו לא ישלחו שוב.

לכן, כאשר אנו במצב שאינו ניתן לדעת מה אחוז איבוד הפקטות, אין ספק שב-DOH נקבל את כלל המידע מהדף המבוקש.

## חלק ב' - Congestion Control

צילומי מסך של ההרצה הראשונה (לפני איבוד):

:Measure

```

raz@raz-VirtualBox: ~/CLionProjects/ex4
#####
raz@raz-VirtualBox:~/CLionProjects/ex4$ ./Measure
num: 1 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 4.838000 millisecond
*****

num: 2 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 1.180000 millisecond
*****

num: 3 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 1.097000 millisecond
*****

num: 4 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 1.025000 millisecond
*****

num: 5 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 0.948000 millisecond

#####
The average of cubic is: 1.400000 millisecond
#####

```

```

raz@raz-VirtualBox: ~/CLionProjects/ex4
num: 1 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1.561000 millisecond
*****

num: 2 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1.647000 millisecond
*****

num: 3 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1.523000 millisecond
*****

num: 4 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1.894000 millisecond
*****

num: 5 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1.196000 millisecond

#####
The average of reno is: 5.000000 millisecond
#####

```

:Sender

```

raz@raz-VirtualBox: ~/CLionProjects/ex4$ ./sender
Socket num: 1
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 2
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 3
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 4
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 5
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 6
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 7
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 8
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 9
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

Socket num: 10
Trying to connect to the server...
Connect to server successfully!
The number of bytes that sent: 2097150
The file was sent
Closing the connection to server!
*****

```

## טבלת זמנים

קובץ מספר	איבוד 30%	איבוד 25%	איבוד 20%	איבוד 15%	איבוד 10%	קובץ מספר
<b>Cubic</b>						
1	63040.801	1998.190000	2534.741000	5.048000	215.45600	1
2	6116.9750	9418.39000	19584.03800	1161.520000	210.24700	2
3	89146.167	13495.56600	13792.07300	432.505000	65.535000	3
4	2665.7530	6400.027000	413.789000	3799.960000	763.34300	4
5	136575.223	17313.55000	2337.768000	1724.987000	17.790000	5
	59508.400	9724.800000	7732.000000	1424.200000	254.00000	<b>ממוצע</b>
<b>Reno</b>						
1	לא מסתיים	92423.51000	61517.53100	468.823000	41.053000	1
2		5552.475000	1234.391000	1803.950000	1698.2150	2
3		82450.44900	9727.897000	13849.29300	2048.1200	3
4		420.937000	2.333000	6590.662000	323.17600	4
5		16174.58200	4182.495000	1924.759000	408.93000	5
		39403.80000	15332.40000	4956.800000	903.93000	<b>ממוצע</b>

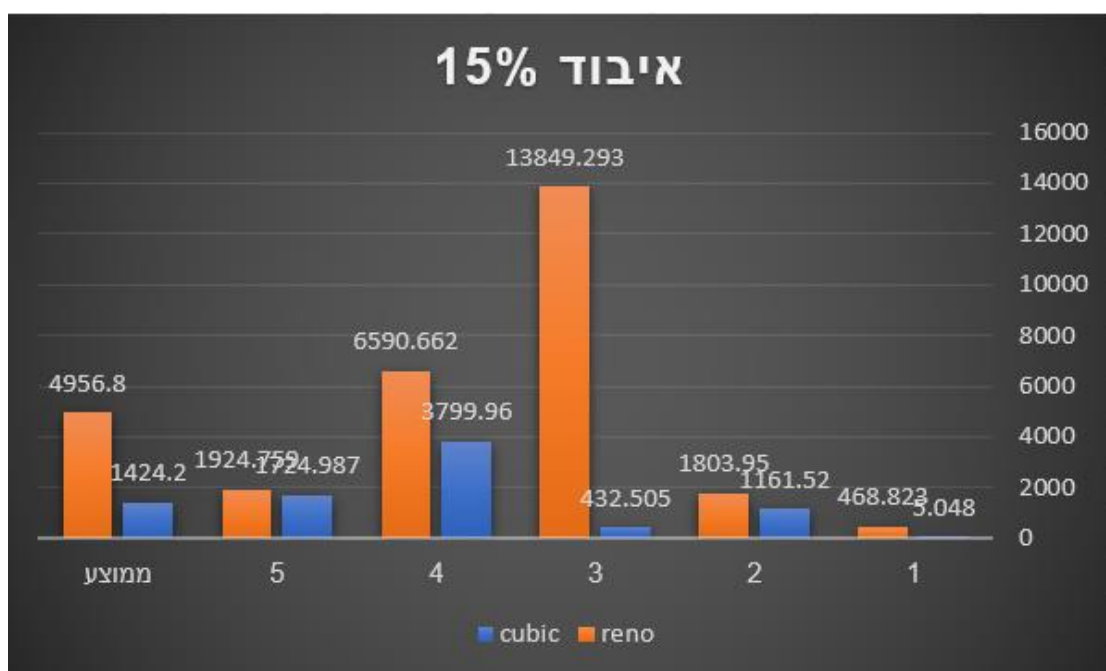
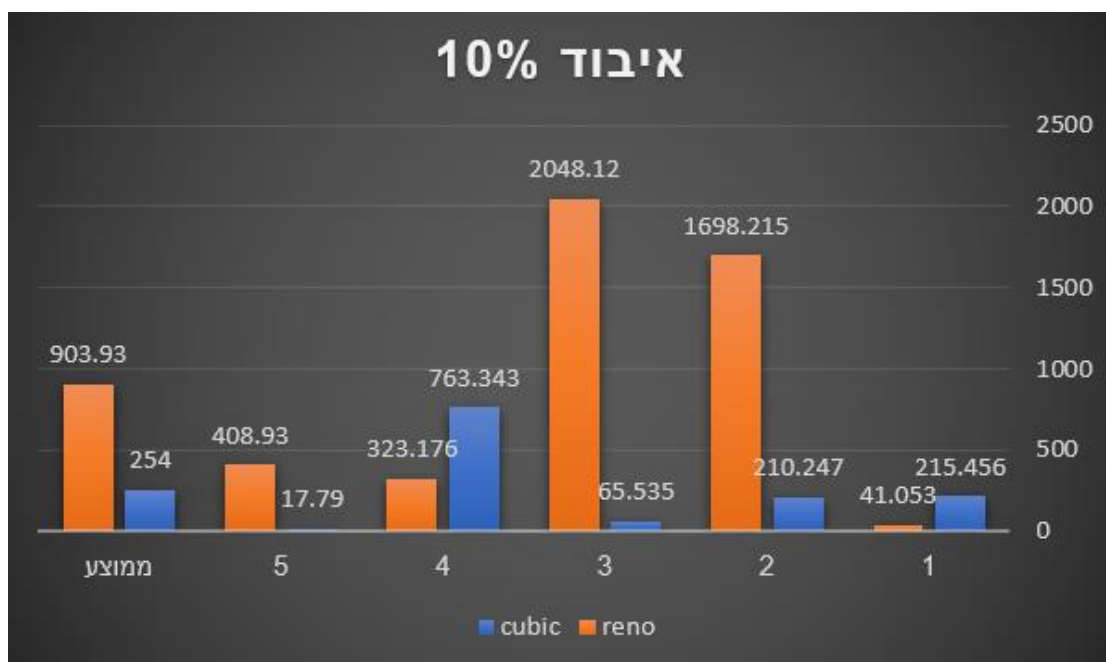
### הסבר להבדלים בין שני האלגוריתמים:

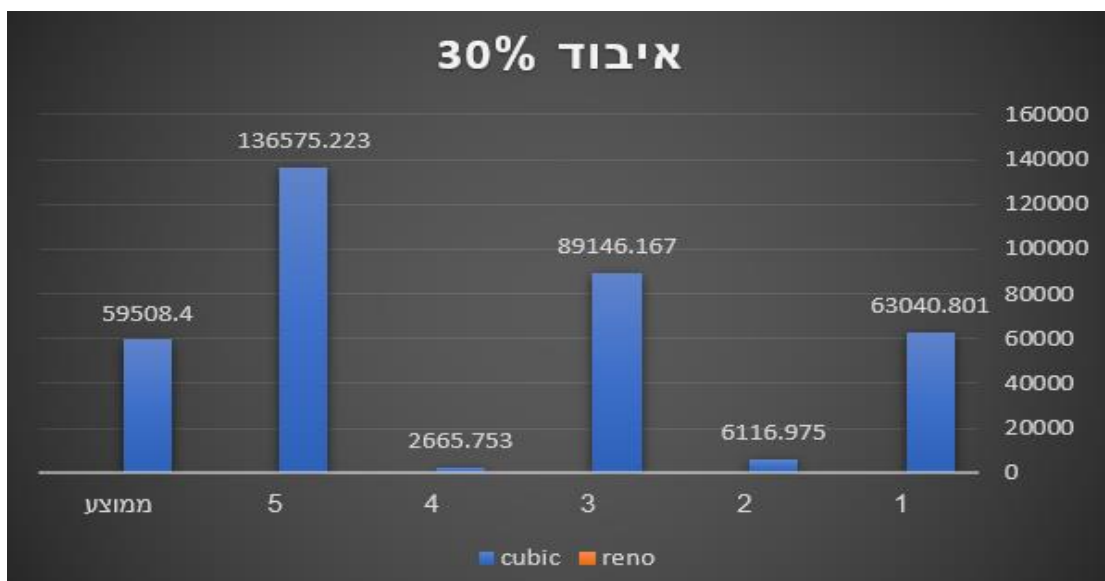
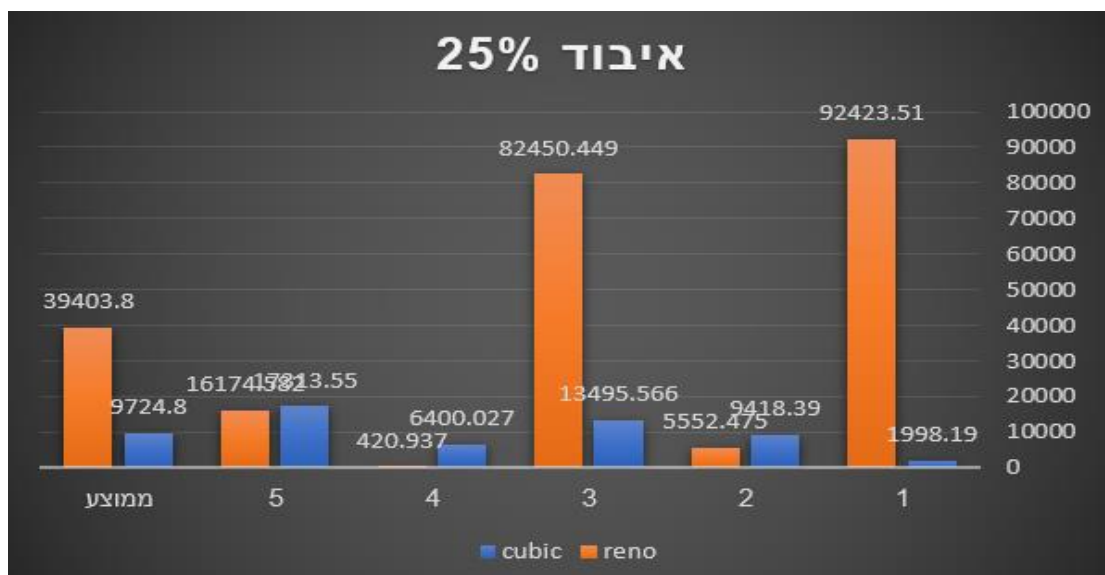
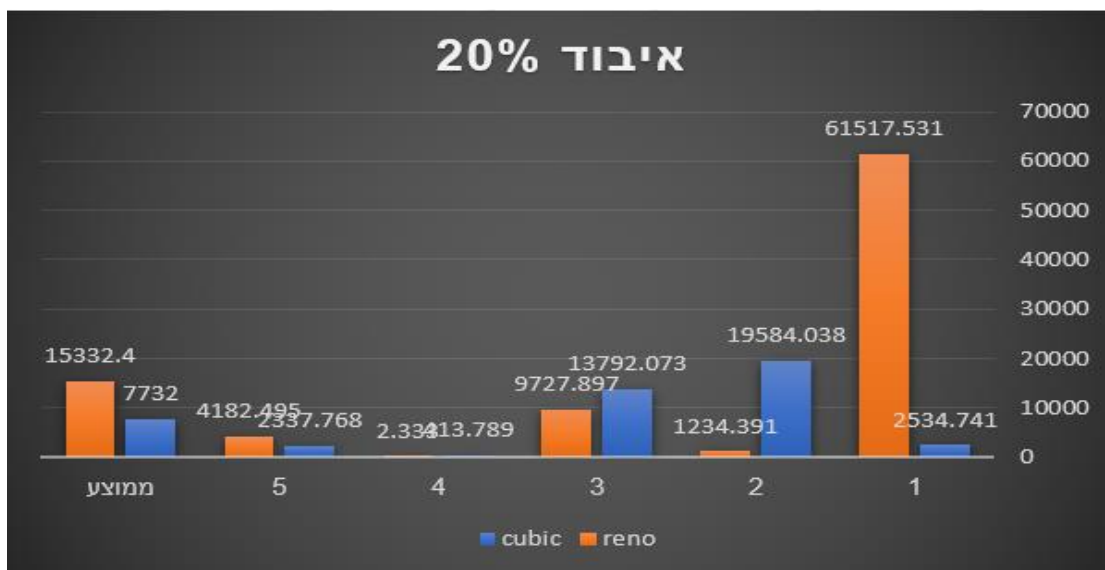
ראשית כל, נאמר ששני האלגוריתמים משתמשים בפרוטוקול TCP הידוע בתור פרוטוקול איטי אך אמין.

שנית, זמן הריצה של שליחת החבילות על ידי האלגוריתם *Reno* איטי יותר מהאלגוריתם *Cubic* מכיוון שהוא משתמש בחיבור איטי יותר בכדי להימנע ממצב שבו נשלחות הרבה פקטות והרשת לא תעמוד בעומס. מנגד, *Cubic* הוא בעל בקרה של עומס ולכן יכול להשתמש בחיבור איטי בהתחלה ואז לעלות למהירות גבוהה ממש של שליחת הפקטות.



## גרפים עבור אחוזי האיבוד





## איבוד 10% צילומי מסך:

```

#####
raz@raz-VirtualBox: ~/CLionProjects/ex4$ sudo tc qdisc add dev lo root netem loss
10%
raz@raz-VirtualBox:~/CLionProjects/ex4$ ./Measure
num: 1 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 215.456000 millisecond
*****

num: 2 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 210.247000 millisecond
*****

num: 3 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 65.535000 millisecond
*****

num: 4 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 763.343000 millisecond
*****

num: 5 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 17.790000 millisecond

#####
The average of cubic is: 254.000000 millisecond
#####
*****

```

```

num: 1 successfully received -reno-
The number of bytes that sent: 2097150
Time: 41.053000 millisecond
*****

num: 2 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1698.215000 millisecond
*****

num: 3 successfully received -reno-
The number of bytes that sent: 2097150
Time: 2048.120000 millisecond
*****

num: 4 successfully received -reno-
The number of bytes that sent: 2097150
Time: 323.176000 millisecond
*****

num: 5 successfully received -reno-
The number of bytes that sent: 2097150
Time: 408.930000 millisecond

#####
The average of reno is: 903.600000 millisecond
#####

```

## איבוד 15% צילומי מסך:

```

raz@raz-VirtualBox:~/CLionProjects/ex4$ ./Measure
num: 1 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 5.048000 millisecond
*****

num: 2 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 1161.520000 millisecond
*****

num: 3 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 432.505000 millisecond
*****

num: 4 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 3799.960000 millisecond
*****

num: 5 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 1724.987000 millisecond

#####
The average of cubic is: 1424.200000 millisecond
#####
*****

```

```

num: 1 successfully received -reno-
The number of bytes that sent: 2097150
Time: 468.823000 millisecond
*****

num: 2 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1803.950000 millisecond
*****

num: 3 successfully received -reno-
The number of bytes that sent: 2097150
Time: 13849.293000 millisecond
*****

num: 4 successfully received -reno-
The number of bytes that sent: 2097150
Time: 6590.662000 millisecond
*****

num: 5 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1924.759000 millisecond

#####
The average of reno is: 4926.800000 millisecond
#####

```

## איבוד 20% צילומי מסך:

```

raz@raz-VirtualBox:~/CLionProjects/ex4$ sudo tc qdisc add dev lo root netem loss
20%
[sudo] password for raz:
raz@raz-VirtualBox:~/CLionProjects/ex4$ ./Measure
num: 1 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 2534.741000 millisecond
*****

num: 2 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 19584.038000 millisecond
*****

num: 3 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 13792.073000 millisecond
*****

num: 4 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 413.789000 millisecond
*****

num: 5 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 2337.768000 millisecond

#####
The average of cubic is: 7732.000000 millisecond
#####
*****

num: 1 successfully received -reno-
The number of bytes that sent: 2097150
Time: 61517.531000 millisecond
*****

num: 2 successfully received -reno-
The number of bytes that sent: 2097150
Time: 1234.391000 millisecond
*****

num: 3 successfully received -reno-
The number of bytes that sent: 2097150
Time: 9727.897000 millisecond
*****

num: 4 successfully received -reno-
The number of bytes that sent: 2097150
Time: 2.333000 millisecond
*****

num: 5 successfully received -reno-
The number of bytes that sent: 2097150
Time: 4182.495000 millisecond

#####
The average of reno is: 15332.400000 millisecond
#####

```

## איבוד 25% צילומי מסך:

```

raz@raz-VirtualBox: ~/CLionProjects/ex4
raz@raz-VirtualBox:~/CLionProjects/ex4$ sudo tc qdisc add dev lo root netem loss 25%
[sudo] password for raz:
raz@raz-VirtualBox:~/CLionProjects/ex4$ ./Measure
num: 1 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 1998.190000 millisecond
*****

num: 2 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 9418.319000 millisecond
*****

num: 3 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 13495.566000 millisecond
*****

num: 4 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 6400.027000 millisecond
*****

num: 5 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 17313.550000 millisecond

#####
The average of cubic is: 9724.800000 millisecond
#####
*****

num: 1 successfully received -reno-
The number of bytes that sent: 2097150
Time: 92423.510000 millisecond
*****

num: 2 successfully received -reno-
The number of bytes that sent: 2097150
Time: 5552.475000 millisecond
*****

num: 3 successfully received -reno-
The number of bytes that sent: 2097150
Time: 82450.449000 millisecond
*****

num: 4 successfully received -reno-
The number of bytes that sent: 2097150
Time: 420.937000 millisecond
*****

num: 5 successfully received -reno-
The number of bytes that sent: 2097150
Time: 16174.582000 millisecond

#####
The average of reno is: 39403.800000 millisecond
#####

```



## איבוד 30% צילומי מסך:

```

raz@raz-VirtualBox: ~/CLionProjects/ex4
raz@raz-VirtualBox:~/CLionProjects/ex4$ ./Measure
num: 1 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 63040.801000 millisecond
*****

num: 2 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 6116.975000 millisecond
*****

num: 3 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 89146.167000 millisecond
*****

num: 4 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 2665.753000 millisecond
*****

num: 5 successfully received -cubic-
The number of bytes that sent: 2097150
Time: 136575.223000 millisecond
*****

#####
The average of cubic is: 59508.400000 millisecond
#####
*****

```

הסבר: היות והקובץ גדול, ולאחר ניסיונות מרובים של הרצות, הריצה של התוכנית לא מסתיימת כאשר ניסינו להריץ את האלגוריתם של RENO, הפרוטוקול משתנה מ-TCP ל-DNS ולאחר התייעצות עם ד"ר עמית דביר התבקשנו לצרף תמונת Wireshark אשר תציג מדוע השליחה לא מסתיימת. בנוסף, צירפנו את הקלטת ה-Wireshark עד למועד העצירה.

The screenshot shows a terminal window on the left and a Wireshark packet capture on the right. The terminal window displays the output of a program that sends data in cubic segments. The output shows five successful receptions of cubic segments, each with a size of 2097150 bytes. The average time for the cubic segments is 59508.400000 milliseconds. The Wireshark packet capture shows the network traffic. The packet list on the left shows a series of TCP segments. A blue arrow points to a TCP segment in the packet list. The packet details on the right show the structure of the selected packet, including the Ethernet II header, Internet Protocol Version 4 header, and User Datagram Protocol header.

.







