

## תקשורת ומחשוב – מטלה 4

### מגישים: חיים גולדפישר 315599563, אור יצחק 208936039

#### חלק א' - DoH

1. כיוון שה-DoH מרכז תעבורת נתוני DNS של כמה שרתים שמאפשרים שימוש ב-DoH זמן הטעינה משתפר בדרך כלל.
2. א'. שימוש ב-DoH יכול לפגוע בחברות אבטחה שמשתמשות בשאילתות DNS שעוברות ברשת לצורך איסוף מידע, בכך ש-DoH מצפין את הנתונים האלו ובכך מונע מהחברות לגשת למידע ולהשתמש בו.
- ב'. אם ברשת שלנו יש DNS filtering שאמור למנוע מאיתנו לגשת לאתרים מסוימים, הוא עלול לא לעבוד כאשר אנו משתמשים ב-DoH כי אנחנו מצפינים את כתובת האתר שאליה אנו רוצים לגשת הפתרון לבעיה ב' הוא לבצע את הסינון עוד ברמת הדפדפן כך ניתן לגשת אל ה-DNS עוד לפני שה-DoH הצפין אותו ולחסום את הגישה אל ה-DNS הזה. בפתרון זה יש להתקין על המחשב שממנו מנסים לגשת לאתר האסור את התוכנה והיא תמנע ממנו לגשת אל האתר. סוג פתרון זה נקרא agent based web filters.
4. א'. **יתרונות** – זה די נוח לשימוש משום שמספיק רק להתקין את התוסף על הדפדפן ולהפעיל את השימוש ב-DoH. **חסרונות** – קיימים דפדפנים שהתוסף לא קיים אצלם. כלומר זה מגביל את השימוש רק בדפדפנים שניתן להתקין בהם את התוסף.
- ב'. **יתרונות** – ניתן לשלוח דרכו שאילתות DNS שאינן מוצפנות ושרת הפרוקסי כבר יצפין אותן וישלח אותן לכל רשת. בנוסף אין צורך לשנות את הגדרות המערכת. **חסרונות** – בגלל שהשאילתה נשלחת כפי שהיא לשרת הפרוקסי, קיים חשש שבזמן השליחה מהמשתמש לשרת היא "תיתפס" – בעיות אבטחה.
- ג'. **יתרונות** – השרת מתעסק אך ורק ב-DoH ללא שירותים אחרים ואין אפשרות לבעיות אבטחה משום שהשרת הוא מקומי ורץ על המכונה. **חסרונות** – נחשבת לשיטה "יקרה" משום שרצה ע"ג המכונה.
- ד'. **יתרונות** – אין כאן תלות בדפדפן כזה או אחר. כלומר השאילתות נשלחות דרך ה-DoH ללא "מתווכים" כלשהם. **חסרונות** – התקנה חיצונית מחייבת נתינת גישה להגדרות של המחשב לתוכנות חיצוניות. הדבר עלול לגרום לבעיות אבטחה.
- לדעתנו, השיטה העדיפה ביותר היא א'.** הדבר נובע מכך שמבחינת בעיות הצפנה (טעמי אבטחה), השיטה הנ"ל היא העדיפה ביותר. משום שהיא רצה על גבי הדפדפן ולא דורשת את הגדרות המחשב. כמו כן, הריצה אינה ע"ג המכונה (פעולה "יקרה"). מבחינת החסרונות שהצגנו, כיום הרוב המוחלט משתמשים בדפדפנים שמאפשרים את התקנת התוסף ולכן החיסרון הנ"ל הינו שולי ביחס ליתרונות.
5. כיוון שה-DoH משתמש בפרוטוקול TCP אם נאבדות פקטות בדרך הוא יבקש אותם עוד פעם עד שיקבל את כל המידע בעוד ש-53D משתמש בפרוטוקול UDP שאיננו מוודא את תקינות ההודעה שקיבל ולכן לא יבחין שחסר מידע.

#### חלק ב' - Congestion Control

יצרנו קובץ בגודל 1 מגה בייט. 2 התוכניות מצורפות למטלה וגם הפאקטות מהוויירשארק בכל אחד מהמקרים.

דוגמה לפלט של הפעלת התוכנית של sender.c (זהה בכל ההרצות):

```
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ ./sender
starting sender
iteration No. 1
Successfull connection!
Successfull connection!
iteration No. 2
Successfull connection!
Successfull connection!
iteration No. 3
Successfull connection!
Successfull connection!
iteration No. 4
Successfull connection!
Successfull connection!
iteration No. 5
Successfull connection!
Successfull connection!
```

כפי שניתן לראות, כל איטרציה מפעילה פעם אחת את "רנו" ופעם אחת את "קיוביק". החיבור לסוקט מתבצע עבור כל אחד מהאלגוריתמים. הפלט הוא אותו פלט בכל המקרים שהיינו צריכים להריץ.

כעת נציג את כל הפלטים שהתקבלו עבור כל אחד מאחוזי האיבוד עבור התוכנית של Measure.c:

0 אחוזי איבוד:

```
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ ./measure
starting measure
Listening to port 7890
iteration No. 1
the current Reno Time is 0.004675
the current Cubic Time is 0.001485
iteration No. 2
the current Reno Time is 0.000503
the current Cubic Time is 0.001333
iteration No. 3
the current Reno Time is 0.001489
the current Cubic Time is 0.001301
iteration No. 4
the current Reno Time is 0.000430
the current Cubic Time is 0.001129
iteration No. 5
the current Reno Time is 0.000543
the current Cubic Time is 0.000656
Average:
the AVG Reno Time is 0.001528
the AVG Cubic Time is 0.001181
closing socket
```

10 אחוזי איבוד:

```
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ sudo tc qdisc add dev lo root netem loss 10%
[sudo] password for goldi:
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ ./mes
starting measure
Listening to port 7890
iteration No. 1
the current Reno Time is 1.277732
the current Cubic Time is 0.236491
iteration No. 2
the current Reno Time is 0.232910
the current Cubic Time is 0.210542
iteration No. 3
the current Reno Time is 0.024451
the current Cubic Time is 1.691434
iteration No. 4
the current Reno Time is 0.000505
the current Cubic Time is 0.223772
iteration No. 5
the current Reno Time is 0.683455
the current Cubic Time is 0.000866
Average:
the AVG Reno Time is 0.443811
the AVG Cubic Time is 0.472621
closing socket
```

### 15 אחוזי איבוד:

```
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ sudo tc qdisc change dev lo root netem loss 15%
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ ./mes
starting measure
Listening to port 7890
iteration No. 1
the current Reno Time is 2.860336
the current Cubic Time is 4.115465
iteration No. 2
the current Reno Time is 0.240352
the current Cubic Time is 1.656143
iteration No. 3
the current Reno Time is 0.267310
the current Cubic Time is 0.032747
iteration No. 4
the current Reno Time is 0.471792
the current Cubic Time is 3.100477
iteration No. 5
the current Reno Time is 2.614520
the current Cubic Time is 0.240471
Average:
the AVG Reno Time is 1.290862
the AVG Cubic Time is 1.829061
closing socket
```

### 20 אחוזי איבוד:

```
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ sudo tc qdisc change dev lo root netem loss 20%
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ ./mes
starting measure
Listening to port 7890
iteration No. 1
the current Reno Time is 0.847955
the current Cubic Time is 1.108470
iteration No. 2
the current Reno Time is 0.016177
the current Cubic Time is 3.536096
iteration No. 3
the current Reno Time is 2.699730
the current Cubic Time is 0.847930
iteration No. 4
the current Reno Time is 16.959727
the current Cubic Time is 1.223607
iteration No. 5
the current Reno Time is 13.016669
the current Cubic Time is 0.000693
Average:
the AVG Reno Time is 6.708051
the AVG Cubic Time is 1.343359
closing socket
```

### 25 אחוזי איבוד (רק קיוביק הצליח לרוץ):

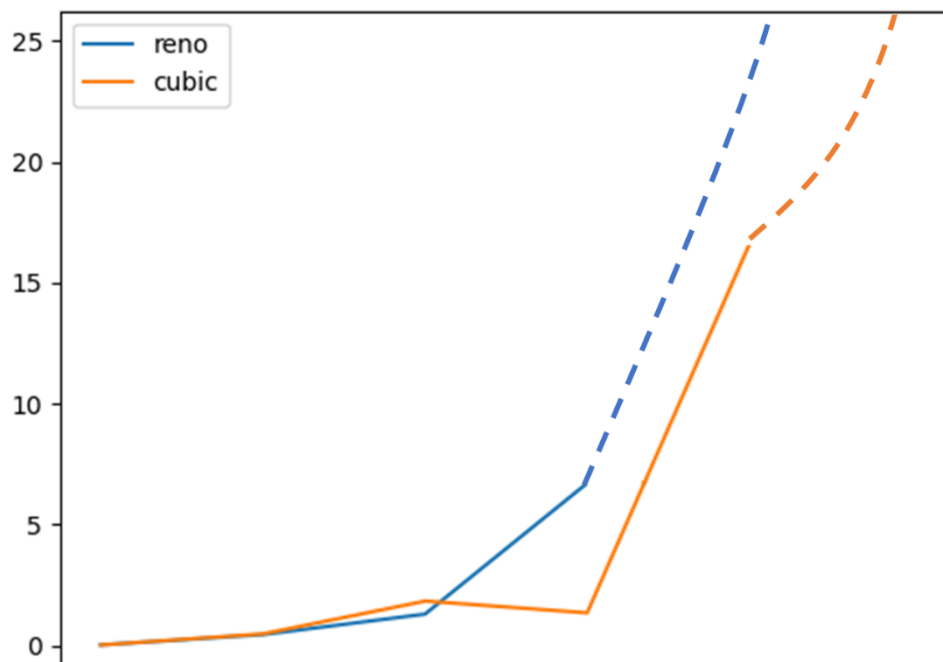
```
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ sudo tc qdisc change dev lo root netem loss 25%
[sudo] password for goldi:
goldi@ubuntu:~/Desktop/My_Projects/Network/Ex4$ ./mes
starting measure
Listening to port 7890
iteration No. 1
the current Cubic Time is 3.551632
iteration No. 2
the current Cubic Time is 0.495680
iteration No. 3
the current Cubic Time is 1.236359
iteration No. 4
the current Cubic Time is 0.211893
iteration No. 5
the current Cubic Time is 77.415593
Average:
the AVG Cubic Time is 16.582232
closing socket
```

30 אחוזי איבוד: לא העלנו משום ששני האלגוריתמים לא נעצרו.

טבלה מסכמת:

	0% איבוד	10% איבוד	15% איבוד	20% איבוד	25% איבוד	30% איבוד
<b>RENO</b>	0.01528	0.443811	1.290862	6.70851	TIMEOUT	TIMEOUT
<b>CUBIC</b>	0.01181	0.472621	1.829061	1.343359	16.582232	TIMEOUT

נציג את ההבדלים בין שני האלגוריתמים באמצעות גרף:



מסקנות: מהגרף נובע שבהתחלה יש ל"רנו" יתרון קטן על פני "קיוביק". החל מנק' מסיומת "קיוביק" הרבה יותר יעיל מ"רנו". החל מ-30% איבוד שניהם לא עצרו. נשים לב שבמצב רגיל בו אין איבוד פאקטות, שני האלגוריתמים יחסית זהים. ההבדל מתרחש כאשר יש איבוד פאקטות. אופן הפעולה של שני האלגוריתמים עד שלב איבוד הפאקטות הוא אקספוננציאלי. כאשר מתרחש איבוד פאקטות, אלגו' "רנו" יקטין את קצב שליחת הפאקטות בחצי ומשם יבור לקצב צמיחה לינארית. כל התקלות נוספת באיבוד יגרום לו להקטין בחצי שוב וכן הלאה. לעומת זאת, אלגו' קיוביק משתמש בשיטה שונה המתסמכת על חיפוש בינארי. נשים לב שכאשר אחוז איבוד הפאקטות הוא גבוה, אלגו' רנו נדרש להקטין את עצמו מספר רק של פעמים כך שהוא הופך לאיטי מאוד, בעוד שאלגו' קיוביק מצליח להתמודד עם הבעיה בצורה טובה יותר באמצעות שימוש ברוחב פס פנוי ע"י חיפוש.