

# בטיחות רשתות

פרויקט גמר:

*Thompson Sampling-Based Opportunistic Routing*

סער בן יוחנה: \*\*\*\*\*

אפי קורנפלד: \*\*\*\*\*

## תוכן עניינים

2		תקציר
3		הקדמה
3	1. תיאור כללי של הבעיה	
5	2. תוצאות עיקריות	
7	3. עבודות נוספות בתחום	
9		מודל המערכת וניסוח הבעיה
12		פתרונות לבעיה
12	1. האלגוריתם המוצע	
17	2. ניתוח ביצועים	
22		סקירת הסימולציה
26	איסוף סטטיסטיקות ונתונים	
26	מקרי בוחן	
27		תוצאות הסימולציה
27	תוצאות הסימולציה עבור טופולוגיה (a)	
34	תוצאות הסימולציה עבור טופולוגיה (b)	
41		מסקנות וביקורת על המאמר
41	1. נקודות חוזקה וחולשה	
42	2. כיווני מחקר עתידיים	
43		נספחים
46		סימוכין
47		הוראות להרצת הסימולציה

## תקציר

פעולת הניתוב מהווה חלק אינטגרלי ובסיסי בתכנון והקמה של רשתות תקשורת. כל משתמש ברשת נדרש לדעת כיצד עליו להעביר את החבילה על מנת שתגיע ליעדה, ברשתות אשר מוקמות על בסיס תשתית קיימת ויציבה, זוהי בעיה פתורה, הפתרון מתקבל ע"י קביעת הניתוב עבור כל רכיב בתוך הרשת בצורה סטטית או ע"י אלגוריתמי ניתוב קיימים למשל: *RIP, OSPF, IS-IS* עבור *intra-domain routing* ו-*BGP* עבור קביעת מדיניות ניתוב כלפי חוץ (*inter-domain routing*).

ניתוב עלולה להיות בעיה מורכבת יותר כאשר עוסקים ברשתות אד-הוק, רשתות אלחוטיות עצמאיות המשתנות בזמן באופן תדיר, ברשתות אלו אין ישות מרכזית אשר קובעת את המדיניות לכלל רכיבי הרשת ומתאמת ביניהם כיצד עליהם לגשת לרשת, יתרה מזאת, כל צומת ברשת מנהלת תקשורת בינה לבין שאר הצמתים הישיגים ממנה ללא כל תשתית מאורגנת, קביעת החלטת הניתוב הינה משימה מאתגרת ברשתות אלו עקב הדינאמיות הגבוהה שמאפיינת אותן, כל צומת נדרש תחילה לוודא שהוא ישיג ליעד והמסלול שלו אליו בכלל קיים באותה נקודת זמן, רק במידה וכן אז הוא יוכל לנסות להעביר את חבילת המידע שלו וכמובן שאין כל ערובה לכך שהחבילה אכן תגיע ליעדה כיוון שמדובר בערוץ אלחוטי בעל הסתברות שגיאה ולכן מתרחש אובדן חבילות.

אחת הדרכים להתמודד עם הבעיה הינה למצוא אלגוריתם אשר תחילה יבדוק את הערוץ ויחכה להזדמנות טובה לשידור, כלומר, נקודה בזמן בה צומת היעד ישיגה מהמקור והסתברות הצלחת השידור הינה גבוהה יחסית, אלגוריתמי ניתוב שמבצעים את הבדיקה המתוארת לפני השליחה נקראים אלגוריתמים מבוססי הזדמנויות (*OR – Opportunistic Routing*), צומת שרוצה לשדר תחילה מפיץ חבילת "מחקר" לכלל הצמתים הישיגים ממנו בתקווה לקבל חזרה חייווי על קיום מסלול תקין ליעד, לאחר קבלת התשובה הוא תשדר לצומת השכן דרכו המסלול עובר, צומת זה יבצע תהליך דומה עד אשר החבילה תגיע ליעדה.

שיטת ניתוב זו נקראת מבוססת הזדמנויות כיוון שברגע שצומת שכן מדווח שהוא ישיג, המקור מנצל את ההזדמנות להעביר את המידע דרכו וייתכן כי אם לא יעשה זאת, בהמשך אחד הלינקים על גבי המסלול ייפול ולא יהיה זמין יותר ואילו צומת המקור יישאר לא ישיג יותר מהיעד.

במאמר הוצגה שיטה לשיפור ביצועים של אלגוריתם מבוסס הזדמנויות ע"י דגימת תומפסון הנקרא *TSOR*, האלגוריתם מיועד לביצוע החלטות ניתוב ע"פ מטריקת מחיר המינימלי ברשתות אד-הוק והוא מבוסס בחלקו על האלגוריתם של בלמן-פורד למציאת מסלולים במרחק מינימלי ע"פ דיווחים של וקטור המרחקים מצמתים שכנים.

בדומה לכל אלגוריתם מבוסס הזדמנויות, גם *TSOR*, מניח שאין לצומת שום מידע מקדים אודות שכניו או הערוץ המשותף עימם, האלגוריתם מגדיר פרוטוקול בעל מספר הודעות בקרה שמטרתן להשתלב בתוך תהליך השידור הסטנדרטי של צומת ובעזרתן להצליח לשערך את אופי הערוץ ולאמוד את המסלול ליעד בעל המחיר המינימלי האפשרי ברגע השידור, החבילות שמתווספות מהוות אינדיקציה עבור צומת המקור לגבי הישגות ליעד, כמו כן מספקות את עדכון המחירים של השכנים בצורה דינאמית ושערוך הפרמטרים של הערוץ. ייחודיות האלגוריתם הינה שיטת הדגימה של תומפסון, עבור כל אינדיקציה מהרשת מעדכנים הצמתים את הפרמטרי הערוץ שלהן (באופן עצמאי ומבוזר).

במאמר מוצגת סימולציה של שתי טופולוגיות רשת אשר מדגימה בתוצאותיה את יעילותו של האלגוריתם *TSOR* ביחס לאלגוריתם אופטימלי וביחס לאלגוריתם מבוסס הזדמנויות אחר הפועל בשיטה דומה, התוצאות הראו כי *TSOR* אכן מצליח להשיג ביצועים מרשימים.

## הקדמה

### 1. תיאור כללי של הבעיה

ישנן מערכות רבות אשר ניתן למדלן ע"י גרף בו הצמתים הינם נקודות התקשרות והקשתות בו מייצגות את הקשרים ביניהם, למשל, עבור מודלים של רשתות תקשורת, הצמתים יכולים לייצג את המשתמשים ברשת והקשתות את התווך המשותף ביניהן, גרף כזה מהווה את טופולוגיית הרשת. הקשרים ברשת יכולים להיות דינאמיים ולהשתנות כפונקציה של הזמן וכאשר משתמש הופך ללא זמין ברשת, אותם קשרים המיוצגים ע"י הקשתות המחוברות אליו מוסרות והמשתמש הופך ללא ישיג מאף משתמש אחר, מצב כזה מתאר טופולוגיית רשת דינאמית.

מערכות המבוססות רשתות אד-הוק המהוות רשתות אלחוטיות דינאמיות (בעלות טופולוגיה המשתנה בזמן) בהן הצמתים מייצגים משתמשים אשר מנהלים אינטראקציה עם שכניהם ומקימים תקשורת באופן עצמאי ובלתי תלוי בשכניהם, ללא ידע מקדים וללא תשתית מוסדרת, ניתנות למידול באופן הנ"ל.

בעיה בסיסית במודלים מסוג זה הינה ביצוע של פעולת הניתוב, כלומר, בחירה עצמאית ומקומית של סדרת צמתי ביניים וצומת יעד אשר יחד יהוו מסלול מאותה צומת מקור ליעד המבוקש. כיוון שהרשת פועלת על בסיס טופולוגיה המשתנה בזמן ישנן מספר בעיות שאיתן צמתים ברשת נדרשות להתמודד:

(1) בעיית סנכרון – הבחירה של המסלול שמבוצעת בכל צומת באופן עצמאי צריכה להיות בראש ובראשונה מסלול אפשרי (ישיג) בגרף וללא מעגלי ניתוב אינסופיים, לכן על מנת שצומת תהיה מסוגלת לקבל החלטה אפשרית שכזו עליה תחילה להשיג מידע מסוים אודות טופולוגיית הרשת, כלומר לממש מנגנון (פרוטוקול) אשר יאפשר לה לחקור ולאסוף מידע אודות מבנה הרשת הנוכחי.

(2) בעיית אופטימיזציה – בדומה לרשתות מבוססות תשתית, כל צומת תשאף לבצע את פעולת הניתוב כך שהיא תמזער עבורה את פונקציית המחיר, אלגוריתם למציאת מסלול ניתוב במחיר מינימלי נדרש לקבל גם את המטריקה לפיה המרחקים מחושבים, לרוב ברשתות פנים ארגוניות מקובל להתייחס למספר ה-*hop-count* – מספר צמתי הביניים, בהן יש לעבור על מנת להגיע ליעד, ייתכן שלכל קשת במסלול יהיה מחיר שונה עבור העברת המידע בו, בנוסף, ייתכן גם כי ישנם פרמטרי רשת אחרים שיהיו חשובים יותר בניתוב, למשל ברשתות זמן-אמת לרוב פרמטר השיהוי יהיה בעל משקל רב יותר מהמרחק הנמדד במספר צמתי ביניים כיוון שלחבילות אלו יש ערך שהולך ופוחת בזמן.

(3) בעיית שערור – כאשר הטופולוגיה משתנה בזמן ובנוסף המידע אודות התווך המשותף בין הצמתים לא ידוע, מתווסף גם ממד הסתברותי לשידור מוצלח, לכן כחלק מהפרוטוקול המבוסס שעל כל צומת להריץ בנפרד על מנת לחקור את הרשת, עליו גם לבצע שערור אודות כל אחד מהלינקים לשכניו בטרם כל יציאה לשידור אחרת, הוא עלול לשלוח את החבילה דרך שכן שבעת צומת היעד כלל לא ישיג ממנו.

הבעיות המוצגות לעיל, כאשר מתקיימות יחד מייצגות ניתוב מבוסס הזדמנויות (*OR (Opportunistic Routing)*, כלומר, צומת אשר ברצונה לבצע ניתוב צריכה לתפוס הזדמנות לשידור – נקודת זמן בה מנגנון הניתוב שלה מפיק מסלול אפשרי וישיג ליעד (בהתאם לידע הנוכחי שלה) המהווה מסלול במחיר מינימלי כך שההסתברות לשידור מוצלח מגולמת בו. נבחין כי שיטת ניתוב פרואקטיבית (עיבוד מקדים של הטופולוגיה ובחירת מסלול מראש) איננה מתאימה לבעיה כיוון שטופולוגית הרשת משתנה בזמן, כמו כן, גם שיטת ניתוב ראקטיבית (חישוב בזמן אמת של המסלול) אינה מתאימה ללא שערך הערוץ כיוון שנדרש להתחשב גם בהסתברות השגיאה בתווך בבחירת המסלול.

המאמר מתמקד בניתוב מבוסס הזדמנויות, כאשר לאף צומת אין ידע מקדים אודות טופולוגית הרשת ההתחלתית או ההסתברות לשידור מוצלח. כדי להתגבר על הבעייתיות של השיטה הראקטיבית, כל צומת נדרש לשערך בעצמו את הסתברות הצלחת שידור בלינקים (הקשורות בגרף) המחוברים אליו, אשר מהוות את התווך המשותף לכל אחד משכניו, שיטת השערך המוצעת במאמר הינה שיטת הדגימה של תומפסון, *TS (Thompson Sampling)*, בה משתמשים בפילוג בטא על מנת לשערך ולקרב את הסתברות הצלחת השידור בערוץ. בהתאם למאמר, נסמן את השיטה המתקבלת ע"י שימוש בשערך *TS* עבור הסתברות הצלחת השידור בערוץ לפתרון בעיית ניתוב מבוסס הזדמנויות באופן הבא:  
*TSOR – Thompson Sampling-Based Opportunistic Routing*.

השיטה *TSOR* מגדירה פרוטוקול לפיו, מתווספות מספר הודעות בקרה בנוסף לשידור חבילות המידע, הודעות הבקרה הנ"ל נועדו לעזור לצמתים ברשת לקבל אינדיקציות אודות שינויים ברשת בעזרתן ניתן לעדכן בהתאם גם את מרחקי השכנים מהיעד וגם את הסתברות הצלחת השידור המשוערך בערוץ.

## 2. תוצאות עיקריות

ביצועי השיטה המוצעת  $TSOR$  נמדדים ביחס לביצועי הפתרון האופטימלי ( $OPT$ ), הפתרון האופטימלי מתקבל כאשר הסתברות הצלחת השידור בערוץ קבועה וידועה מראש עבור כל אחד מהצמתים ברשת, במצב זה, כל משתמש יכול להתחשב בהסתברות זו בבחירת המסלול ליעד ותוך מספר סופי של חבילות בקרה למצוא את המסלול האופטימלי עבורו ליעד. שיטה נוספת המוצגת במאמר אשר ביצועיה מושווים לביצועים של  $TSOR$  היא שיטת  $DORL$  ( $Distributed Opportunistic Routing with Learning$ ), הרעיון מאחורי שיטה זו דומה לרעיון באלגוריתם  $TSOR$  והוא שימוש בהודעות בקרה על מנת לחקור את הרשת ולסנכרן בין המשתמשים, שני סוגי החבילות בהן מבוצע שימוש באלגוריתם  $DORL$  הינם:

(1) חבילת גילוי ( $Exploration$ ) – חבילות הגילוי מופצות על גבי הלינקים בהם השימוש מבוצע לעיתים רחוקות במטרה לגלות את הסתברות הצלחת השידור העדכנית על גביהם ולוודא שאם קיים מסלול אפשרי במרחק קצר יותר נוכל לעדכן ולבחור בו, תדירות חבילות הגילוי באלגוריתם תלויות בפרמטר מסוים  $G$ .

(2) חבילת ניצול הזדמנות ( $Exploitation$ ) – חבילות ניצול ההזדמנות מכילות את המידע המיועד עבור צומת היעד והן מועברות תמיד במסלול הקצר ביותר הנוכחי שידוע לצומת.

נבחין כי קיים דמיון יחסית משמעותי בין  $TSOR$  לבין  $DORL$  בשיטת העבודה הכללית ובגישה לפתרון, לכן ההשוואה לאלגוריתם זה במובן הביצועים צפויה לספק מדד יחסי משמעותי ורלוונטי בשלב ניתוח התוצאות.

על מנת להשוות את ביצועי האלגוריתמים נבחרו שלושה מדדים:

i. תמורה ממוצעת – לכל חבילה  $m$  מוגדר ערך תגמול  $Reward_m$  במידה והיא מגיעה ליעדה בהצלחה ואפס אחרת, שליחת הודעה לצומת  $i$  מסוים כרוכה בתשלום מחיר  $Cost_i$ , נסמן את קבוצת המשתמשים בהם עברה ההודעה  $m$  ב-  $\mathcal{L}_m$  ונגדיר את התמורה עבור חבילה  $m$  באופן הבא:

$$Payoff(m) = \left\{ Reward_m \cdot \mathbb{I}_{[m \text{ reached to dest}]} - \sum_{i \in \mathcal{L}_m} Cost_i \right\}$$

התמורה הממוצעת, מהווה מיצוע של כל ערכי התמורות על פני כלל ההודעות ברשת.

ii. חרטה ממוצעת – חרטה עבור הודעה  $m$  מוגדרת להיות ההפרש בין תוחלת התמורה בפתרון האופטימלי בו הסתברות השגיאה ידועה, לבין תוחלת התמורה בפתרון האלגוריתם הנבדק  $A$ :

$$Regret(m) = \{Payoff_{OPT}(m) - Payoff_A(m)\}$$

החרטה הממוצעת, מהווה מיצוע של כל ערכי החרטה על פני כלל ההודעות ברשת.

iii. **מרחק ממוצע מהמקור ליעד** – מטריקת המרחק המשוערכת אשר מגלמת בתוכה גם את המחיר עבור שימוש בקבוצת המשתמשים  $\mathcal{L}_m$  וגם את ההסתברות המשוערכת לשידור מוצלח. נציין כי ע"פ המאמר חישוב המרחק מבוסס אלגוריתם בלמן-פורד, כלומר, מבוסס על מחיר הלינק לשכן והמרחק המדווח ממנו ליעד.

המדדים הנ"ל בין היתר תלויים במספר הדילוגים (*hop count*) בין צומת המקור ליעד ולכן מהווים מטריקה אפשרית להערכת הביצועים של האלגוריתמים מעלה. ע"פ תוצאות האנליזה התיאורטית שבוצעה בעבודת המחקר במאמר הוזכרו החסמים הבאים לאלגוריתם *TSOR* עבור  $N$  צמתים ו- $N^2$  לינקים:

חסמים על מדד החרטה (מרחק מהפתרון האופטימלי):  
חסם תחתון על מדד החרטה המצטברת -  $O(N^3)$   
חסם עליון על מדד החרטה המצטברת -  $O(N^3 \cdot 4^{N_{max}} \cdot \log(M) + N\tau_{max} \cdot \lambda)$   
כאשר  $N_{max}$  הוא מספר השכנים המקסימלי בטופולוגיה,  $M$  הוא מספר החבילות שהופצו,  $\tau_{max}$  הוא הזמן המקסימלי לעדכון ו- $\lambda$  הוא קצב שליחת החבילות.

חסמים על זמן הריצה של האלגוריתם:  
אלגוריתם *TSOR* מחולק לשלושה שלבים, שלב האתחול, שלב ראשון ושלב שני.  
חסם עליון על זמן הריצה בשלב הראשון הינו -  $O(N_{max} \cdot \log(N_{max}))$   
חסם עליון על זמן הריצה בשלב השני הינו -  $O(2^{N_{max}} + N_{max} \cdot \log(N_{max}))$

חסם עליון על זמן הריצה של אלגוריתם *DORL* -  $O(N \cdot G \cdot \log(M))$   
כאשר  $G$  הינו פרמטר המווסת את מספר הודעות הגילוי באלגוריתם ונקבע באופן פרופורציוני למספר הלינקים, בגרף מלא:  $G \propto N^2$ .

לצורך השוואה, נבחין כי בניגוד לאלגוריתם *TSOR* שזמן הריצה שלו אינו תלוי במספר החבילות העוברות ברשת, זמן הריצה של אלגוריתם *DORL* אכן תלוי בפרמטר  $M$ , לפיכך קיים איזשהו ערך סף מסוים  $M'$  עבור כמות החבילות ברשת שהחל ממנו, *TSOR* ישיג זמן ריצה טוב יותר.

### 3. עבודות נוספות בתחום

**אלגוריתם אופטימלי – *index policy***, מהווה בסיס להערכת ביצועים של פתרונות מוצעים. האלגוריתם פותר את בעיית הניתוב במודל רשת שבו יש הסתברויות הצלחת שידור לכל לינק, ושנעשה שימוש ב-*broadcast* ע"מ לקבוע את המסלול. כותבי המאמר מראים שבהינתן הסתברויות ההצלחה לשידור בלינקים, ניתן למצוא את המסלול האופטימלי ע"י תחזוק מרחק של כל משתמש מהיעד, ובחירת השכן הזמין בעל המרחק הקצר ביותר ליעד להיות ה-*next hop*. (ניתוב מבוסס *distance vector*). (מתוך אזכור [6] במאמר)

על בסיס האלגוריתם האופטימלי ישנם מספר עבודות שמגדירות מטריקות שונות ומודדות את ה"מרחק" בצורות שונות:

- **GeRaf (Geographic Random Forwarding)** – המרחקים עבור הניתוב מתבססים על המרחקים הגיאוגרפיים של המשתמשים. כדי למנוע התנגשויות תחילה המקור שולח הודעת *RTS (Request to send)*, כל השכנים הישירים יישלחו הודעת *CTS (Clear to send)*, ה-*next hop* נבחר על פי המיקום היחסי שלו בין השולח ליעד. מדד ההערכה שלהם הוא הספק השידור הנצרך ביחס אל ההשהייה. (מתוך אזכור [7] במאמר).

- **ExOR (Extremely Opportunistic Routing)** – באלגוריתם זה החבילות משודרות ב-*broadcast*, וכל משתמש מקבל עדיפות (*priority*) שונה בהתאם למרחק מהיעד. המרחק מתבסס על כמות השידורים הממוצעת ממשתמש ליעד עד להגעת חבילה (*ETX – expected transmission count*) ולכן הוא פונקציה של הסתברויות הצלחת שידור בלינקים אשר במאמר מניחים שהן ידועות מראש. כדי לבחור את ה-*next hop*, חבילת המידע תכיל את העדיפויות של השכנים של השולח ומתוך קבוצת השכנים שקיבלו את חבילת המידע השכן בעל העדיפות הגבוהה ביותר יעביר אותה הלאה. (מתוך אזכור [8] במאמר).

- **DORCD (Distributed Opportunistic Routing policy with Congestion Diversity)** גם באלגוריתם זה החבילות משודרות ב-*broadcast* כדי לקבוע מי השכנים הישירים. ה-*next hop* נבחר על בסיס עומסים, כלומר השכן בעל העומס הנמוך ביותר ייבחר, במטרה להבטיח השהייה מינימלית. לצורך שיערוך איכות הלינקים, משתמשים בחבילות ייעודיות (*probes*) המגדילות את ה-*overhead*. (מתוך אזכור [11] במאמר).

באלגוריתמים אלו יש צורך בהודעות בקרה נוספות על מנת לקבוע ולתאם מי הוא ה-*next hop*, בעקבות כך יורדת התעבורה ברשת, לכן פותחו אלגוריתמי *OR* מבוססי *NC (Network-coding)*. *NC* היא טכניקה בה מקודדים אצל משתמשי הביניים כמה הודעות המיועדות לאותו משתמש יחד ושולחים את החבילה המקודדת. היעד מפענח את הקידוד ומקבל את החבילות, כך נחסך מספר ההודעות הכולל הנשלחות ברשת. לרוב הקידוד והפיענוח מבוססים על ביצוע פעולת *XOR*.

- **HCOR (High-throughput Coding-aware Opportunistic Routing)** – משתמשים בקידוד חבילות כדי להגדיל את התעבורה. לאחר מציאת השכנים הישירים (*OR*) המשתמשים הנבחרים לביצוע הקידוד הם אלו עם המחיר הנמוך ליעד, כאשר המחיר מתבסס על כמות השידורים הממוצעת ממשתמש ליעד עד להגעת החבילה (*ETX*) בדומה ל-*ExOR*, גם כאן ההנחה היא כי הסתברויות הצלחת השידור בלינקים ידועות מראש. (מתוך אזכור [17] במאמר).



- **CAOR (Coding-aware Opportunistic Routing)** – אלגוריתם מבוסס קידוד שבו הוצעו שיפורים על מנת לוודא כי משתמשים יכולים לבצע פיענוח של חבילות המגיעות אליהם, על ידי שיערוך של החבילות שהגיעו אליהם עד כה. גם באלגוריתם זה המחיר מבוסס על  $ETX$  והסתברויות להצלחת שידור בלינקים ידועות מראש. (מתוך אזכור [18] במאמר).

- **PipelineOR (Pipelined Opportunistic Routing)** – אלגוריתם מבוסס קידוד שמאפשר למקור לשלוח הודעות לפני שקיבל  $ACK$ , ע"י העברת האחריות על העברת החבילה לשכן הבא. מטריקת המחיר של המסלולים היא  $ETX$ , והסתברויות השידור על הלינקים ידועות על ידי ביצוע  $PING$  ( $ICMP$ ) בין המשתמשים באופן מחזורי. מקור שמעוניין לשלוח קובץ מחלק אותו ל- $segments$  (חלקים מההודעה) כאשר מספר החבילות קבוע מראש. החבילות מקודדות ומועברות לשכנים, כאשר השכן עם ה- $ETX$  הנמוך ביותר ליעד הוא המפיץ הבא. אצל כל משתמש הסגמנטים מקודדים מחדש וכך מתאפשר לשלוח לשלוח מספר חבילות מבלי לחכות ל- $ACK$ , וההשהייה הממוצעת ברשת פוחתת. (מתוך אזכור [21] במאמר).

- **UNIV (Universal Opportunistic Routing)** – במאמר מוצע להשתמש בקידוד על חבילות ה- $feedback$ , וכך להקטין את העומס הנוצר ברשת מחבילות אלו, תוך ביצוע שיערוך בזמן-אמת של הסתברויות השידור בלינקים. המטריקה מתבססת על מתן "קרדיט" למשתמשים אשר הצליחו לנתב את החבילה ליעד ושימוש באלו עם הקרדיט הגבוה ביותר. אלגוריתם אפקטיבי לייצוב התעבורה, אך לא אופטימלי ליעדים אחרים כמו מזעור ההשהייה ברשת. (מתוך אזכור [26] במאמר).

- **AdaptOR (Adaptive Opportunistic Routing)** – משתמש באלגוריתם  $\varepsilon - Greedy$  לבחירת ה- $next\ hop$ . כלומר, בהסתברות  $\varepsilon$  נבחר שכן רנדומלית מקבוצת השכנים הישיגים (נקבעת לפי  $ACK$  בדומה לאלגוריתם במאמר שלנו), ובהסתברות  $1 - \varepsilon$  נבחר השכן הישיג בעל המרחק הקצר ליעד, כאשר מטריקת המרחק יכולה להיות הספק שידור, השהייה או  $hop\ count$ . הפרמטר  $\varepsilon$  מתעדכן תוך כדי הריצה והוא פונקציה של מספר השכנים הישיגים. האלגוריתם תוכנן להתמודד עם הסתברויות הצלחה של שידור שאינן ידועות, ומצליח להתכנס לתוצאות של האלגוריתם האופטימלי לאחר מספר מספיק של הודעות שנשלחו ברשת. בעצם בחירת שכן רנדומלי (בהסתברות  $\varepsilon$ ) מהווה חקר של הרשת ומאפשרת למצוא מסלולים חדשים שיכולים להיות טובים יותר מהמסלול אליו האלגוריתם התכנס באותה נקודת זמן (חרטה ממוצעת -  $O(N^4 4^{N_{max}} \log M)$ ). (מתוך אזכור [28] במאמר).

האלגוריתם  $TSOR$  מתכנס גם הוא לתוצאות אופטימליות ועושה זאת בסדר גודל של  $N$  פחות מאשר  $AdaptOR$ , כלומר הוא יותר ישים ברשתות גדולות (מתכנס לחרטה ממוצעת בסדר גודל של  $O(N^3 4^{N_{max}} \log M)$ ).

## מודל המערכת וניסוח הבעיה

מודל המערכת הינו רשת אד-הוק אלחוטית בעלת  $N$  משתמשים:  $\mathcal{N} = \{1, 2, \dots, N\}$ . לשם פשטות כותבי המאמר מתמקדים בניתוח מקרה של זרימה בודדת, מקור ויעד יחידים. נסמן את צומת המקור ב-1 ואת צומת היעד ב- $N$ , כאשר בלי הגבלת הכלליות, הם יכולים להיות כל אחד ממשתמשי הרשת, המטרה הינה לנתב רצף של חבילות ממוספרות  $m = 1, \dots, M$  מהמקור ליעד, כאשר הניתוב נדרש להתבצע ללא ידע מקדים אודות טופולוגיית הרשת ופרמטרי הלינקים (התווך המשותף בין הצמתים).

כל אחד ממשתמשי הרשת  $n \in \mathcal{N}$  משדר את ההודעה בתצורת *broadcast*, כלומר, מפיץ אותה לכל שכניו, כאשר השידור בין משתמשים ממודל בצורה הסתברותית (השידור יכול להיכשל בהסתברות מסוימת). ברשת אלחוטית ההסתברות תלויה במרכיבים סביבתיים כגון: מרחק בין הצומת המשדרת לצומת הקולטת, תנועת אובייקטים במרחב, החזרות, השתקפויות, בליעות ורעש משידורים מקבילים באותו תא שטח, לפיכך, זמינות השכנים לקליטת הודעות עלולה להשתנות בזמן. נגדיר  $\mathcal{N}(n) \subseteq \mathcal{N}$  להיות קבוצת השכנים הפוטנציאליים של משתמש  $n$ , הסתברות הצלחת השידור על גבי לינק אלחוטי עבור המשתמש ה- $n$  מתקבלת באופן הבא:

$$P_n := \{Pr(S|n) : \forall S \subseteq \mathcal{N}(n)\}$$

כאשר  $Pr(S|n)$  היא ההסתברות שֶרֶק המשתמשים בקבוצה  $S$  קיבלו את השידור של  $n$ , מכאן שאוסף הקבוצות  $S$  הינן זרות בזוגות, לכן המאורעות הינם מאורעות זרים ומתקיים:

$$\sum_{S \subseteq \mathcal{N}(n)} Pr(S|n) = 1$$

נשים לב כי אם בנוסף מתקיים שהלינקים בין כל שני משתמשים הם בלתי תלויים אחד בשני, אז ניתן לפשט את פונקציית ההסתברות ולהגדיר אותה עבור כל זוג באופן הבא:

$$P_n := \{Pr(n'|n) : \forall n' \in \mathcal{N}(n)\}$$

החבילות נשלחות ללא עותקים כפולים, כלומר בכל נקודת זמן רק משתמש אחד אחראי על שליחת חבילת המידע, חבילת מידע נעצרת כאשר הגיעה ליעדה או לחילופין כאשר ערך ה- $TTL$  שלה התאפס.

*TTL (Time-To-Live)* - הינו שדה בחבילה אשר מגדיר את מספר הקפיצות המקסימלי שהחבילה תעבור ברשת, בכל מעבר בין משתמשים ערך ה- $TTL$  יורד באחד, וכאשר מגיע ל-0 החבילה נזרקת. מבחינת המודל, כאשר חבילה מגיעה ליעדה בהצלחה, מתקבל תגמול (*Reward*) שמוגדר ע"י הפרמטר  $R$ , אם החבילה לא הגיעה ליעדה ונזרקה בצומת ביניים בדרך, התגמול הוא 0. באופן מפורש, התגמול של חבילה  $m$  מוגדר באופן הבא:

$$r_m = R \cdot \mathbb{I}_{[m \text{ reached to dest}]}$$

לכל אחד ממשתמשי הרשת משויך מחיר  $c_n$  אשר תפקידו לכמת ערך מספרי המייצג את העלות עבור פעולת שידור המידע ליעד, בדרך כלל נהוג להתאים אותו למידול אחד או יותר מהפרמטרים הבאים: הספק השידור, השהייה מצטברת, מרחק כולל ועוד.

במאמר מניחים כי  $c_n$  בלתי תלוי בחבילות המידע הנשלחות ומגדירים אותו להיות בצורה אחידה להיות ה-*hop count*, כלומר בפועל:  $c_n = 1, \forall n \in \mathcal{N}$ . ההצדקה להגדרה זו נשענת על כך שאם חבילות המידע בעלות גודל קבוע אז כל פונקציית מחיר אשר מבוססת על הספק או השהייה בשידור תניב גם כן ערך קבוע.

נסמן ב- $\mathcal{L}_m$  את קבוצת כל המשתמשים במסלול הניתוב של החבילה  $m$ , המחיר הכולל של ניתוב החבילה הינו:  $\sum_{n \in \mathcal{L}_m} c_n$ , נקבל שהתמורה, עבור ניתוב חבילה  $m$  הוא:

$$r_m - \sum_{n \in \mathcal{L}_m} c_n$$

כאשר במודל שבו המחיר הוא 1 לכל משתמש, מתקבל שמחיר המסלול הוא ערך ה-*hop count* של החבילה והתגמול הוא כפי שהוגדר קודם. לפיכך, התמורה הממוצעת עבור ניתוב של  $M$  החבילות ממשתמש 1 למשתמש  $N$  מוגדרת באופן הבא:

$$\mathcal{J}_m := \mathbb{E} \left[ \sum_{m=1}^M \left[ r_m - \sum_{n \in \mathcal{L}_m} c_n \right] \right]$$

כלומר, מיצוע התמורה המצטברת עבור כל הודעת מידע שנשלחה.

המטרה הינה לפתח אלגוריתם ניתוב אשר ימקסם את התמורה הממוצעת של כל אוסף ההודעות  $M$ , תחת ההנחה שהסתברויות הלינקים  $P_n$  ידועות מראש, פתרון אופטימלי היה מבצע *index policy* (אזכור [6] במאמר), כלומר בוחר את הניתוב על בסיס ה-*next hop*, השכן בעל המרחק המינימלי מהיעד, בחירה זו מתבססת על משוואות בלמן-פורד. נסמן את המרחק האופטימלי של כל משתמש ברשת מצומת היעד ב-  $V^*(n)$ .

חישוב המרחקים עבור המקרה האופטימלי מתקבל באופן הבא:

$$V^*(N) = -R$$

$$V^*(n) = \min \left\{ 0, \left\{ c_n + \sum_{S \subseteq \mathcal{N}(n)} Pr(S|n) \left( \min_{n' \in S} V^*(n') \right) \right\} \right\}, \forall n \in \mathcal{N}, n \neq N$$

עבור צומת היעד, המרחק שלו מעצמו מוגדר להיות המרחק המינימלי ברשת  $(-R)$ , עבור כל שאר המשתמשים חישוב המרחק מתקבל ע"י סכימה של מחיר שליחת החבילה לשכן  $(c_n)$  עם המרחק הממוצע של השכנים שלו -  $\sum_{S \subseteq \mathcal{N}(n)} Pr(S|n) \left( \min_{n' \in S} V^*(n') \right)$ . המרחק  $V^*(n)$  הוא התמורה הממוצעת שיכולה להתקבל משליחת הודעה מ- $n$  לכיוון היעד תוך שימוש ב-*index policy* (בהיפוך סימן). השכן שייבחר בתור ה-*next hop* יהיה זה בעל המרחק הקצר ביותר ליעד.

חשוב לציין ש-  $V^*(n)$  מוגדר להיות המינימום בין 0 לבין המרחק המתקבל ע"י השכנים וזאת בכדי למנוע מצב שבו התמורה משליחת חבילת מידע הינה שלילית, כלומר, עלות שליחת החבילה עולה על התגמול ממנה. במקרים אלו נעדיף שלא להשתמש במסלול ולוותר על שליחת ההודעה.

מהגדרת המרחקים הללו נובע שככל שהמרחק של משתמש מסוים מהיעד נמוך יותר כך יש לו יכולת יותר טובה להעביר את החבילה ליעד, לכן נעדיף להשתמש בו.

בגרסה הריכוזית ישנו בקר אשר תחילה מחשב את  $V^*(n)$  עבור כל  $n$ , לאחר מכן השולח מפיץ את החבילה בשידור *broadcast* על מנת לקבוע את קבוצת השכנים הישיגים שלו -  $S$ , ממנה נבחר בשבן בעל המרחק המינימלי בתור ה-*next hop*, כך התהליך יחזור על עצמו עד שהחבילה תגיע ליעדה או לחילופין תיזרק.

באופן דומה, ניתן לבצע *index policy* בצורה מבוזרת ע"י שליחת הודעות בקרה בין השכנים, אשר יעזרו לכל אחד ואחד מהם לאסוף מידע אודות טופולוגית הרשת ואודות שכניהם כחלופה למידע שהיה מתקבל ע"י הבקר בגרסה הריכוזית.

המודל במאמר מתמקד בהסתברויות לינק לא ידועות ולכן פתרון אפקטיבי יהיה כזה שיהיה מסוגל ללמוד את הסתברויות הלינקים, תוך מקסום התמורה. כותבי המאמר משתמשים במדד החרטה (*Regret*) על מנת למדוד את הפיחות בביצועים בין פתרון מוצע כלשהו לבין הפתרון האופטימלי, נסמן את התמורה הממוצעת עבור האלגוריתם האופטימלי ב-  $J_m^*$  (מתקבל בשיטת *index policy*), בנוסף נסמן את התמורה הממוצעת של הפתרון המוצע ב-  $J_m$ , לכן החרטה הממוצעת עבור  $M$  חבילות המידע הינה:

$$\mathcal{R}_M = J_m^* - J_m$$

כאשר השאיפה של כותבי המאמר הייתה לעצב אלגוריתם אשר ימזער את החרטה הממוצעת תוך שמירה על סיבוכיות זמן-ריצה נמוכה.

#### קשיים בפיתוח האלגוריתם:

קשיים עיקריים בתכנון אלגוריתם העונה על הבעיה המוצגת הינם:

- מציאת איזון בין חקר הלינקים (כאמור ההסתברויות אינן ידועות מראש) לבין מקסום התמורה בצורה דינאמית, כלומר על האלגוריתם להיות מסוגל לשערך את טיב הערוץ בצורה מספיק טובה.
- ברשתות אלחוטיות מסוג אד-הוק אין ישות מתאמת (בקר מרכזי) ולכן על האלגוריתם לעבוד בצורה מבוזרת (התמודדות עם בעיית הסנכרון).
- על האלגוריתם לספק חסם עליון מספיק הדוק על מדד החרטה הממוצעת שכן נרצה שככל שמספר חבילות המידע ברשת יגדל כך ישאף ערך החרטה הממוצעת ל-0, במילים אחרות, נרצה אלגוריתם אשר מספק פתרון אפשרי וישים כך שהמרחק בין הפתרון הגרוע ביותר המתקבל מהפעלתו לבין הפתרון האופטימלי בעזרת ישות מתאמת חסום ע"י ערך הדוק ככל הניתן.

## פתרונות לבעיה

### 1. האלגוריתם המוצע

כותבי המאמר מציגים כפתרון את האלגוריתם:

#### *TSOR: Thompson Sampling-Based Opportunistic Routing*

תכנון אלגוריתם *TSOR* הוא בהשראת אלגוריתמי בלמן-פורד מבוזרים, ההבדל העיקרי הינו שאלגוריתמי בלמן-פורד מבוזרים מספקים מענה עבור המקרים בהם מחירי הלינקים ידועים מראש בעוד ש-*TSOR* תוכנן לתת מענה על המודל ההסתברותי בו לכל לינק ישנה הסתברות הצלחה לשידור שאיננה ידועה ולכל צומת משויך מחיר קבוע.

*Thompson Sampling (TS)* הינו כלי לשערוך פרמטרים של התפלגויות לא ידועות על בסיס כלל בייס (*Bayes' theorem*), הכלי תחילה מניח התפלגות אפריורית מסוימת ואז מעדכן אותה עם דגימות מההתפלגות הלא ידועה ולאחר מספיק עדכונים (חיוויים) צפוי להניב ריאליזציות אשר מתואמות לפרמטרים האמיתיים של ההסתברות האפוסטריורית.

נשים לב שבמודל זה אנו נדרשים לשערך ממוצעים של התפלגויות ברנולי, ניתן להסתכל על הסתברויות הלינקים כמתפלגות ברנולי - ישנה הסתברות להצלחת שידור והמשלימה לה הינה הסתברות השגיאה, הממוצע של התפלגות ברנולי הוא בדיוק ההסתברות להצלחה וזה הפרמטר אותו אנו שואפים לשערך -  $P_n$ .

שיערוך של ממוצע פילוג ברנולי ע"י *TS* מתבצע באופן הבא: מניחים התפלגות אפריורית *Beta* על הממוצע - הבחירה בהתפלגות *Beta* היא מכיוון שהיא מהווה *conjugate prior* להתפלגות ברנולי, כלומר, אם פונקציית ה-*likelihood* הינה ברנולי וההתפלגות האפריורית היא *Beta*, אז גם ההתפלגות האפוסטריורית היא *Beta*, באופן מפורש, עבור התפלגות  $Beta(\alpha, \beta)$ , ניתן לעדכן את הפרמטרים בצורה הבאה:  $\alpha + X$ ,  $\beta + 1 - X$ . כאשר  $X$  היא דגימה מתוך התפלגות ברנולי הלא ידועה.

לצורך המחשה בלבד, ניקח מקרה פרטי בו אין הסתברויות שגיאה על הלינקים, כלומר ההסתברות להצלחת שידור הינה 1, במקרה זה כל דגימה  $X$  תקבל ערך 1 בהסתברות 1, מכאן שהעדכונים יניבו ערכי  $\alpha$  עולים, כמו כן הממוצע של התפלגות *Beta* הינו  $\frac{\alpha}{\alpha+\beta}$ , לכן נקבל לאחר מספר מסוים של עדכונים שהממוצע שואף ל-1 ובהתאם, הריאליזציות מההתפלגות הנ"ל יהיו בקירוב 1, וזהו אכן הממוצע עבור התפלגות ברנולי בערוץ ללא שגיאות.

כותבי המאמר משתמשים ב-*Thompson Sampling* לשערוך הסתברויות ההצלחה של שידור בלינקים באופן הבא:

לכל משתמש  $n$  ולכל קבוצת שכנים ישיגים  $S \subseteq \mathcal{N}(n)$  מניחים הסתברות אפריורית  $Beta(\alpha_{n,S}, \beta_{n,S})$  לכל הסתברות לינק  $Pr(S/n)$  בהתאמה, כאשר הערכים מאותחלים כך:  $\alpha_{n,S} = \beta_{n,S} = 1$ , נסמן ב- $A$  את קבוצת השכנים הישיגים בנקודת זמן ספציפית (הקבוצה משתנה בזמן), עדכון הפרמטרים יתבצע כך:

$$\begin{cases} \alpha_{n,S} = \alpha_{n,S} + 1 & \text{if } S = A \\ \beta_{n,S} = \beta_{n,S} + 1 & \text{otherwise} \end{cases}$$

כלומר, עבור קבוצת שכנים ישיגים נקדם את ערך ה- $\alpha$  המתאים, ועבור כל השאר נקדם את  $\beta$ . בעת חישוב המרחק של צומת מסוים  $n$ , תחילה הוא דוגם מתוך ההסתברות האפוסטריורית:  $\theta(S|n) \sim \text{Beta}(\alpha_{n,S}, \beta_{n,S})$ , לאחר מספיק עדכונים הדגימות צפויות להתכנס למוצע התפלגות ברנולי המתאימה להסתברות ההצלחה לשידור בלינק, בהינתן הדגימה חישוב המרחק הוא בדומה לחישוב של האלגוריתם האופטימלי:

$$V(N) = -R$$

$$V(n) = \min \left\{ 0, \left\{ c_n + \sum_{S \subseteq \mathcal{N}(n)} \theta(S|n) \left( \min_{n' \in S} V(n') \right) \right\} \right\}, \forall n \in \mathcal{N}, n \neq N$$

מבאן נוכל להסיק שככל שההסתברות האפוסטריורית תהיה מדויקת יותר, כך נתקרב לפתרון האופטימלי.

#### שלבי האלגוריתם:

כותבי המאמר מחלקים את אלגוריתם *TSOR* לשלושה חלקים:

- שלב אתחול
- שלב ראשון – הפצת החבילה מהמקור ליעד
- שלב שני – שליחת והפצת הודעות *NACK/ACK*, ועדכון מרחקים.

כל צומת שומר אצלו את הפרמטרים של התפלגות ה-*Beta*, המרחק שלו מהיעד  $V(n)$ , ומרחקי השכנים שלו מהיעד  $V(n, n')$ .

#### סימונים:

סימון	משמעות
$\mathcal{N}; N$	מספר המארחים (צמתים); קבוצת הצמתים
$\mathcal{N}(n)$	קבוצת השכנים של צומת $n$
$\mathcal{M}; M$	מספר החבילות; קבוצת החבילות
$Pr(S n)$	ההסתברות שרק הצמתים ב- $S$ יקבלו שידור broadcast מהצומת $n$ בהצלחה
$P_n$	התפלגות ההסתברות על הלינקים של צומת $n$ : $P_n := \{Pr(S n): \forall S \subseteq \mathcal{N}(n)\}$
$\text{Beta}(\alpha_{n,S}, \beta_{n,S})$	התפלגות <i>Beta</i> אפריורית על $Pr(S n)$
$\Theta_n$	התפלגות ההסתברות המשוערכת על הלינקים של צומת $n$ : $\Theta_n := \{\theta(S n): \forall S \subseteq \mathcal{N}(n)\}$
$R$	התגמול על חבילה שהגיעה ליעד
$r_m$	התגמול על חבילה $m$ אם הגיעה ליעד, $0$ אחרת
$c_n$	מחיר שליחת חבילה בצומת $n$
$t_m$	ערך ה- <i>TTL</i> הנוכחי של חבילה $m$
$\mathcal{L}_m$	קבוצת הצמתים במסלול הניתוב של חבילה $m$
$\mathcal{R}_M$	החרטה המצטברת על $M$ חבילות
$V^*(n)$	המרחק האופטימלי של צומת $n$ מהיעד, כפי שחושב ע"י שימוש ב $P_n$
$V(n)$	המרחק המשוערך של צומת $n$ מהיעד, כפי שחושב ע"י שימוש ב $\Theta_n$
$V(n, n')$	המרחק של השכן של $n'$ , כפי ששמור אצל $n$

### הודעות בקרה באלגוריתם ותפקידן:

- LACK (Link Ack) – הודעה שתפקידה ליידע את השולח כי ההודעה שלו התקבלה בהצלחה, משמשת לעדכון פרמטרי התפלגות ה-Beta ועדכון קבוצת השכנים הישיגים באותה נקודת זמן.
- ACK/NACK – תפקידן לעדכן כי ההודעה הגיעה/לא הגיעה ליעד, מתחילות את תהליך עדכון המרחקים ברשת.
- LCFM (Local Confirmation Message) – תפקידה הינו להכריז על זהות המשתמש עבור ה-*next hop*.

### פירוט שלבי האלגוריתם:

(0) אתחול:

#### **Algorithm 1** Algorithm Initialization for Node $n$

```
1:  $\alpha_{n,S} = \beta_{n,S} = 1, \forall S \subseteq \mathcal{N}(n);$ 
2: if  $n \neq N$  then
3:    $V(n) = 0;$ 
4: else
5:    $V(n) = -R;$ 
6: end if
7: for all  $n' \in \mathcal{N}(n)$  do
8:   if  $n' \neq N$  then
9:      $V(n, n') = 0;$ 
10:  else
11:     $V(n, n') = -R;$ 
12:  end if
13: end for
```

בשלב האתחול מבצעים השמה של ערכים התחלתיים לכל המשתמשים, כל ערכי ה- $\alpha$  וה- $\beta$  מאותחלים להיות 1, בנוסף לכל צומת שאיננה צומת היעד המרחק מאותחל ל-0, עבור צומת היעד המרחק מוגדר להיות  $-R$  (שורות 1-5).  
הסיבה שהיעד מקבל מרחק  $-R$  נובעת מכך שהמרחקים מהווים את התמורה הממוצעת שמשתמש  $n$  מרוויח משליחת החבילה ליעד בסימן הפוך, לכן אם היעד יישלח לעצמו חבילה הוא בהכרח ירוויח תמורה של  $R$  ולכן התמורה הממוצעת היא  $R$  והמרחק המתאים הינו  $-R$ , נבחין כי זהו גם המרחק הקטן ביותר האפשרי באלגוריתם הנ"ל.  
בנוסף מאתחלים לכל משתמש גם את טבלת המרחקים לפי השכנים (שורות 7-12), לכל שכן שאינו היעד מאתחלים ל-0, אם השכן הוא בן היעד, נעדכן את המרחק שלו להיות  $-R$ .

(1) שלב ראשון - הפצת חבילות מידע:

**Algorithm 2** Part 1 of TSOR for Node  $n$

```

1: if Receive packet  $m$  then
2:   Respond a LACK message with the estimated distance  $V(n')$ ;
3:   if Receive an LCFM from node  $n' \in \mathcal{N}(n)$  then
4:     if  $n$  is the next hop appointed by  $n'$  then
5:       Broadcast packet  $m$  locally to its neighbors;
6:       if Receive LACKs then
7:         Observe the available neighbors  $A \subseteq \mathcal{N}(n)$  and
           their estimated distance values  $V(n'), \forall n' \in A$ ;
8:         Update the beta distributions based on (5);
9:         Broadcast an LCFM with node index  $I := \arg \min_{n' \in A} V(n')$ ;
10:      end if
11:    end if
12:  end if
13:  Drop packet  $m$ ;
14: end if

```

תחילת ההפצה מתרחשת כאשר המקור מפיץ את חבילת המידע בתצורת *broadcast*, כל שכן שמקבל את החבילה מגיב עם הודעת LACK בה מצוין גם המרחק שלו מהיעד ( $V(n')$ ) (שורות 1-2).

צומת שקיבל חבילות LACK, קובע את קבוצת השכנים הישיגים ממנו  $A$  ומעדכן את ערכי התפלגות ה-Beta, העדכון מתבצע באופן הבא:

$$\begin{cases} \alpha_{n,S} = \alpha_{n,S} + 1 & \text{if } S = A \\ \beta_{n,S} = \beta_{n,S} + 1 & \text{otherwise} \end{cases}$$

הצומת עובר על כל המרחקים שהשכנים פרסמו לו ובוחר שכן מתוך הקבוצה  $A$  שמרחקו מהיעד מינימלי, כלומר בוחר את:  $\arg \min_{n' \in A} V(n')$  ומפיץ הודעת LCFM בה מצוין מיהו השכן הנבחר, (שורות 6-9).

כאשר השכנים מקבלים את הודעת ה-LCFM הם בודקים אם הם נבחרו, השכן שנבחר יפיץ את ההודעה הלאה ב-*broadcast* והתהליך ימשיך באופן זהה ממנו עד אשר ההודעה תגיע ליעדה או תיפול (עקב התאפסות TTL או היעדר שכנים ישיגים). (שורות 3-5). שכן שלא נבחר לא ימשיך להפיץ את החבילה וכך מובטח שעותק אחד בלבד שלה מופץ בכל רגע נתון.

ניתן לסכם בקצרה את תהליך ההפצה כך: תחילה המקור מפיץ את ההודעה לשכניו, כל מי שקיבל את ההודעה מחזיר LACK. המקור בוחר מבין כל מי שהחזיר LACK את השכן עם המרחק המינימלי ליעד ומפיץ הודעת LCFM בה הוא בוחר בו כ-*next hop*. המשתמש שיקבל את ה-LCFM ונבחר ימשיך את התהליך, כך עד שההודעה תיעצר ביעד או בצומת ביניים אחר.



(2) שלב שני - שליחה והפצה של הודעות ACK/NACK + עדכון מרחקים:

**Algorithm 3** Part 2 of TSOR for Node  $n$

```

1: if Packet  $m$  terminates at  $n$  then
2:   Send an ACK if  $n = N$  or a NACK otherwise;
3: end if
4: if Receive an ACK or NACK from node  $n'$  then
5:   Respond a LACK to node  $n'$ ;
6:   Observe the value of node  $n'$ ,  $V(n')$ ;
7:   if  $V(n') = V(n, n')$  then
8:     Continue;
9:   end if
10:  if  $n \neq N$  then
11:    Update the maintained estimated distance value of node
         $n'$ ,  $V(n, n')$ ;
12:    Sample  $\theta(S|n) \sim \text{Beta}(\alpha_{n,S}, \beta_{n,S}), \forall S \subseteq \mathcal{N}(n)$ ;
13:    Update  $V(n)$  with  $V(n, n'), \forall n' \in \mathcal{N}(n)$  based on (6);
14:  end if
15:  Broadcast the ACK or NACK with  $V(n)$  locally;
16:  if Receive LACKs then
17:    Observe the available neighbors  $A \subseteq \mathcal{N}(n)$ ;
18:    Update the beta distributions based on (5);
19:  end if
20: end if

```

הודעה  $m$  תיעצר באחת משתי אפשרויות, הראשונה היא כאשר ההודעה תגיע ליעדה ולאחר מכן תישלח הודעת ACK, השנייה הינה כאשר היא נזרקת באחת מצמתי הביניים, במקרה זה תישלח הודעת NACK. (שורות 1-2).  
הודעות אלו מופצות ב-broadcast והטיפול בהן זהה, הן מכילות בתוכן את המרחק של המפיץ (הצומת בה נעצרה ההודעה) מהיעד -  $V(n')$ .  
משתמש שמקבל ACK/NACK, תחילה מגיב עם LACK על מנת שהשכן שלו יוכל לעדכן את פרמטרי התפלגות Beta של הלינק, בדומה לעדכון בשלב הקודם (שורות 16-18).  
לאחר שליחת הודעת ה-LACK, המשתמש בוחן את המרחק  $V(n')$ , אם מרחק זה שונה מהמרחק המעודכן אצלו  $V(n, n')$ , הוא יבצע עדכון פרמטרים (נשים לב שהיעד לא מבצע עדכוני פרמטרים כלל) – תחילה מעדכנים את המרחק מהשכן:  $V(n, n') = V(n')$ , לאחר מכן דוגמים מהתפלגות ה-Beta הנוכחית ומחשבים את המרחק  $V(n)$  באופן הבא:

$$V(n) = \min \left\{ 0, \left\{ c_n + \sum_{S \subseteq \mathcal{N}(n)} \theta(S|n) \left( \min_{n' \in S} V(n') \right) \right\} \right\}$$

(שורות 7-14).

לבסוף, הצומת שקיבל ACK/NACK, מפיץ אותו הלאה ב-broadcast לכל שכניו, וכך תהליך העדכון נמשך עד אשר ערך ה-TTL מתאפס.

## 2. ניתוח ביצועים

לצורך הערכת ביצועי האלגוריתם  $TSOR$  נעשה שימוש במדד החרטה ( $Regret$ ) - שמייצג את ההפסד בשליחת החבילות בשימוש באלגוריתם לעומת האופטימלי. תחילה כותבי המאמר מחשבים תיאורטית את ה- $LRB$ -lower regret bound שהוא חסם תחתון לבעיית ה- $OR$  המוצגת במודל. החסם המחושב הוא נכון לגבי כל הטופולוגיות והלינקים, מכאן שהוא מחושב עבור המקרה הגרוע ביותר. לאחר מכן תבוצע השוואה של החסם העליון של החרטה ( $URB$  - upper regret bound) של אלגוריתם  $TSOR$  לעומת ה- $LRB$ . אם הם קרובים סימן שהחרטה של  $TSOR$  מתכנסת לאופטימלי.

חישוב החסם התחתון ( $LRB$ ) - לצורך חישוב החסם מבצעים המרה של בעיית ה- $OR$  לבעיית  $CMAB$  - combinatorial multi-armed bandit (אזכור [32]). בעיית  $MAB$  מתוך אזכור [31] היא בעיה בה סוכן מרים ידית אחת מתוך סט ידיות בכל נקודת זמן. כל ידית שמורמת מניבה תגמול מהתפלגות לא ידועה. מטרת הסוכן היא למקסם את התגמולים לאורך זמן סופי. בעיית ה- $CMAB$  היא הרחבה של  $MAB$  שבה ניתן להרים מספר ידיות בכל נקודת זמן. בעיות אלו בעצם מייצגות את ה- $tradeoff$  בין חקר הרשת לניצול הרשת. לאחר ההמרה, החסם התחתון לבעיית ה- $CMAB$  יהיה גם החסם התחתון של  $OR$ . כותבי המאמר מציינים כי הדוגמא המוצגת באיור הבא ידועה כיום כדוגמא שמניבה את ה- $LRB$  הכי נמוך, כלומר היא ה- $worst case$  שנמצא עד היום.

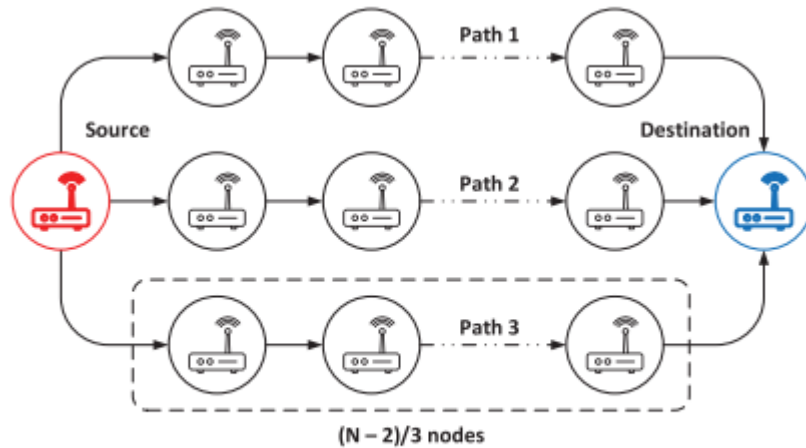


Fig. 1. The three-path wireless network.

בדוגמא ניתן לראות מקור ויעד, כאשר המקור רוצה לשדר חבילות ליעד על פי המודל שהצגנו. קיימים שלושה מסלולים, כל אחד מהם מכיל  $\frac{N-2}{3}$  משתמשי ביניים ו- $\frac{N+1}{3}$  לינקים (כולל לינקי המקור והיעד). ניתן להניח בלי הגבלת הכלליות כי מספר צמתי הביניים והלינקים הינו שלם. לכל שני משתמשים  $n$  ו- $n'$ , נסמן ב- $Pr(n'|n)$  את הסתברות הצלחת השידור בלינק, בשליחה מ- $n$  ל- $n'$ . כל ההסתברויות בלתי תלויות ואינן ידועות מראש. לכל משתמש מחיר  $c_n$  לשליחת חבילה. המרה לבעיית  $CMAB$ : כל לינק מייצג זרוע. מכיוון שהסתברות ההצלחה על כל לינק קטנה מ-1, כל משתמש צריך לשלוח את החבילה לפחות פעם אחת, במחיר  $c_n$ . התגמול המשוך לכל זרוע הוא ההופכי של מכפלת מספר השידורים- $k$  והמחיר:

$$r(n, n') = \frac{1}{k \cdot c_n}$$

מכאן שהתמורה הממוצעת ניתנת לחישוב באופן הבא:

$$\bar{r}(n, n') := \sum_{k=1}^{\infty} \frac{1}{k \cdot c_n} (1 - \Pr(n'|n))^{k-1} \Pr(n'|n)$$

שזהו ממוצע של מחיר שידור של  $k-1$  שידורים לא מוצלחים ועוד שידור מוצלח.

מטרת בעיית ה- $CMAB$  היא לבחור אחד מהמסלולים – בעצם  $\frac{N+1}{3}$  זרועות, כך שהתגמול הממוצע המצטבר של  $M$  חבילות יהיה מקסימלי. מקסום התגמול שקול למינימיזציה של מחיר המסלול כפי שהוצג במודל.

עבור המקרה בו התמורות הממוצעות של הלינקים הן:

$$\bar{r}(n, n') = \begin{cases} 0.5 & \text{הלינק במסלול 1} \\ 0.5 - \frac{3}{(N+1)^2} & \text{אחרת} \end{cases}$$

מתקבל שלכל  $N > 2$  בדוגמת שלושת המסלולים, החסם התחתון על החרטה הינו:

$$LRB = \frac{(N+1)^3}{18} \log M$$

ההוכחה מתבססת על ההמרה לבעיית  $CMAB$  והעובדה שהחסם התחתון לבעיה זו ידוע (מתוך אזכור [39])

$$\lim_{M \rightarrow \infty} \frac{\mathcal{R}_M}{\log M} \geq \frac{\left((N+1) - \frac{N+1}{3}\right) \cdot \frac{N+1}{3}}{4 \frac{1}{N+1}} = \frac{(N+1)^3}{18}$$

חישוב חסם עליון של אלגוריתם  $TSOR$ :

נסמן ב- $\lambda$  את קצב שליחת ההודעות מהמקור ו- $N_{max} := \max_{n \in \mathcal{N}} |\mathcal{N}(n)|$  הוא מספר השכנים

המקסימלי האפשרי למשתמש  $n$ .

הנחה 1: ה- $TTL$  של ה- $ACK$  ו- $NACK$  מספיק גדול כך שכל משתמש ברשת יקבל אותם לפחות פעם אחת. ההנחה היא לטובת הניתוח, נשים לב שברשתות קטנות זה מתרחש בהסתברות גבוהה, וברשתות גדולות יותר צריך  $TTL$  גדול כדי שיתקיים.

למה 1: יהי

$$\Delta \in \left(0, \min_{n \in \mathcal{N}} \left\{ \min_{n' \in \mathcal{N}(n)} |V^*(n) - V^*(n')| \right\}\right)$$

אם מתקיים:

$$\mathcal{K} := \left\{ |V^*(n) - V(n)| \leq \frac{\Delta}{2}, \forall n \in \mathcal{N} \right\}$$

אז החרטה היא 0.

הוכחת הלמה בנספח [1].

למה 1 נותנת תנאי מספיק לכך שלא תהיה חרטה, מכאן שהאירוע המשלים שבו יש חרטה הינו:

$$\bar{\mathcal{K}} := \left\{ |V^*(n) - V(n)| \geq \frac{\Delta}{2}, \exists n \in \mathcal{N} \right\}$$

מכיוון שהמרחקים מחושבים בצורה מבוזרת, נדרש זמן נוסף עד שהם מתעדכנים. נסמן  $\tau_{max}$  את זמן העדכון המקסימלי, כך שבאשר מעדכנים את החבילה ה  $(m + \tau_{max} \cdot \lambda)$ , כל המשתמשים עדכנו את המרחקים בהתבסס על המידע הנלמד מ- $m$  החבילות הראשונות.

למה 2:

יהי

$$\alpha := \sum_{n=1}^N \frac{R}{c_n}, \quad B := \frac{8\alpha^2 4^{N_{max}} \log M}{\Delta^2} + \tau_{max} \cdot \lambda$$

לכל

$$M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot 2^{N_{max}}}} \right\rceil, \tau_{max} \geq 0, N_{max} > 0, \lambda > 0$$

מתקיים שלאחר שליחת  $B$  חבילות, ההסתברות למאורע  $\bar{\mathcal{K}}$  (שיש חרטה) קטנה מ  $\frac{4N \cdot 2^{N_{max}}}{M}$ . הוכחת הלמה בנספח [2].

מלמה 2 נובע כי אם נבטיח שכל משתמש יקבל  $ACK$  או  $NACK$  לפחות  $\frac{8\alpha^2 4^{N_{max}} \log M}{\Delta^2}$  פעמים, ההסתברות שיש חרטה נמוכה מאוד. ניתן להסיק מכך כי ניתן להפחית את כמות ה- $ACK$  וה- $NACK$  הנשלחים באלגוריתם, שגורמים לסיבוכיות חישוב גבוהה.

בהינתן למות 1 ו-2 ניתן לחשב את החרטה של  $TSOR$  באופן הבא:

טענה 1:

לכל

$$M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot 2^{N_{max}}}} \right\rceil, \tau_{max} \geq 0, N_{max} > 0, \lambda > 0$$

החרטה של  $TSOR$  חסומה מלמעלה ע"י:

$$O(N^3 4^{N_{max}} \log M + N \tau_{max} \cdot \lambda)$$

הוכחה בנספח [3].

מהטענה נובע שהחרטה הממוצעת של  $TSOR$  לאחר  $M$  חבילות, היא באותו סדר גודל כמו החסם התחתון  $LRB$  שהצגנו, בתוספת הביטוי  $\lambda \cdot \tau_{max}$  שנובע מזמן ההתכנסות.

מסקנה 1:

אם מתקיים

$$\tau_{max} \cdot \lambda \leq N^2 4^{N_{max}} \log M$$

אז החרטה הממוצעת חסומה ע"י:

$$O(N^3 4^{N_{max}} \log M)$$

הוכחה בנספח [4]. ניתן להשיג תנאי זה ע"י הורדת הקצב  $\lambda$ , או לשפר את הרשת כך ש- $\tau_{max}$  יקטן.

מסקנה 2:

אם הלינקים בין כל שני שכנים הם בלתי תלויים אחד בשני, אז אם:

$$M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot N_{max}}} \right\rceil, \forall N_{max} > 0$$

אם בנוסף מתקיים

$$\tau_{max} \cdot \lambda \leq N^2 \cdot N_{max}^2 \log M$$

אז חסם עליון לחרטה הממוצעת הינו:

$$O(N^3 \cdot N_{max}^2 \log M)$$

ההוכחה מתבססת על כך שאם הלינקים בלתי תלויים, מתקיים:

$$P_n := \{Pr(n'|n) : \forall n' \in \mathcal{N}(n)\}$$

ואז ניתן להחליף את  $2^{N_{max}}$  ב- $N_{max}$  בהוכחות של למה 2 וטענה 1.

מכאן, שאם מבטיחים שבין כל המשתמשים הלינקים הם בלתי תלויים אחד בשני ומספר השכנים הוא לא גדול מדי, ניתן להפחית את כמות החבילות הנדרשות ב-TSOR על מנת להגיע להתכנסות, והוא יכול להגיע לחרטה ממוצעת קטנה מאוד.

$$LRB = \frac{(N+1)^3}{18} \log M$$

לעומת

$$URB = O(N^3 \cdot N_{max}^2 \log M)$$

ניתן לראות שאכן אם כמות השכנים קטנה יחסית החסם העליון קרוב מאוד לחסם התחתון, מה שמבטיח שהחרטה הממוצעת ב-TSOR תהיה נמוכה.

## ניתוח סיבוכיות זמן ריצה וזיכרון:

### שלב האתחול:

כל אחד מהמשתמשים מאתחל את טבלת מרחקי השכנים, שהיא בגודל  $N_{max}$  וכן את הפרמטרים של התפלגות  $Beta$  לכל אחת מהקבוצות  $S$ . מכיוון שקיימות  $2^{N_{max}}$  קבוצות כאלו זמן הריצה של שלב זה הינו:

$$O(2^{N_{max}} + N_{max}) = O(2^{N_{max}})$$

עבור לינקים בלתי תלויים, נקבל שמספר הקבוצות  $S$  הינו  $N_{max}$ . במקרה זה זמן הריצה של האתחול הוא:

$$O(N_{max})$$

### שלב הפצת חבילות מידע:

בחירת ה next hop - בשלב זה נדרש לעבור על המרחקים שפורסמו ע"י השכנים הישיגים ששלחו Lack ולבחור את המינימלי. בשימוש באלגוריתמי מיון יעילים נקבל זמן ריצה:

$$O(N_{max} \log N_{max})$$

כאשר  $N_{max}$  הוא מספר השכנים המקסימלי האפשרי. שאר הפעולות בשלב זה – שליחת LACK או LCFM הן מיידיות, כלומר בזמן ריצה  $O(1)$ .

### שלב הפצת ACK/NACK ועדכון מרחקים:

עדכון המרחק כולל מעבר על כל הקבוצות  $S$ , ומציאת המשתמש בעל המרחק המינימלי ב- $S$ . מיון מרחקי המשתמשים לוקח  $O(N_{max} \log N_{max})$ , ומספר הקבוצות  $S$  המקסימלי הינו  $2^{N_{max}}$ , לכן זמן הריצה הינו

$$O(2^{N_{max}} + N_{max} \log N_{max})$$

במקרה בו הלינקים בלתי תלויים אחד בשני נצטרך לעבור פעם אחת על כל השכנים, כלומר  $N_{max}$  איטרציות, וזמן הריצה קטן להיות:

$$O(N_{max} + N_{max} \log N_{max}) = O(N_{max} \log N_{max})$$

### זמן ריצה כולל:

ניתן לראות שעבור לינקים בלתי תלויים, נקבל זמן ריצה כולל של האלגוריתם לכל משתמש:

$$O(N_{max} \log N_{max})$$

### ניתוח זיכרון:

כל משתמש מתחזק את טבלת המרחקים של השכנים מהיעד, מכאן שסיבוכיות הזיכרון הינה

$$O(N_{max})$$

## סקירת הסימולציה

הסימולציה שהוצגה במאמר הינה סימולציה נומרית בתוכנת *MATLAB*, היא יעילה על מנת להראות את התכנסות האלגוריתם, ביצועיו התיאורטיים ונכונותו ביחס לפתרון האופטימלי וביחס לאלגוריתם *DORL* שהוזכר קודם.

אנו בחרנו לבצע את הסימולציה בסימולטור רשת *Omnet++* על מנת לוודא ולאשש את התוצאות גם כאשר מבצעים מימוש מלא של הפרוטוקול ולא רק מימוש תיאורטי של פעולת האלגוריתם *TSOR* בצורה נומרית.

הסימולציה מדמה רשת אד-הוק מבוססת בה כל צומת מייצגת משתמש ברשת, המצב ההתחלתי של כל משתמש הוא בהתאם לשלב האתחול של אלגוריתם *TSOR*, בדומה לסימולציה במאמר, הסימולציה מדגימה את ההתכנסות למסלול במחיר מינימלי בין צומת המקור לצומת היעד ע"י מימוש אלגוריתם *TSOR*.

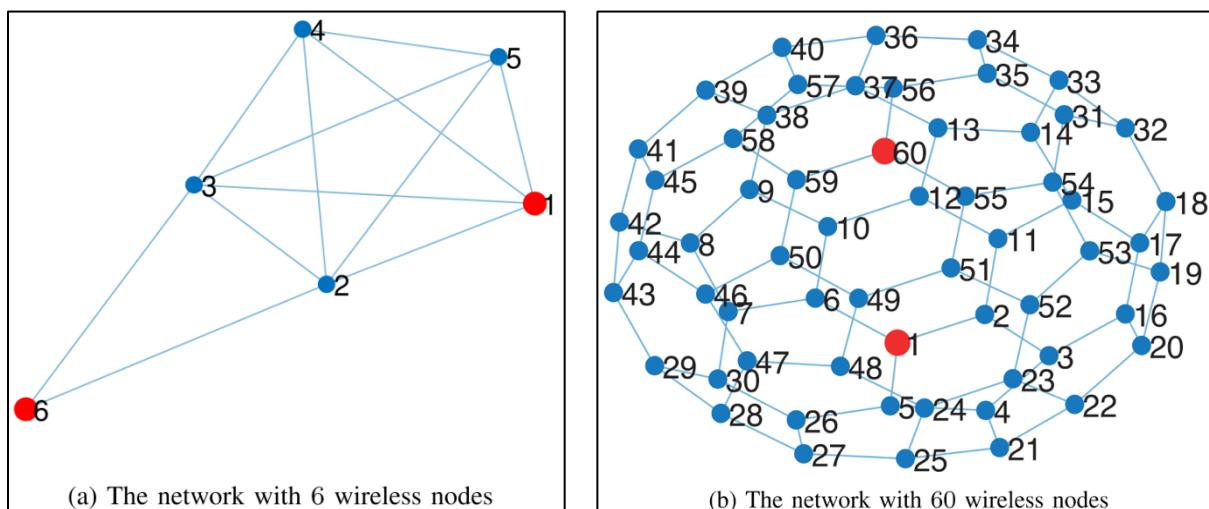
כל משתמש ברשת מחזיק 3 טבלאות:

- טבלה הממפה עבור כל אחד משכניו את פרמטרי הערוץ המתעדכנים במהלך הריצה.
- טבלת המרחקים המשוערכים ליעד דרך כל אחד מהשכנים.
- טבלה הממפה עבור כל שכן את שער הגישה לערוץ המשותף אתו.

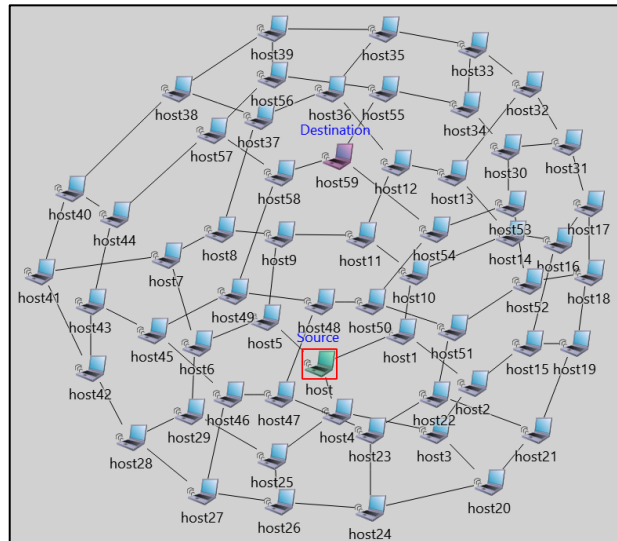
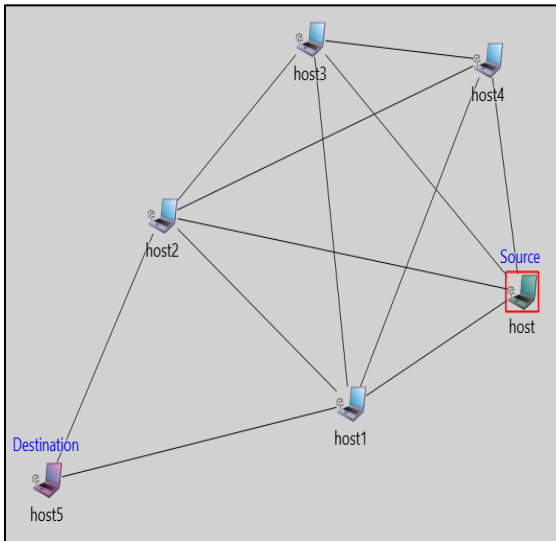
לצרכי מעקב ווידוא נכונות הפונקציונאליות במהלך כתיבת הסימולציה, הפרדנו את הודעות ה-*LACK* הנובעות מהחלק הראשון של האלגוריתם ואת אלה הנובעות מהחלק השני, הוספנו לכל אחת מהן אינדיקטור מספרי המציין איזו מהן נשלחה בערוץ, בנוסף, ע"פ האלגוריתם המוצג מעלה, הודעות מסוג *LCFM* אמורות להישלח בשידור *broadcast*, אנו בחרנו מטעמי יתירות ההודעות בסימולציה לשלוח אותן ליעד הנבחר בשידור *unicast*, שכן הודעות אלו נשלחות לאחר קבלת הודעת *LACK (Type 1)* המציינת כי הלינק ביניהם תקין בנקודות הזמן הנוכחית והזמן בין שני האירועים הנ"ל (בסימולציה) הינו אפסי, לכן לדעתנו שידור *unicast* ברוב המקרים צפוי לספק תוצאה מספקת.

כמו כן, בהתאמה לאופן ביצוע הסימולציה במאמר, הנחנו כי הלינקים הינם בלתי תלויים, הנחה זו מפשטת את מורכבות החישוב ומביאה לחסמים ההדוקים שהופיעו בניתוח במאמר.

הטופולוגיות שנבחנו בסימולציית המאמר (Fig. 2):

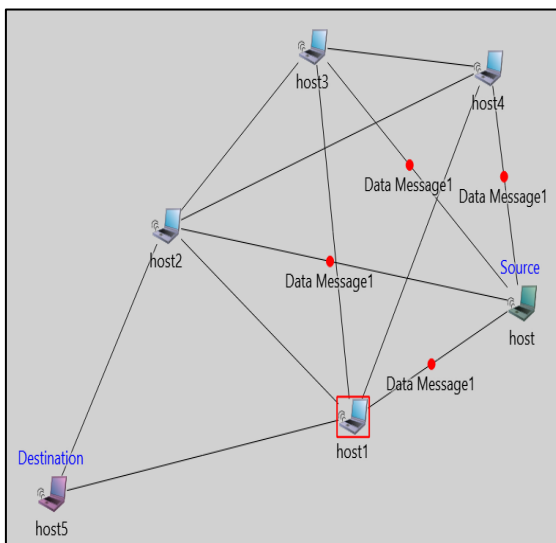


## בהתאם, הטופולוגיות בסימולציה שלנו:



**\*\* הערה:** בסימולציה שלנו מספור המשתמשים מתחיל מאפס.

## המצב ההתחלתי בסימולציה:



```

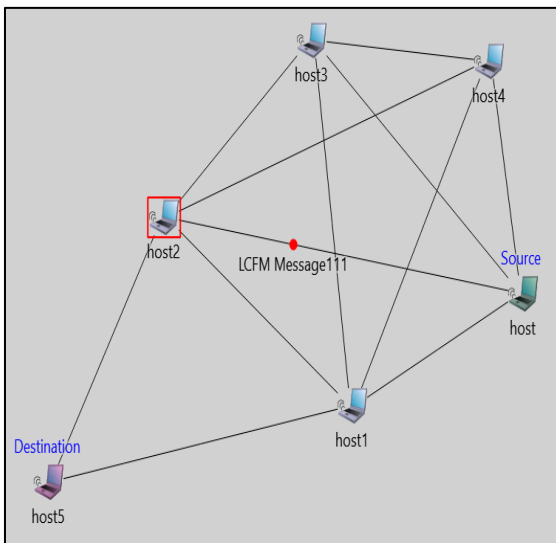
Initializing module Network1.host, stage 0
INFO (Host)Network1.host:host index: 0
INFO (Host)Network1.host:-----
INFO (Host)Network1.host:Network1.host Beta table:
INFO (Host)Network1.host:Network1.host Neighbor: 1 alpha: 1 Beta: 1
INFO (Host)Network1.host:Network1.host Neighbor: 2 alpha: 1 Beta: 1
INFO (Host)Network1.host:Network1.host Neighbor: 3 alpha: 1 Beta: 1
INFO (Host)Network1.host:Network1.host Neighbor: 4 alpha: 1 Beta: 1
INFO (Host)Network1.host:-----
INFO (Host)Network1.host:Network1.host Neighbors distances table:
INFO (Host)Network1.host:Network1.host Neighbor: 1 Distance: 0
INFO (Host)Network1.host:Network1.host Neighbor: 2 Distance: 0
INFO (Host)Network1.host:Network1.host Neighbor: 3 Distance: 0
INFO (Host)Network1.host:Network1.host Neighbor: 4 Distance: 0
INFO (Host)Network1.host:-----
INFO (Host)Network1.host:Network1.host Neighbors gates table:
INFO (Host)Network1.host:Network1.host Neighbor: 1 Gate: 0
INFO (Host)Network1.host:Network1.host Neighbor: 2 Gate: 1
INFO (Host)Network1.host:Network1.host Neighbor: 3 Gate: 2
INFO (Host)Network1.host:Network1.host Neighbor: 4 Gate: 3
    
```

**\*\* הערה:** הטבלה מתייחסת לצומת המקור.

בתחילת הסימולציה המקור מפיץ את הודעת המידע הראשונה בשידור *broadcast* לכל שבביו, בשלב הבא, בהתאם לאלגוריתם, הוא יקבל בחזרה הודעות *LACK (Type 1)* משכביו שקיבלו את הודעת המידע בהצלחה ויעדכן בהתאם את הפרמטרים של הסתברות הצלחת השידור בערוץ מול כל שכן ואת המרחק הנוכחי של כל שכן מהיעד.

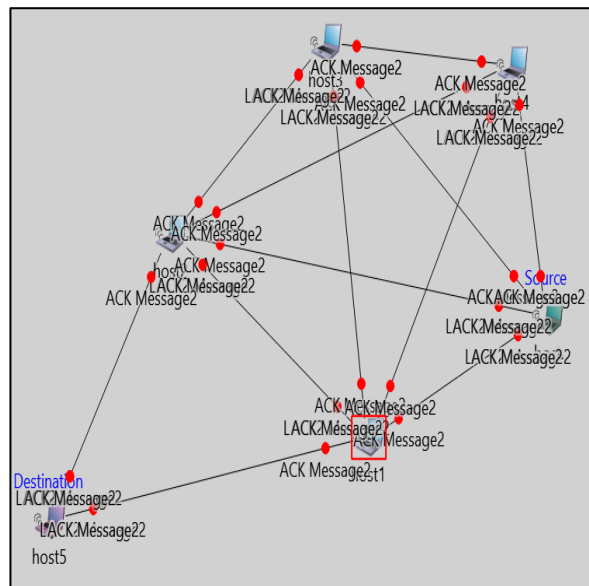
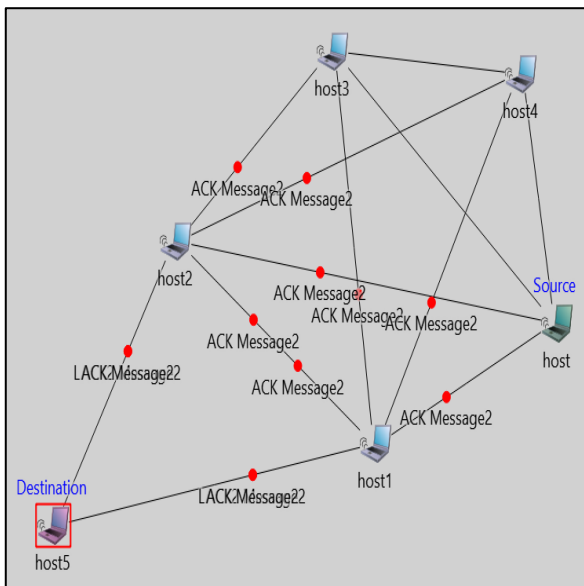


בשלב הבא ייבחר את השכן דרכו קיים מסלול ישיג במחיר מינימלי וישלח לו הודעת *LCFM*.



```
INFO:-----
INFO:Network1.host Handling Lack type 1 Message Sequence: 111
INFO:0: Lack1 count: 4
INFO:Recieved all Lack1
INFO:Network1.host Neighbor: 2 Dist: -7.07071
INFO:Network1.host Neighbor: 1 Dist: -7.03916
INFO:Network1.host Neighbor: 3 Dist: -4.85087
INFO:Network1.host Neighbor: 4 Dist: -4.83649
INFO:-----
```

ניתן לראות כי בהודעה מספר 111, לאחר קבלת כלל הודעות ה-*LACK (Type 1)*, ע"פ ערכי הטבלה העדכנית בצומת המקור, *host 2* הוא השכן בעל המרחק המינימלי ליעד ולכן המקור יישלח לו הודעת *LCFM*, כפי שניתן לראות בתמונה המצורפת מעלה.  
כך התהליך יחזור על עצמו עד אשר הודעת המידע תגיע ליעדה (*host 5*), במידה ותגיע תקינה, היעד יפיץ בשידור *broadcast* הודעת *ACK* לכל שכניו, כפי שניתן לראות בתמונות המצורפות:



כל שכן שיקבל בהצלחה את הודעת ה-*ACK*, יבצע עדכון פרמטרים אצלו בטבלאות, יעדכן את המרחק המעודכן של היעד ממנו בהודעה וימשיך להפיצה עד להגעתה למקור.  
אם בדרך ערך ה-*TTL* ירד ל-0 או שכל הלינקים נפלו, הצומת בה נעצרה ההפצה תפיץ הודעת *NACK* לכל שכניה עם הפרמטרים העדכניים עבורה.

דוגמא לעדכון המרחקים לאחר הודעת ACK אצל  $host1$  (שכן של היעד):

```

** Event #145337 t=204.402047988758 Network1.host1 (Host, id=3) on ACK Message110 (pkt, id=146432)
INFO:-----
INFO:Network1.host1 Handling ACK Message: 110
INFO:Sending Lack type 2 Message: 1 -> 4
INFO:Updating V, old: -7.07877
INFO:Neighbor: 0 beta params, alpha= 2590 beta= 638
INFO:Caluculating, beta realization: 0.80908 neighbor distance: -4.82676 overall: -3.90524
INFO:Neighbor: 2 beta params, alpha= 2619 beta= 609
INFO:Caluculating, beta realization: 0.810826 neighbor distance: -7.09791 overall: -5.75517
INFO:Neighbor: 3 beta params, alpha= 2602 beta= 626
INFO:Caluculating, beta realization: 0.80497 neighbor distance: -4.86581 overall: -3.91683
INFO:Neighbor: 4 beta params, alpha= 2582 beta= 646
INFO:Caluculating, beta realization: 0.805697 neighbor distance: -4.92023 overall: -3.96421
INFO:Neighbor: 5 beta params, alpha= 2634 beta= 594
INFO:Caluculating, beta realization: 0.812301 neighbor distance: -10 overall: -8.12301
INFO:New V: -7.12301

```

בהרצה זו, הסתברות השגיאה בערוץ הינה קבועה והוגדרה להיות  $PER = 0.1$ , מחיר כל הקשתות הוא 1, נבחין כי במצב זה  $host1$  נמצא במרחק קפיצה אחת מהיעד, ההסתברות שלו להצלחה במצב זה הינה שליחה בכיוון אחד של הודעת המידע וגם קבלה מוצלחת של הודעת ה-ACK, לכן התוצאה התיאורטית עבור הסתברות הצלחת השידור בלינק הינה:

$$p_{succ} = (1 - PER)^2 = 0.81$$

נבחין כי לאחר 110 הודעות,  $host1$  מתכנס בקירוב טוב יחסית לתוצאה התיאורטית עבור כל אחד מהלינקים עם שכניו, כמו כן, בסימולציה שלנו ערך התגמול הוגדר להיות  $R = 10$  בדומה לסימולציה המקורית במאמר, ועדכון המרחק שלו מהיעד מתקבל באופן הבא:

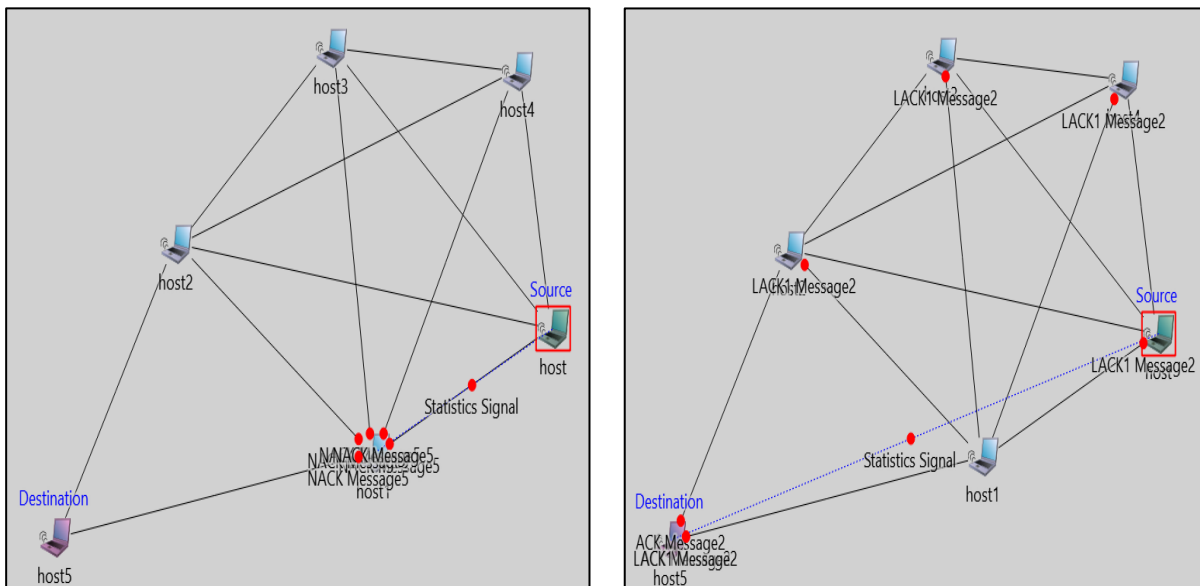
$$V_{host1} = c_{(1,5)} + p_{succ} \cdot (-R) = 1 + 0.81 \cdot (-10) = -7.1$$

## איסוף סטטיסטיקות ונתונים

איסוף הנתונים בסימולציה בוצע ע"י שימוש בסיגנל בקרה הנשלח ב-*sendDirect*, (מתודה בסימולטור המאפשרת לשלוח הודעה ישירה ממודול אחד לאחר).

בכל פעם כאשר הודעת מידע מגיעה בהצלחה לצומת היעד (בהתאם להגדרה במאמר) או כאשר היא נופלת במהלך ההפצה אצל אחד מבין צמתי הביניים, נשלחת הודעת בקרה ישירה (איננה על בסיס הלינקים של הסימולציה) לצומת המקור, אשר הטיפול בה הוגדר להיות ביצוע תיעוד של המרחק הנוכחי של המקור מהיעד, מחירי החבילה (מגלם בתוכו את אורך המסלול) והתגמול עליה. הינו  $R$  או 0.

שליחת סיגנל בקרה לאיסוף סטטיסטיקה בייצור *ACK* (מימין) ובייצור *NACK* (משמאל):



## מקרי בוחן

במהלך בניית הסימולציה בחרנו לבחון 3 מקרי בוחן שונים:

- **ערוץ אידיאלי** – הסתברות השגיאה בכל אחד מהלינקים בין הצמתים הוגדר להיות אפס, לפיכך חבילות אינן צפויות ליפול כלל (למעט מניעת מעגלים ע"י איפוס ערך ה-*TTL*).
- **הסתברות שגיאה קבועה** – כלל הלינקים בסימולציה יוגדרו להיות בעלי הסתברות שגיאה של  $PER = 0.1$ .
- **הסתברות שגיאה מפולגת אחיד** – באופן דומה לסימולציה המקורית במאמר, נרצה להגריל הסתברות הצלחת שידור המפולגת אחיד בתחום:  $(0.1, 0.9)$  ולהשוות את התוצאות המתקבלות אל מול תוצאות המאמר.

הערה: בכל אחד ממקרי הבוחן מחירי הקשתות הוגדרו להיות 1, לכן המחיר מיוצג ע"פ מספר ה-*hopcount* בחבילה.

## תוצאות הסימולציה

בסיום ההרצות, ייצאנו את הסטטיסטיקות שאספנו לקבצי CSV וביצענו את חישוב התמורה והחרטה הממוצעים מחוץ לסימולטור, את החישובים ויצירת הגרפים של התוצאות ביצענו בשפת Python.

### תוצאות הסימולציה עבור טופולוגיה ( $\alpha$ )

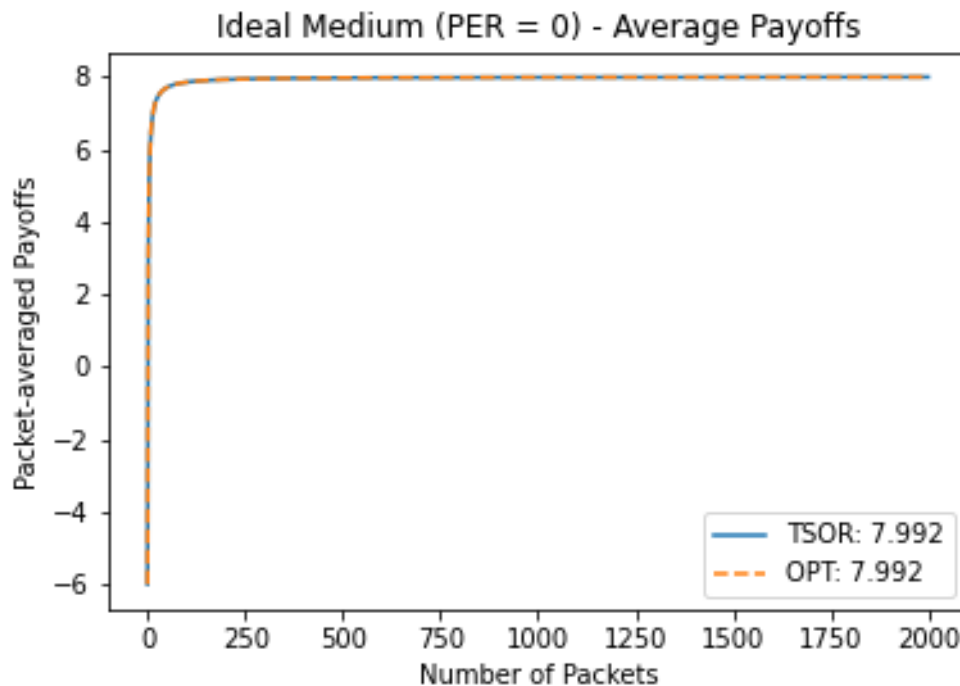
#### • ערוץ אידיאלי:

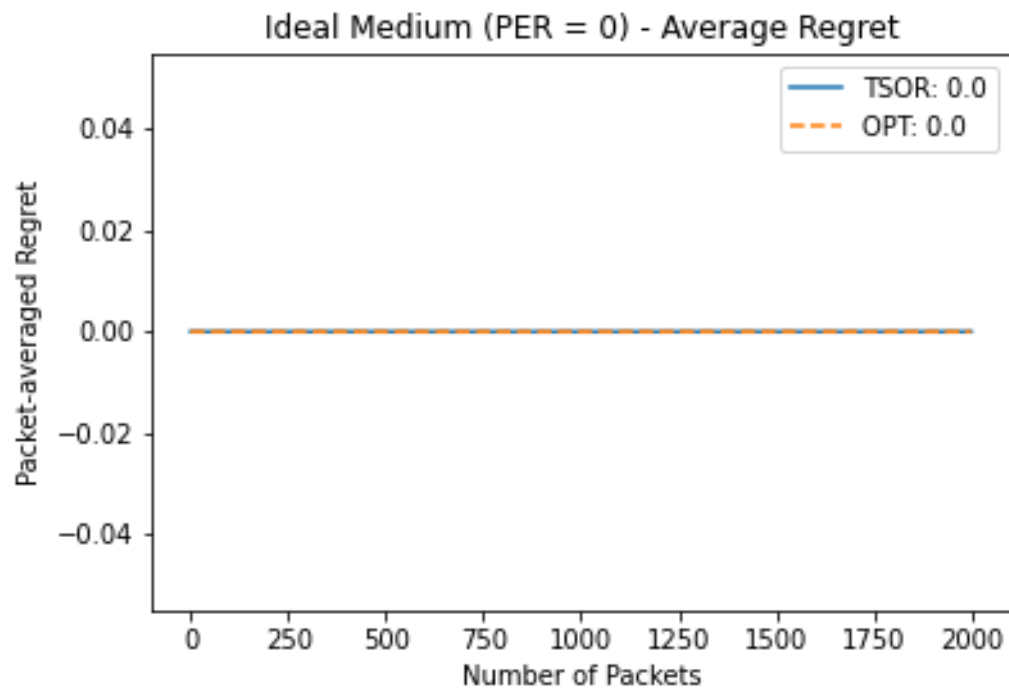
בערוץ אידיאלי אנו צופים כי אף חבילה לא תיפול, כלומר, ככל שיגדלו מספר ההודעות בערוץ, כך יגדל מספר הודעות ה-ACK שיייוצרו, בעוד שמספר הודעות ה-NACK יישאר 0, בהתאם כל משתמש יעדכן את הפילוג שלו ע"י הגדלה של  $\alpha$  ב-1 בעוד הפרמטר  $\beta$  יישאר 1 (תנאי ההתחלה שלו), נבחין כי מתקיים:

$$E[Beta(\alpha, \beta = 1)] = \frac{\alpha}{\alpha + 1}, \quad \lim_{\alpha \rightarrow \infty} \frac{\alpha}{\alpha + 1} = 1$$

כלומר, נצפה שהחל מנקודה מהודעה מסוימת, כל הריאליזציות של הפילוג יהיו באזור של 1, ומכאן שהפתרון המתקבל צפוי להיות שקול לפתרון האופטימלי בו ההסתברות לשידור מוצלח ידועה.

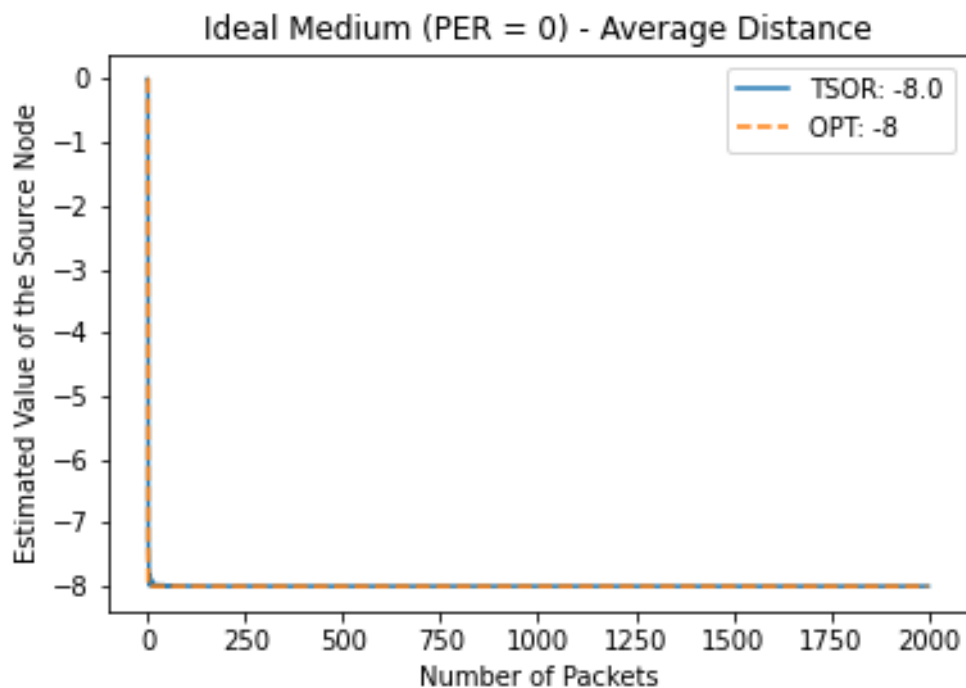
התוצאות המתקבלות עבור תמורה ממוצעת וחרטה ממוצעת:





נבחין כי התוצאות אכן מתיישבות עם הציפייה ואלגוריתם *TSOR* אכן מתכנס לפתרון האופטימלי, התמורה הממוצעת המתקבלת בהרצה הנ"ל הינה 7.992 וכך גם של האלגוריתם האופטימלי לכן החרטה הממוצעת הינה 0.

התוצאה המתקבלת עבור מרחק משוערך מהמקור ליעד:



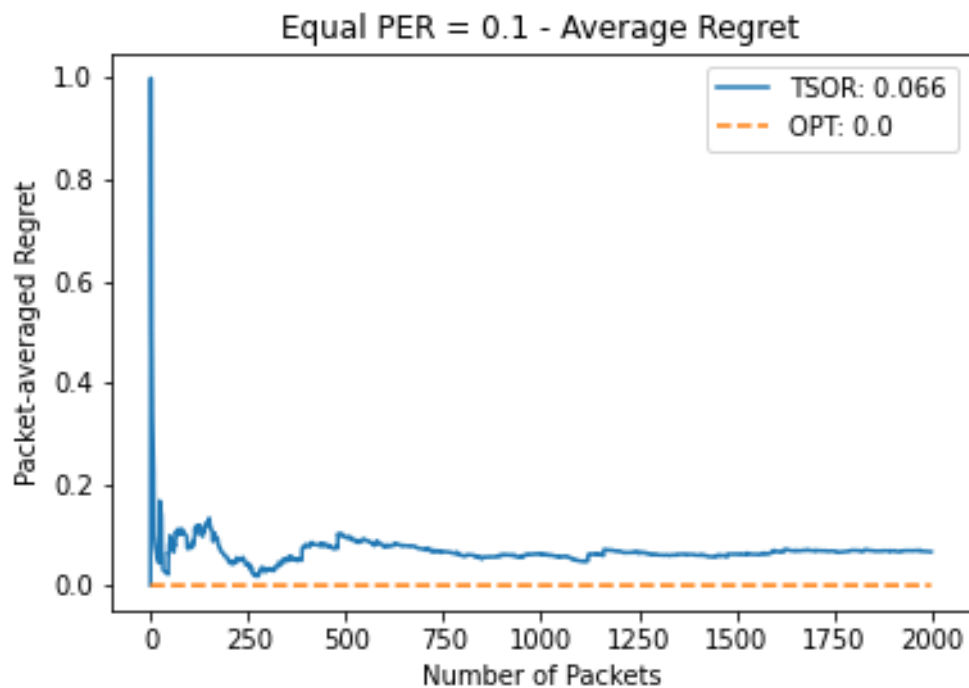
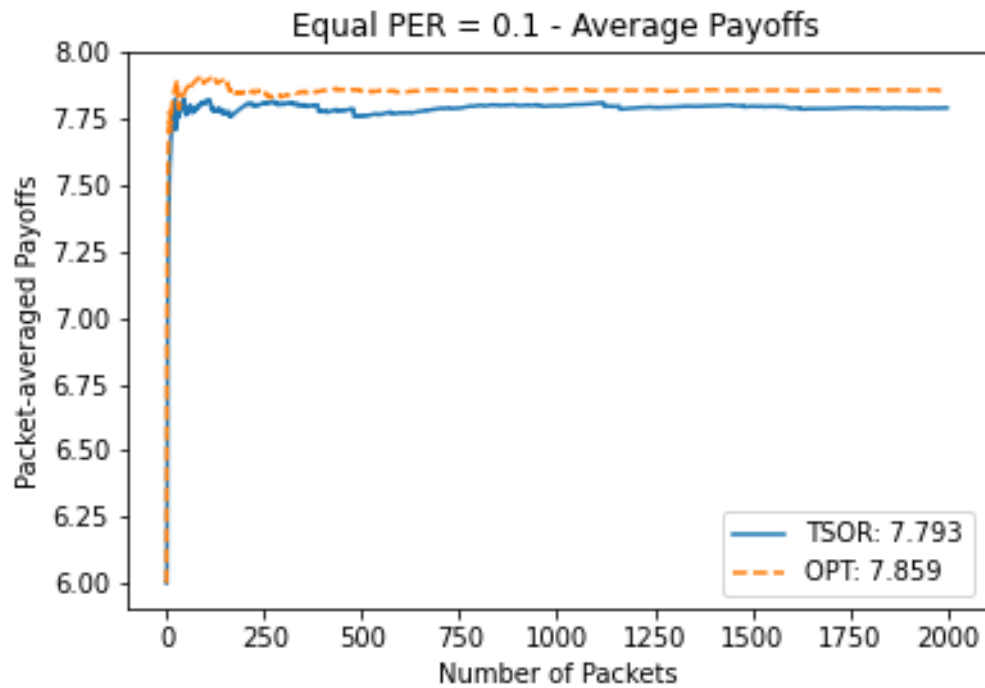
בהתאם, המרחק המשוער בין המקור ליעד באלגוריתם *TSOR* התכנס לערך האופטימלי.

- הסתברות שגיאה קבועה:

כאשר כל הערוצים הינם בעלי הסתברות שגיאה קבועה של 0.1, נצפה לקבל התכנסות של פילוג הערוץ בכל אחד מהמשתמשים למומצע ברנולי המתאים:

$$E[Beta(\alpha, \beta)] = \frac{\alpha}{\alpha + \beta} \rightarrow (1 - 0.1)^2 = (0.9)^2 = 0.81$$

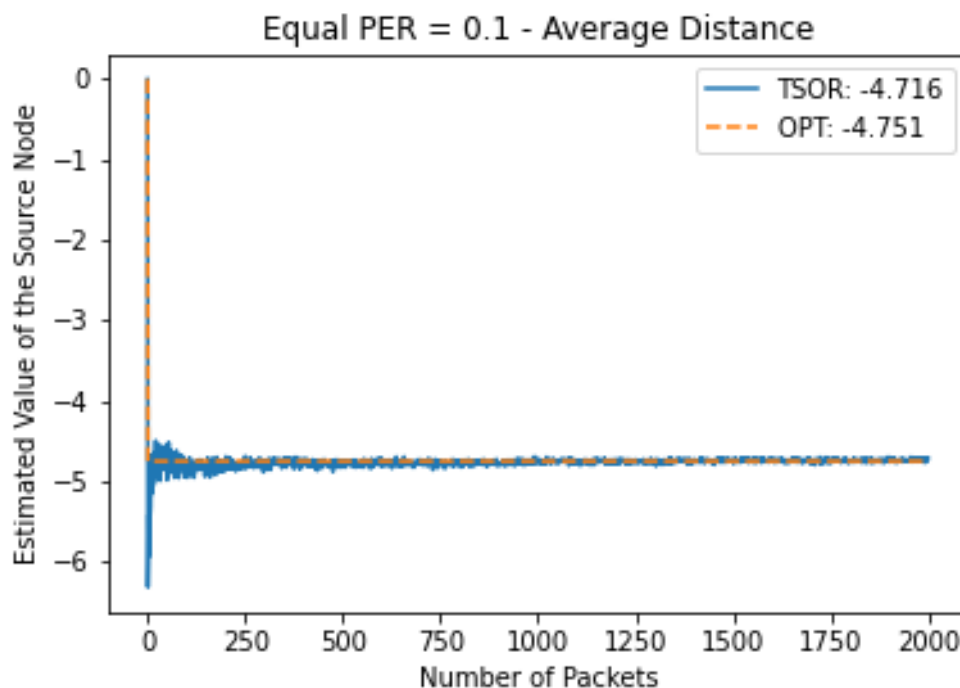
התוצאות המתקבלות עבור תמורה ממוצעת וחרטה ממוצעת:



נבחין כי התוצאה אכן הגיונית, כאשר הסתברות השגיאה גדולה מאפס, שערך הערוץ אינו מושלם ומהווה קירוב בלבד להסתברות הצלחת השידור האמיתית, לכן הפתרון האופטימלי (בו הסתברות זו ידועה מראש לכלל המשתמשים) משיג תמורה מעט טובה יותר וחרטה ממוצעת של 0 (ע"פ הגדרה).

מצד שני, נבחין לאחר מספר קטן יחסית של חבילות מידע, אלגוריתם TSOR מתכנס בקירוב טוב יחסית לערכים של האלגוריתם האופטימלי, נציין בנוסף כי בהתאם לסימולציה שבוצעה במאמר מספר הודעות המידע הוגבל ל-2,000, לכן אנו מניחים כי ככל שניתן לסימולציה לרוץ אלגוריתם TSOR יקרב את הפילוג בצורה עדינה ומדויקת יותר להסתברות הצלחת השידור האמיתית ולכן ילך ויתכנס לביצועים האופטימליים.

התוצאה המתקבלת עבור מרחק משוערך מהמקור ליעד:



נבחין כי גם ע"פ תוצאות המרחק הממוצע של המקור מהיעד, ניתן לראות שהאלגוריתם האופטימלי משיג מרחק מינימלי יותר (סימן שלילי), אך הערך אליו מתכנס אלגוריתם TSOR הינו מאוד קרוב אליו, אחוז השגיאה לאחר 2,000 הודעות הינו:

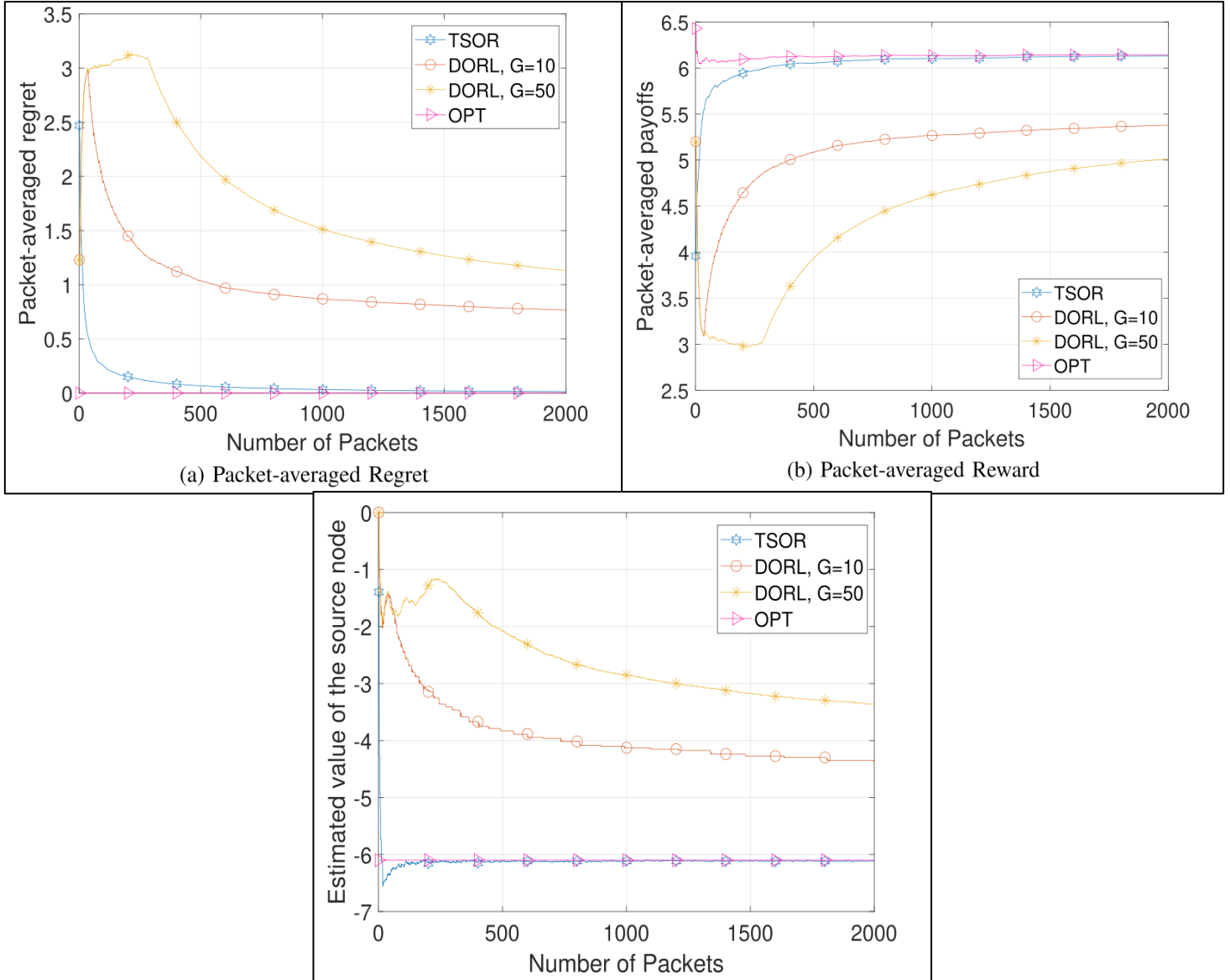
$$\left| \frac{Dist_{OPT} - Dist_{TSOR}}{Dist_{OPT}} \right| = \frac{4.751 - 4.716}{4.751} = \frac{35}{4751} \sim 0.007\%$$

באופן דומה לנימוקים מעלה, אנו מניחים שאם לא נגביל את מספר חבילות המידע וניתן לסימולציה להמשיך לרוץ נגיע לאחוז שגיאה הולך וקטן והפתרון המתקבל ישאף לערך האופטימלי.

נשים לב כי שני מקרי הבוחן הבסיסיים שנועדו עבור בדיקת שפיות עצמית אכן מתנהגים כפי שציפינו, כעת נעבור למקרה הבוחן השלישי המדמה את תוצאות הסימולציה במאמר.

• הסתברות שגיאה מפולגת אחיד:

תוצאות הסימולציה כפי שהן מוצגות במאמר (Fig. 3, 5):



על מנת להתאים את תוצאות הסימולציה לתוצאות המאמר, נבחר את פילוג ההסתברות השגיאה בערוץ כך שיתקיים שההסתברות להצלחת שידור תהיה מפולגת אחיד בתחום  $(0.1, 0.9)$ , נבצע חישוב לאחור ונמצא את הטווח הדרוש עבור התפלגות השגיאה:

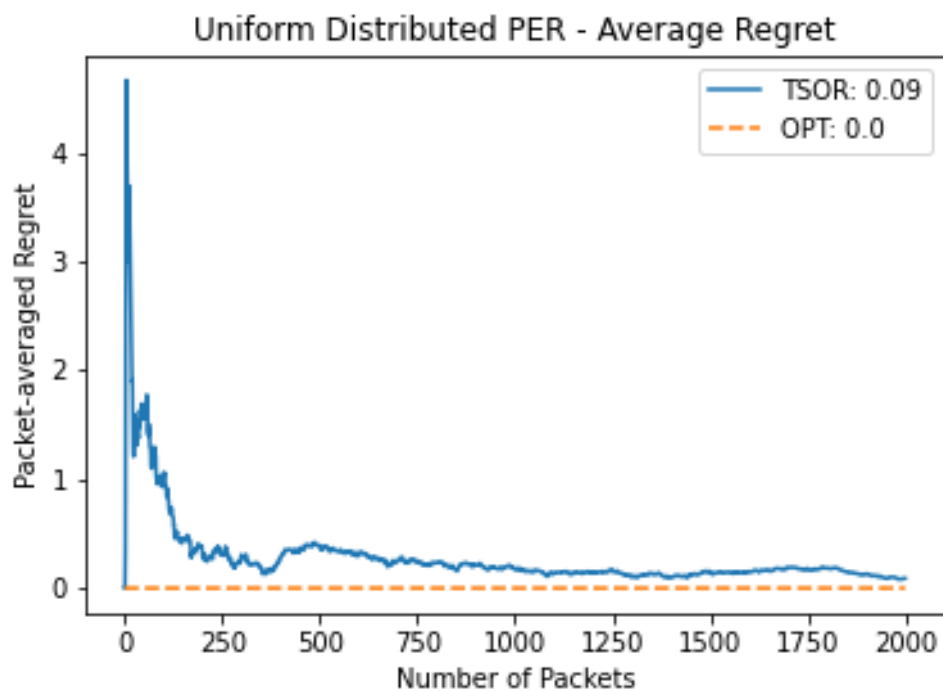
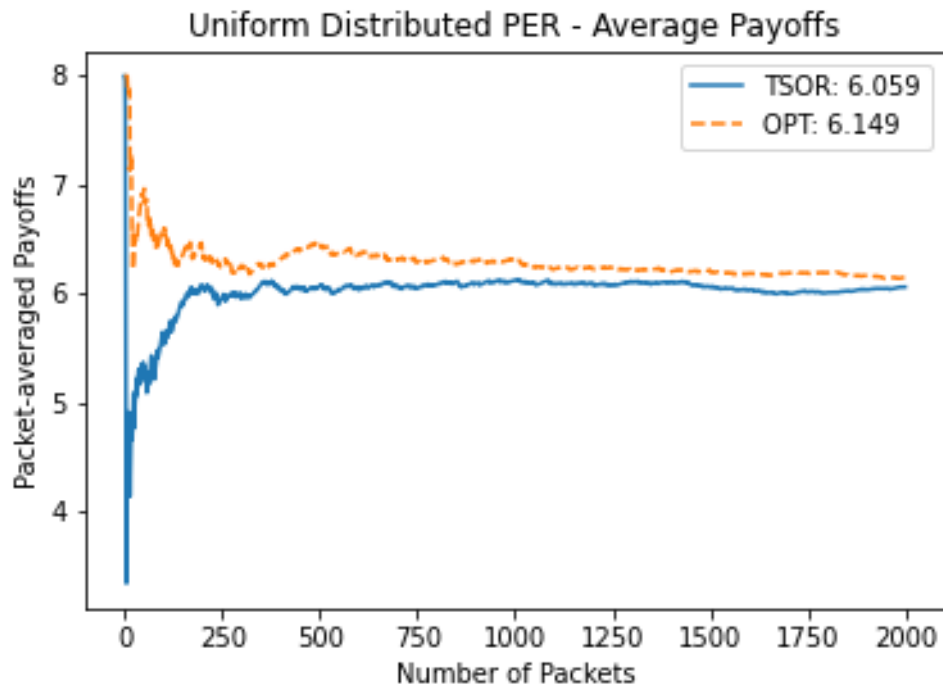
$$p_{succ} = (1 - PER)^2, \quad 0.1 \leq (1 - PER)^2 \leq 0.9$$

$$0.05 \approx 1 - \frac{3}{\sqrt{10}} \leq PER \leq 1 - \frac{1}{\sqrt{10}} \approx 0.68$$



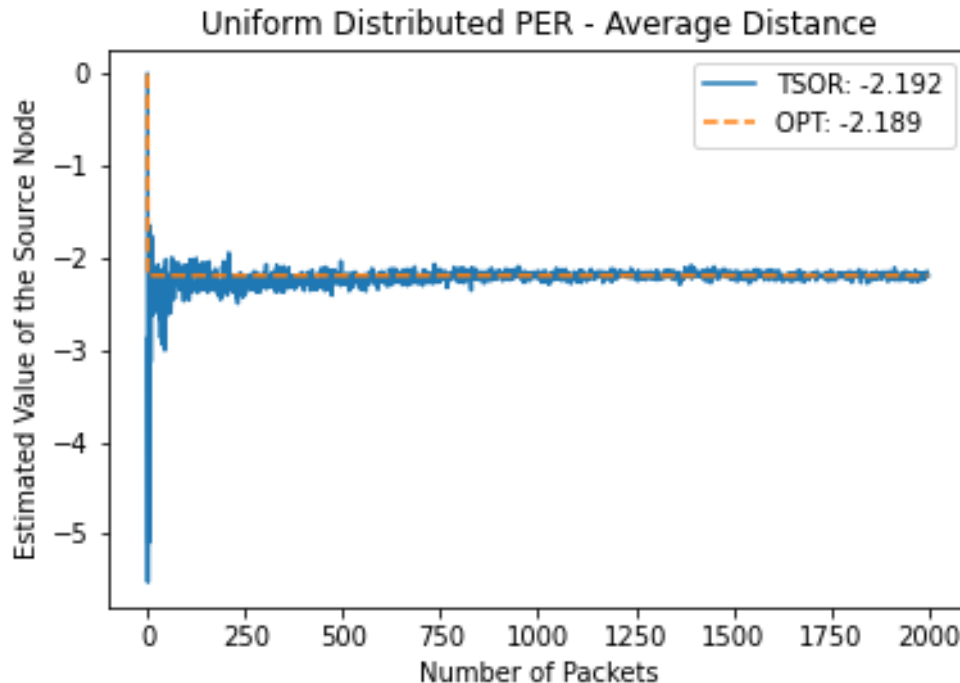
בהתאם לתוצאת החישוב הנ"ל בחרנו את הסתברות שגיאה בערוץ מתוך פילוג אחיד בתחום:  $(0.05, 0.68)$ .

התוצאות המתקבלות עבור תמורה ממוצעת וחרטה בסימולציה שלנו:



נבחין כי גם עבור המקרה בו הסתברות השגיאה על הלינקים איננה קבועה, אלגוריתם TSOR מצליח להשיג תוצאה טובה יחסית בהשוואה לתוצאות האלגוריתם האופטימלי, מכאן נוכל להסיק כי שיטת הדגימה של תומפסון אכן מצליחה לשערך את הערוץ באופן אפקטיבי. בהתאם ניתן לראות כי האלגוריתם האופטימלי התכנס לערך 6.149 שהוא בקירוב הערך שהתקבל גם בתוצאות המאמר, בהתאם נראה גם כי ערך החרטה הממוצעת שואף לאפס כפי שקיבלנו.

התוצאה המתקבלת עבור מרחק משוערך מהמקור ליעד:



אחוז השגיאה לאחר 2,000 הודעות הינו:

$$\left| \frac{Dist_{OPT} - Dist_{TSOR}}{Dist_{OPT}} \right| = \frac{2.192 - 2.189}{2.189} = \frac{3}{2189} \sim 0.00137\%$$

נבחין כי הערך אליו התכנסו האלגוריתם האופטימלי יחסית רחוק מהערך בתוצאות המאמר, אנו מניחים כי ההבדל נובע מכך שבסימולציה, האלגוריתם האופטימלי התכנס למסלול במחיר המינימלי שאיננו המסלול הקצר ביותר (במונחים של *hop count*), וזאת כתוצאה מריאליזציות בעלות הסתברות שגיאה גבוהה יחסית באופן ספציפי על פני אותם לינקים השייכים למסלול הקצר ביותר (הביסוס להנחה הנ"ל היא ע"פ טענותיהם של כותבי המאמר בפסקת הדיון בתוצאות).

נציין בנוסף כי במאמר הסימולציה שבוצעה הייתה נומרית לעומת סימולציית הרשת שביצענו. אנו מאמינים כי התמודדנו עם מקרים שלא קיימים בסימולציה נומרית, כמו לדוגמא מצב שבו הודעה הגיעה ליעד, אך ה-ACK עליה לא התקבל אצל שכני היעד והם הפיצו NACK. הדבר מוביל לקושי מבחינת איסוף הסטטיסטיקה שכן במאמר התגמול הוגדר להיות ברגע שההודעה מגיעה ליעד, אך מבחינת תקינות הפרוטוקול, הפצת ה-NACK תקינה שכן מבחינת שכני היעד ההודעה לא הגיעה אליהם. בנוסף, נדרש לממש מנגנון של סנכרון הודעות אשר יוודא היעדר התנגשויות ואת מהלך האלגוריתם התקין.

לפי דעתנו ייתכן כי הפערים הללו הובילו להבדלים מספריים בתוצאות, אך מכיוון שהגדרנו שהאלגוריתם האופטימלי ירוץ תחת אותם התנאים כמו TSOR, המגמה הכללית נשמרת.

### תוצאות הסימולציה עבור טופולוגיה (b)

ההרצה עבור הטופולוגיה הגדולה (60 משתמשים) בסימולטור רשת אשר מבצע בפועל שליחה של כל ההודעות המפורטות באלגוריתם (בניגוד לסימולציה הנומרית שבוצעה במאמר) מהווה משימה חישובית מורכבת ודורשת משאבי חישוב שלא היו זמינים עבורנו, למרות זאת, ניסינו לשנות מעט את הגדרות הסימולציה שתוארו במאמר על מנת לבחון את המימוש.

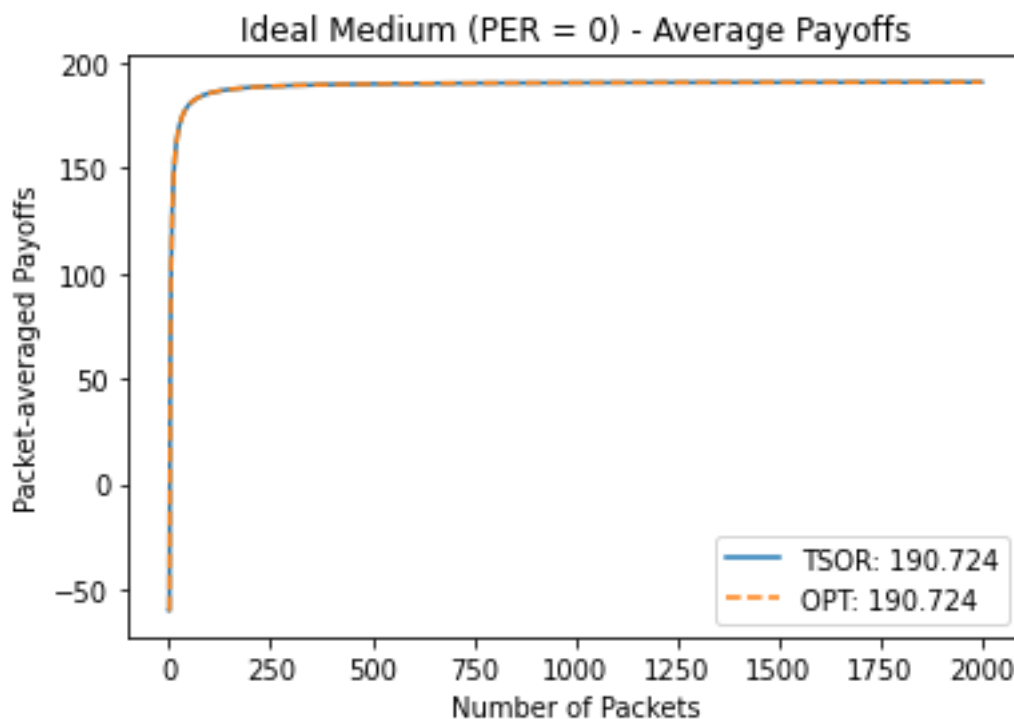
השינויים שביצענו ספציפית עבור הטופולוגיה המורחבת הינם:

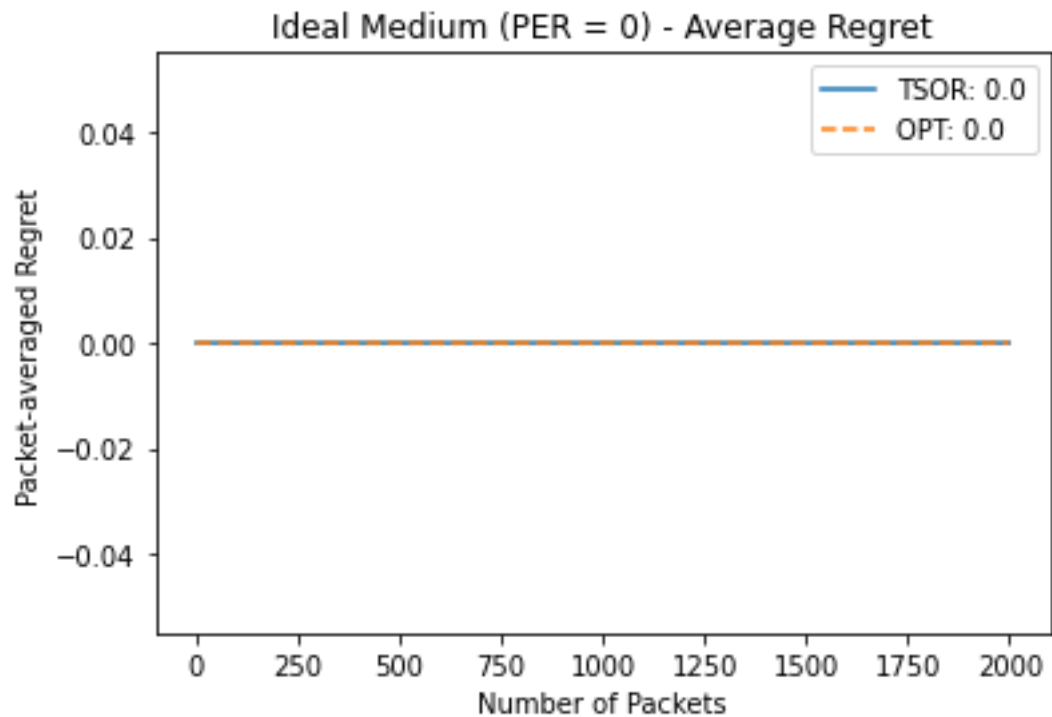
- הקטנת ערך ה- $TTL$  עבור הודעות  $ACK/NACK$  מ-60 ל-13: מספר ההודעות בעת שידור  $broadcast$  גדל אקספוננציאלית כפונקציה של מספר השכנים וערך ה- $TTL$ , בטופולוגיה המורחבת המסלול הקצר ביותר (במונחים של  $hop\ count$ ) בין צומת המקור לצומת היעד הינו באורך 9, לכן ניסינו למצוא ערך  $TTL$  וערך  $seed$  מתאים כך שמצד אחד הודעות המידע יוכלו להגיע ליעד ומצד שני שנוכל להראות את ההתכנסות ולכן בחרנו בערך 13, זמן הריצה ערך כ-60 דקות על גבי שרת באוניברסיטה בעל חומרה חזקה מזו של מחשבינו האישיים.
- על מנת לאזן את ההשפעה של הפחתת ערך ה- $TTL$ , בחרנו גם לצמצם את הקטע מתוכו מגרילים את הסתברות השגיאה בכדי לאפשר להודעות המידע אכן להגיע ליעדן.

ברור לנו שכתוצאה מהשינויים שביצענו, התוצאות אינן משקפות היטב את יעילות האלגוריתם אך החלטנו להציג אותן בכל זאת על מנת להראות את ההתכנסות.

#### • ערוץ אידיאלי:

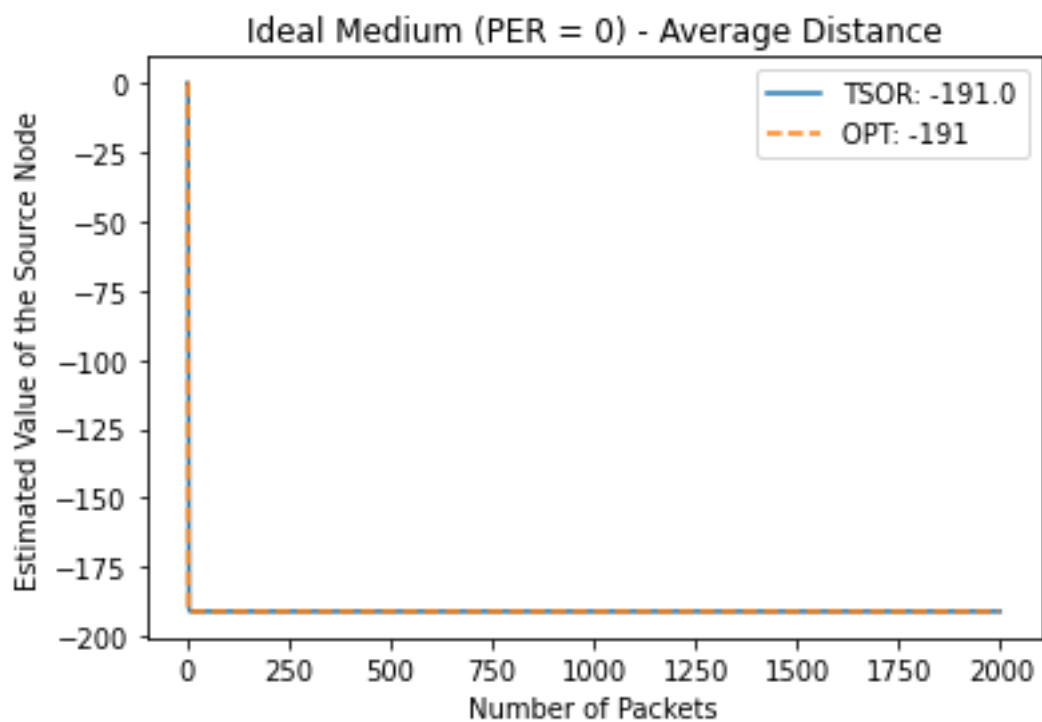
כאן הסתברות הצלחת השידור הינה 1, אין חבילות מידע שנופלות בדרך, לכן בהתאם לתוצאות עבור הטופולוגיה המצומצמת אנו מצפים לראות שהאלגוריתם האופטימלי יבחר במסלול הקצר ביותר (באורך 9) וכך גם אלגוריתם  $TSOR$ .





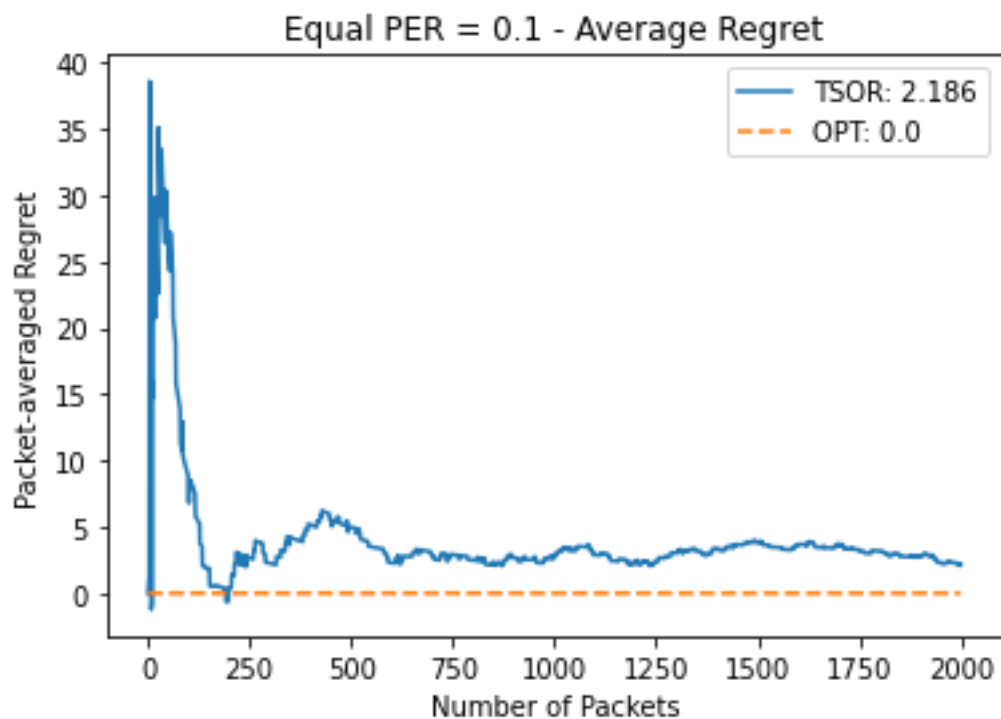
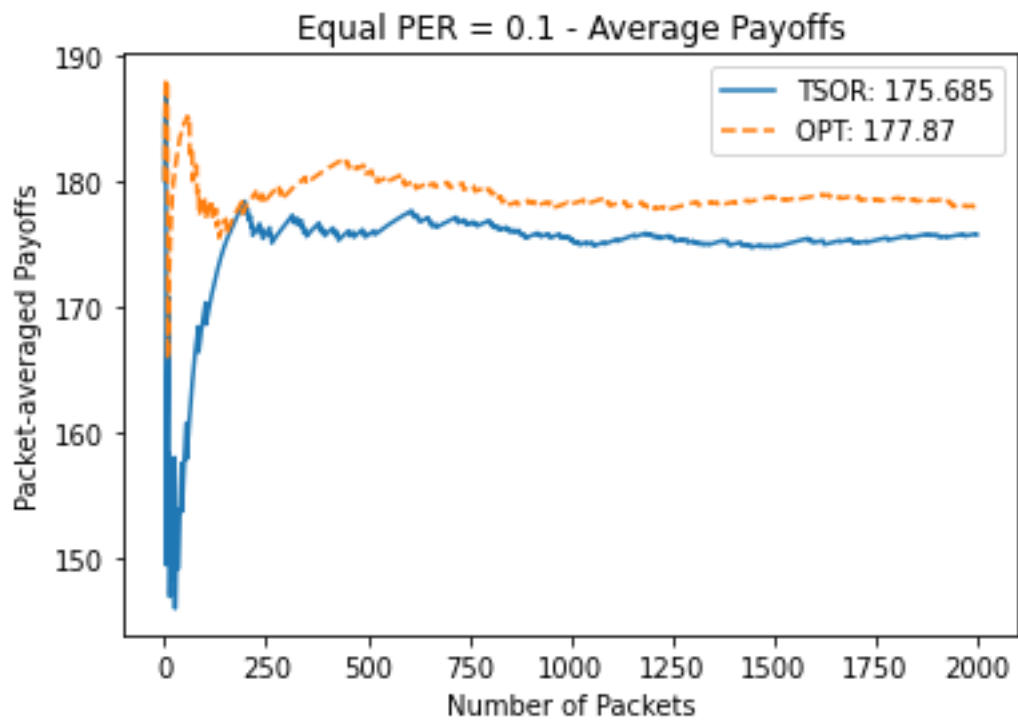
נבחין כי בהתאם לציפייה, כיוון שערך ה- $TTL$  הוגדר להיות גדול ממש מאורך המסלול הקצר ביותר בטופולוגיה  $TSOR$  מצליח להתכנס בדיוק מלא לתוצאות האלגוריתם האופטימלי.

התוצאה המתקבלת עבור מרחק משוערך מהמקור ליעד:



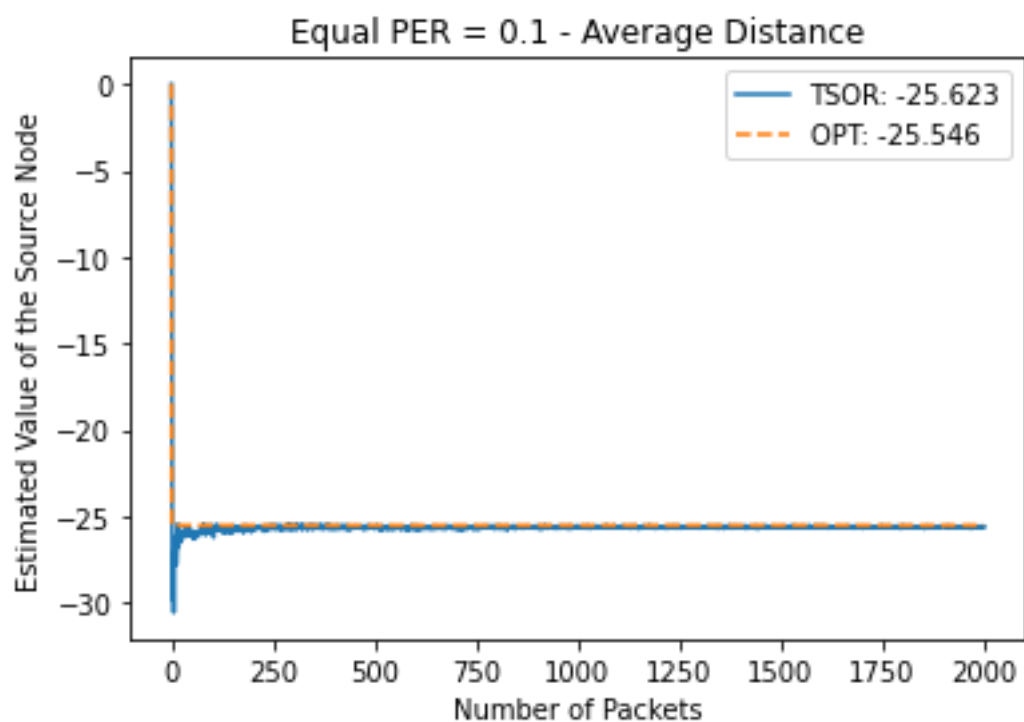
בהתאם, ניתן לראות כי המרחק המינימלי שהושג מצומת המקור ליעד בשני האלגוריתם הינו בקירוב טוב זהה.

- הסתברות שגיאה קבועה:



נבחין כי בעת אלגוריתם *TSOR* מתכנס בצורה פחות מדויקת ופחות מהירה בהשוואה לטופולוגיה המצומצמת אך ניתן לראות בבירור כי הוכחת ההתכנסות אכן מתקיימת וזאת על אף השינויים ב-*TTL* אשר משפיעים באופן ישיר על מהירות הגילוי של הטופולוגיה עבור כל אחת מהצמתים.

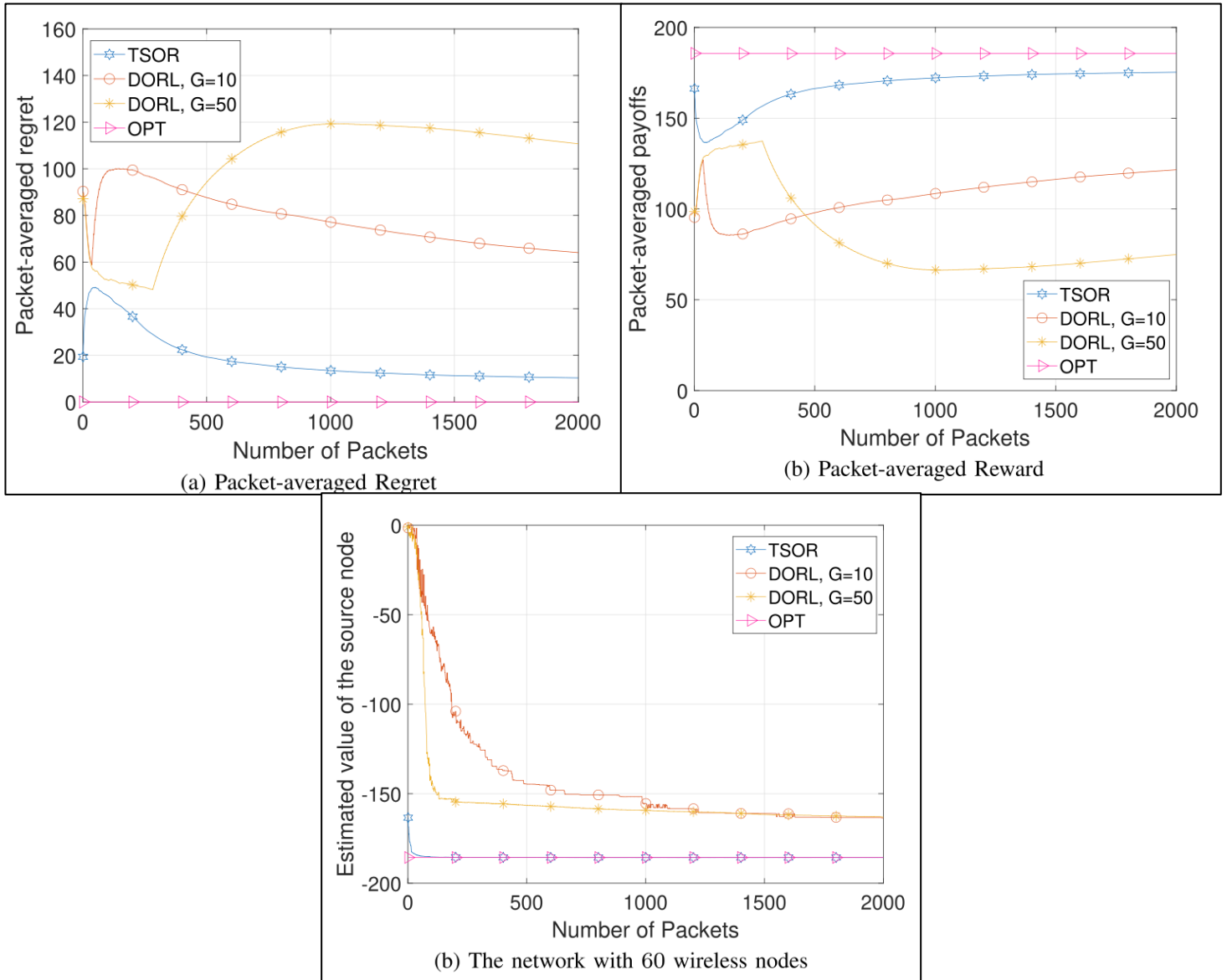
התוצאה המתקבלת עבור מרחק משוערך מהמקור ליעד:



נבחין כי למרות ההבדל היותר משמעותי בין האלגוריתם האופטימלי לאלגוריתם *TSOR* בערך התמורה הממוצעת וערך החרטה הממוצעת, עדיין שני האלגוריתמים מתכנסים בצורה יחסית מובהקת על ממוצע מרחק זהה בקירוב טוב בין המקור ליעד.

• הסתברות שגיאה מפולגת אחיד:

תוצאות הסימולציה כפי שהן מוצגות במאמר (Fig. 4, 5):

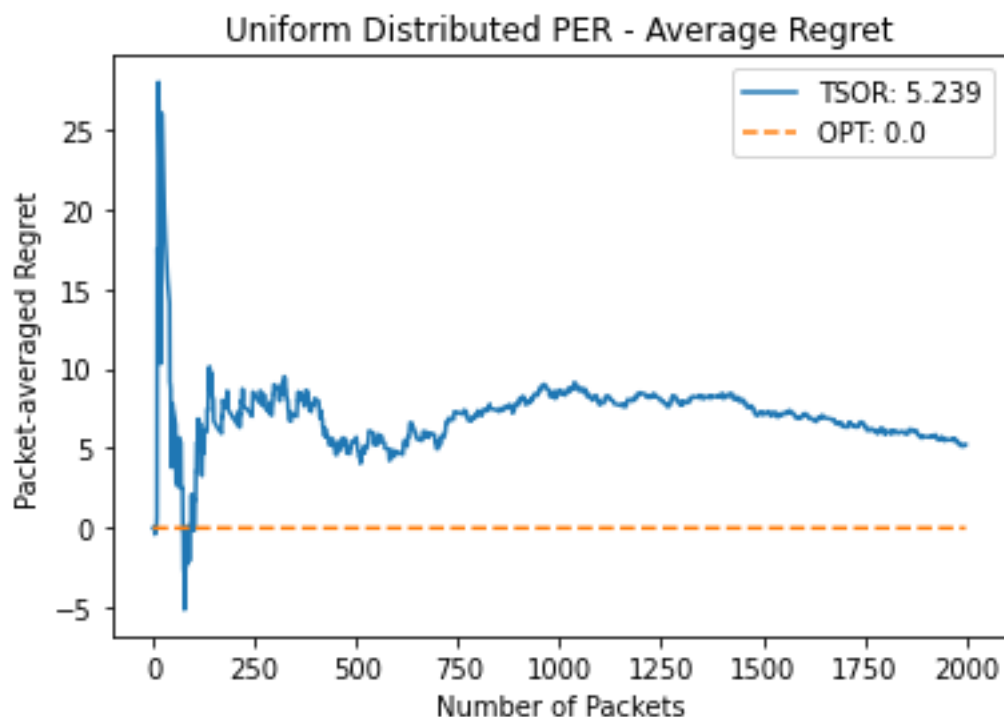
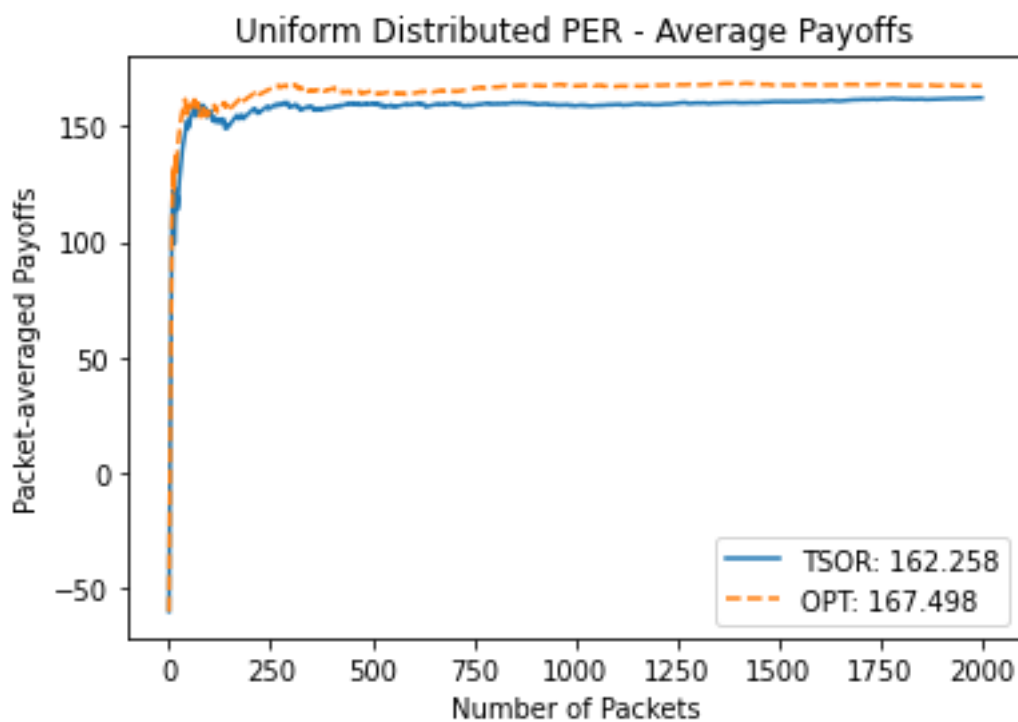


כפי שתיארנו בהגדרת הסימולציה עבור הטופולוגיה הנ"ל, בחרנו לשנות את הקטע מתוכו מוגרלת הסתברות השגיאה על כל אחד מהלינקים ועבור הקטע: (0.05,0.25) קיבלנו שהודעות המידע אכן מצליחות להגיע ליעדן למרות שינוי ערך ה-TTL.

באופן מפורש:

$$0.05 \leq PER \leq 0.25$$

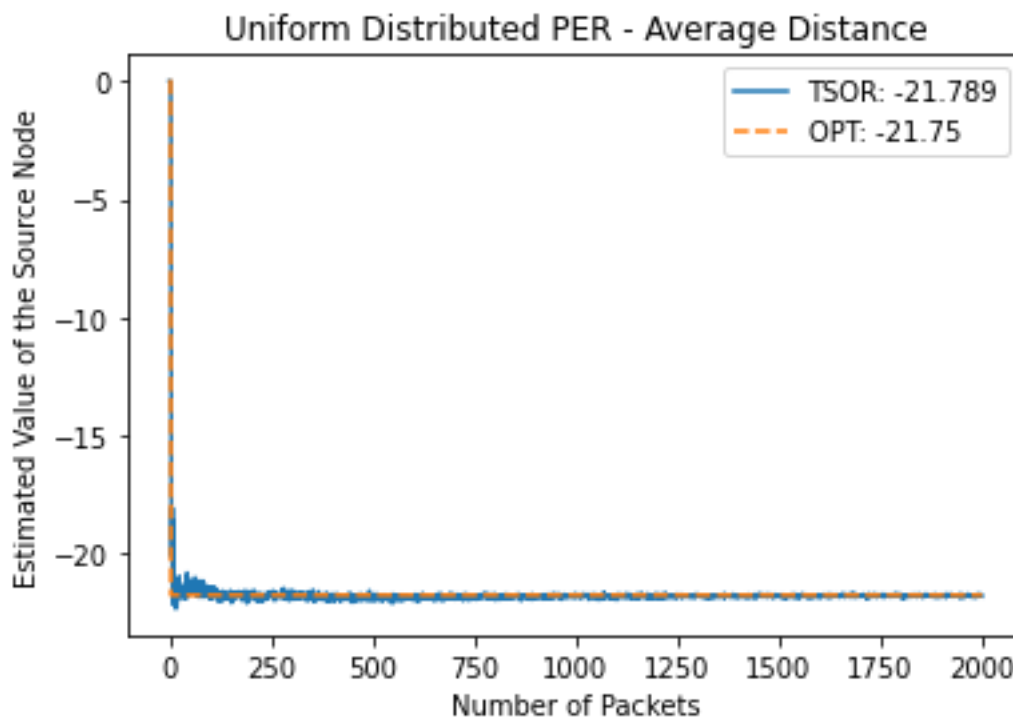
התוצאות המתקבלות עבור תמורה ממוצעת וחרטה בסימולציה שלנו:



כתוצאה מהקטנת ה-TTL בעצם מנענו מכל הצמתים להכיר את כלל הצמתים סביבם, לפיכך כל צומת מחזיקה מידע חלקי אודות הרשת וייתכן כי האלגוריתם האופטימלי כלל לא יצליח להתייבב ולהתכנס למסלול הקצר ביותר ברשת, בפרט הדבר נכון גם עבור אלגוריתם TSOR, אך ניתן לראות



כי על אף הבחנה זו שניהם מצליחים להשיג תמורה ממוצעת קרובה יחסית ובהתאם מתקבלת התנהגות של ערך החרטה הממוצעת בתוצאותינו ביחס למגמה המוצגת בתוצאות המאמר. התוצאה המתקבלת עבור מרחק משוערך מהמקור ליעד:



כאמור, ייתכן כי האלגוריתם האופטימלי לא בהכרח יצליח להתכנס למסלול האופטימלי ברשת בעקבות הקטנת ערך ה- $TTL$ , אך בהתאם לערך ה- $TTL$  שהגדרנו האלגוריתם האופטימלי ייבחר במסלול שהינו הקצר ביותר האפשרי ברשת תחת אותה המגבלה. כעת, נבחין כי שני האלגוריתמים התכנסו למרחק ממוצע זהה בקירוב טוב, מכאן אנו מסיקים כי הם התכנסו לבסוף לאותו המסלול – שהוא המסלול הקצר ביותר האפשרי שחושב על בסיס האלגוריתם האופטימלי.

אנו מניחים כי ההבדל הקטן בתוצאה בסופית נובע מזמן ההתכנסות של  $TSOR$ , שכן בניגוד לאלגוריתם האופטימלי אין בידיו את הסתברויות הצלחת השידור המדויקות מהאיטרציה הראשונה ולכן נדרשות מספר חבילות מידע נוספות על מנת לגלות את הרשת ולהיות מסוגלים לבחור במסלול הרצוי.

ניתן לראות גם בגרף כי העקומה הכחולה (המתארת את המרחק הממוצע של המקור מהיעד במהלך ריצת האלגוריתם) מבצעת מספר קפיצות מסביב לערך ההתכנסות של האלגוריתם האופטימלי עד להתייצבות.

לסיכום, על אף שנאלצנו לבצע שינויים במימוש הסימולציה ועל אף שהתוצאות המספריות בתוצאות המתקבלות מעט רחוקות מאלו המוצגות במאמר, לדעתנו הן עדיין מצליחות לספק הוכחת נכונות לאלגוריתם  $TSOR$  ואכן מתיישבות היטב עם הציפייה שלנו.

## מסקנות וביקורת על המאמר

### 1. נקודות חוזקה וחולשה

#### נקודות חוזקה:

- ✓ המאמר כתוב בצורה מסודרת וברורה. תיאור מודל הבעיה, מקרה הבוחן, גישת הפתרון נכתבו בפירוט רב והקלו על הבנת האנליזה המתמטית והרעיון המרכזי.
- ✓ לפי דעתנו הבחירה בניתוח חסמים הייתה נכונה שכן היא משקפת בצורה מיטבית ביצועים של אלגוריתם. ניתוח החסמים שבוצע במאמר היה מקיף ותרם להבנת הנכונות של האלגוריתם המוצע.
- ✓ מדדי ההערכה של האלגוריתם שנבחרו והשוואה לאלגוריתם אופטימלי ואלגוריתמים דומים באופי פעולתם אפשרו לנו להבין את תוצאות הסימולציות שבוצעו במאמר
- ✓ תיאור האלגוריתם הכללי מפורט היטב, דבר שאפשר לנו לממש אותו בעצמנו בסימולטור רשת בניגוד לסימולציה נומרית ב-MATLAB ולאשש את נכונותו.

#### נקודות חולשה:

- × הסימולציה המתוארת במאמר בוצעה ב-MATLAB ולכן הינה נומרית בלבד, אינה מתמודדת עם האתגרים הקיימים בסימולציית רשת למשל- התנגשות הודעות על הלינקים, מצב שבו אין מסלול ליעד בנקודת זמן מסוימת. כמו כן תיאור הפרוטוקול לא מכסה מקרי קצה שנתקלנו בהם בסביבת המקור ובסביבת היעד כפי שתיארנו בדיון בתוצאות.
- × לדעתנו שלב האתחול באלגוריתם הינו מעט בעייתי, שכני היעד מאתחלים את המרחק ממנו כך:  $V(n, n') = -R$ . דבר העלול להוביל לכך שבעת קבלת ACK מהיעד, המרחק המפורסם ממנו יהיה זהה למרחק המעודכן אצלם, ההפצה תיעצר וכך גם עדכון המרחקים ברשת.
- × על פי הגדרת הפרוטוקול, אין מניעת מעגלים באופן ישיר, אלא רק מניעת ניתובים אינסופיים ע"י שימוש ב-TTL. לדעתנו, ניתן להפחית בצורה משמעותית את overhead, ע"י שימוש במנגנון מסוג זה, למשל path vector, אשר מכיל בתוך ההודעה את הדרך שעברה וכך מזהה מעגלים.

## 2. כיווני מחקר עתידיים

- חסם החרטה העליון שהוצג במאמר הינו ריבועי בגודל הרשת ואקספוננציאלי במספר השכנים המקסימלי. מכאן שברשתות גדולות החרטה עלולה להיות מאוד גבוהה. יש לחקור את סקיילביליות האלגוריתם- כלומר ביצועיו ברשתות בגדלים שונים, כדי למצוא חסמים הדוקים יותר עבור גדלים שונים של רשתות.
  - נבחין כי האלגוריתם נבחן על שתי טופולוגיות ספציפיות שאינן כלליות מספיק לדעתנו, ולכן כשלב מקדים ליישומי ברשתות אמיתיות, נדרש לבחון את התנהגותו על מגוון רחב של טופולוגיות רשת שונות בכדי לספק מסקנות כלליות יותר.
  - כפי שצינו קודם לכן, מספר הודעות הבקרה הרב על פי הגדרת האלגוריתם מהווה תקורה משמעותית על הרשת, ועבור רשתות אמיתיות שאינן סימולטיביות יביא להפחתת הביצועים. כפי שסקרנו בעבודות הנוספות בתחום, ישנן שיטות להפחתת הודעות הבקרה וניתן לחקור הטמעת שיטות אלו באלגוריתם המוצע.
- לסיכום, לדעתנו כותבי המאמר ביצעו עבודה טובה ומקיפה מבחינת בחירת המודל, והגדרת האלגוריתם וניתוחו. כמו כן, הצליחו להעביר את עיקרי הדברים בצורה מופשטת וברורה גם עבור האנליזה המתמטית המורכבת. ישנם מספר מקרי קצה שלדעתנו לא פורטו בצורה מספיק טובה כתוצאה מהבחירה לבצע סימולציה נומרית ולכן ישנם חלקים בפרוטוקול שנשארו לא מוגדרים, אנו מאמינים כי מדובר במקרים שאינם מאוד מורכבים, אך הכרחיים לטיפול עבור מימוש של הפרוטוקול בתנאי רשת מציאותיים. על ידי ביצוע סימולציות רשת נרחבות כפי שתיארנו בכיווני המחקר, ניתן לקבל מושג טוב יותר על מידת פוטנציאל המימוש של האלגוריתם ברשתות אמיתיות.

## נספחים

[1]. למטה 1: יהי

$$\Delta \in \left(0, \min_{n \in \mathcal{N}} \left\{ \min_{n' \in \mathcal{N}(n)} |V^*(n) - V^*(n')| \right\}\right)$$

אם מתקיים:

$$\mathcal{K} := \left\{ |V^*(n) - V(n)| \leq \frac{\Delta}{2}, \forall n \in \mathcal{N} \right\}$$

אז החרטה היא 0.

הוכחה:

נניח בשלילה כי קיימים שני משתמשים  $n$  ו  $n'$  כך שמתקיים:

$$V^*(n) > V^*(n'), V(n) < V(n')$$

כאשר מתקיים  $\mathcal{K}$ , נקבל:

$$\begin{aligned} V(n) - V(n') &= (V(n) - V^*(n)) + (V^*(n') - V(n')) + V^*(n) - V^*(n') \\ &> -\frac{\Delta}{2} - \frac{\Delta}{2} + \Delta = 0 \end{aligned}$$

■ בסתירה להנחה.

הלמה מתבססת על כך שסדר הבחירה של המשתמשים לפי האלגוריתם המשערך זהה לסדר הבחירה של האלגוריתם האופטימלי.

[2]. למטה 2:

יהי

$$\alpha := \sum_{n=1}^N \frac{R}{c_n}, \quad B := \frac{8\alpha^2 4^{N_{max}} \log M}{\Delta^2} + \tau_{max} \cdot \lambda$$

לכל

$$M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot 2^{N_{max}}}} \right\rceil, \tau_{max} \geq 0, N_{max} > 0, \lambda > 0$$

מתקיים שלאחר שליחת  $B$  חבילות, ההסתברות למאורע  $\bar{\mathcal{K}}$  (שיש חרטה) קטנה מ  $\frac{4N \cdot 2^{N_{max}}}{M}$ .

הוכחה:

הוכחת הלמה מתפרשת על פני 3 עמודים במאמר, נפרט את המעברים העיקריים בה:

במהלך ההוכחה נשתמש באי שוויון שהוכח במאמר באזכור [40]:

$$|V^*(n) - V(n)| \leq \alpha \max_{n' \in \mathcal{N}} \sigma(P_{n'}, \theta_{n'})$$

כאשר

$$\alpha = \sum_{n=1}^N \frac{R}{c_n}, \quad \sigma(P_{n'}, \theta_{n'}) := \frac{1}{2} \sum_{S \subseteq N(n)} |Pr(S|n) - \theta(S|n)|$$

מכאן שאם מתקיים

$$\alpha \max_{n' \in \mathcal{N}} \sigma(P_{n'}, \theta_{n'}) \leq \frac{\Delta}{2}$$

נקבל לפי למה 1 שאין חרטה.

מכאן שאם יש חרטה, בהכרח מתקיים  $\alpha \max_{n' \in \mathcal{N}} \sigma(P_{n'}, \theta_{n'}) > \frac{\Delta}{2}$ .

ההסתברות שמתקיים מאורע זה הינה:

$$Pr\left(\alpha \max_{n' \in \mathcal{N}} \sigma(P_{n'}, \theta_{n'}) > \frac{\Delta}{2}\right) \leq \sum_{n=1}^N \sum_{S \subseteq N(n)} B(n)$$

כאשר

$$B(n) = Pr\left(\underbrace{Pr(S|n) - \theta(S|n)}_{c_1(n)} < -\frac{\Delta}{|2^{N(n)}|\alpha}\right) + Pr\left(\underbrace{Pr(S|n) - \theta(S|n)}_{c_2(n)} > \frac{\Delta}{|2^{N(n)}|\alpha}\right)$$

חסם שהוכח עבור  $Pr(C_1(n))$ :

$$Pr(C_1(n)) \leq \frac{1}{M} + \frac{1}{M^4}$$

והוא נכון לאחר שמספר ההודעות שנשלח הינו  $\lambda \cdot \tau_{max} + \frac{8\alpha^2 4^{N_{max}} \log M}{\Delta^2}$   
חסם שהוכח עבור  $Pr(C_2(n))$ :

$$Pr(C_2(n)) \leq \frac{1}{M^2} + \frac{1}{M^4}$$

לבסוף נקבל:

$$\begin{aligned} Pr\left(\alpha \max_{n' \in \mathcal{N}} \sigma(P_{n'}, \theta_{n'}) > \frac{\Delta}{2}\right) &\leq \sum_{n=1}^N \sum_{S \subseteq N(n)} B(n) \leq \sum_{n=1}^N \sum_{S \subseteq N(n)} \frac{1}{M} + \frac{1}{M^2} + \frac{2}{M^4} \\ &\leq \frac{4N2^{N_{max}}}{M} \end{aligned}$$

כלומר

$$Pr(\bar{\mathcal{K}}) \leq \frac{4N2^{N_{max}}}{M}$$

■

[3]. טענה 1:  
לכל

$$M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot 2^{N_{max}}}} \right\rceil, \tau_{max} \geq 0, N_{max} > 0, \lambda > 0$$

החרטה של TSOR חסומה מלמעלה ע"י:

$$O(N^3 4^{N_{max}} \log M + N \tau_{max} \cdot \lambda)$$

הוכחה:

לפי למה 1, אם מתקיים מאורע  $\bar{\mathcal{K}}$  אז יש חרטה. נסמן ב  $\mathcal{F}_m$  את ההיסטוריה לאחר ניתוב החבילה  $m$  ונסמן ב  $\Delta_{max}$  את החרטה המקסימלי בשליחת חבילה אחת. מכיוון שמחיר שליחת חבילה  $c_n$  לכל משתמש  $n$ , נקבל  $\Delta_{max} = O(N)$ .  
הינו  $c_n$  לכל משתמש  $n$ , נקבל  $\Delta_{max} = O(N)$ .  
מכאן שניתן לבטא את החרטה כך:

$$\begin{aligned} \mathcal{R} &\leq \sum_{m=1}^M Pr(\bar{\mathcal{K}} | \mathcal{F}_{m-1}) \Delta_{max} \leq B \cdot \Delta_{max} + \sum_{m=B+1}^M Pr(\bar{\mathcal{K}} | \mathcal{F}_{m-1}) \Delta_{max} \\ &\leq B \cdot \Delta_{max} + \sum_{m=B+1}^M \frac{4N2^{N_{max}}}{M} \Delta_{max} \leq B \cdot \Delta_{max} + 4N2^{N_{max}} \Delta_{max} \\ &= O(N^3 4^{N_{max}} \log M + N \tau_{max} \cdot \lambda) \end{aligned}$$

כאשר ההצבה בסכום נובעת מלמה 2. ■

[4]. מסקנה 1:  
אם מתקיים

$$\tau_{max} \cdot \lambda \leq N^2 4^{N_{max}} \log M$$

אז החרטה הממוצעת חסומה ע"י:

$$O(N^3 4^{N_{max}} \log M)$$

הוכחה:

עבור

$$\tau_{max} \cdot \lambda \leq N^2 4^{N_{max}}$$

מתקיים:

$$\begin{aligned} O(N^3 4^{N_{max}} \log M + N \tau_{max} \cdot \lambda) &\leq O(N^3 4^{N_{max}} \log M + N^3 4^{N_{max}} \log M) \\ &= O(N^3 4^{N_{max}} \log M) \end{aligned}$$

■

- [6] C. Lott and D. Teneketzis, "Stochastic routing in ad-hoc networks," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 52–70, Jan. 2006.
- [7] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance," *IEEE Trans. Mobile Comput.*, vol. 2, no. 4, pp. 349–365, Oct. 2003.
- [8] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 133–144, Oct. 2005.
- [11] A. Bhorkar, M. Naghshvar, and T. Javidi, "Opportunistic routing with congestion diversity in wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1167–1180, Apr. 2016.
- [17] L. Hai, H. Wang, J. Wang, and Z. Tang, "HCOR: A high-throughput coding-aware opportunistic routing for inter-flow network coding in wireless mesh networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, p. 148, Dec. 2014.
- [18] K. Chung, Y.-C. Chou, and W. Liao, "CAOR: Coding-aware opportunistic routing in wireless ad hoc networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 136–140.
- [21] Y.-J. Lin, C.-C. Huang, and J.-L. Huang, "PipelineOR: A pipelined opportunistic routing protocol with network coding in wireless mesh networks," in *Proc. IEEE 71st Veh. Technol. Conf.*, Jun. 2010, pp. 1–5.
- [26] A. Khreishah, I. M. Khalil, and J. Wu, "Universal opportunistic routing scheme using network coding," in *Proc. 9th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Jun. 2012, pp. 353–361.
- [28] A. A. Bhorkar, M. Naghshvar, T. Javidi, and B. D. Rao, "Adaptive opportunistic routing for wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 243–256, Feb. 2012.
- [31] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2011, pp. 2249–2257.
- [32] S. Wang and W. Chen, "Thompson sampling for combinatorial semi-bandits," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 5114–5122.
- [39] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari, "Tight regret bounds for stochastic combinatorial semi-bandits," in *Proc. Artif. Intell. Statist. (AISTATS)*, 2015, pp. 535–543.
- [40] T. Javidi and D. Teneketzis, "Sensitivity analysis of an optimal routing policy in an ad hoc wireless network," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1303–1316, Aug. 2004.

## הוראות להרצת הסימולציה

בשלב הראשון יש לחלץ את תיקיית הפרויקט מקובץ zip המצורף.

לאחר מכן יש לפתוח את סביבת העבודה של TONET++ , ולבחור file (1) ואז Open Projects from file system (2) .  
בחלון שנפתח יש להקיש על Directory (3), לנווט אל תיקיית הפרויקט שחילצנו ולבסוף להקיש finish (4).

תיקיית הפרויקט אמורה להופיע כעת ברשימת הפרויקטים בסביבת העבודה (5).  
נווט אל קובץ ה omnetpp.ini בתיקייה ונקיש עליו פעמיים (6).

לאחר שהקובץ נפתח נקיש על החץ הירוק בשורת הפקודות Run (7), אם יקפוץ חלון נקיש Yes.

כעת ייפתח מסך הסימולציה בו נוכל לבחור איזה רשת להריץ ואת הפרמטרים שלה. בדוגמא זו נפעיל את רשת 1 עם הסתברויות שגיאא בלינקים per=0.1 ואלגוריתם TSOR (8).

נקיש על OK וטופולוגית הרשת תופיע (9).

כעת ניתן להריץ את הסימולציה ע"י לחיצה על Run (10).

בסוף הסימולציה הסטטיסטיקות יישמרו בתיקיית results. ניתן לייצא אותן לקובץ csv ולנתח עם סקריפט הפייתון שצירפנו.

סיכום התהליך מוצג בתמונות הבאות:

