תקשורת באינטרנט - אביב תשפ"ג

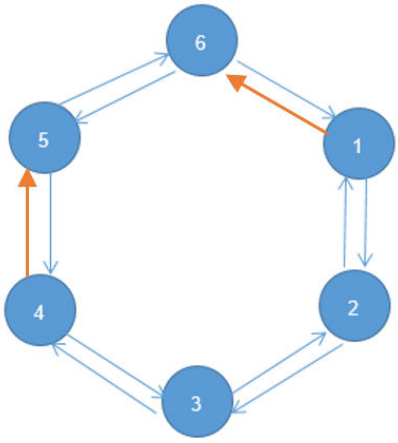
תרגיל בית 5

תאריך הגשה: **6.7.2023**
האחראי על התרגיל: נדב, דוא"ל **nadav.adir@campus.technion.ac.il**
נמקו היטב אך בקצרה את כל תשובותיכם. תשובה לא מנומקת לא תזכה במלוא הניקוד!
ההגשה הינה בזוגות בלבד , אלקטרונית באתר בפורמט **Zip**. יש לצרף קובץ **PDF** עם התשובות והגרפים וכן את קובץ ה **Python** שמכיל את ההרצה האחרונה שביצעתם.

שאלה 1 - MAX/MIN FAIRNESS

[30 נקודות]

נתונה הטופולוגיה הבאה:

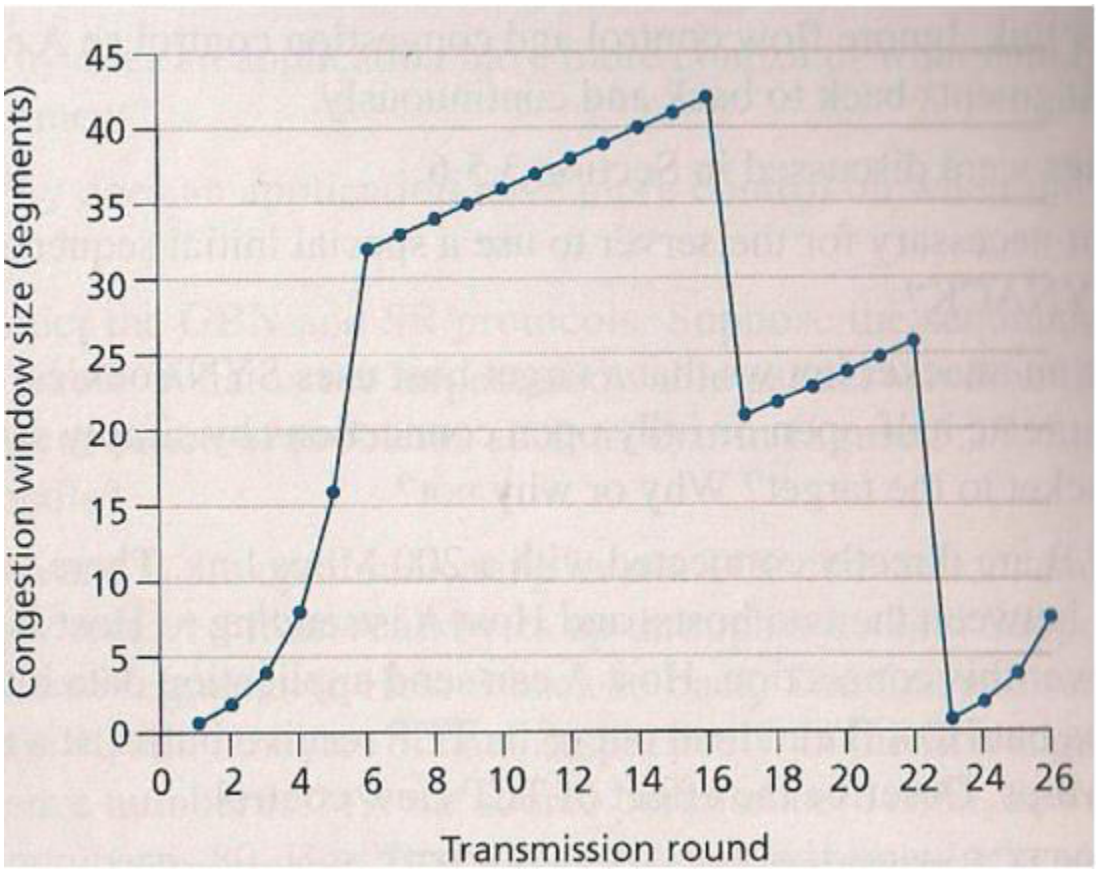


הקווים הם חד כוונים.
קיבולת הקווים הכחולים 1 Mbps, קיבולת הקווים האדומים 10 Mbps.
ישנם שני flows מכל אחת מהצמתים 1, 2, 3, 4, 5 לצומת 6, אחד בכל כיוון (עם ונגד כוון השעון).
בצורה דומה יש שני flows מצמתים 2,3,4 לצומת 5, ושני flows מצומת 3 לצומת 4.
מצאו את זרימת max/min fair ברשת, הראו את ה-bottleneck link בכל שלב.
רמז: ניתן להתייחס לבעיה כשתי בעיות בלתי תלויות.

שאלה 2 - TCP RENO CONGESTION CONTROL

[40 נקודות]

התייחסו לגרף הבא שמתאר את גודל החלון ב- TCP כפונקציה של הזמן (מחזור שליחה). הגרף מתייחס לפרוטוקול TCP Reno. בכל השאלות הבאות, נמקו את תשובתכם בקצרה.



1. באילו מחזורי שליחה ה-ssthresh השתנה ולאילו ערכים?
2. מהם אינטרוולי מחזורי השליחה בהם מצב הפעולה של האלגוריתם הוא slow start?
3. נניח שאחרי מחזור השליחה ה - 26 אנו מגלים שחבילה בודדת נאבדה. הדבר מתגלה באמצעות 3 duplicate ACKs . מה יהיה בעקבות כך cwnd ו-sstresh?
4. אחרי מחזור השליחה ה - 16, האם איבוד החבילה התגלה ע"י time out או ע"י שלושה duplicate ACKs
5. אחרי מחזור השליחה ה - 22, האם איבוד החבילה התגלה ע"י time out או ע"י שלושה duplicate ACKs
6. מהם אינטרוולי מחזורי השליחה בהם מוד הפעולה של האלגוריתם הוא congestion avoidance?
7. מהו ה ssthresh ההתחלתי (בתחילת פעולת האלגוריתם)?
8. באיזו מחזור שליחה נשלח סגמנט מספר 100 (הנח שהשליחה התחילה מסגמנט מספר 0)?

שאלה 3 - TCP CONGESTION CONTROL

[30 נקודות]

בתרגול הוזכרו Reno ו-New Reno כפרוטוקולים שמזהים גודש ומנסים לנצל את רוחב הפס הזמין בצורה אופטימלית.

1) כיצד פרוטוקולים אלה מזהים גודש? מה ההבדל בין Reno לבין New Reno?

עיינו ב-RFC 6297 section 2 וענו על השאלות הבאות:

2) כיצד TCP Vegas מזהה גודש ברשת?
3) מדוע כאשר ב-bottleneck link מסוים המשותף למספר flows הרצים מעל TCP עם אלגוריתם למניעת גודש של TCP Vegas ה-throughput גדול יותר ביחס ל-TCP Reno?
4) מה קורה כאשר יש שימוש גם ב-TCP Vegas וגם ב-TCP Reno באותו ה-link?

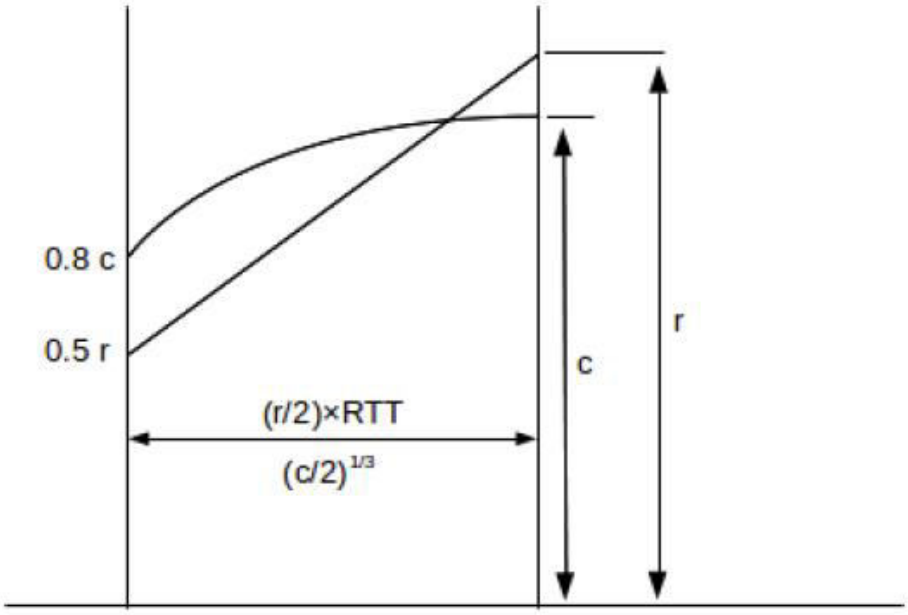
אלגוריתם ל-congestion control פופולרי מאוד שמופץ ב-windows נקרא Compound TCP . עיינו ב- RFC section 3 שמתאר את האלגוריתם.

5) מה הם 2 סוגי החלונות שאלגוריתם זה מתחזק?
6) מה היתרון המרכזי של האלגוריתם על TCP Reno?
7) מה היתרון המרכזי של האלגוריתם על TCP Vegas?

שאלה 4 - TCP CUBIC CONGESTION CONTROL

[בונוס - 15 נק]

הניחו שחיבור TCP RENO מתחרה עם חיבור TCP CUBIC על אותו הלינק (אין תעבורה אחרת על הלינק).
בנוסף הניחו כי כאשר הלינק מגיע למצב של congestion איבוד חבילות מתרחש בשני החיבורים (לא יכול להיות מצב שחבילה הלכה לאיבוד בחיבור אחד ובשני לא).
במצב היציב, חלון הcwnd יראה כך:



One tooth, TCP Cubic v TCP Reno

נסמן ב-**c** את cwnd המקסימלי של חיבור הTCP CUBIC, וב-**r** את cwnd המקסימלי של TCP RENO.
M זו תקרת הרשת, כלומר מתרחש איבוד של חבילה כאשר סכום גדלי החלונות של שני החיבורים מגיע ל-M.
זמן המחזור (הזמן שעובר בין שני איבודים של חבילות) של חיבור TCP RENO הוא r/2 * RTT.
זמן המחזור של חיבור TCP CUBIC הוא $(c/2)^{1/3}$.

א. אם M=200 ו-RTT=50 ms=0.05 sec, הראו כי מגיעים למצב היציב כאשר r=130.4 ו- c = M – r = 69.6.
ב. מצאו את ערכי r, c במצב היציב כאשר M=1000 ו-RTT=50 ms, הסבירו כיצד מצאתם ערך זה.
ב. מצאו את ערכי r, c במצב היציב כאשר M=1000 ו-RTT=100 ms, הסבירו כיצד מצאתם ערך זה.

* יש להתעלם מתכונת הTCP Friendly של TCP CUBIC.
* לסעיפים ב, ג מומלץ להיעזר בWolfram או תכנה דומה.

שאלה 5 – Mininet TCP Cubic Lab – לא להגשה עקב העיכובים במהלך הסמסטר

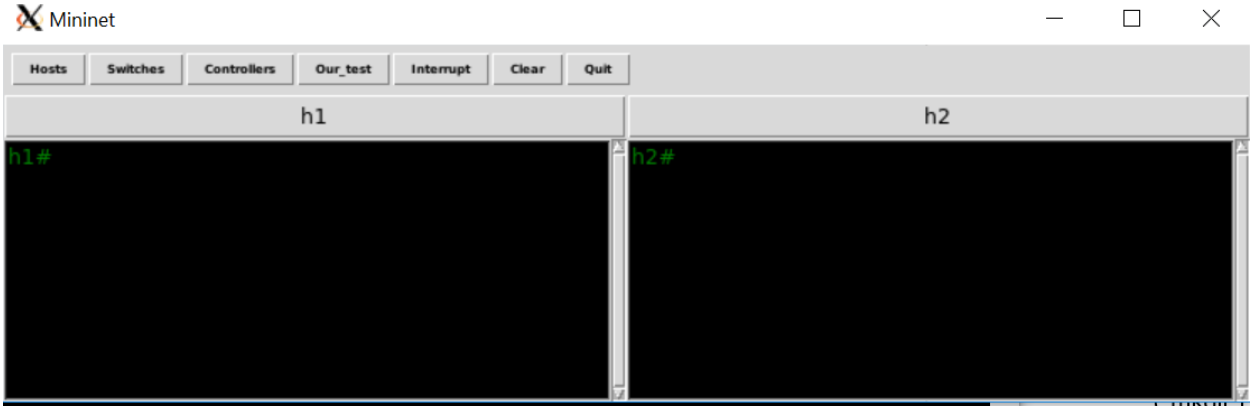
[0 נקודות]

במעבדה זו נחקור את פרוטוקול TCP-Cubic, שמהווה את ליבת ה Transport ברשת האינטרנט. נעשה זאת באמצעות בנייה ב Mininet של טופולוגיית רשת ונבחן את התנהגות TCP תחת תנאים שונים כגון: Traffic Congestion וכן שיהוי ואיבוד תעבורה. הבחינה תהיה באמצעות הכלי iperf3 שהוא למעשה כלי נפוץ כיום לניטור תעבורת רשת.

קיראו על הכלי והאופציות שלו ב https://linux.die.net/man/1/iperf וכן ב:
http://software.es.net/iperf/invoking.html#iperf3-manual-page

א. הכנות:

1. הפעילו את המכונה הוירטואלית מהתרגיל הקודם וודאו חיבור ל Internet
2. התקינו iperf3 באמצעות sudo apt-get install iperf3
3. ודאו חיבור Host-only Adapter והפעילו שוב המכונה הוירטואלית
4. צרו מחיצה בשם consolesOutput תחת /home/mininet/mininet/examples (השתמשו בפקודה mkdir)
5. פתחו ssh session (עם X11 forwarding enable) והריצו את xming. העתיקו את קובץ ה script המצורף לתרגיל tcp-cubic-test.py אל ספריית mininet/examples שנמצאת תחת מחיצת ה home שלכם. את ההעתקה אפשר לעשות באמצעות תוכנת WinSCP .
6. הריצו את tcp-cubic-test.py באופן הבא:
   mininet@mininet-vm:~/mininet/examples$ sudo python ./tcp-cubic-test.py
7. אם הכל כשורה תקבלו ב xming את החלון הבא:



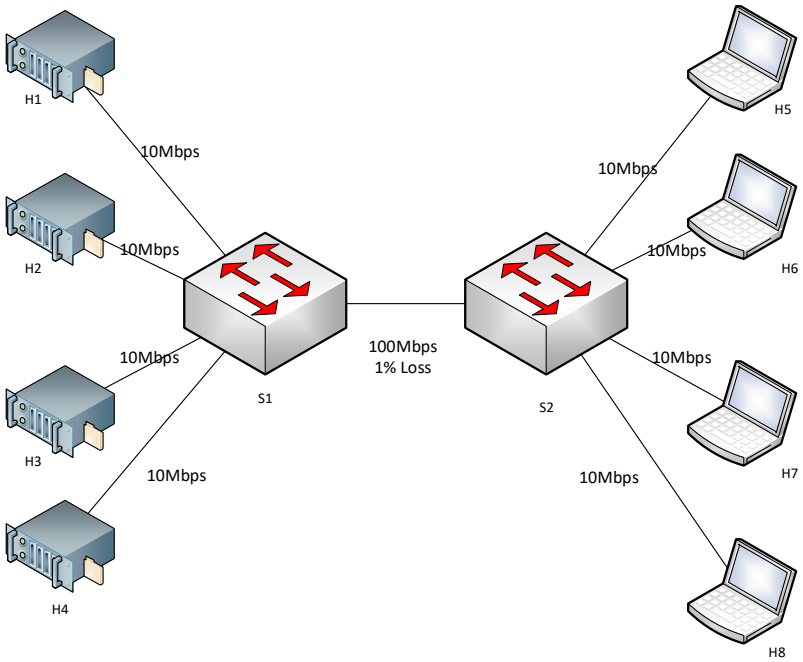
8. בקובץ tcp-cubic-test.py ישנן דוגמאות בליווי הערות ו TODO כיצד יש לבנות טופולוגיה ב mininet וכן כיצד להריץ את iperf3 לאחר לחיצה על הכפתור Our_test . תוצאת ההרצה בכל host$_x$ (x=n/2..n-1) תכתב לקובץ /home/mininet/mininet/examples/consolesOutput/host$_x$.
9. כל הרצה של iperf צריכה להתבצע בתנאים הבאים:
   a. אורך ההרצה: 60 שניות
   b. אינטרוול: 1 שנייה
10. יש לשמור תוצאות כל הרצה

שימו לב: יש ללחוץ על Quit בסיום כל הרצה כדי לגרום לפירוק תקין של טופולוגית mininet .

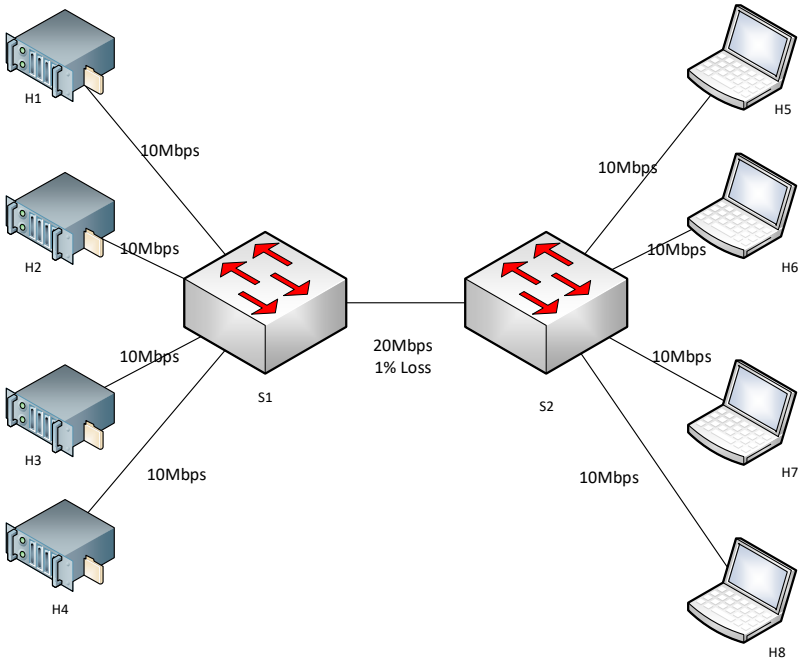
11. כדי להוריד link בטרמינל של switch השתמשו ב "sudo ifconfig eth1 down" כדי להעלות חזרה "sudo ifconfig eth1 up".
בכל ההרצות נשתמש בטופולוגית DoubleSwitch

ב. הרצות:

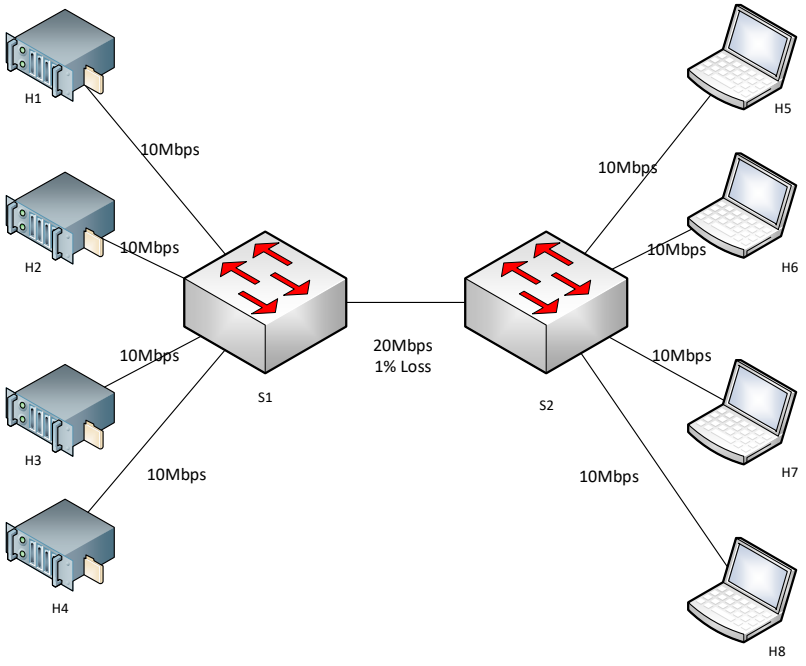
a. Non Congested Link



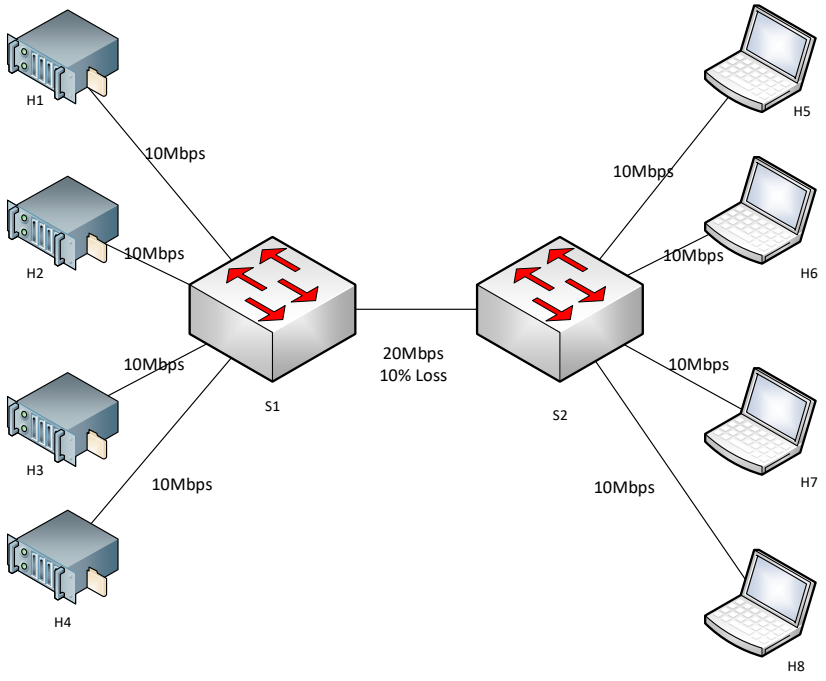
b. Non Congested Link בשילוב הפלה ל 2 שניות של הקו בין s1 ל s2 בשניה ה 20 ובשניה ה 40.
רוחב הפס הוא 100Mbps (בתמונה כתוב 20 בטעות)



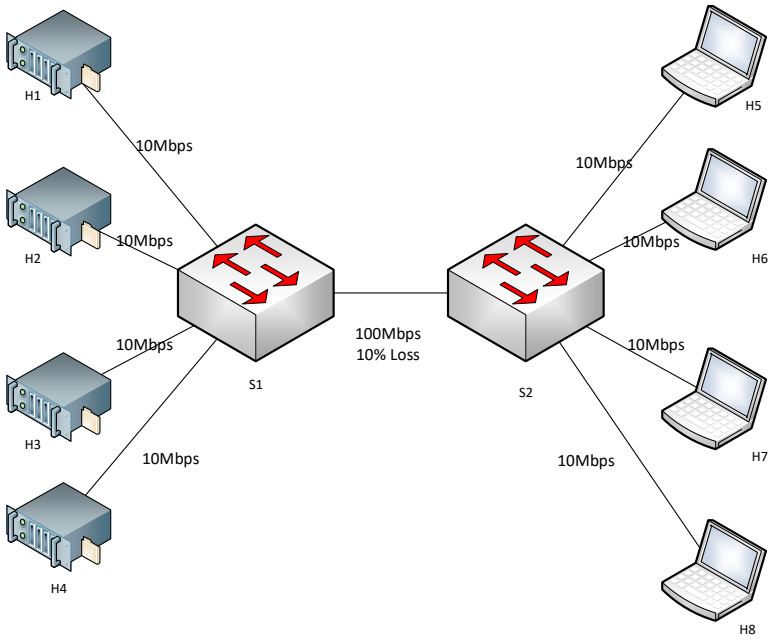
Congested Link .c



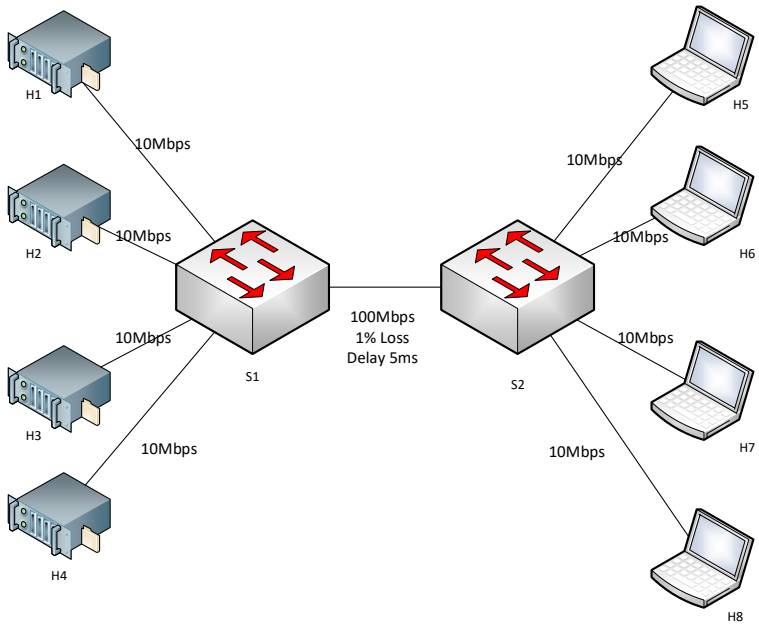
Loss Rate=10% עם Congested Link .d



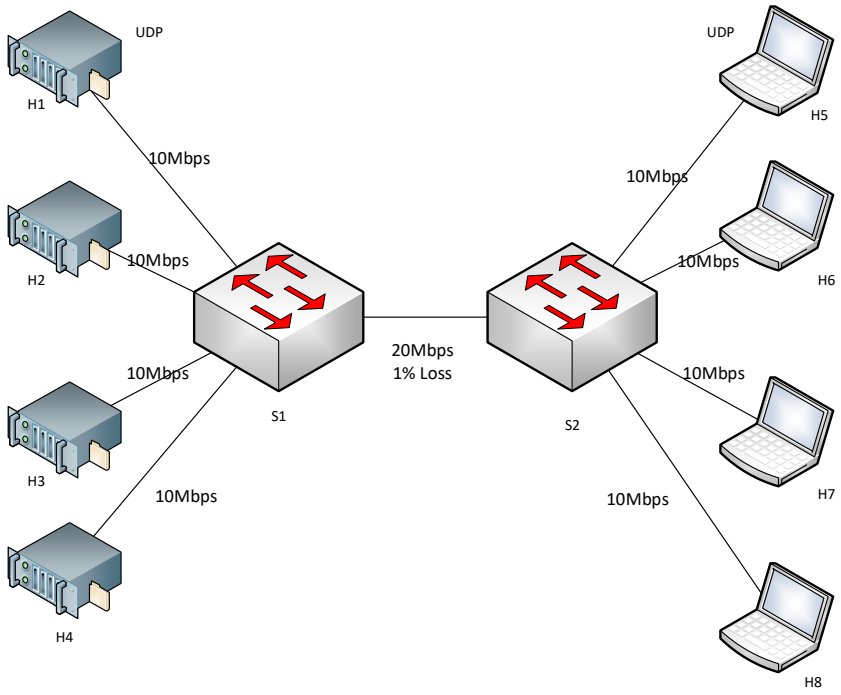
Loss Rate=10% .e



Delay=5ms .f



g. Congested Link עם זוג אחד שמריצים UDP ו 3 זוגות עם TCP



ג. לכל אחת מההרצות נדרשים הגרפים הבאים (בעזרת Excel) לכל אחד מהפרמטרים :

a. השתנות Bandwidth כתלות בזמן

b. השתנות cwnd כתלות בזמן

c. השתנות ssthreshold כתלות בזמן (https://tools.ietf.org/html/rfc8312#section-4.5)

שימו לב: אם הפרוטוקול שמר על fairness באותה הרצה – מספיק לשרטט גרף של host אחד, אחרת את כולם.