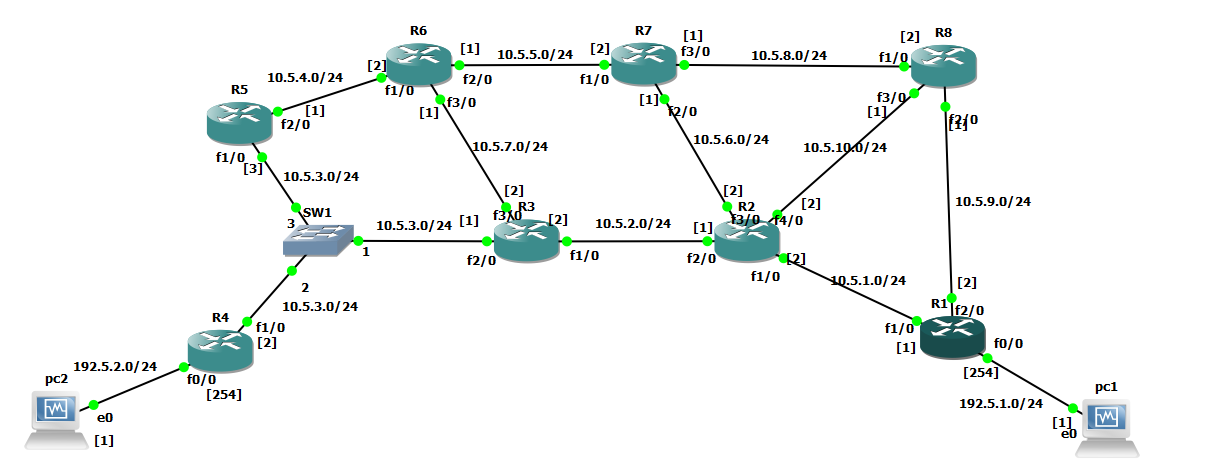
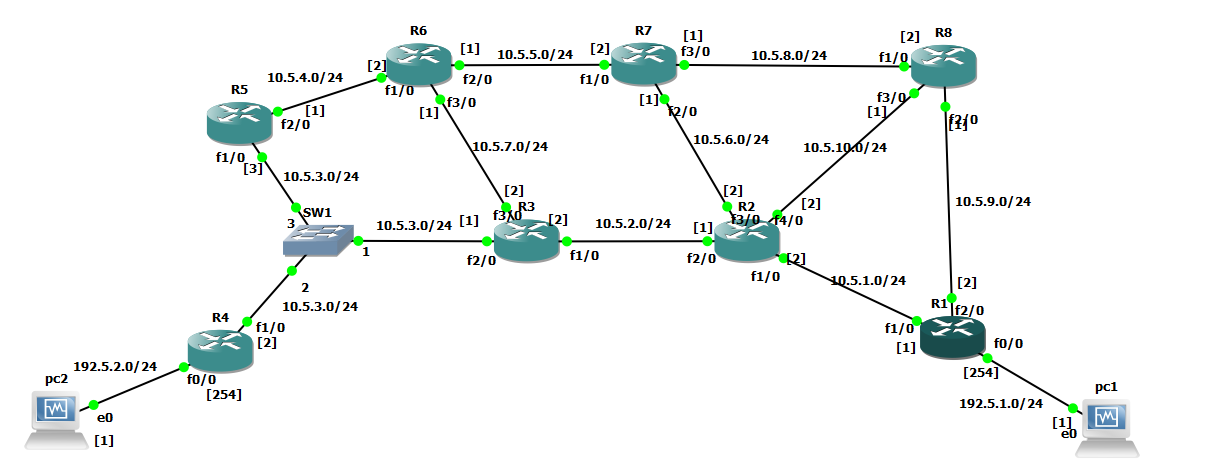
**Lab 8**

Shir Moshe - 318492667

Nadav Biran - 316468834

1. **Network Configuration – Topology 11 (base on topology 9)**

|  |  |
| --- | --- |
| Pc1 | *route add default gw 192.5.1.254* |
| Pc2 | *route add default gw 192.5.2.254* |
|  | *configure terminal*  *interface FastEthernet0/0*  *shutdown*  *no ip route 192.168.2.0 255.255.255.0*  *exit*  *configure terminal*  *interface FastEthernet0/0*  *ip address 192.5.1.254 255.255.255.0*  *no shutdown*  *end* |
|  | configure terminal  router ospf 1  network 192.5.1.0 0.0.0.255 area 1  passive-interface FastEthernet 0/0  interface FastEthernet 0/0  ip address 192.5.1.254 255.255.255.0  no shutdown  end |
|  | configure terminal  router ospf 1  network 192.5.2.0 0.0.0.255 area 1  passive-interface FastEthernet 0/0  interface FastEthernet 0/0  ip address 192.5.2.254 255.255.255.0  no shutdown  end |

1. **Transmitting data with UDP**
   1. 



* 1. Start Wireshark on R3 – R2, set filter to display packets that include :

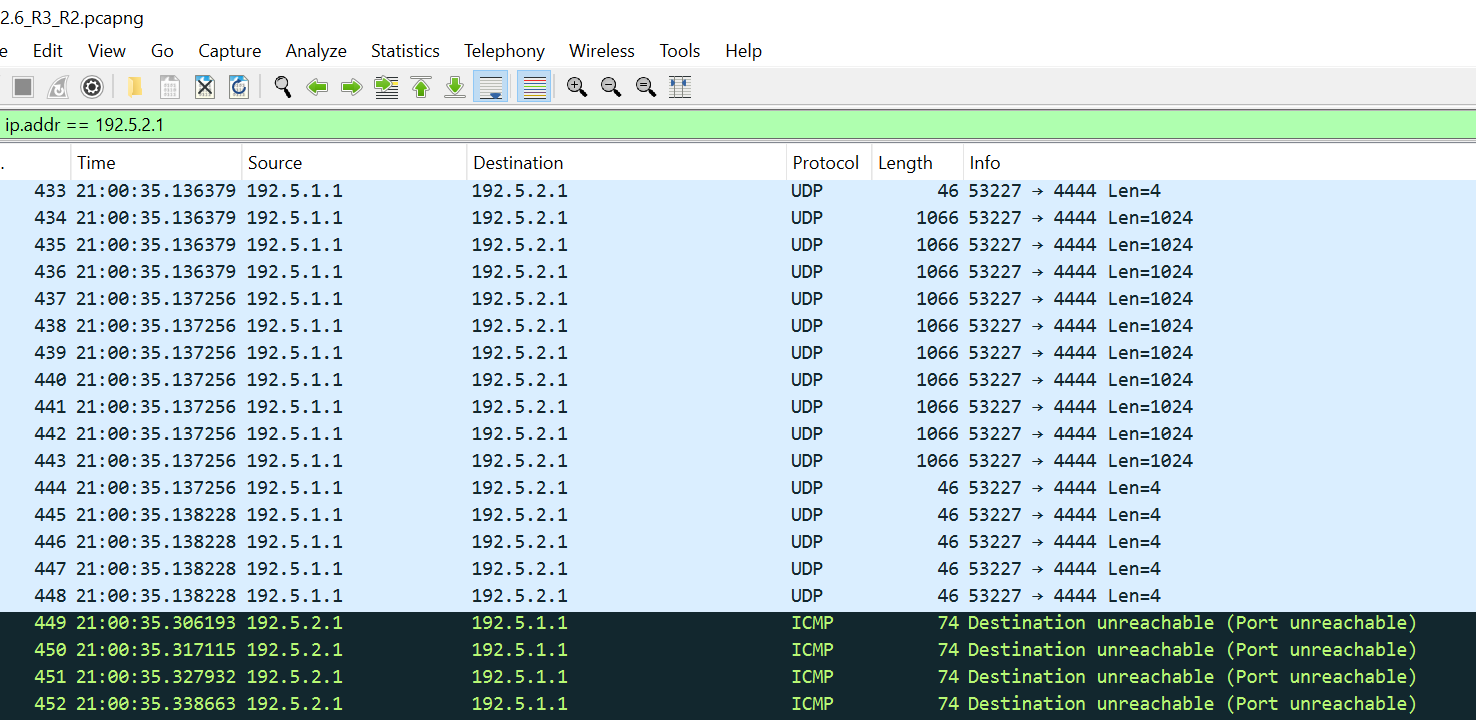
|  |
| --- |
| *Ip == 192.5.2.1* |

* 1. **On PC2**, start a ttcp receiver that receives UDP traffic with the following command:

|  |
| --- |
| *ttcp –r –l1024 –n10 –p4444 –u* |

* 1. **On PC1**, start a ttcp sender that transmits UDP traffic by typing:

|  |
| --- |
| *ttcp –t –l1024 –n10 –p4444 –u 192.5.2.1* |

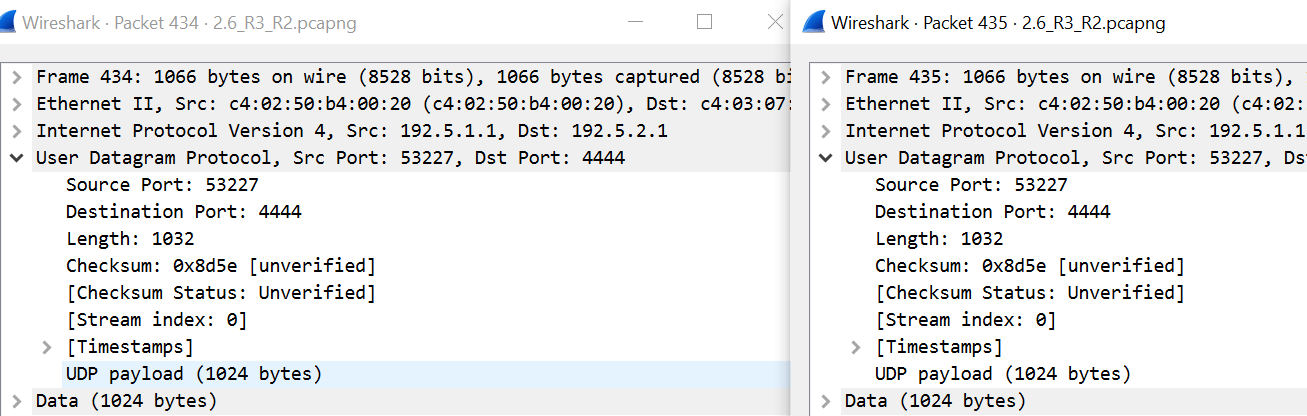
* 1. Stop the Wireshark capture and save all the captured traffic for the exercise below.
  2. 

גודל ה payload = 1024 bytes, נשלחו בסה"כ 10 (לא כולל הפתיח/סוף) הודעות UDP.

לכל datagram נשלחה הודעה אחת.



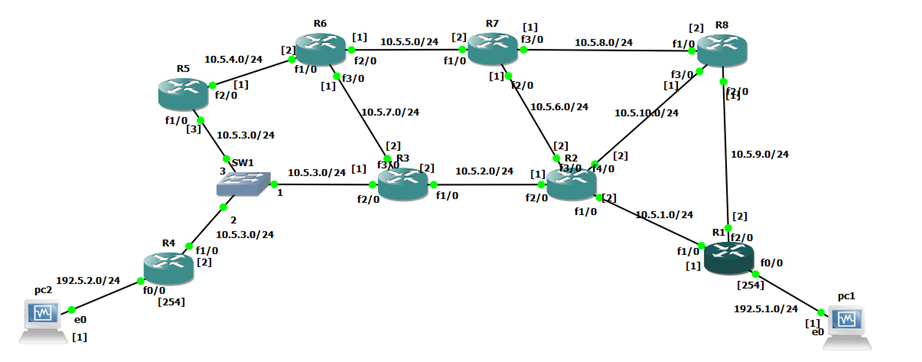
כל ההדרים אותו דבר : 2 פקטות לדוגמא:



הפקודה ttcp מגרילה מספר.

סה"כ יש בהודעה 1066 byte , מתוכם יש 1024 byte של data.  
כלומר כל ההדר (ip, ethernet, udp) מוסיפים

ויש 10 הודעות כאלה זהות לכל המידע.

1. **Transmitting data with TCP**
   1. 



* 1. Start Wireshark on R3 – R2, set filter to display packets that include:

|  |
| --- |
| *Ip == 192.5.2.1* |

* 1. **On PC2**, start a ttcp receiver that receives UDP traffic with the following command:

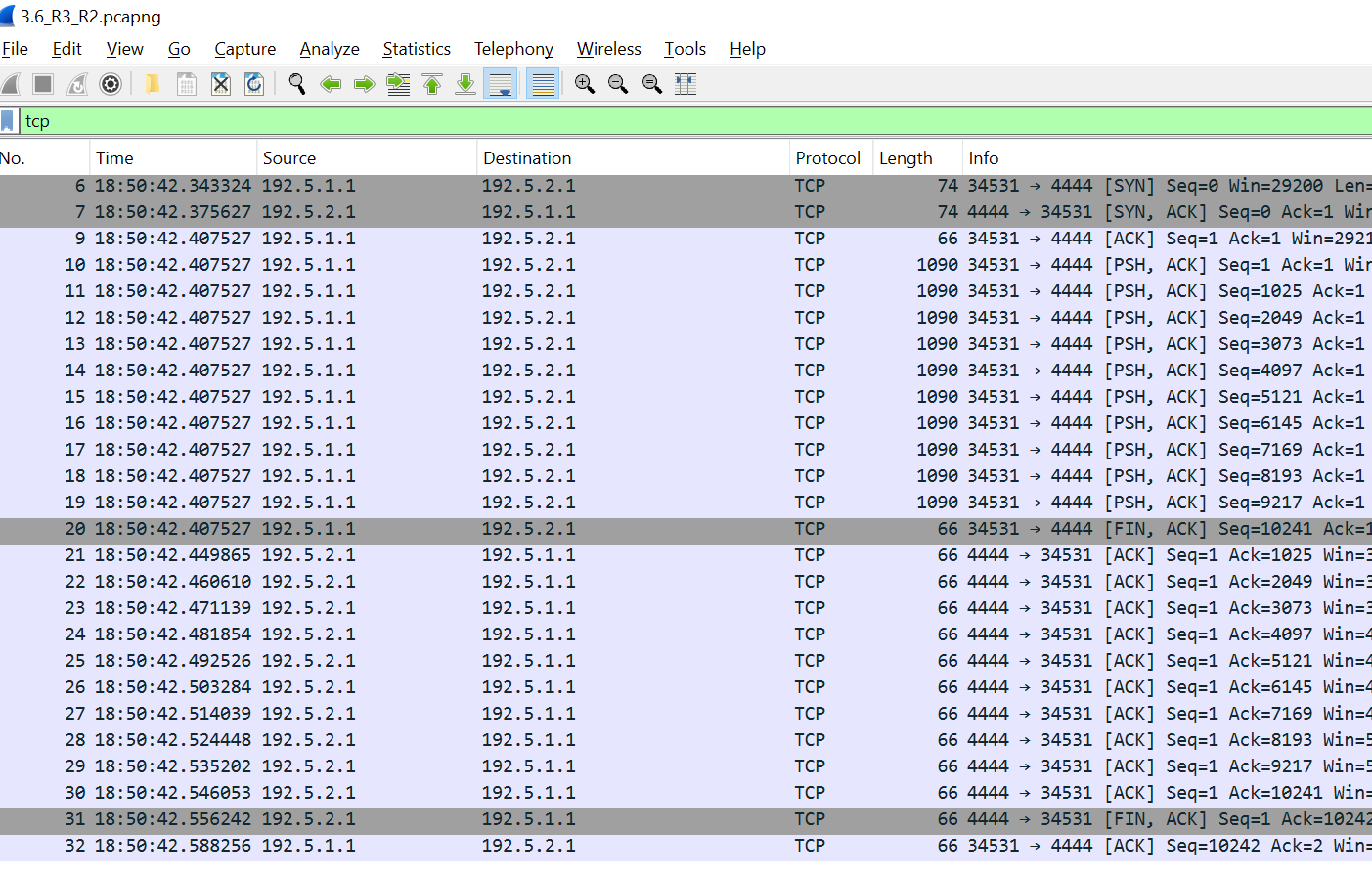
|  |
| --- |
| *ttcp –r –l1024 –n10 –p4444* |

* 1. **On PC1**, start a ttcp sender that transmits UDP traffic by typing:

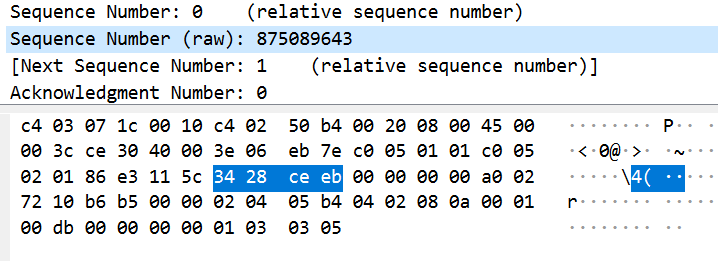
|  |
| --- |
| *ttcp –t –l1024 –n10 –p4444 –D 192.5.2.1* |

* 1. Stop the Wireshark capture and save all the captured traffic for the exercise below.

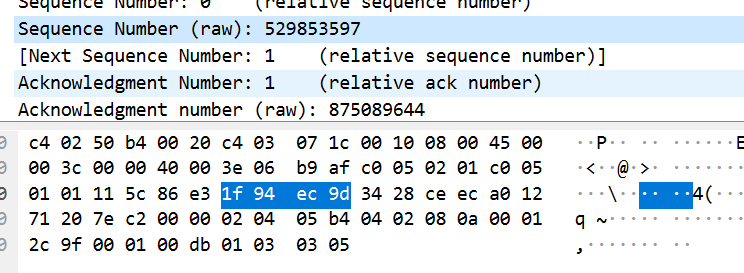
3 לחיצת ידיים, 10 פקטות מידע, ועוד 10 פקטות ACK ועוד 2 של FIN-ACK לסגירה TCP נשלחו



* 1. Sender – initial sequence number = 34 28 ce eb = 875089643



Receiver - initial sequence number = 1f 94 ec 9d = 529853597



The rage:

* 1. sequence number = 875089644

זה לא 0 בגלל שהתחלנו מ ISN שהגרלנו בהודעת SYN הראשונה ואז התקדמנו בהתאם לכמות הביייטים שנשלחו בחבילה

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. 1. sender – SIN

השולח שולח הודעה שהוא רוצה להתחבר למקבל

2. receiver – SIN ACK

המקבל מאשר את הבקשה מהשולח, ושולח בקשת התחברות מעצמו

3. sender - ACK

השולח שולח אישור על הצעת החברות מהמקבל

* 1. packets do not carry a payload:
  2. flag:  
     SYN – בקשת התחברות

ACK – אישור על קבלת הודעה

PSH – להגיד למקבל שיעביר את המידע ישר לאפליקציה

Table

Description automatically generatedFIN - בקשת התנתקות

* 1. overhead:

* 1. sender:

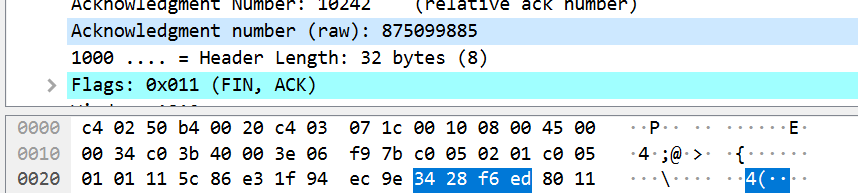
התחלנו ב 875,089,643 והגענו ל 875,099,885 – ההפרש: 10,242.

המספר הסידורי מצביע בכל פעם על הבייט הבא ברצף ביייטי המידע שצריכים להישלח

נשים לב שהודעות SYN,FIN נספרות כל אחת כ 1 בייט מרצף הדתא.

בהתחלה מצביע על הבייט הראשון, שהוא SYN

אז אחרי הודעת SYN המספר יצביע על הבייט הבא, כלומר יעלה באחד

אחרי הודעת דתא, שנשלחו 1024 בייטים המספר יעלה ב1024 ובסוף להודעת הפין יעלה באחד שוב

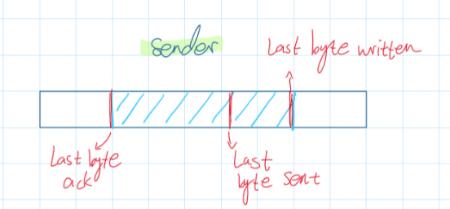
Receiver:

המקבל אין לו דתא לשלוח, אז הוא מגריל מספר בהתחלה, ואז שולח SYN ומעלה ב 1 בייט את המספר שלו, ואז שולח אקים שלא מעלים כלום ובסוף שולח FIN שמסיים את הכל, אז סה"כ רק עלינו במספר אחד.

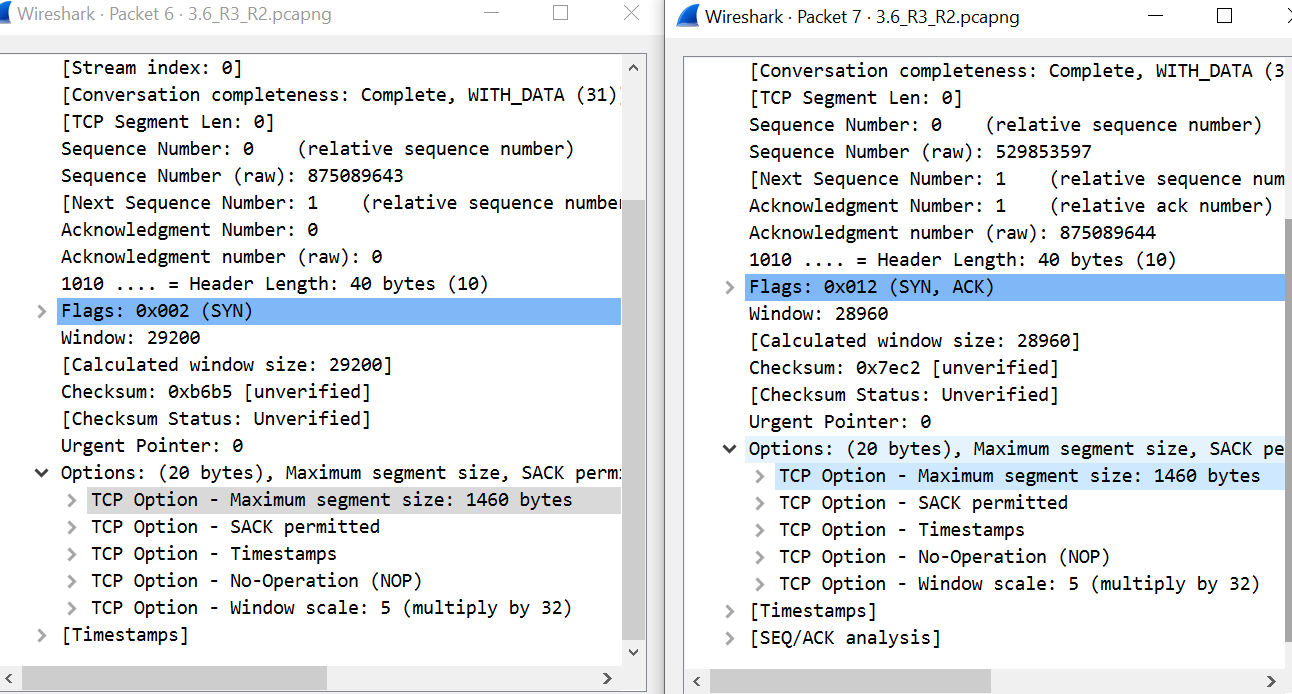


* 1. Window:

Window- כל אחד מפרסם את גודל החלון שנשאר לו, כלומר כמה מקום עוד יש לו בבאפר לקבל מידע. (עד שלא יגיע ACK החלון לא יזוז ולאט לאט יגמרו החבילות שניתן לשלוח בחלון)



* 1. בשלב הלחיצת ידיים שולחים כל אחד מהמחשבים את גודל הסיגמנט שהם מוכנים לקבל בכל חבילת TCP.



1. **Comparing basic UDP and TCP data transfers**

ב udp – 420

ב tcp - 1676

כלומר ב TCP יש יות מידע מסביב

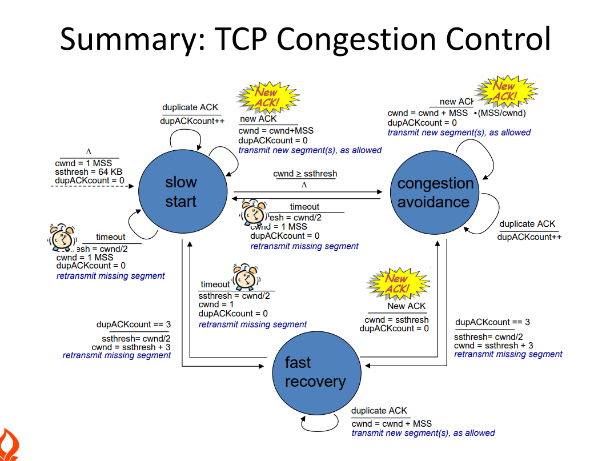
|  |  |
| --- | --- |
| Udp – 1066  אורך ה datagram הוא 1032 בייטים:   1. Source port – 2 2. Destination port – 2 3. Length – 2 4. Checksum- 2 5. Payload – 1024 | Tcp – 1090  אורך ה segment 1056 בייטים:   1. Source port – 2 2. Destination port – 2 3. Sequence number – 4 4. Ack – 4 5. Header length – 1 6. Flag - 1 7. Window – 2 8. Checksum – 2 9. Urgent pointer – 2 10. Options – 12 11. Payload – 1024 |

* 1. Flow control

TCP:

Flow control זה מה שקובע את גודל הפקטות ששולחים, מושפע מהגודל חלון, ונקבע בהתאם אליו כדי למנוע גודש על הערוץ, או גודש על הבאפר של המקבל.

משתמש ב slow start, congestion avoideance, fast retransmit

Slow start – מעלה בתחילה את גודל החלון בצורה אקספוננציאלית על כל ACK שמגיע עד שיש התנגשות או שמגיע ל טרש הולד (איזה גבול עליון של גודל חלון)

Congestion avoidance – אם הגענו לטרש הולד, אז נעבור לעליה עדינה בצורה ליניארית

Fast recovery – אם הייתה התנגשות, נגיע למצב זה בו נקטין את החלון בחצי, ואז נחזור לעליה ליניארית.

UDP:

אין לו flow control, אז הוא פשוט שולח הודעות לפי ה MTU – גודל מקסימלי של הודעות בהינתן רשת מסוימת. מחלק מראש את כל החבילות לגודל הזה.

**TCP and UDP (Part B): Real world connection management**

1. **TCP Fairness and advanced Wireshark statistics**
   1. The first PC - **PC\_RCV** will be used to receive two TCP streams.

The two other PCs - **PC\_TX1, PC\_TX2** are used to send them.

* 1. Copy ttcp to your desktop of all the PCs.
  2. Check the IP addresses of all the PCs you are using. Make a screen capture of the IP address using ifconfig.
  3. On **PC\_RCV** start Wireshark with capture filter exclusively for PC\_TX1 and PC\_TX2.
  4. On **PC\_RVC** run two separate terminals and change directory (cd) to the Desktop folder (where you put the file).
  5. Execute the "chmod 777 ttcp" if necessary (לתת הרשאות)
  6. On PC\_RVC run ttcp in listening mode on each terminal, as follows:

./ttcp –r –l1024 –n200000 –p4444

./ttcp –r –l1024 –n200000 –p4445

* 1. On PC\_TX1 run:

./ttcp –t –l1024 –n200000 –p4444 IP\_of\_PC\_RCV

* 1. On PC\_TX2, wait 3 seconds and run:

./ttcp –t –l1024 –n200000 –p4445 IP\_of\_PC\_RCV

* 1. Stop capture traffic and generate the following graphs:
     1. An IO Graph, where the download stream from PC\_TX1 is green and the download stream from PC\_TX2 is blue
     2. Two Stevens graphs (one for each download stream).
     3. Create graphs as described in the appendix.