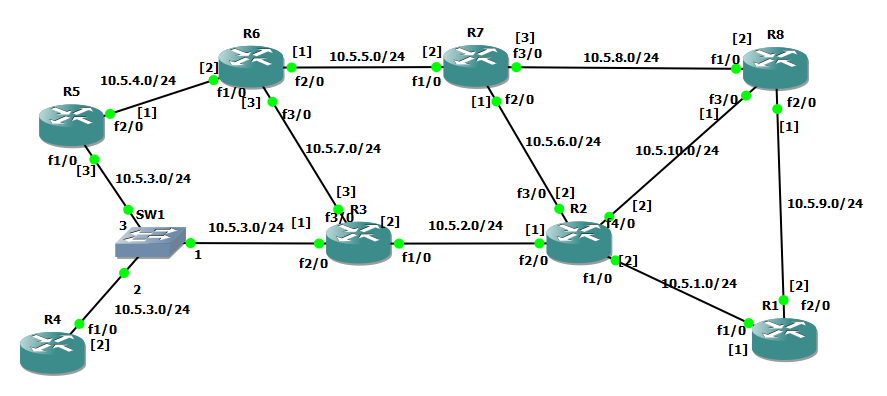
**LAB 7 - Intra AS Routing: RIP**

Shir Moshe - 318492667

Nadav Biran - 316468834

15%

85%

**1.Topology 7 Configuration – Topology 7**

R8

|  |  |  |
| --- | --- | --- |
| *configure terminal*  *interface FastEthernet1/0*  *ip address 10.5.8.2 255.255.255.0*  *no shutdown*  *end* | *configure terminal*  *interface FastEthernet2/0*  *ip address 10.5.9.1 255.255.255.0*  *no shutdown*  *end* | *configure terminal*  *interface FastEthernet3/0*  *ip address 10.5.10.1 255.255.255.0*  *no shutdown*  *end* |

R1

|  |
| --- |
| *configure terminal*  *interface FastEthernet2/0*  *ip address 10.5.9.2 255.255.255.0*  *no shutdown*  *end* |

R2

|  |
| --- |
| *configure terminal*  *interface FastEthernet4/0*  *ip address 10.5.10.2 255.255.255.0*  *no shutdown*  *end* |

R7

|  |
| --- |
| *configure terminal*  *interface FastEthernet3/0*  *ip address 10.5.8.1 255.255.255.0*  *no shutdown*  *end* |

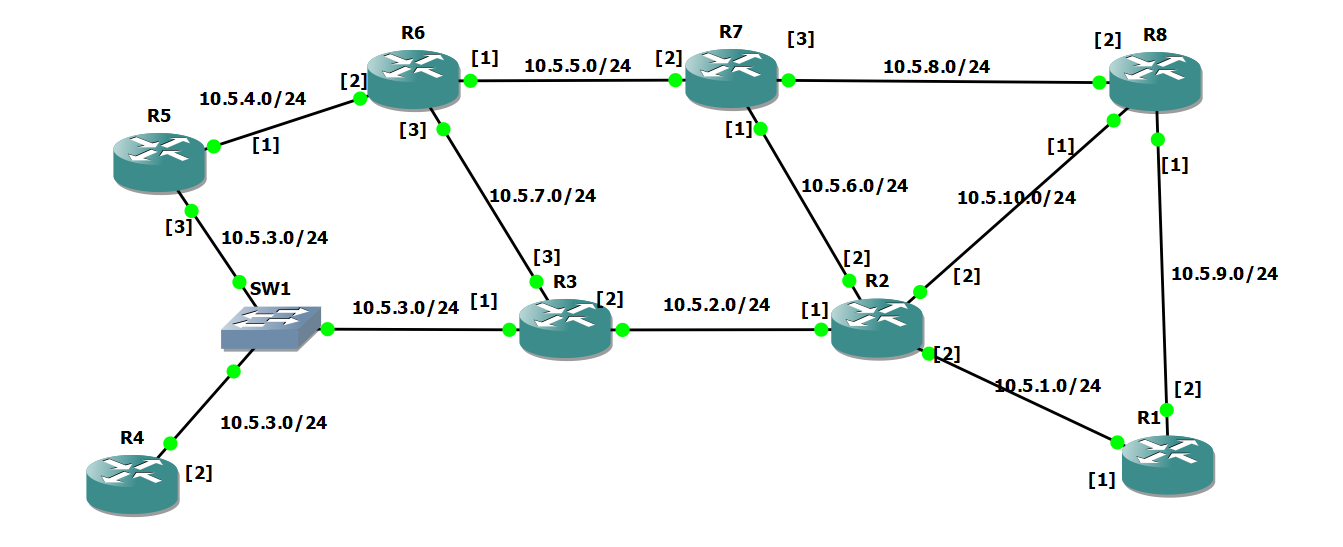
R6

|  |
| --- |
| *configure terminal*  *interface FastEthernet3/0*  *ip address 10.5.7.1 255.255.255.0*  *no shutdown*  *end* |

R3

|  |
| --- |
| *configure terminal*  *interface FastEthernet3/0*  *ip address 10.5.7.2 255.255.255.0*  *no shutdown*  *end* |

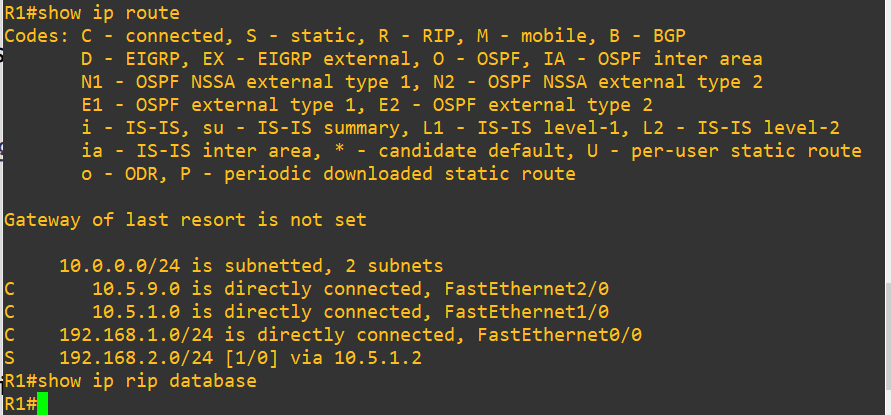
**2.Configuring RIP version 2 on Cisco Routers** –Topology 8

2.1

2.2

*R1#show ip route*

*R1#show ip rip database*



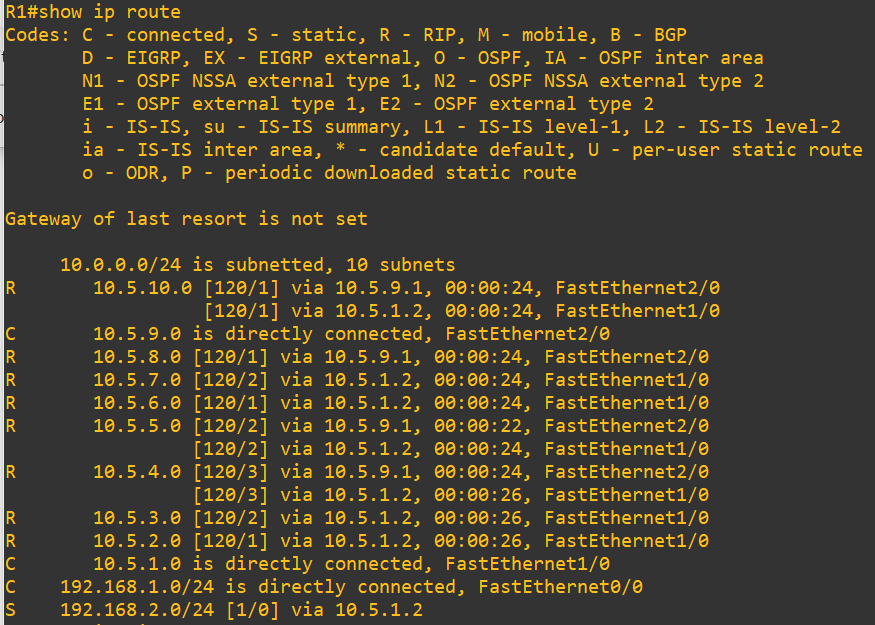
2.4

לכתוב את זה לכל הראוטרים (לקנפג אותם שישתמשו בRIP , וישחקו עם כל מי שב netrork 10.5.0.0)

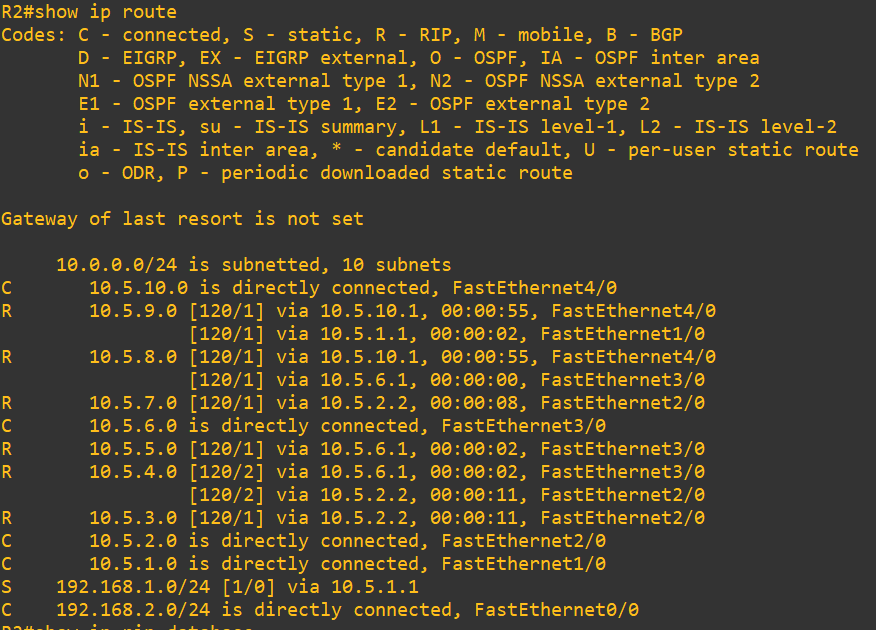
|  |
| --- |
| *configure terminal*  *router rip*  *version 2*  *no auto-summary*  *network 10.5.0.0*  *end* |

2.6

Text

Description automatically generatedR1: *show ip route show ip rip database*

Text

Description automatically generatedR2: *show ip route show ip rip database*

2.7

Text

Description automatically generated*R2#show ip protocols*

2.11

R1#configure terminal

R1(config)#router rip - enable RIP.

R1(config-router)#version 2 - change the version of the protocol to 2

R1(config-router)#no auto-summary – turn off automatic summarizations

(config-router)#network 10.5.0.0 – advertise only this subnet

R1(config-router)#end

2.12

1. נעשה את הפקודה *show ip rip database* ואם נראה שם את הרשת 10.5.0.0 אז הכל התכנס

2. נשלח פינג ונראה אם הכל מחובר.

2.13

איי פי של היעד – 224.0.0.9

רק מי שרוצה לשחק מקבל את ההודעות ולא סתם כולם.

2.14

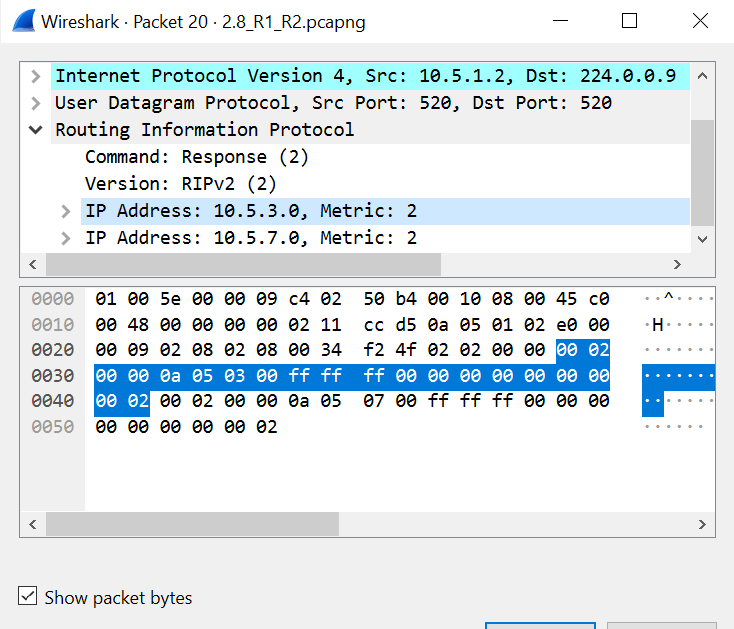
כן, אפשר לראות בWireshark שהתוכן בפקטות מתעדכן.

2.15

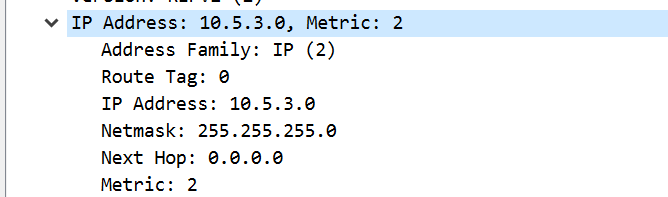
שני סוגים :

1. request - מבקש מהראוטר השכן את התוכן של הטבלת ניתוב שלו
2. response – הטבלה שכל ראוטר שולח (של עצמו) מפרסם לשכנים. נשים לב שזה מגיע כל כמה זמן ולא רק כשיש request.

2.16

כל שורה שנשלחת היא 14 בייטים.

2.17

1. Network address: - היעד שאליו מגיעים

2. Subnet mask:- של היעד

3. Next hop: - דרך מי הוא מגיע לשם

4. Metric: - כמה הופים יש בדרך

5. Timer:- הזמן לפני שהראוטר יוצא מהטבלה.

2.18

1. Routing Protocol is "rip" – איזה פרוטוקול ממומש פה.
2. Outgoing update filter list for all interfaces is not set - אין חוקים לשליחת הודעות של flow control
3. Incoming update filter list for all interfaces is not set – אין חוקים לשליחת הודעות של flow control
4. Sending updates every 30 seconds, next due in 20 seconds – כל 30 שניות תשלח הודעת עדכון, הבאה תשלח בעוד 20 שניות.
5. Invalid after 180 seconds, hold down 180, flushed after 240 – אם R2 לא יקבל הודעת עדכון מR1 תוך 180 שניות – כל המסלולים מR1 יהפכו ללא שמישים. אם אחרי 240 שניות עדיין לא קיבלנו עוד עדכון – נמחק מהטבלה את המסלולים שמגיעים מR1.
6. Default version control – שולח ומקבל בגרסה 2 של RIP.
7. FastEtherne1/0 2 2 – הפורט מוציא הודעות בגרסה 2 ומקבל הודעות של גרסה 2.
8. Automatic network summarization is not effect – אין פה קיצור של הטבלת ניתוב (נגיד לאחד שורות עם פרפיקס דומה)
9. Maximum path: 4 – אפשר להחזיק עד 4 מסלולים לאותו יעד.
10. Routing for Networks: 10.0.0.0 - מספר איזה רשת משתתפת בפרסומים של RIP.
11. Routing Information Sources: - נותן מידע על כל כתובת איי פי והמחיר של המרחק שלה. בנוסף מתי היא התעדכנה לאחרונה.

Text

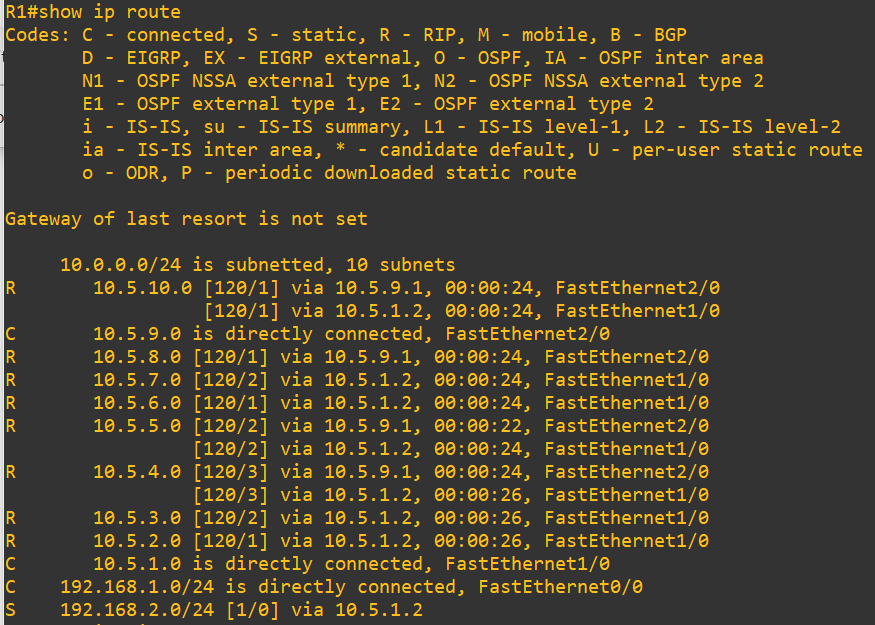
Description automatically generated

2.19

שוני בין שתי הטבלאות- טבלת ריפ וטבלת ניתוב :

הטבלה של RIP מתמלאת ע"י עדכון של הפרוטוקול של RIP בלבד, בכל םעם שמישהו שמשתתף בRIP מתעדכן הטבלה מתעדכנת.   
לעומת זאת, טבלת הניתוב של הראוטר מתעדכנת ע"י כל מיני פרוטוקולים (כולל ניתוב סטטי למשל). ניתן לראות בתחילת כל שורה את סוג הפרוטוקול שאחראי על הבאת אותו מידע. (למשל R זה RIP).

Text

Description automatically generatedנשים לב שאצלנו לדוגמא, בטבלת ניתוב כתובים המחשבים שחיברנו בצורה סטטית במעבדה הקודמת, ואילו בטבלת RIP הם לא מופיעים (כי הם לא שיחקו במשחק).

2.20

הפעלנו wireshark, ואז קינפגנו את כל הראוטרים להיות RIP, חיכינו שהרשת תתכנס ושמרנו את הפיק אפ:

לבדוק עם בירן אם אצלו יש פקטות כמו שצריך, צריך למצוא 2 פקטות אחת של R1 אחת של R2 (אחרי ההתכנסות – נראה שיש להן את כל המידע על כל הראוטרים), למצוא אם יש2 IP מסויים שנמצא בשתי הטבלאות (של R1 ושל R2) ואז לראות אם אותו IP מופיע גם בטבלאות ניתוב.

20.21

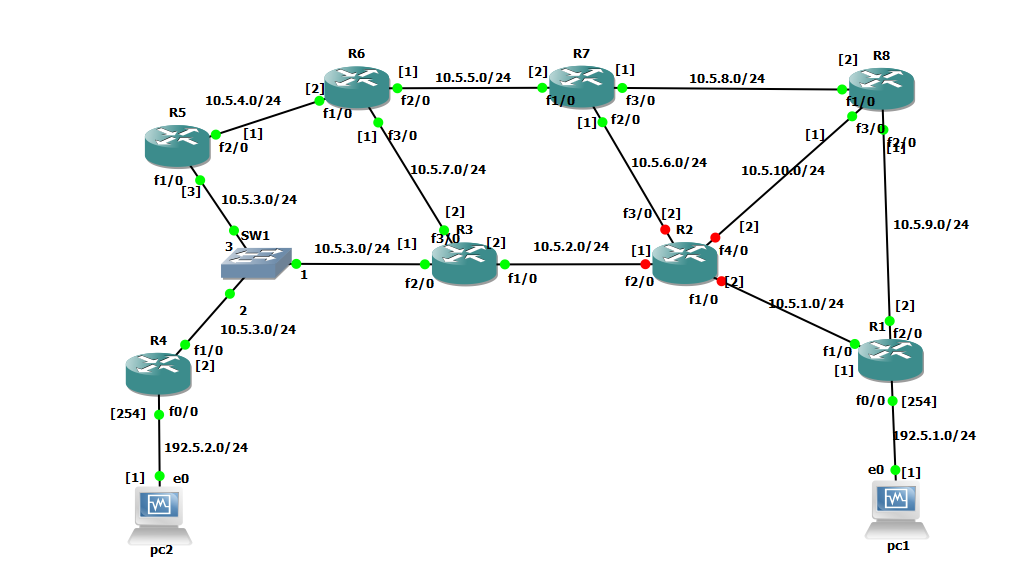
נסביר את הטבלת ניתוב של R2:  
דוגמא:

R 10.5.9.0 [120/1] via 10.5.10.1, 00:00:55, FastEthernet4/0  
 [120/1] via 10.5.1.1, 00:00:02, FastEthernet1/0

* C/R/S : מציינים באיזה פרוטוקול השתמש הראוטר כשקיבל את המידע אודות אותה רשת. מחובר ישירות (c ), ע"י RIP (r) או הוכנס סטטי (s).
* 10.5.9.0 : אותה כתובת IP שלמדנו את הניתוב אליה.
* [120/1]: 120 מציין את רמת האמינות של הדרך, כמה שיותר נמוך יותר טוב.   
  1 מציין כמה הופים בדרך, (10.5.9.0 זה בין R8 לR1 אפשר ושניהם מחוברים ישירות לכן אפשר להגיע אליה דרך הופ אחד). הראוטר לוקח בחשבון את שניהם כשהוא בוחר את הדרך העדיפה.
* Directly/via: איך היא מחוברת, באופן ישיר או דרך אחד IP אחרת.
* 00:00:55 – כמה זמן עבר מאז העדכון האחרון של השורה הזאת.
* FastEthernet4/0 – דרך איזה פורט לצאת.
* Text

  Description automatically generated\*\* אפשר לראות שיש בטבלה דרך חלופית באותו משקל.

1. **Measuring RIP version 2 Convergence Time –** Topology 8.1



**3.3**

\*\* קודם למחוק את הקינפוג של המחשבים שנשאר מלפני (ממעבדה קודמת):

למחוק רשת שמחוברת ישירות- נוציא את הפורט משימוש:

*configure terminal*

*interface FastEthernet0/0*

*shutdown*

*no ip route 192.168.1.0 255.255.255.0*

*exit*

**3.4**

נקבע IP סטטי חדש לשני המחשבים:

|  |  |
| --- | --- |
| **Pc1:**  *nano /etc/network/interfaces*  *auto eth0*  *iface eth0 inet static*  *address 192.5.1.1*  *netmask 255.255.255.0*  **ctrl+x  y enter reboot**  *route add default gw 192.5.1.254* | **Pc2:**  *nano /etc/network/interfaces*  *auto eth0*  *iface eth0 inet static*  *address 192.5.2.1*  *netmask 255.255.255.0*  **ctrl+x  y enter reboot**  *route add default gw 192.5.2.254* |

**3.5**

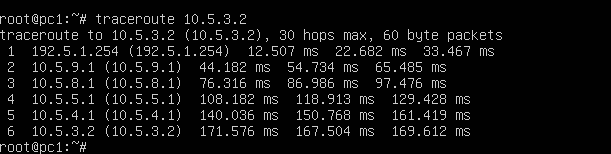
R1 *(configure the network 192.5.1.0 on rip – passive, configure the port with new ip)*

|  |
| --- |
| *configure terminal*  *router rip*  *network 192.5.1.0*  *passive-interface FastEthernet 0/0*  *exit*  *interface FastEthernet 0/0*  *ip address 192.5.1.254 255.255.255.0*  *no shutdown*  *end* |

**3.6**

**Pc1** *(check traceroute from pc1 to R4- port 1/0)*

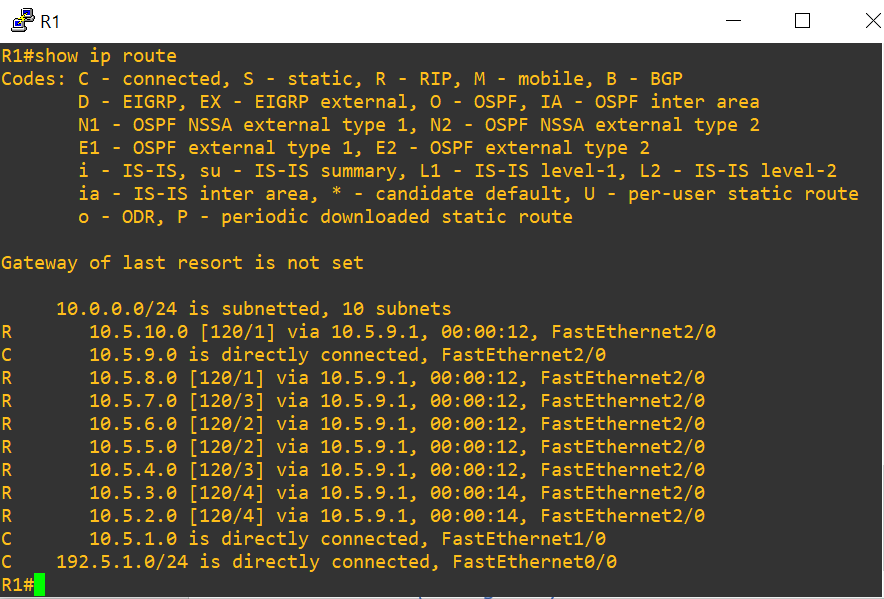
|  |
| --- |
| *traceroute 10.5.3.2* |



**3.7**

**R1** *(routing table)*

|  |
| --- |
| *show ip route* |



**3.8** Start Wireshark on link: **R1-R8**, **R8- R7, R4- R5 (via switch)**

**3.8** Wireshark -> View -> Time Display Format -> Time Of Day.

**3.10**

**pc1** *(ping from PC1 to PC2)*

|  |
| --- |
| *ping 192.5.2.1* |

**3.11**

**R4** *(configure the network 192.5.2.0 on rip – passive, configure the port with new ip)*

|  |
| --- |
| *configure terminal*  *router rip*  *network 192.5.2.0*  *passive-interface FastEthernet 0/0*  *exit*  *interface FastEthernet 0/0*  *ip address 192.5.2.254 255.255.255.0*  *no shutdown*  *end* |

**3.12** Wait for the network convergence. The ping should start working properly.

Text

Description automatically generated**3.13** Stop the ping command using Ctrl + C.

**3.14**

Stop capture and save the wireshark pcaps to files.

**3.15**

Look again at the routing table of R1 and save the output.  
**R1** *(routing table)*

|  |
| --- |
| *show ip route* |

Text

Description automatically generated

**3.17**

כששלחנו את הפינג בהתחלה מpc1 לpc2 ההועדה הגיעה בדיפולט גט וואיי לR1.  
R1 זרק אותה, כי אין לR1 בטבלת ניתוב את הרשת 192.5.2.0, זאת משום שR4 אפילו לא קונפג ולא יודע שמאחוריו יש את הרשת 192.5.2.0.

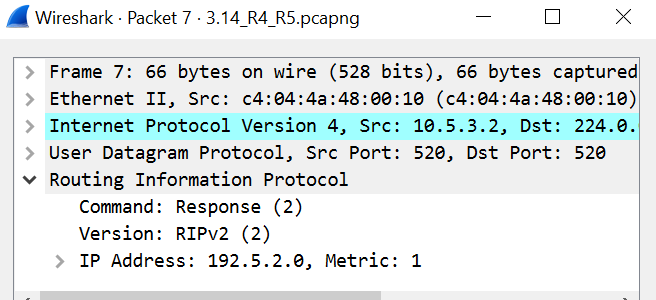
**3.18**

לאינטרפסיים שקינפגנו כpassive מחוברים רק מחשבים- ללא ראוטרים. בנוסף מדובר ברשת שונה שלא כלולה במשחק של הRIP שהגדרנו (10.5.0.0) , לכן לשלוח למחשבים את ההודעות של הRIP זה סתם בזבוז של רוחב פס. בנוסף מדובר בבעיית בטיחות אם לא רוצים שהמחשב בקצה ידע על כל המסלולים ומי הולך לאן.

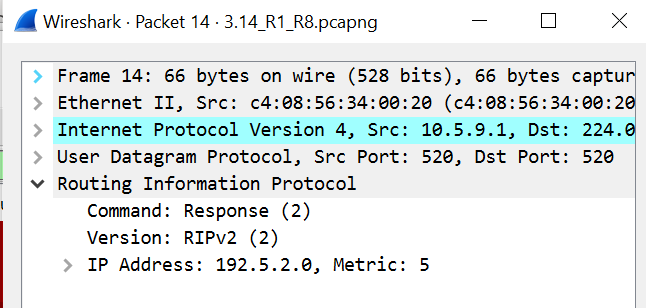
**3.19**

* Link between R4 – R5 (via switch): packet #7, time: 18:51:51

R4 בפורט f1/0 מפרסם לראשונה את הרשת בשעה 18:51:51, רואים שהוא מפרסם עם מרחק = 1. (כי הרשת היא בפורט אחר שלו. זה הפרסום הראשון שמכניס את הרשת למשחק אחרי שקינפגנו את R4.)



* Link between R1 - R8: packet #14, time: 18:52:04

R8 בפורט f2/0 מפרסם לראשונה את הרשת בשעה 18:52, עם מרחק = 5.

* Link between R7 - R8: packet #6, time: 18:52:00

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

Description automatically generatedR7 פורט f3/0 מפרסם את הרשת לראשונה בשעה 18:52, עם מרחק = 4.

לכן מהרגע שהתחלנו לפרסם, עד שהמידע פורסם לR1 לקח

הזמן שלוקח להודעה לעבור דרך הופ יחיד ניתן לחישוב ע"י:

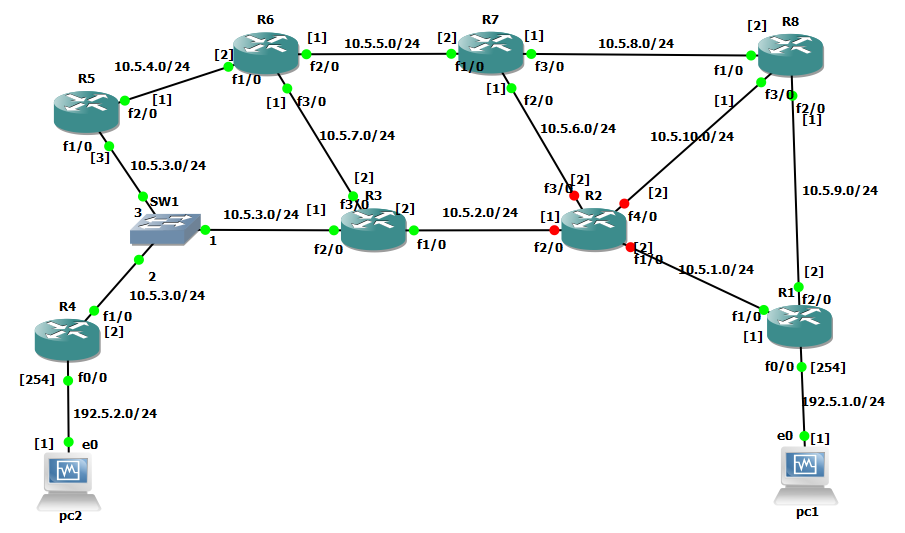
*(הזמן שהפקטה עם המידע הראשוני על הרשת יצאה מR7 עד שיצאה לראשונה מr8)*

**3.20**

בRIP 2 משתמשים בשני דברים שעוזרים להתכנסות מהירה יותר של הרשת:  
1. Poisoning – כשיש ראוטר שיוצא משימוש מי שמזהה את זה ישר מעדכן את השכנים שלו, לעומת RIP1 שרק מוחק אותו מהטבלה שלו.

2. trigger updates – ברגע שיש למישהו עידכון בטבלה הוא ישר מפיץ הודעה ולא מחכה להודעת הבאה (כל 30 שניות), בrip1 הוא יחכה אין את העדכון המיידי הזה.

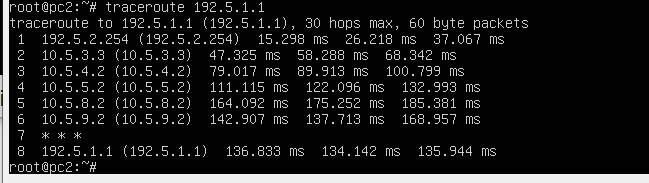
**4.Good News Propagation (Adding a Router)** - Topology 8.2

**4.1**

**4.3**   
\*\*verify the default gw, may need to use: *route add default gw 192.5.2.254*

**Pc2: traceroute from PC2 to PC1**

|  |
| --- |
| *traceroute 192.5.1.1* |



\*\*היינו צריכים להריץ שוב כדי לקבל טבלה בלי הכוכביות (לפעמים מריצים כמה פעמים עד שנלמדות הטבלאות ARP כמו שצריך)

**4.4** Start Wireshark on link **R4 -SWITCH,** **R3 – R2, R1 – R2**

**4.5 Pc2:** Issue an infinite ping from PC2 to PC1

|  |
| --- |
| *ping 192.5.1.1* |

**4.6 R2:** Start R2 router   
wait for the network to converge.

**4.7 R1, R4** :verify that the network was conveneds (נבדוק שאכן הטבלה מכילה את הכל)

|  |
| --- |
| *show ip route* |

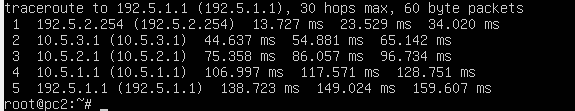
**4.8 Pc2:** Stop the ping

|  |
| --- |
| Ctrl+c |

**4.9** Stop capture and save the Wireshark pcaps to file

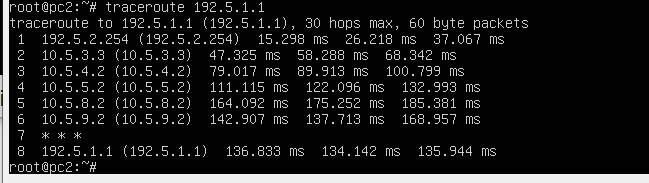
**4.10 Pc2:** Use traceroute again from PC2 to PC1

|  |
| --- |
| *traceroute 192.5.1.1* |



**4.11**

נחקור את ההבדל בין המסלול PC2-PC1 לפני ואחרי הוספת R2:

Before:

|  |  |
| --- | --- |
| **לפני:**   1. R4 2. R5 3. R6 4. R7 5. R8 6. R1 7. PC1 | **אחרי:**   1. R4 2. R3 3. R2 4. R1 5. PC1 |

\*\*היינו צריכים להריץ שוב כדי לקבל טבלה בלי הכוכביות (לפעמים מריצים כמה פעמים עד שנלמדות הטבלאות ARP כמו שצריך)

Text

Description automatically generatedAfter:

**4.12**

הפינג נשלח כל הזמן, גם לפני שהדלקנו את R2 וגם אחרי:

**4.13**

* נחפש את ההודעה הראשונה בה R2 מפרסם שהוא התחיל לעבוד:  
  בלינק R2-R1 נסנן הודעות RIP עם כתובת IP במקור של R2

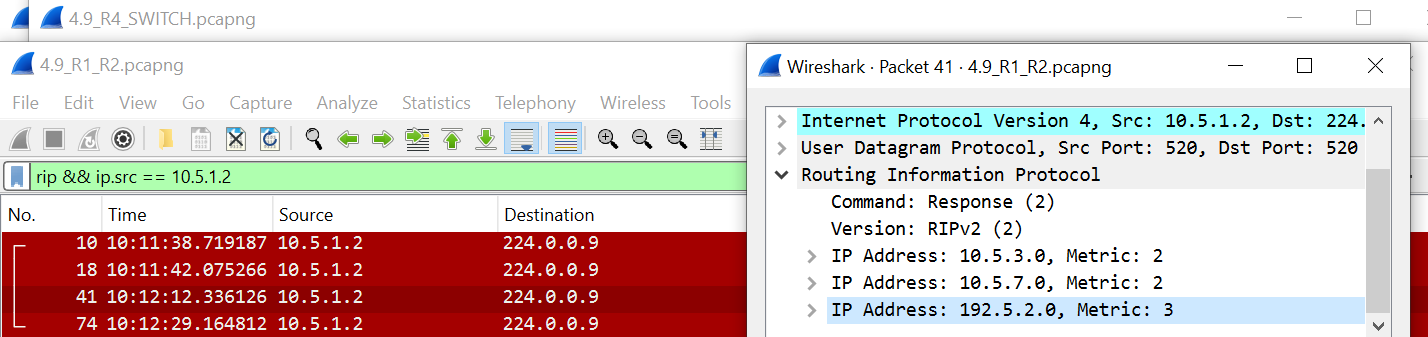
Graphical user interface, application, table, Excel

Description automatically generatedrip && ip.src == 10.5.1.2

קיבלנו הודעה ראשונה בשעה 10:11:38

* נחפש את ההודעה הראשונה בה R2 מפרסם את הרשת של PC2 אבל במרחק 3 (כלומר לאחר שסיפר ל R3 שיש דרכו דרך וכו')

בלינק R2-R1 נחפש הודעת RIP עם כתובת מקור של R2 שלראשונה מפרסמת את הרשת עם metri=3:

rip && ip.src == 10.5.1.2

קיבלנו הודעה ראשונה בשעה : 12:12:10

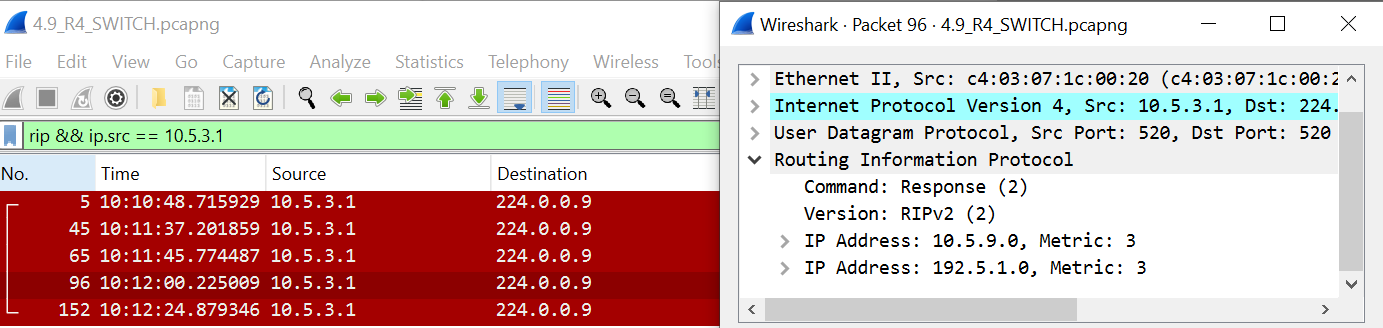
**סה"כ קיבלנו 34 שניות עד ש R2 פרסם ל R1 שהוא מכיר את הרשת:**

**4.14**

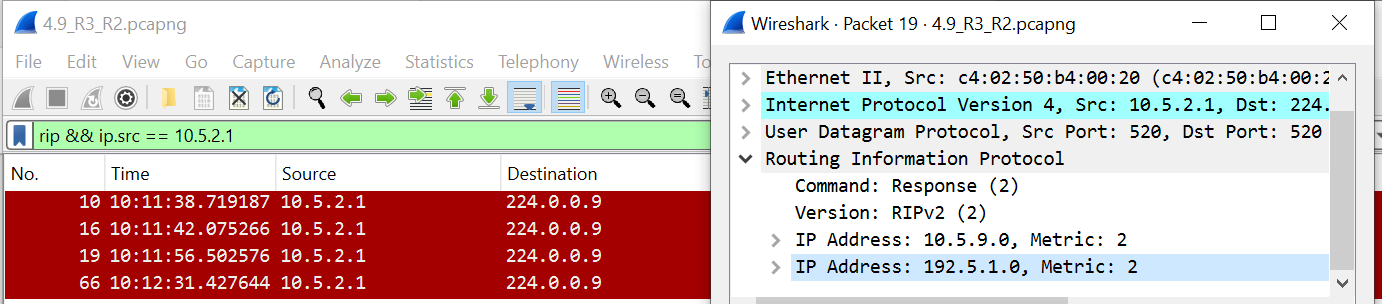
נחפש פקטת RIP שמעדכנת לראשונה על השינוי והדרך הקצרה יותר

**R4 – SWITCH:**

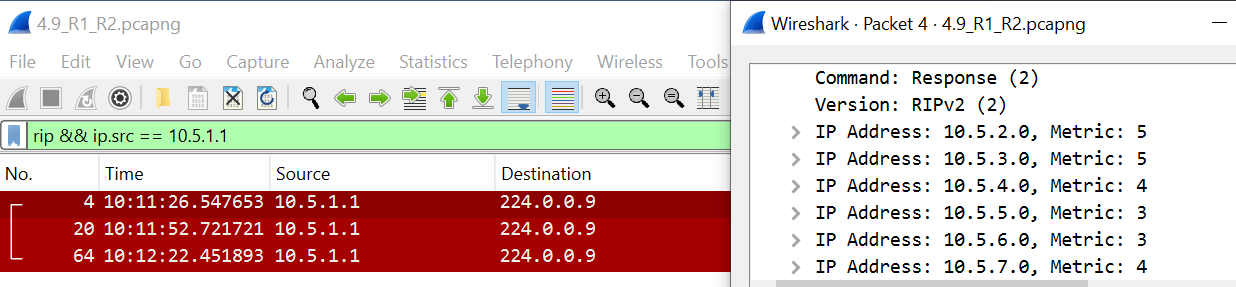
פקטה 96, נשלחת מ R3 ומפרסמת לראשונה שיש לה דרך ל PC1 באורך 3.   
לאחר ש R4 ילמד את המידע הזה הוא יעביר את הפינג דרך R3



**R3 - R2:**

פקטה 19, נשלחת מ R2 מפרסמת לראשונה שיש לה מסלול ל PC1 באורך 2

**R2 – R1:**

פקטה 4 מ R1 לראשונה מפרסמת ל R2 שיש לה מסלול של 1 ל PC1, R2 לומד ויעביר את ההודעות דרכו מעכשיו.

**4.15**

* בהתחלה R2 פרסם לראשונה הודעת RIP REQUEST עם טבלה ריקה
* אחכ קיבל מהשכנים שלו את הטבלאות שלהם
* R2 הפיץ את הטבלה שהוא למד אחרי הנתונים מ R1,R3
* ואז שוב השכנים מעדכנים בעדכונים שהם קיבלו
* ואז שוב R2 מעדכן מחדש את הטבלה שלו ושוב מפיץ אותה.

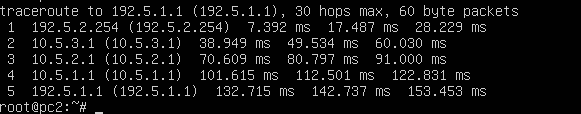
**5.Bad News Propagation (Router Down)** - Topology 8.3

**5.4** Use traceroute from PC2 to PC1

\*\*verify the default gw, may need to use: *route add default gw 192.5.2.254*

**Pc2: traceroute from PC2 to PC1**

|  |
| --- |
| *traceroute 192.5.1.1* |



**5.5** Start Wireshark on **R4 - SWITCH, R3 -R2, R1 - R8**

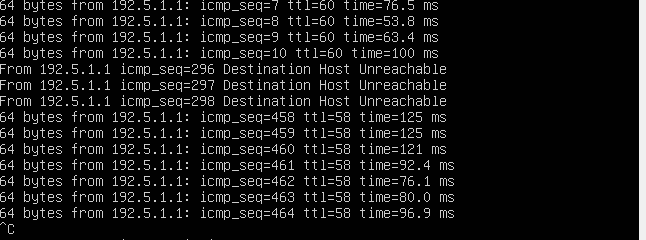
**5.6 Pc2:** Issue an infinite ping from PC2 to PC1

|  |
| --- |
| *ping 192.5.1.1* |

**5.7** open computer's clock, stop the R2 write down the “stop time” (in resolution of seconds):

**06:13:39**

**5.8** When the router stops, the ping fails. Wait for the network to converge until the ping returns to work properly (6 minutes).



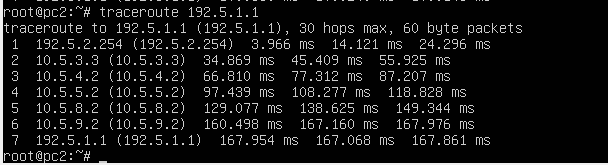
**5.9 Pc2:** Stop the ping

|  |
| --- |
| Ctrl+c |

**5.10** Stop capture and save the wireshark pcaps to files

**5.11 Pc2: traceroute from PC2 to PC1**

|  |
| --- |
| *traceroute 192.5.1.1* |



**5.12**

נתאר את המסלול מ PC2 ל PC1 לפני ואחרי הכיבוי של R2:

A screen shot of a computer

Description automatically generated with low confidenceBefore:

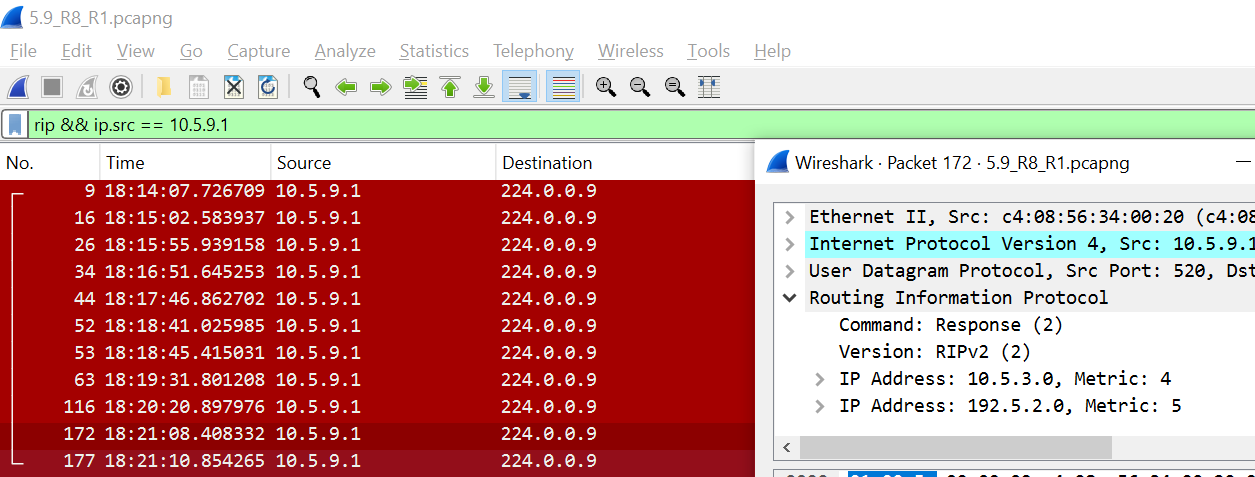
|  |  |
| --- | --- |
| **לפני:**   1. R4 2. R3 3. R2 4. R1 5. PC1 | **אחרי:**   1. R4 2. R5 3. R6 4. R7 5. R8 6. R1 7. PC1 |

Text

Description automatically generatedAfter:

**5.13**

נבדוק כמה זמן לקח לרשת להתכנס מהרגע שכיבינו את R2:

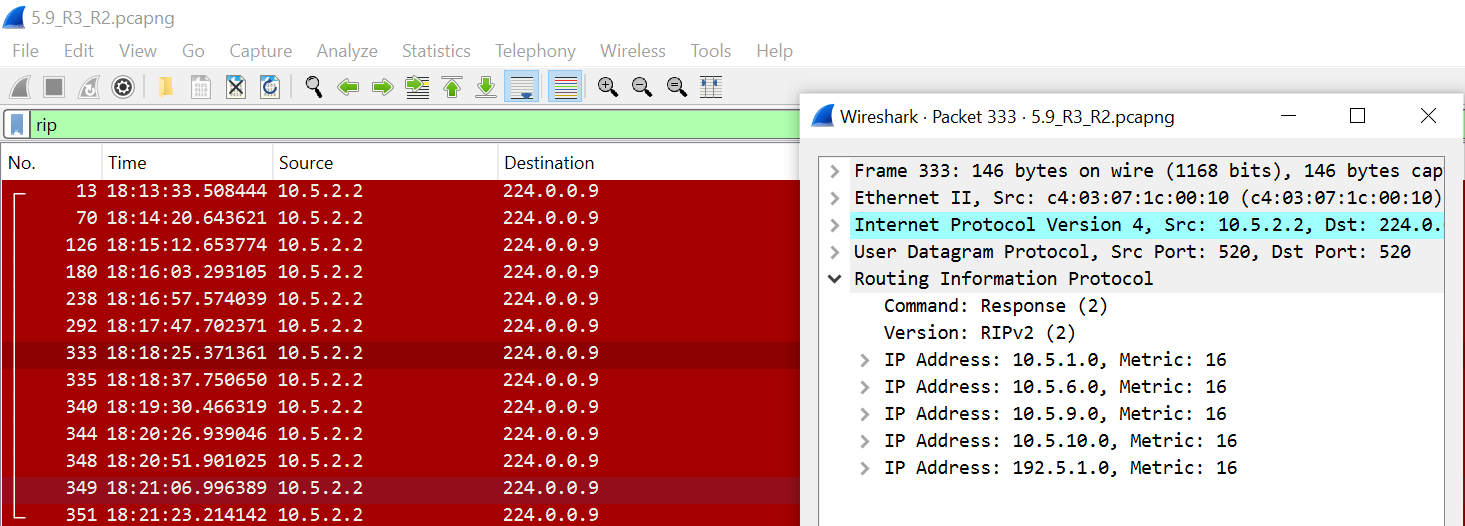
נחפש בקצבים של הwireshark את ההודעת RIP שגרמה לפינג להצליח להגיע ל PC1 וגם לחזור ממנו.

הזמן בו כיבינו את R2 :

R8 מפרסם את PC2 עם מטריקה 5 בפעם הראשונה ב: 18:21:08

כלומר לקח לרשת להתכנס:

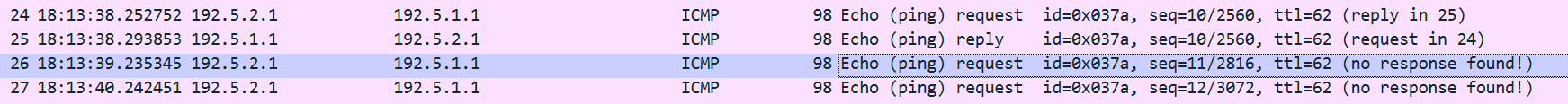
**5.14**

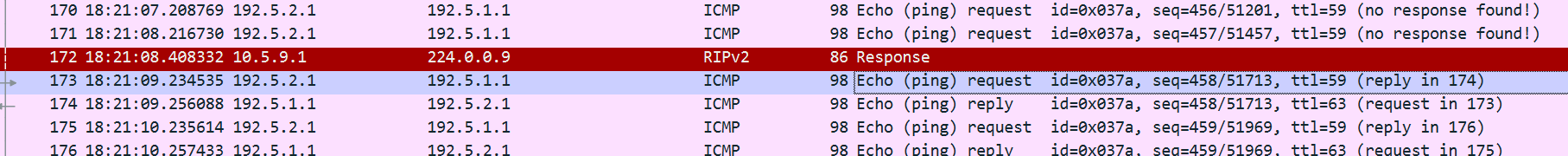
**בלינק R2-R3** צריך ללכוד את הפקטה בה R3 הבין ש R2 אבד. ונמדוד כמה זמן לקח לו להבין את זה:

כאשר R2 נכבה R3 עדיין המשיך לשלוח דרכו הודעות במשך זמן ה holddown (180 שניות) ע"מ לאפשר לו להתאושש אם זה היה סתם נפילה נקודתית.   
אחרי שעבר ה holddown R3 מבין שאין לו גישה דרך R2 לכל מי שעד עכשיו היה מאחוריו ומפרסם שהוא במרחק 16 מהם – כלומר לא נגיש.

הזמן בו נשלחת ההודעה הזו: אז סה"כ לקח לו להבין

**5.15**

\*\*\* צריך להשתמש בסטטיסטיקה של הפינג, אחרי הקונטרול c אבל לא צילמתי את זה, אז בדקנו ב wireshark\*\*\* - אם רוצים סטטיסטיקה צריך:  
- נקבל את המידע שאומר כמה פינגים נפלו מתוך כל הפינגים שנשלחו  
- וזאת כמות השניות שבהן הפינג לא עבד, כלומר הזמן בו הרשת עוד לא התכנסה  
עכשיו נבדוק את זמן ההתכנסות בעזרת הפינג, פינג נשלח כל שניה, נחפש מתי הפינג התחיל ליפול ומתי הוא חזר לעבוד כראוי:  
נסתכל על הלינק R2-R3:   
אפשר לראות שהפינג נשלח כראוי עד 18:13:39 שם כבר לא התקבלה תגובה לפינג, וזה אכן הזמן בו כיבינו את הראוטר.

בלינק של R8 -R1 אפשר לראות שהפינג חזר לעבוד בשעה 18:21:09 כשאר הרשת מצאה מסלול חלופי.  
נחשב את ההפרש:

**5.16**

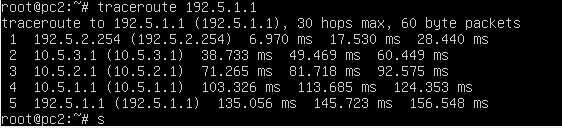
Explain, what in the protocol supports the waiting time you calculated?

**6.Bad News Propagation (Interface Down)** – Topology 8.4

**6.3**   
\*\*verify the default gw, may need to use: *route add default gw 192.5.2.254*

**Pc2: traceroute from PC2 to PC1**

|  |
| --- |
| *traceroute 192.5.1.1* |

 **6.4 Pc2:** Issue an infinite ping from PC2 to PC1

|  |
| --- |
| *ping 192.5.1.1* |

**6.5** Shut down the interface f2/0 of router R2.

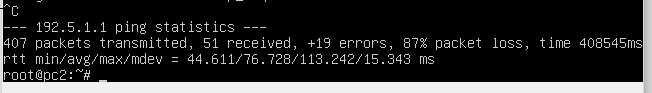
**R2:**

|  |
| --- |
| *configure terminal*  *interface FastEthernet2/0*  *shutdown*  *no ip route 10.5.2.0 255.255.255.0*  *exit* |

**6.6** Wait for the network to converge until the ping returns to work properly.

**6.7 Pc2:** Stop the ping

|  |
| --- |
| Ctrl+c |



**6.8 Pc2: traceroute from PC2 to PC1**

|  |
| --- |
| *traceroute 192.5.1.1* |

Text

Description automatically generated

**6.9**

מרגע שסגרנו את האינטרפייס אף פינג לא יכל לעבור, הזמן שלקח בשניות לרשת להתכנס שוב שווה למספר הפיניגים שנפלו שכן כל שניה נשלח פינג

**6.10**

|  |
| --- |
|  |
| Text  Description automatically generated |

|  |  |
| --- | --- |
| **לפני:**   1. R4 2. R3 3. R2 4. R1 5. PC1 | **אחרי:**   1. R4 2. R5 3. R6 4. R2 5. R1 6. PC1 |

**7.Mechanisms to prevent "Count to Infinity Problem" –** Topology 8.5

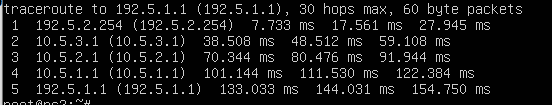
Diagram

Description automatically generated with medium confidence**7.2.** Start only: PC1, PC2, R1, R2, R3, R4

**7.3.** \*\*verify the default gw, may need to use: *route add default gw 192.5.2.254*

**Pc2: traceroute from PC2 to PC1**

|  |
| --- |
| *traceroute 192.5.1.1* |



**7.4.** Start Wireshark on **R4- SWITCH , R3 -R2 , and on R1- R2**

**7.5.** **Pc2:** Issue an infinite ping from PC2 to PC1

|  |
| --- |
| *ping 192.5.1.1* |

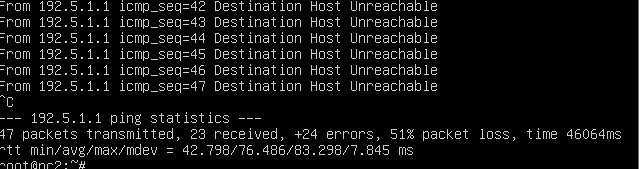
* 1. **Open computer's clock,** Shut down the interface f1/0 of router R2. **11:13:41**  
     **R2:**

|  |
| --- |
| *configure terminal*  *interface FastEthernet1/0*  *shutdown*  *no ip route 10.5.2.0 255.255.255.0*  *exit* |

* 1. The ping will fall. After several attempts that result in "Destination Host Unreachable", stop the ping.

**Pc2:** Stop the ping

|  |
| --- |
| Ctrl+c |



* 1. Stop capture and save the wireshark pcaps to files.

updates – מספר ישר ולא מחכה ל30 שניות של העדכון הקבוע

split horizon – לא מדווח למי שעובר דרכו על המסלול הטוב שלו

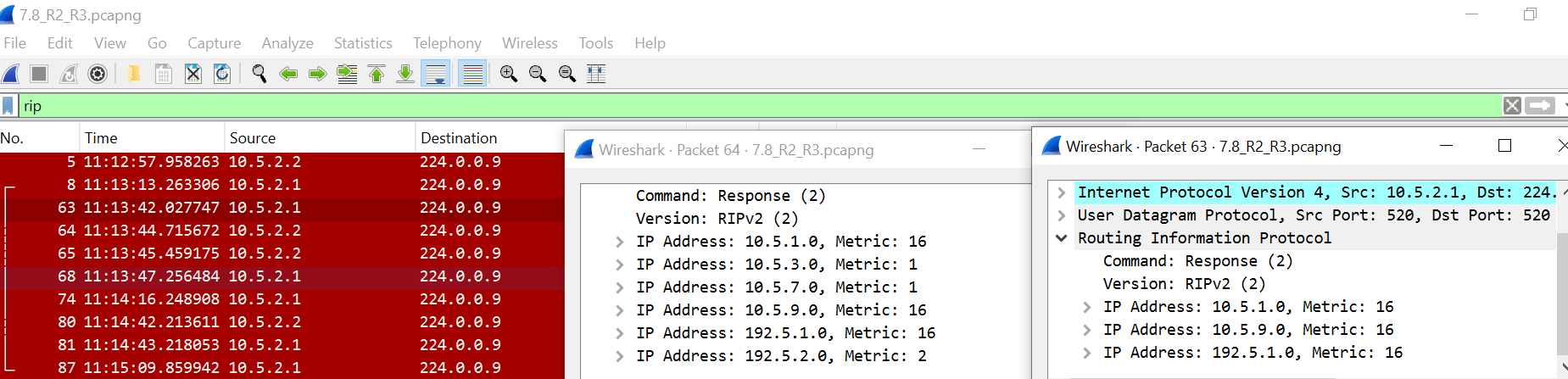
route poisoning – מדווח למי שהמסלול עובר דרכו שהמרחק הוא 16

אם היינו משתמשים ב RIPV1 שאין לו את המרכיבים האלו מה היה קורה:

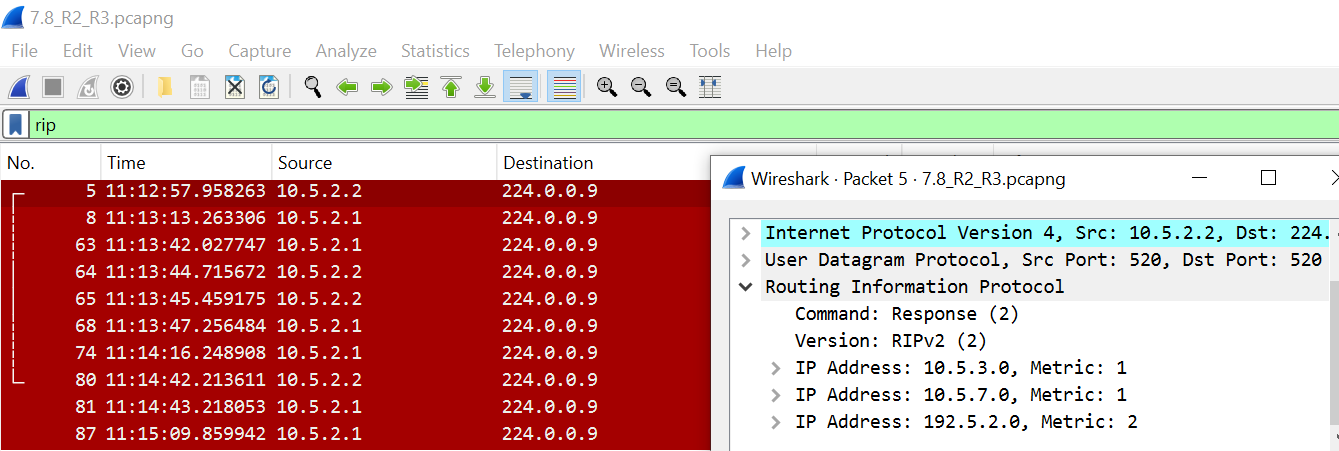
* בגלל שאין **trigger update** ברגע שה interface נפל R2 לא מעדכן אף אחד ומחכה לעדכון הבא.
* בינתיים R3 ממשיך לחשוב שהדרך ל PC1 היא דרך R2 והיא באורך 2.
* בגלל שאין **split horizon** R3 פרסם קודם ל R1 שיש לו דרך ל PC1 באורך של 2.
* ברגע שהיינו מכבים את הinterface של R2 אז הניתוב של R2 ל PC1 היה מתבטל והוא היה מחפש דרך חלופית טובה יותר.
* בגלל ש R3 פרסם לו שיש לו דרך באורך 2, R2 היה מנתב את החבילות ל PC1 דרך R3.
* R3 היה מחזיר אותן ל R2 כי הוא לא יודע על השינוי אצל R2 (כי אין **trigger update**)
* שוב R2 היה מחזיר אותן ל R3 וככה היינו נתקעים בלופ עד העדכון הבא של R2 שידווח שאין לו קשר ל PC1.
* \*\*\*\* אם היו המנגנונים **split horizon** וגם **route poisoning** אז R3 היה מפרסם ל R2 שהדרך שלו ל PC1 היא באורך 16. ואז R2 בכלל לא היה חושב על האופציה להחליף את הניתוב להיות דרך R3.

1. צריך לזהות ב wireshark 2 סיטואציות שונות שנגרמו כתוצאה מהשימוש ב trigger update (נדגיש את הזמן בן נשלחה ההודעה)
   1. בשעה 11:13:41 כיבינו את ה interface, אפשר לראות ששניה אח"כ R2 פרסם דרך שאר ה interfaces שלו שנפל לו החיבור ל PC1 (packet#63)

בלינק R2-R3 אפשר לראות שאיך R2 מפרסם (packet#63) שאין לו גישה ל PC1 ישר (שניה אח"כ) נשלחת הודעה מ R3 (packet#64) שאומרת שגם לה אין גישה ל PC1

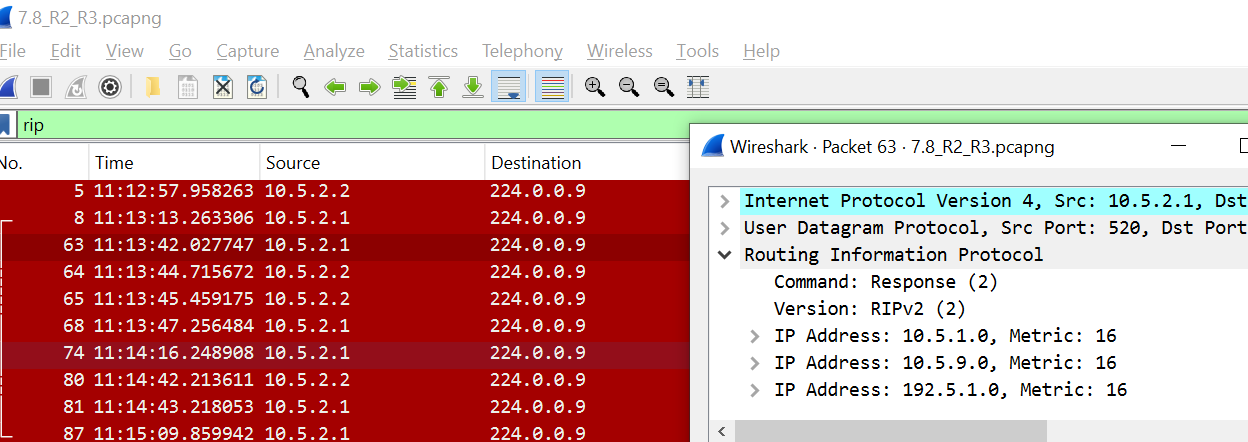
תמונה מלינק R2-R3:

1. סיטואציה אחת שמראה שימוש ב split horizon

נסתכל על הודעה ששולח R3, אפשר לראות שאין שם את הניתוב ל PC1 כי ההודעה הזאת תגיד גם ל R2 שדרכו הניתוב עובר, לכן R3 לא מפרסם לו את זה.

1. סיטואציה אחת שמראה שימוש ב route poisoning

כש R2 מפרסם ל R3 שנפל לו הקשר ל PC1 אז הוא מפרסם את כל הניתובים דרכו באורך 16.

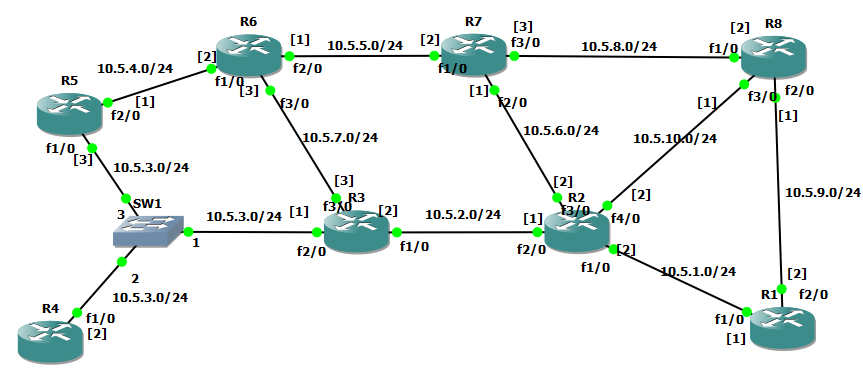


* 1. For each such situation, explain how the relevant mechanism contributes with prevention of count-to-infinity problem.
  2. ברגע שיש תקלה R2 ישר מעדכן, מונע מהשכנים שלו להעביר לו הודעות סתם למקום שכבר אין לו חיבור אליו.
  3. ברגע R3 מבין שיש תקלה ל R2 הוא ישר מעדכן גם אצלו וגם את השכנים שלו שכל מי שרוצה להגיע למקומות שעד עכשיו נותבו דרך R2 שלא ינסה דרכו.

בגלל במיידיות אנחנו חוסכים את ההודעות המיותרות שיעברו בפינג פונג כי אף אחד חוץ מ R2 לא יודע שהייתה תקלה.

1. אם R3 היה מספר ל R2 שיש לו ניתוב באורך 2 ל PC1 אז ברגע שה interface נפל R2 היה מעביר את ההודעות ל R3, R3 היה מחזיר לו אותן והיה נוצר פינג פונג.
2. אם R2 לא היה מפרסם ל R3 שכל הניתובים שנפלו הם באורך 16, אז R2 היה מגדיל את המרחק שלו ל 3, ואז R3 היה מגדיל ל4 וככה הפינג פונג היה נמשך לנצח. בגלל שהוא אמר 16 – כלומר לא ישיג, אז R3 לא יכול עוד להגדיל את המרחק שלו ל PC1 בפעם הראשונה כבר בפינג פונג והוא היה נעצר.

**Intra AS Routing (Part B): OSPF**

**8.Configuring OSPF on Cisco Routers** – Topology 9

**8.4.** Start Wireshark on **R4 – SWITCH**

**8.5.**  Enable OSPF on all routers:

|  |  |  |
| --- | --- | --- |
| **R1**  *configure terminal*  *router ospf 1*  *router-id 10.5.0.1*  *network 10.5.0.0 0.0.255.255 area 1*  *end* | **R2**  *configure terminal*  *router ospf 1*  *router-id 10.5.0.2*  *network 10.5.0.0 0.0.255.255 area 1*  *end* | **R3**  *configure terminal*  *router ospf 1*  *router-id 10.5.0.3*  *network 10.5.0.0 0.0.255.255 area 1*  *end* |
| **R4**  *configure terminal*  *router ospf 1*  *router-id 10.5.0.4*  *network 10.5.0.0 0.0.255.255 area 1*  *end* | **R5**  *configure terminal*  *router ospf 1*  *router-id 10.5.0.5*  *network 10.5.0.0 0.0.255.255 area 1*  *end* | **R6**  *configure terminal*  *router ospf 1*  *router-id 10.5.0.6*  *network 10.5.0.0 0.0.255.255 area 1*  *end* |
| **R7**  *configure terminal*  *router ospf 1*  *router-id 10.5.0.7*  *network 10.5.0.0 0.0.255.255 area 1*  *end* | **R8**  *configure terminal*  *router ospf 1*  *router-id 10.5.0.8*  *network 10.5.0.0 0.0.255.255 area 1*  *end* |  |

**8.6.** Wait for the network convergence.

**8.7.** Stop capture and save the Wireshark pcap to file.

**8.8.** You should now be able to ping any router from any other router, **R1:**

|  |  |
| --- | --- |
| R2  R3  R4  R5  R6  R7  R8 | *ping 10.5.1.2*  *ping 10.5.2.2*  *ping 10.5.3.2*  *ping 10.5.3.3*  *ping 10.5.4.2*  *ping 10.5.6.1*  *ping 10.5.9.1* |

**8.9.** Perform "write" on each router, save the topology (as Topology 9) and ZIP it.

|  |
| --- |
| *write* |

**8.10.** Leave the network running, in order to answer the exercise questions.

**8.11.** Explain the meaning of each command from step 5:

*configure terminal*

*router ospf 1 -* מספר התהליך בו הראוטר משתתף

*router-id 10.5.0.1* - ID של הראוטר בתוך התהליך של ה OSPF

*network 10.5.0.0 0.0.255.255 area 1 -* הראוטר מפרסם באזור 1, בו המשתתפים ילמדו על הרשת 10.5.0.0/16 בהודעות OSPF

*end*

**8.12.**

OSPF משתמש ישירות בשכבת ה network, ושולח פקטות של IP

**8.13.**

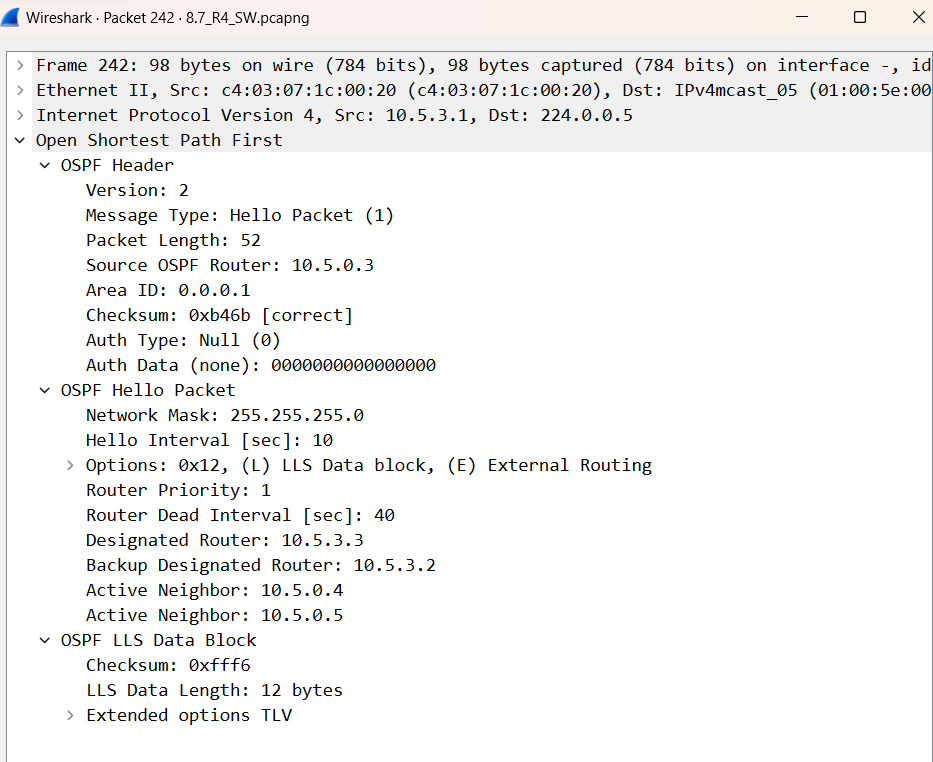
נחפש ב wireshark את פקטת ה OSPF הכי גדולה מכל סוג.

לכל פקטה נסביר מה המטרה שלה, ונסביר מה הפרוטוקול לומד ממנה (main field) :

1. **Hello packet**

**משמשת לתחזוק וזיהוי של השכנים, כל אחד שולח הודעות כאלה לשכנים שלו במחזורי זמן קבועים.(10 שניות)**

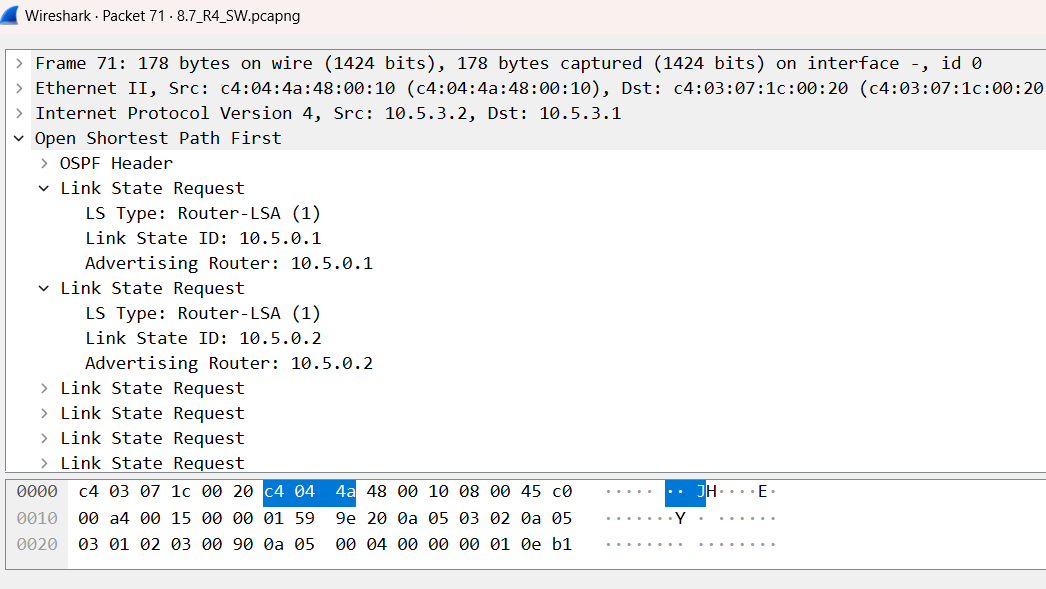
**הפקטות האלה מכילות מידע על ה network של המכשיר, על האזור שלו, ועל ה ID.**

הודעה מ R3:

1. **Link State Ack – בפקטות אלו מועבר המידע על מצב הלינקים. (כתגובה ל update)  
   R4 פרסם הודעה עם כל הלינקים שהוא מכיר:**

**Graphical user interface, text

Description automatically generated**

1. **Link State Requests- אם רוצים לבקש מידע ספציפי מהשכן (כלומר לבקש הועדת LSA ספציפית), אם לאותו שכן אין את המידע הוא יכול להעביר את הבקשה הלאה.**
2. **Graphical user interface, text, application

   Description automatically generatedLink State Updates - מפרסם את המרחקים שלמד משאר הראוטרים ברשת, מי שמקבל הודעה כזאת מעדכן את עצמו.**
3. **DBD -**  מוודא שיש לראוטרים את כל המידע ולא פספסו כלום

**Graphical user interface, text, application

Description automatically generated**

**8.14.**   
**R1:** 10.5.3.0/24

|  |  |
| --- | --- |
|  | *show ip ospf database network 10.5.3.3*  באמצעות הפקודה למעלה ניתן לראות השיוך בין הרשת לראוטרים. |
|  | *show ip ospf database router 10.5.0.3*  בשלב זה מסתכלים לאיזה רשתות ראוטר 3 מתחבר |
|  | *show ip ospf database network 10.5.2.2* |
|  | *show ip ospf database router 10.5.0.2* |
|  | *show ip ospf database network 10.5.1.2* |
|  | *show ip ospf database router 10.5.0.1*    חקרנו אחורה מהיעד אל ראוטר 1 בחזרה דרך המידע על גבי ראוטר1 בלבד. מהרשת לנתב בכל איטרציה, התקדמנו הופ אחר הופ עד שהגענו לנקודת המוצא. |

**8.15.**

בשיטת OSPF אין היררכיה ועל כן כל המידע נמצא ברשות כל הראוטרים המשתתיפ כל הזמן (כל עוד הרשת מכונסת)

|  |  |
| --- | --- |
| *show ip ospf database* |  |

**9. Measuring OSPF Convergence Time**

9.4. Add and configure the *PCs* according to the figure. Set appropriate default gateway for each *PC*.

|  |
| --- |
| \*\* ייתכן ויהיה צורך למחוק בR2 את הקינפוג של המחשבים שנשאר מלפני (ממעבדה קודמת) לבדוק לפני אם הקישור קיים בעזרת show ip route:  *configure terminal*  *interface FastEthernet0/0*  *ip address 192.168.1.254 255.255.255.0*  *shutdown*  *end* |

*end* נקבע IP סטטי לשני המחשבים, ונגדיר default gateway עבורם.  
את הראוטרים נקבע בסעיפים הבאים

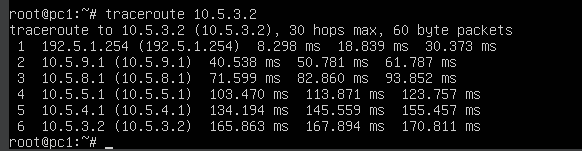
|  |  |
| --- | --- |
| PC1  nano /etc/network/interfaces  auto eth0  iface eth0 inet static  address 192.5.1.1  netmask 255.255.255.0  **ctrl+x  y enter reboot**  route add default gw 192.5.1.254 | PC2  nano /etc/network/interfaces  auto eth0  iface eth0 inet static  address 192.5.2.1  netmask 255.255.255.0  **ctrl+x  y enter reboot**  route add default gw 192.5.2.254 |

9.5. show OSPF routing process.

|  |
| --- |
| ***R1:***  *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* |

9.6. Use *traceroute* from *PC1* in order to verify that your network convened.

|  |
| --- |
| *traceroute 10.5.3.2* |



9.7. Start Wireshark on *R1* interface *f2/0*, on *R8* interface *f1/0* and on *R4* interface *f1/0*.

9.8. Change the time display format of the *Wireshark's* time Column, to "Time of day".

9.9. Issue an infinite *ping* from *PC1* to *PC2*. You should see that the ping failed.

|  |  |
| --- | --- |
| *ping 192.5.2.1* |  |

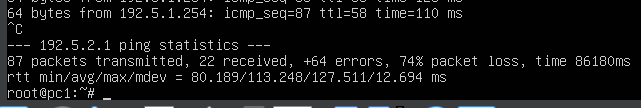
9.10. Similarly, to *R1*, now configure the network **interface *f0/0* of *R4***and set as *passive-interface*. Add the new network to the OSPF routing process.

|  |
| --- |
| ***R4:***  *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* |

9.11. Wait for the network convergence. The ping should start working properly.

9.12. Stop the ping command.

|  |
| --- |
| Ctrl+C |



9.13. Stop capture and save the *wireshark pcaps* to files.

9.14. Perform "*write*" on routers *R1* and *R4*, and save the topology for later exercises.

9.14.1. Keep the project open for question 9.17 .

9.15. Describe the *OSPF* distribution process of the information about the new added subnet. Pay attention to times and to the chronological order of the packets. Refer to information that the packets contain.

|  |
| --- |
| ברגע שR4 מקבל את המידע שהסאבנט של PC2 מחובר אליו, הוא מייד שולח הודעה UPDATE OSPF על כך ליתר הראוטרים  בתוכן ההודעה הוא מספר להם על הלינק החדש: |
| המידע זורם ברשת, ועובר בין הראוטרים השונים:  R7-r8  pc1-r1 |
| לבסוף מכל העדכון הנל חוזרים AKC לR4 |
| לאחר כל העדכונים אפשר לראות שהפינגים מתחילים לזרום: |

9.16. Measure how long it takes for *OSPF* information to spread throughout the entire network, and how long it takes for the information to pass through a single hop.

|  |
| --- |
| מרגע שליחת העדכון ועד חזרת הא האק האחרון זה הזמן שלקח לרשת להתעדכן הוא 2.5225 שניות. אילו הכוונה היא לכמה זמן לקח לעדכון להגיע להופ הרחוק ביותר אז |
| נסתכל על הזמן שלקח לראוטר 8 להעביר את העדכון לראוטר 1: 1.7757 עד 1.7862 נקבל 0.0105 שניות בין הופסה להופלה |

9.17. Find out how *OSPF* calculates the *metric* for an interface. Explain what causes the difference in *metric* between the advertised entries. For that purpose, use the following command and explain how it can help to find the reason. *show interfaces FastEthernet* ***interface-number***

|  |  |
| --- | --- |
| להלן שלושת הממשקים של ראוטר 1.  ***show******interfaces*** |  |
|  |
|  |
| ***show******ip******ospf******database******router***  The Cost formula is reference bandwidth divided by interface bandwidth. The default reference bandwidth of 100 Mbps is used for OSPF cost calculation.  מהנוסחא הל"נ ניתן לראות כי המטריקה אל המחשב תהא 10 וגדולה פי 1 מהמטריקו בין הראוטרים. |  |

**10. Good News Propagation (Adding a Router)**

First make shure defulte gw is set:

|  |  |
| --- | --- |
| PC1  route add default gw 192.5.1.254 | PC2  route add default gw 192.5.2.254 |

10.2. Start all the network components, **except for R2**.

10.3. Use *traceroute* from *PC2* to *PC1* in order to verify that your network convened.

|  |
| --- |
| *traceroute 192.5.1.1* |

Text

Description automatically generated with low confidence

10.4. Start *Wireshark* on ***R4* interface *f1/0***, on ***R3* interface *f1/0***and on ***R1* interface *f1/0***.

10.5. Issue an infinite *ping* from *PC2* to *PC1*.

|  |
| --- |
| *ping 192.5.1.1* |

10.6. **Start router *R2***and wait for the network to converge. Notice that when we start *R2*, we provide a better route between the two *PCs*. Observe *Wireshark* on links *R3-R2* and *R2-R1* to see when the ICMP packets (the Request and the reply) start to pass through the new route.

10.7. Look at the routing table of *R1* and *R4* and verify that the network was convened.

|  |  |
| --- | --- |
| R1 | R4 |

10.8. Stop the ping command using **Ctrl+C** and save the ping statistics.

10.9. Stop capture and save the *wireshark pcaps* to files

----------------------

10.10. Did the network changes affect the ping application? If so, how many packets were lost?

הפינגים לא נאבדו   
Text

Description automatically generated

זאת משום שבכל רגע נתון הם הטבלאת ניתוב מכילה מסלולים מלאים.

10.11. Using the *Wireshark pcaps*, measure how long it takes the network to converge. Explain, how did you calculate this?

|  |
| --- |
| נמדוד זמן בין הודעת העדכון הראשונה של ראוטר 4:  ויד האק האחרון שהוא שלח:  זמן עד להתכנסות המערכת: |

10.12. At the pcap of subnet 10.x.3.0, follow the update packets that spread in the network because the router started. Use the display filter: “*ospf.msg.lsupdate and ip.dst == 224.0.0.5*”. Describe the update packets which spread in the network, and the reason for each one.

|  |
| --- |
| **97**: ראוטר5 מעביר הודעת עדכון על השכנים של ראוטר2 (בניהם ראוטר3 שממהר מיד אחרכך להעביר את הבשורה)  **99**: ראוטר5 מעביר עדכון מראוטר3 על השכנים החדשים שלו    **101:** כנ"ל מראוטר 7    וכן הלאה.. משידועות המטריקות של השכנים לראוטר 4 יש מספיק מידע בשביל לנתב את עצמו באופן המהיר ביותר. |

1. **Bad News Propagation (Router Down)**

Now, you **turn off a router** and see how *OSPF* reacts and how the network converges accordingly.

**Do:**

11.3. Start all the network components.

11.4. Use *traceroute* from *PC2* to *PC1* in order to verify that your network convened.

|  |
| --- |
| *traceroute 192.5.1.1* |

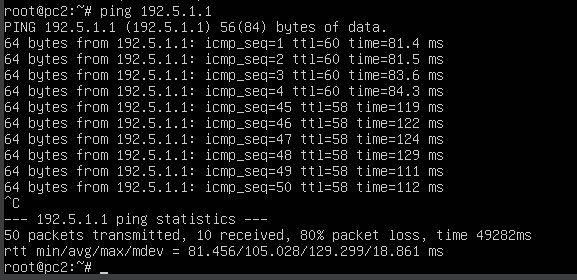
11.5. Issue an infinite *ping* from *PC2* to *PC1*.

|  |
| --- |
| *ping 192.5.1.1* |

11.6. Stop the router *R2*.

11.7. When the router stops, the ping fails. Wait for the network to converge until the ping returns to work properly. That will happen when an alternate path is found and may take several minutes.

11.8. Stop the ping command using **Ctrl+C** and save the ping statistics



11.9. Using the ping statistics, calculate the time it took *OSPF* to converge.

אנו יודעים כי פינג נשלח בכל שניה, וכן ידוע לנו כי נשלחו עשרה פינגים לאורך 50 שניות. אזי שלא נשלחו פיניגים במשך 40 שניות אז זה הזמן אשר לקח למערכת להתכנס מחדש.

11.10. Explain, what in the protocol supports the waiting time you calculated?

בשיטת OSPF בין הראוטרים נשלח hello בכל 10 שניות. הdead interval הוא פרק זמן ארוך פי 4 כלומר 40 שניות שאם בהן ראוטר מסויים לא מעביר hello אז הוא בטח מת או הלך לישון או שהוא בעונש, בכל אופן צריך למצוא לו אלטרנטיבה. 40 שניות שקיבלנו להתכנסות כנראה בגלל שכיבינו את הראוטר ממש רגע לפני השליחה הבאה של הhello שלו, וזה "קיצר" אז זמן האינטרוול המת. שכן ראינו שמספיקות לנו שניות מעטות מרגע הבנת הבעיה ועד להתכנסות בפועל.

1. **Hierarchical Routing in OSPF**

**DO:**

12.2. Start *Wireshark* on *R4* interface *f1/0*.   
  
12.3. Enable *OSPF* on all routersConfigure the routers in this order: ***R4, R5, R3, R6, R7, R2, R8, R1***. Note that the configuration of the border routers is a little different. Follow the examples.   
In *OSPF*, networks can be divided into areas, to significantly reduce the amount of topological information routers have to learn. In *OSPF* all areas must be connected to *Area 0* which is known as the *backbone* area.   
  
You will now define *Hierarchical OSPF* with two areas (*area 1* and *area 2*) connected through a *backbone* area (*area 0*). Routers that connect two areas are called *area border routers*.

|  |
| --- |
| **R4** router inside *area 1*:  *configure terminal*  *router ospf 1*  *router-id 10.5.0.4*  *network 10.5.0.0 0.0.255.255 area 1*  *end* |
| **R5** router inside *area 1*:  *configure terminal*  *router ospf 1*  *router-id 10.5.0.5*  *network 10.5.0.0 0.0.255.255 area 1*  *end* |
| **R3** border router:  *configure terminal*  *router ospf 1*  *router-id 10.5.0.3*  *network 10.5.3.0 0.0.0.255 area 1*  *network 10.5.2.0 0.0.0.255 area 0*  *network 10.5.7.0 0.0.0.255 area 0*  *end* |
| **R6** border router:  *configure terminal*  *router ospf 1*  *router-id 10.5.0.6*  *network 10.5.4.0 0.0.0.255 area 1*  *network 10.5.5.0 0.0.0.255 area 0*  *network 10.5.7.0 0.0.0.255 area 0*  *end* |
| **R7** border router:  *configure terminal*  *router ospf 1*  *router-id 10.5.0.7*  *network 10.5.5.0 0.0.0.255 area 0*  *network 10.5.6.0 0.0.0.255 area 0*  *network 10.5.8.0 0.0.0.255 area 2*  *end* |
| **R2** border router:  *configure terminal*  *router ospf 1*  *router-id 10.5.0.2*  *network 10.5.2.0 0.0.0.255 area 0*  *network 10.5.6.0 0.0.0.255 area 0*  *network 10.5.1.0 0.0.0.255 area 2*  *network 10.5.10.0 0.0.0.255 area 2*  *end* |
| **R8** router inside *area 2*:  *configure terminal*  *router ospf 1*  *router-id 10.5.0.8*  *network 10.5.0.0 0.0.255.255 area 2*  *end* |
| **R1** router inside *area 2*:  *configure terminal*  *router ospf 1*  *router-id 10.5.0.1*  *network 10.5.0.0 0.0.255.255 area 2*  *end* |

12.6. Wait for the network convergence.

12.7. Stop capture and save the *wireshark pcap* to file.

12.8. You should now be able to ping any router from any other router. Test your network by using *ping* from some router to all the others.

|  |  |
| --- | --- |
| R1  R2  R3  R4  R5  R6  R7  R8 | *ping 10.5.1.1*  *ping 10.5.1.2*  *ping 10.5.2.2*  *ping 10.5.3.2*  *ping 10.5.3.3*  *ping 10.5.4.2*  *ping 10.5.6.1*  *ping 10.5.9.1* |

12.9. **Perform** "*write*" on each router, save the topology (as *Topology 10*) and ZIP it.

ענה על השאלות הבאות הדבק לדוח:

Look at the *Wireshark pcap* using the display filter:

“*ospf.msg.lsupdate and ip.dst == 224.0.0.5*”.

12.11. Using the *Wireshark pcap*, describe which information routers in area 1 received about area 2 and which information they received about the backbone area.

|  |
| --- |
| בפקטה הנ"ל ניתן לראות שראוטר5 מקבל מידע אודות לינק 10.5.7.0 שאליו יש להגיע דרך ראוטר6, הפקטה אינה מכילה מידע אודות קיומה של רשת0. |
| נסתכל על פקטה אחרת שמדווחת על לינק 10.5.10.3 השייך לאיזור 2, הפקטה מלמד כי אל הלינק יש להגיע דרך ראוטר3 אך אינה מכירה את איזור2 ואינה מכירה את הרואטרים השייכם אליו |
| נמצא כי ראוטרים אשר אינם שיייכים לאותו אזור לא מכירים זה את זה או אזורים אחרים, הם יודעים רק דרך איזה ראוטר עליהם לגשת ללינקים אחרים. |

12.12. Locate on the *Wireshark pcap* the *Router-LSA* update that *R3* advertised. Which interfaces it advertises? Why?

|  |
| --- |
| ראוטר3 מכיר ומפרסם את הרשות המקושרות אליו, נתבונן בפקטה הבאה: |
| הראוטר מפרסם את הרשתות המחוברות אליו דרך ממשקים 1/0 ו 3/0  למה? |

12.13. Display the *OSPF database* of *R4* and compare it to the one you saved earlier, at exercise “Configuring OSPF on Cisco Routers”. Discuss the differences.

|  |  |
| --- | --- |
| Configuring OSPF on Cisco Routers: | Hierarchical Routing in OSPF: |
| ההבדלים: 1. כאשר הOSPF הוא היררכי ראוטר4 אינו מכיר באופן אישי את כל הראוטרים ברשת, הוא מכיר רק את אלו שחולקים לינקים מהאיזור שלו.  2. הניתוב שהראוטר מכיר הוא עד לנקודה בה מתחלף האיזור, הראוטר מכיר את הניתוב לכל הליניקים עד הנקודה בה החבילה יוצאת מהאיזור שלו. | |