מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

שאלה מספר 1:

UDP Server

```
1  from socket import socket, AF_INET, SOCK_DGRAM
2
3  s = socket(AF_INET, SOCK_DGRAM)
4  source_ip = '0.0.0.0'
5  source_port = 12345
6  s.bind((source_ip, source_port))
7  while True:
8      data, sender_info = s.recvfrom(2048)
9      print "Message: ", data, " from: ", sender_info
10     s.sendto(data.upper(), sender_info)
```

1. מייבא ממודל הSocket את פונקציות פתיחת הסוקט מsocket את סוג כתובות האייפי שאנחנו נתעסק בהן, IPv4 מAF_INET וSOCK_DGRAM נותנן לנו את פרוטוקל הUDP שבו נשתמש.
3. נפתח סוקט חדש שמקבל כתובות אייפי מסוג IPv4 , שעובד בפרוטוקל UDP.
4,5. נגדיר את כתובת האייפי להיות 0.0.0.0, כלומר השרת מאזין על כל כתובות האייפי של כל כרטיסי הרשת הלוקליים. ונבחר פורט רנדומלי שאינו בשימוש להאזין לו (12345).
6. נקשר את כתובת האייפי והפורט שבחרנו לסוקט שפתחנו כדי שנאזין דרכו.
7. נריץ לולאה אינסופית בכדי להריץ את השרת ללא הפסקה (אלא אם כן סגרנו בכח את התכנית)
8. נשמור את המידע שמתקבל לסוקט (נאפשר לו לקבל עד 2048 בייטים), הdata זו ההודעה עצמה, בסטרינג, וsender_info זו הכתובת שממנה התקבלה ההודעה.
9. נדפיס את ההודעה שהתקבלה ואת כתובת המקור שלה
10. כעת נשלח את אותה ההודעה, לאחר שנמיר אותה לאותיות גדולות, חזרה אל השולח.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

UDP Client

```
1  from socket import socket, AF_INET, SOCK_DGRAM
2
3  s = socket(AF_INET, SOCK_DGRAM)
4  dest_ip = '127.0.0.1'
5  dest_port = 12345
6  msg = raw_input("Message to send: ")
7  while not msg == 'quit':
8      s.sendto(msg, (dest_ip, dest_port))
9      data, sender_info = s.recvfrom(2048)
10     print "Server sent: ", data
11     msg = raw_input("Message to send: ")
12 s.close()
```

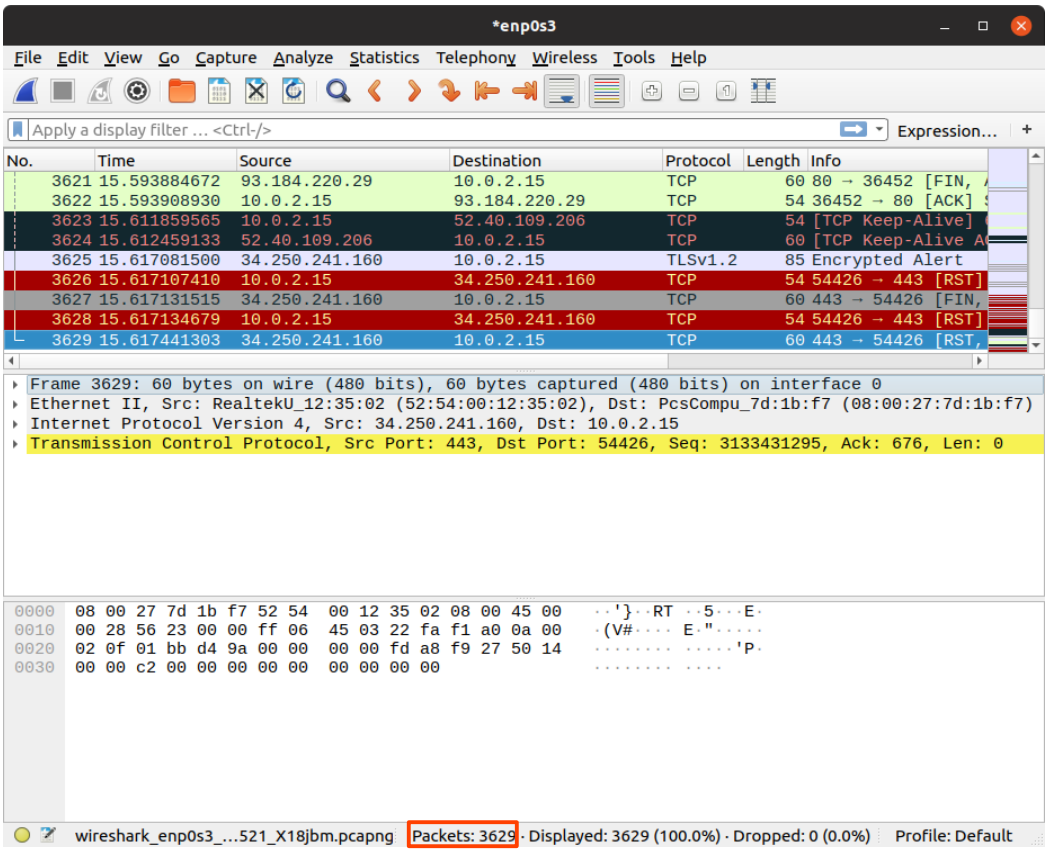
1. מייבא ממודל הSocket את פונקציות פתיחת הסוקט מsocket את סוג כתובות האייפי שאנחנו נתעסק בהן, IPv4 מAF_INET וSOCK_DGRAM נותנן לנו את פרוטוקל הUDP שבו נשתמש.
3. נפתח סוקט חדש שמקבל כתובות אייפי מסוג IPv4 , שעובד בפרוטוקל UDP.
4,5. כתובת האייפי של שרת היעד שלנו, במקרה זה היא כתובת הloopback כיוון שהשרת נמצא על אותה המכונה. כמו כן כתובת הפורט ששרת היעד מאזינה עליה.
6. נקבל כאינפוט מהמשתמש את ההודעה שהוא רוצה לשלוח.
7. כעת, עד שלא נשלח הודעה של quit, התוכנה תבקש מהמשתמש הודעות, ותשלח אותן לשרת.
8. אנחנו שולחים דרך הסוקט את ההודעה. נעביר לפונקציה השליחה כארגומנטים את ההודעה ואת כתובת האייפי והפורט שהשרת מאזין דרכן.
9. נקבל דרך הסוקט שלנו את התשובה של השרת, נשמור אותו למשתנה הdata שהוא תוכן ההודעה וsender_info שהם פרטי השרת שענה לנו.
10,11. נדפיס את ההודעה שקיבלנו (שהיא ההודעה ששלחנו, מומרת לאותיות גדולות) ואז נבקש מהשתמש את ההודעה הבאה שהוא רוצה לשלוח ונחזור שוב על התהליך.
12. כאשר המשתמש שלח הודעת quit ויצאנו מהלולאה, נסגור את הסוקט ונפנה את הפורט שלנו כיוון שאין בו יותר שימוש, והתכנית נסגרה.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

שאלה מספר 2:

א.

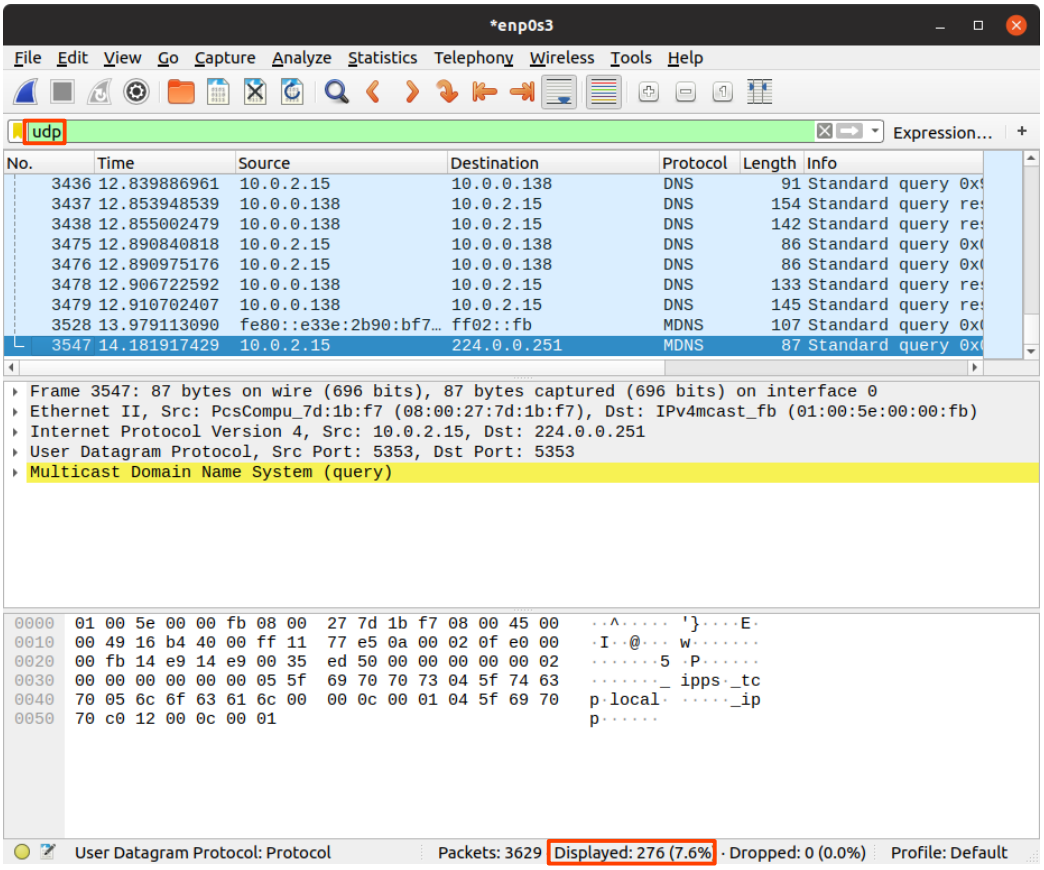


מספר החבילות שהוסנפו בפתיחת שני אתרים הוא 3629.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

ב.

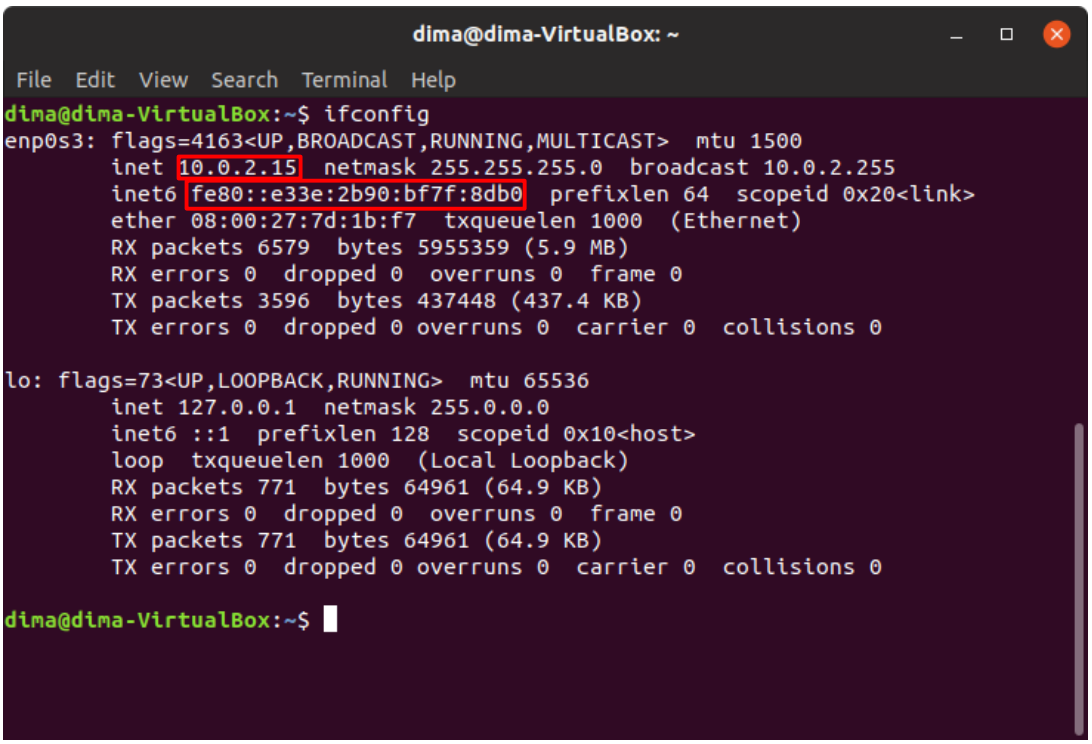


כאשר הפעלנו את הפילטר כדי שיראה רק חבילות שנשלחו בפרוטוקל UDP, ראינו שרק 7.6% מסך החיבלות נשלחו בפרוטוקל הזה (276 מתוך 3629). כמו כן אפשר לראות שהרוב המוחלט, אם לא כל החבילות, הן Queries לשרתי DNS.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

ג.

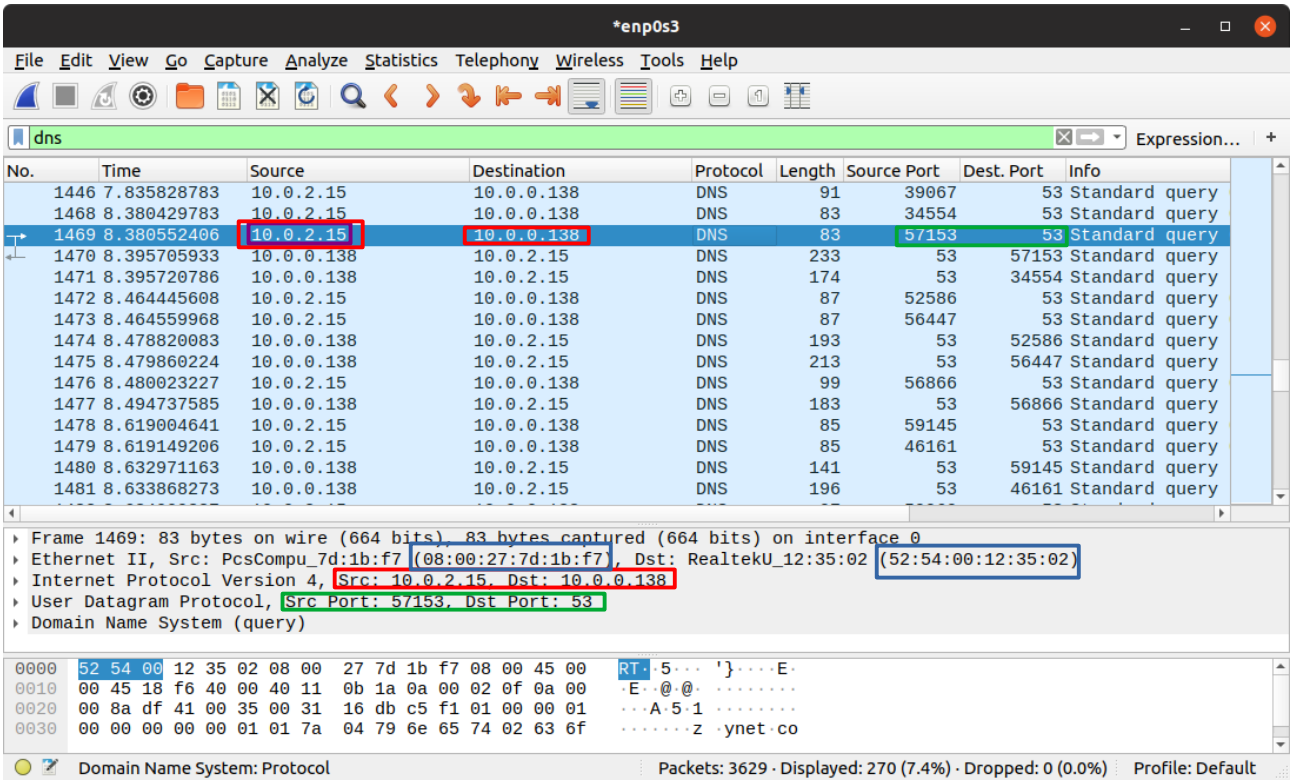


כתובת האייפי ברשת היא 10.0.2.15 כאשר מדובר על פרוטוקול IPv4 ו fe80::e33e:2b90:bf7f:8db0 כאשר מדובר בפרוטוקול IPv6.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

ד.



– החבילה שבחרתי נשלחה מהמחשב שלי אל השרת. ניתן לקבוע זאת כיוון שכתובת האייפי שמוגדרת כמקור החבילה היא כתובת האייפי שהראנו שמשוייכת אליי בסעיף הקודם (10.0.2.15).

– ניתן לראות שהחבילה נשלחה מפורט 57153 אל פורט 53. הלקוח האזין לפורט 57153, שהוא פורט המקור והשרת האזין לפורט 53. (הסופתי עמודות פורט מקור/יעד לשם נוחות)

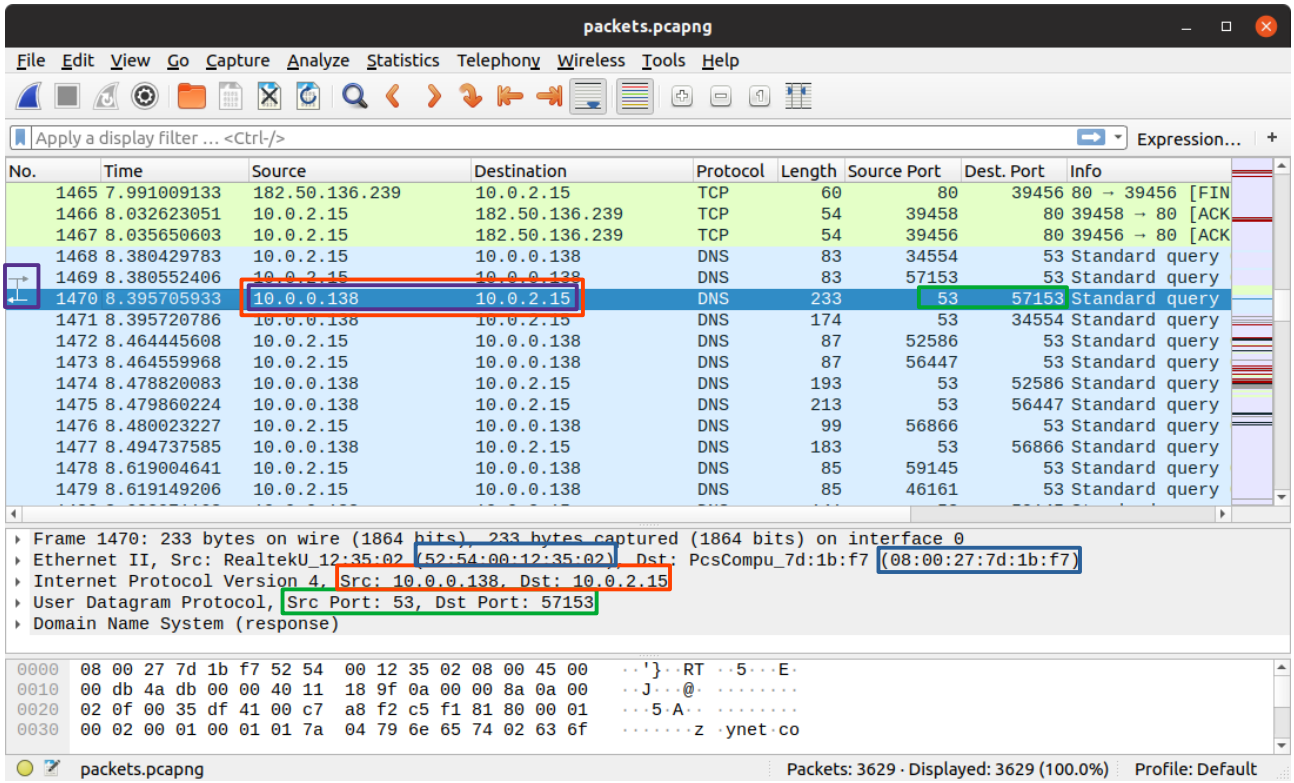
– כתובת האייפי של השולח היא 10.0.2.15 וכתובת האייפי של היעד היא 10.0.0.138

– כתובת הMAC של השולח היא 08:00:27:7d:1b:f7 והכתובת של היעד היא 52:54:00:12:35:02

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

ה.



– החבילה המתאימה לחבילה הקודמת היא זו שמסומנת, ניתן לראות שהחצים הם בין החבילה הקודמת לזו, וכן מקור ויעד האייפי והפורט הפוכים מהקודמת.

– הפורט שהחבילה נשלחה ממנו הוא פורט 53 והפורט שאליו החבילה נשלחה הוא פורט 57153.

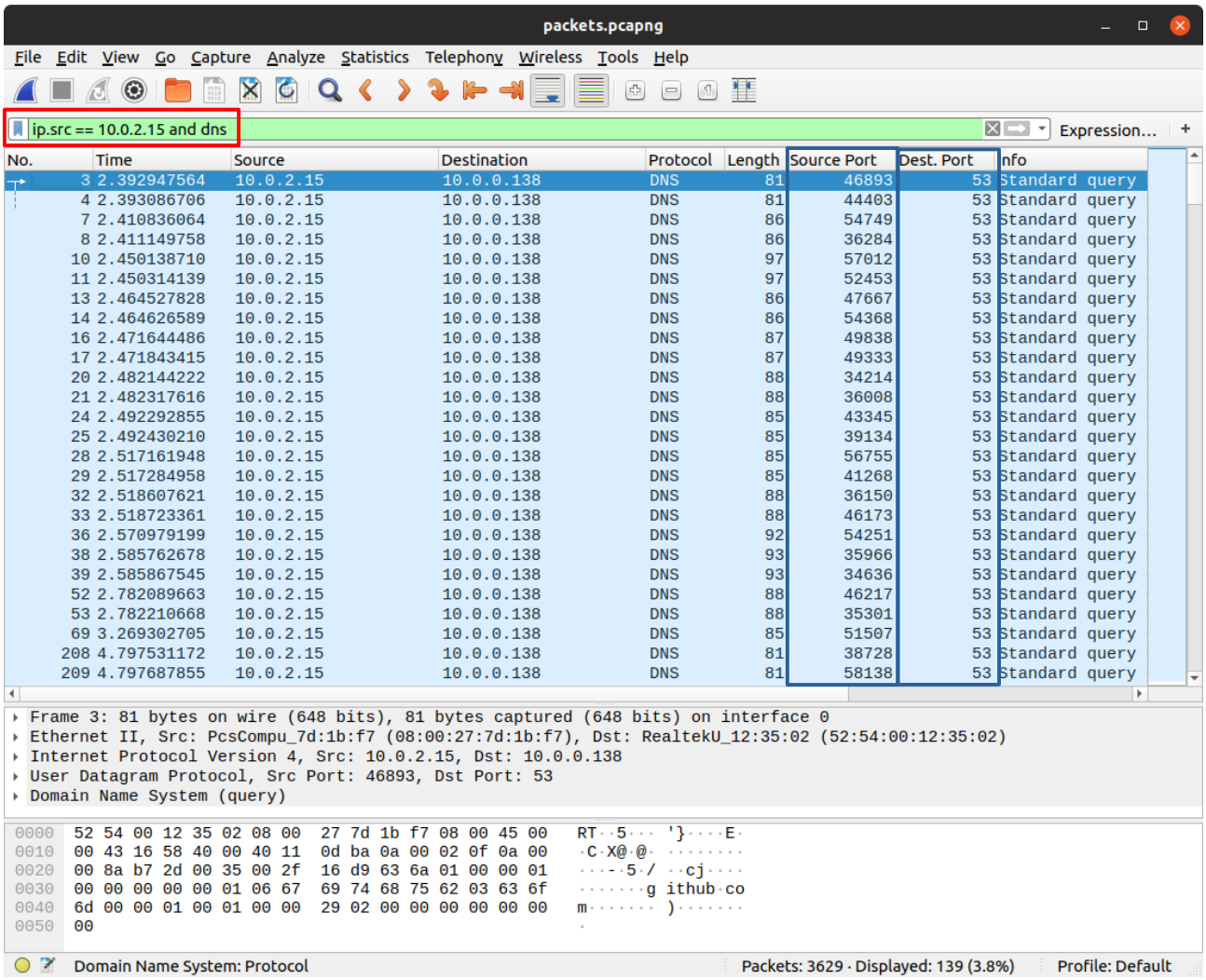
– כתובת האייפי של המקור היא 10.0.0.138 וכתובת האייפי של היעד היא 10.0.2.15

– כתובת הMAC של המקור היא 52:54:00:12:35:02 וכתובת היעד היא 08:00:27:7d:1b:f7

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

.I



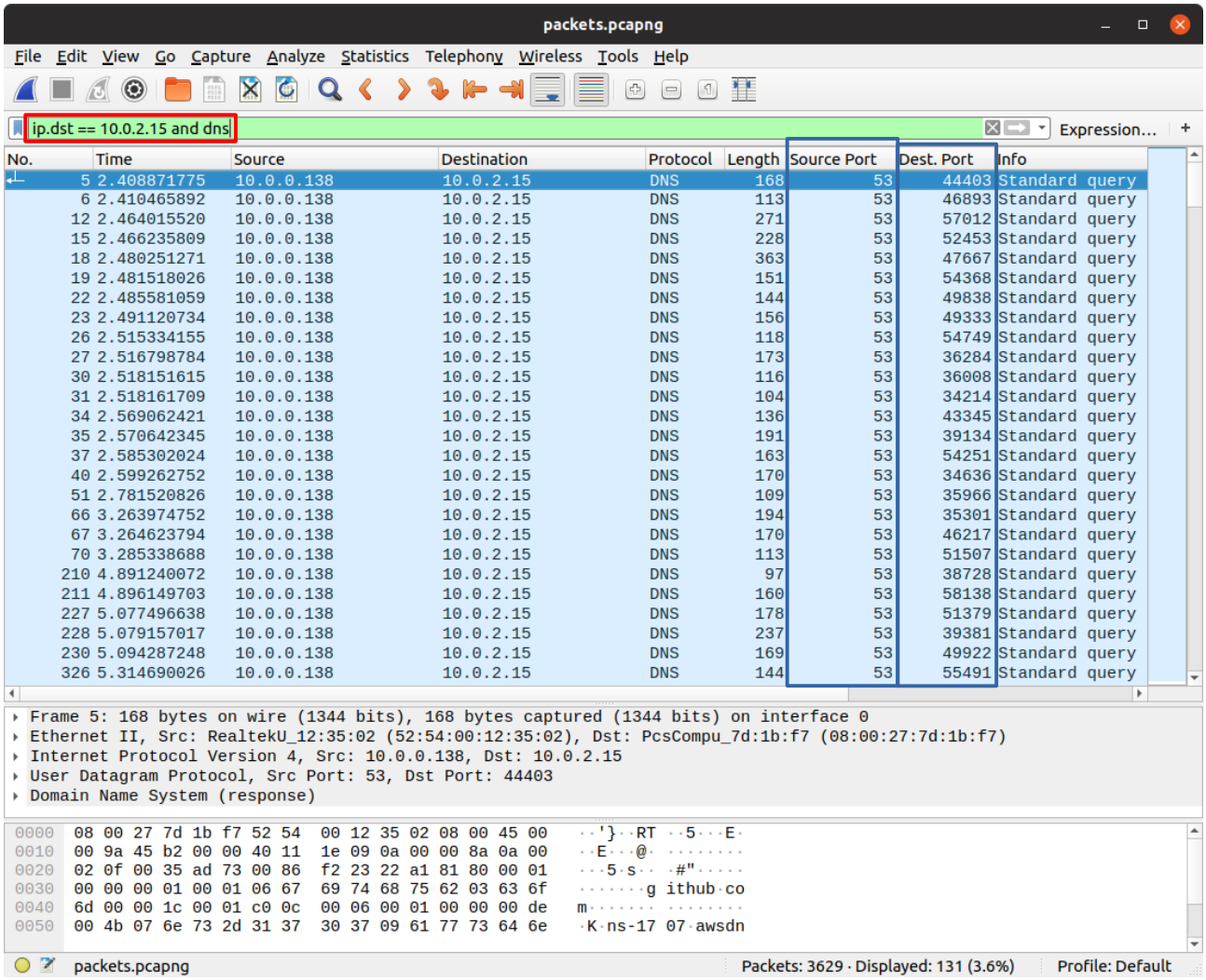
כעת סיננו לפי החבילות שנשלחו מהמחשב שלי (אייפי המקור הוא 10.0.2.15) ושהן בפרוטוקול DNS. ניתן לראות שבכל query שנשלח מהמחשב שלי אל היעד, היעד של שרת הDNS תמיד מאזין על פורט 53, ואילו כל שאילתא שאני שולח, נשלחת מסוקט עם פורט רנדומלי.

המשמעות כאמור היא שתעבורת הDNS query עוברת תמיד דרך פורט 53 כשהיא מגיעה לשרת או יוצאת ממנו.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

.I



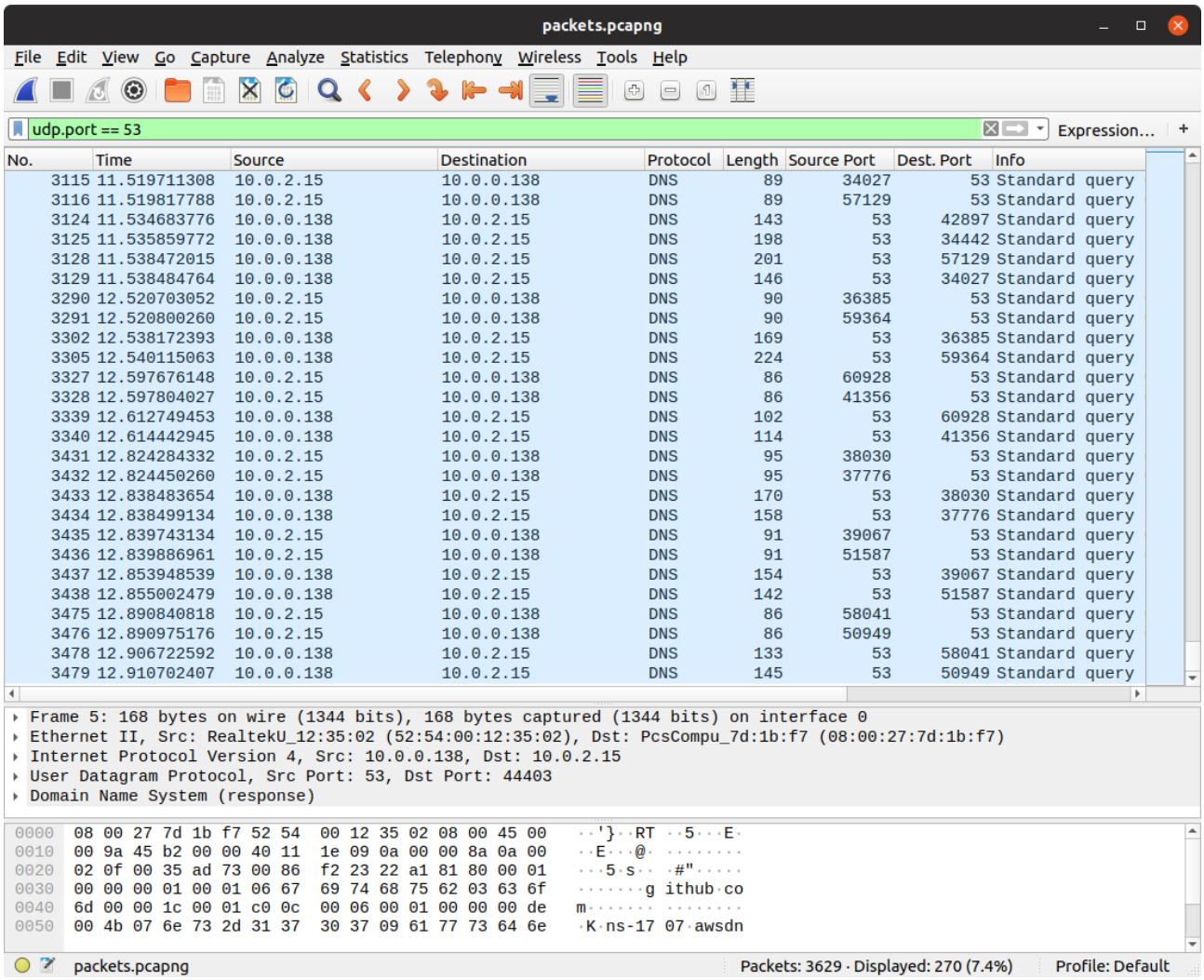
אם נסתכל בתעבורה כאשר כתובת האייפי שלנו היא היעד, ניתן לראות אשרור לטענה הקודמת. כל התעבורה של שאליתאות הDNS יוצאת מפורט 53, ומגיעה אלינו כאשר אנחנו מאזינים דרך פורט רנדומלי.

מוטב גם לציין כי כל חבילה בשאליתא עוברת בפורטוקול הUDP. לפי בדיקה באינטרנט מסתבר כי DNS Queries משתמשים בUDP כל עוד גודל החבילה לא עולה על 512 בייטים (וכפי שניתן לראות מהצילום, אף חבילה לא עברה את הגודל הזה – לפי עמודת הLength). אם והייתה שאליתא שהתשובה אליה גדולה מהגודל הזה, אז ככל הנראה היא הייתה עוברת בפרוטוקול TCP, אלא אם כן בוצע איזשהו extension לUDP המאפשר שליחה של חבילות גדולות יותר.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

ד.



משני הסעיפים הקודמים ניתן להסיק שנוכל לסנן את כל חבילות שנשלחו בפורטוקול DNS אם נסנן לפי כל החבילות שנשלחו ל או מפורט 53 לפי UDP. ואכן, אם נשווה את הצילום של הסינון למעלה אל הצילום בתת סעיף **ד'** נוכל לראות כי מספר החבילות המוצגות זהה.

מבוא לרשתות תקשורת – תרגיל 1

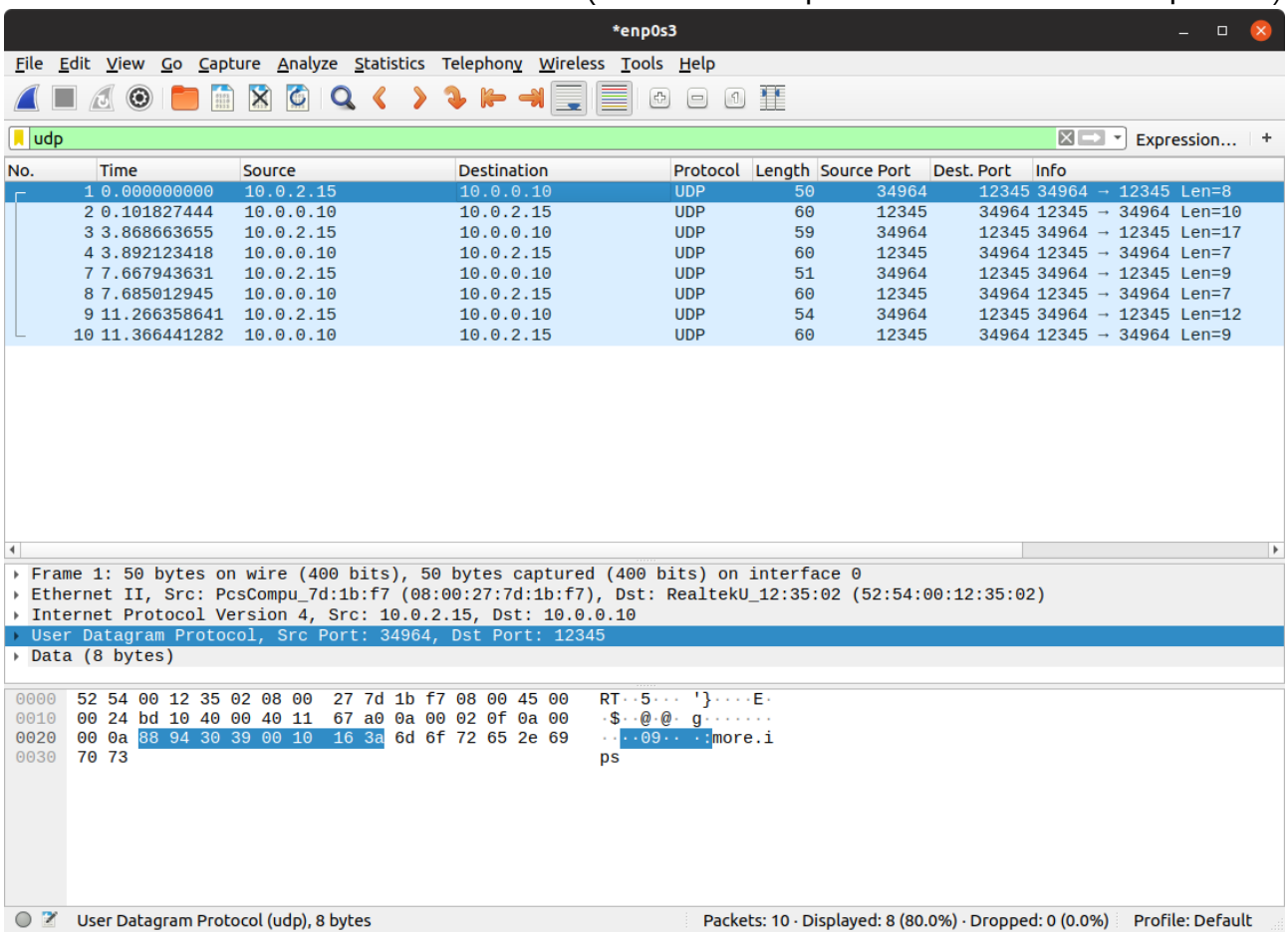
דמיטרי זינקביץ ת.ז 317301596

שאלה 3:

הרצתי את הClient על מכונה וירטואלית שמריצה Ubuntu על הדסקטופ, ואת צד השרת על לפטופ שמריץ גם הוא Ubuntu.

זו התעבורה שנתפסה בWireshark בצד הלקוח לאחר שליחת ארבעה DNS Queries:

(שתי פקטות לא רלוונטיות נתפסו לכן סיננתי לפי UDP)



– ניתן לראות שלאחר 4 queries קיבלנו 8 חבילות, כלומר לכל שאליתא יש חבילה ששולחת את הבקשה, וחבילה שמקבלים שמחזירה את התשובה.

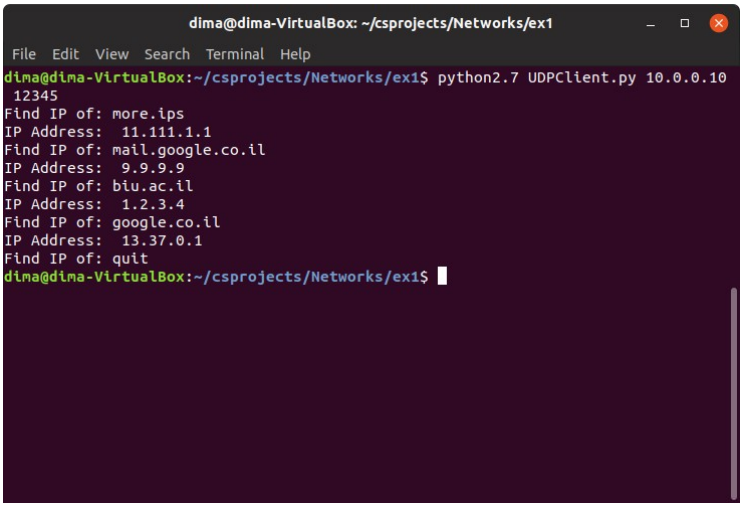
– פורט המקור הוא 34964, שזהו פורט רנדומלי שקושר לסוקט של הקליינט. הוא לא השתנה כיוון שלא סגרנו את הסוקט עד ליציאת התוכנה. במידה והיינו פותחים את הקליינט מחדש ומבצעים עוד queries, הן הינו נשלחות דרך פורט רנדומלי אחר. פורט היעד, 12345, הוא הפורט שנקבע ונשלח כארגומנט לצד השרת בעת פתיחת השרת.

– אייפי המקור הוא האייפי של המכונה הוירטואלית ברשת (10.0.2.15) ואייפי היעד (10.0.0.10) הוא האייפי של המחשב השני ברשת שהעברנו כארגומנט לקליינט אחרי שבדקנו מה האייפי עצמו ע"י הרצת ifconfig במחשב של השרת.

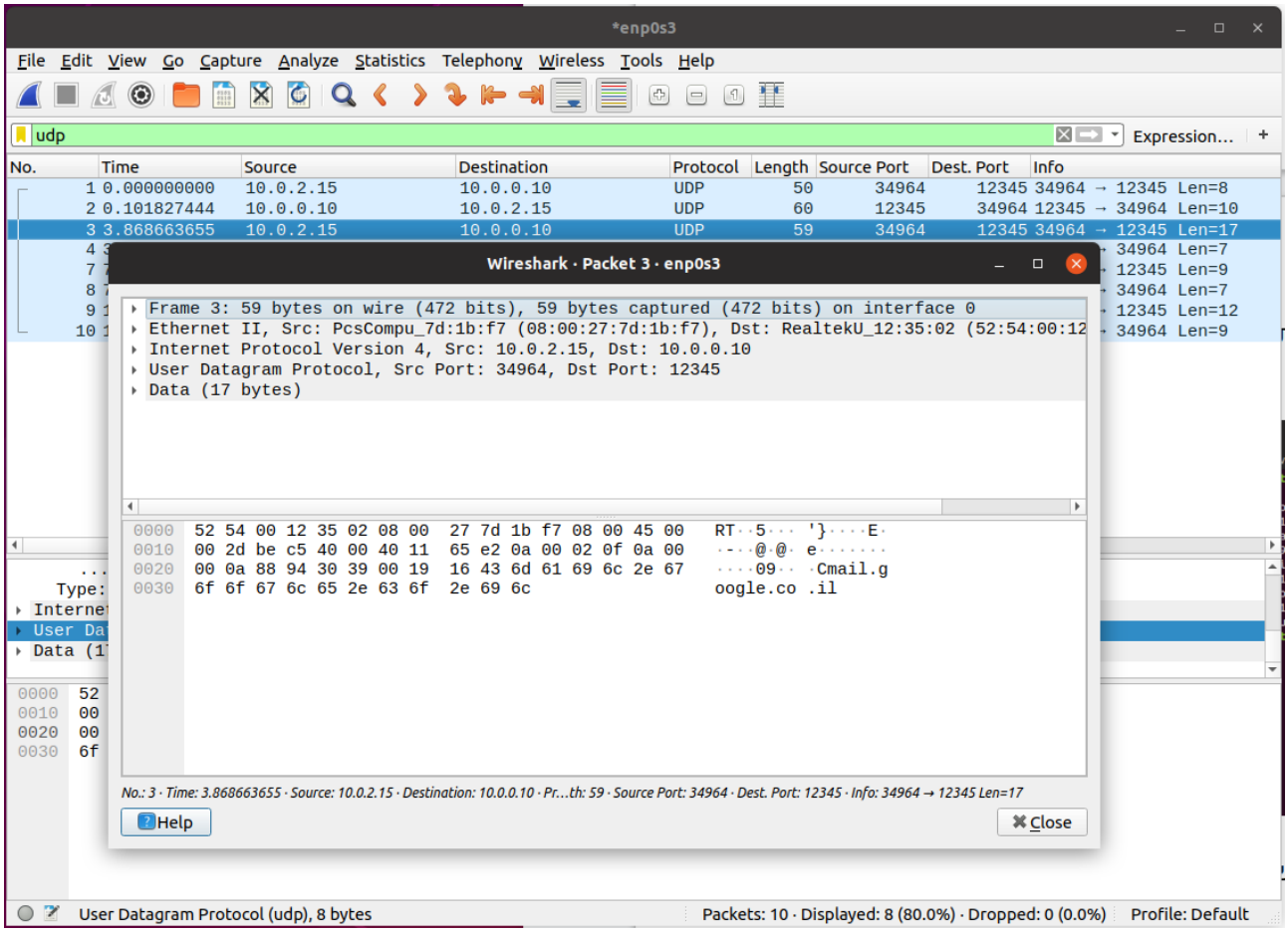
מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

הפלט של צד הלקוח הוא:



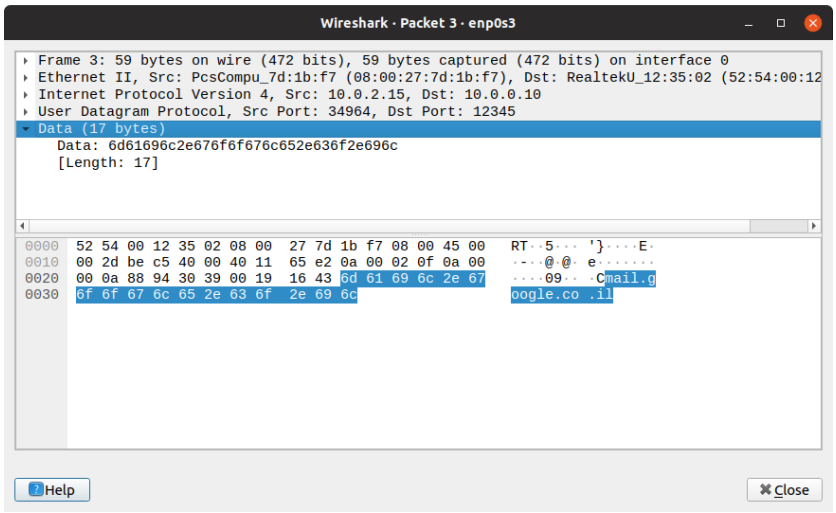
נתבונן יותר לעומק באחד מהQueries ובתשובה של השרת אליו:
נקח את החבילה השניה שנשלחה מצד הלקוח



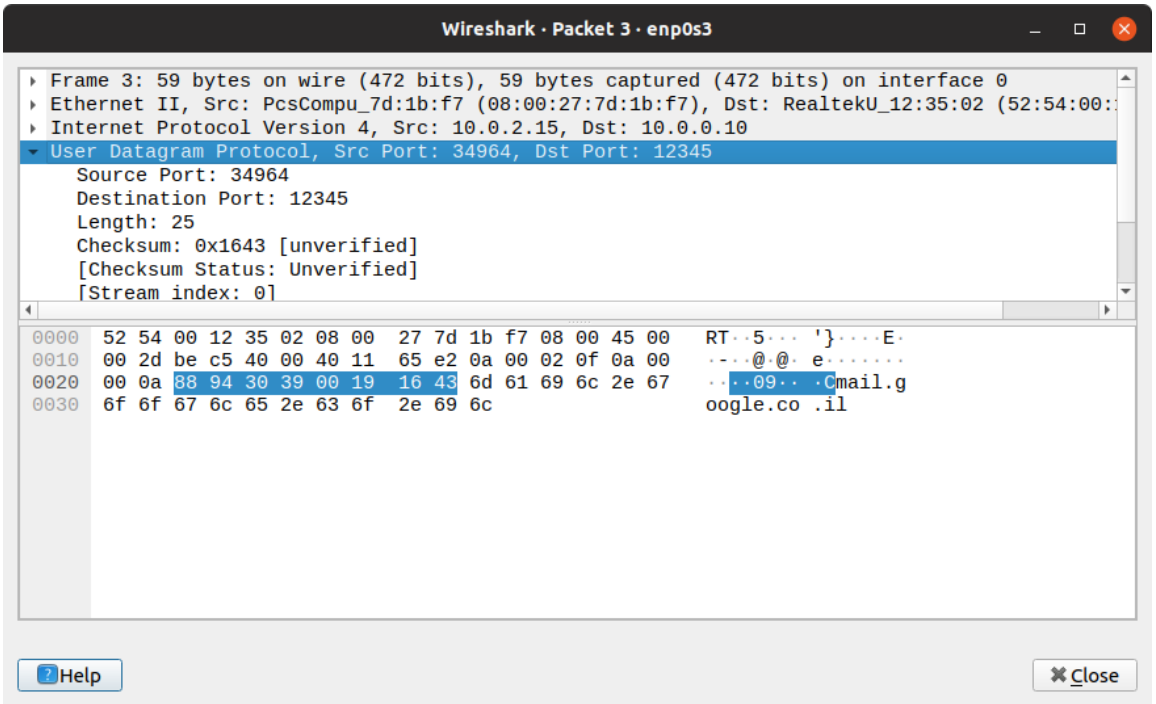
מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

ניתן לראות שהמידע שהחבילה מעבירה, במקרה שלנו הכתובת של האתר שאנחנו מעוניינים לקבל את האייפי שלה, היא החלק האחרון בפקטה שמורכבת מheaders שמורכבים עליה בכל שכבות התעבורה.



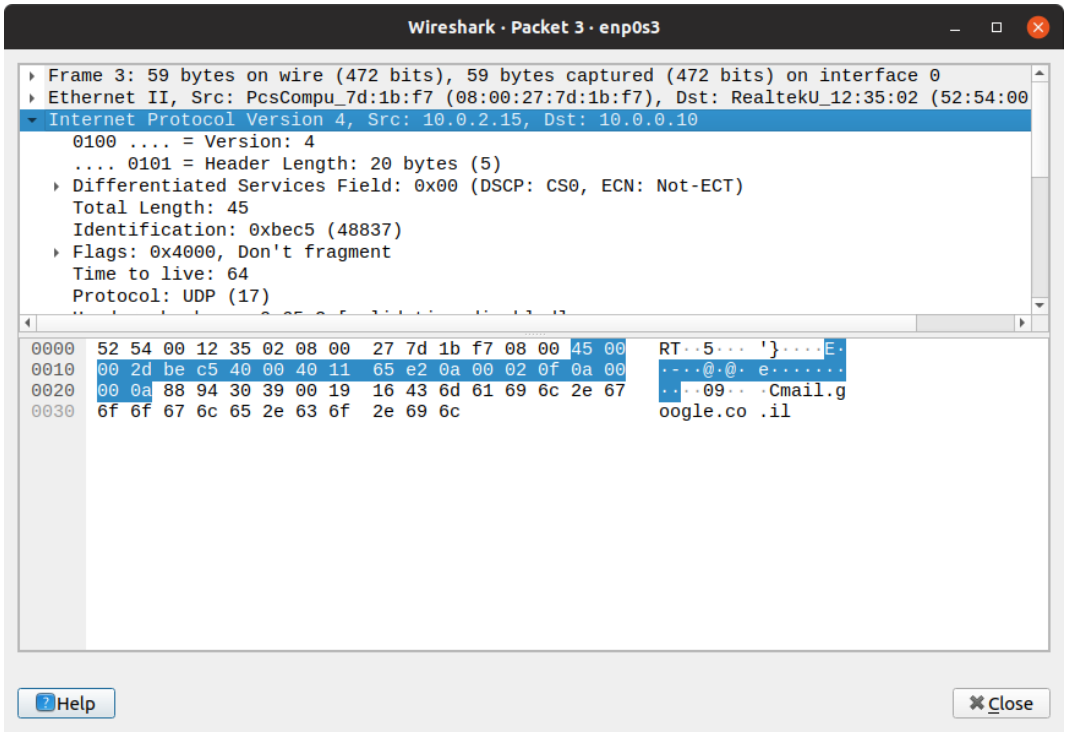
החלק של המידע מועבד בApplication Layer פשוט למחרוזת של בתים שתועבר.
לאחר מכן אנחנו יורדים לTransport Layer, בה מתווסף למידע שאנחנו שולחים האדר בו מוגדר שהוא נשלח בפרוטוקול UDP ומתווסף לו להאדר פורט המקור והיעד.
כמו כן מתווסף צקסאם של ההאדר וגודל החבילה בבייטים, כעת פיסת המידע נקראת Segment.



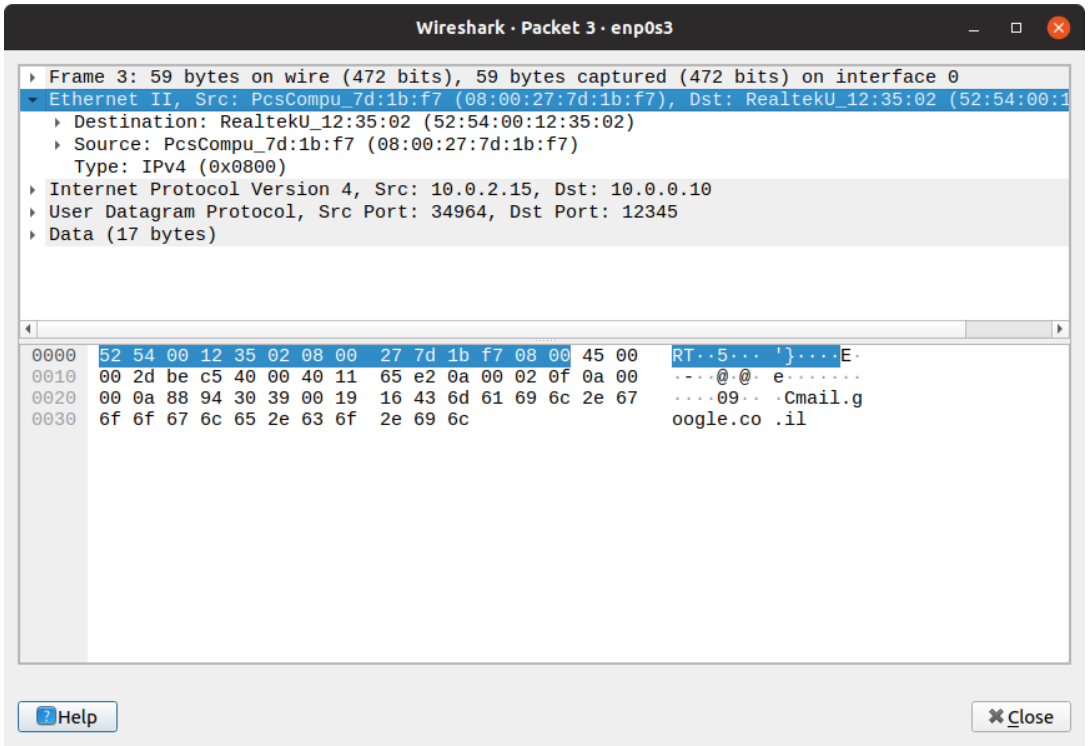
מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

אחרי שכבת התעבורה אנחנו עוברים לNetwork Layer. כעת מתווסף למידע שלנו האדר שמגדיר מאיזה IP ולאיזה IP המידע נשלח וכן האם מדובר בIPv4 או בIPv6, הSegment כעת נקרא Packet. ניתן לראות גם שבתוך השכבה מתווסף הTTL של החבילה.



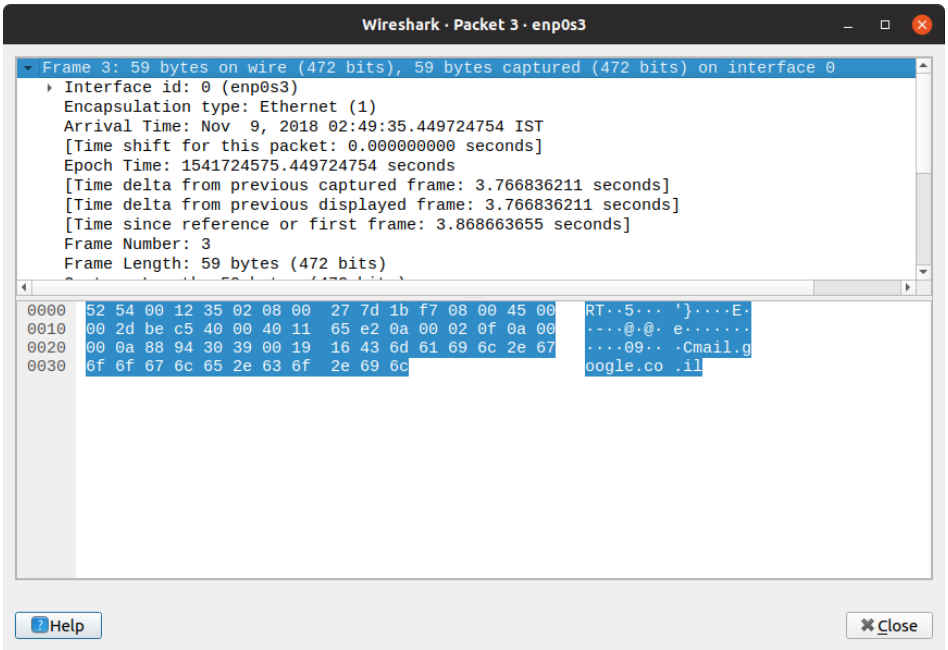
אחרי שכבת הרשת אנחנו עוברים לLink Layer. בשכבה זו מתבצע מיפוי לכתובות MAC. ניתן לראות את כתובת הMAC של המקור ושל היעד:



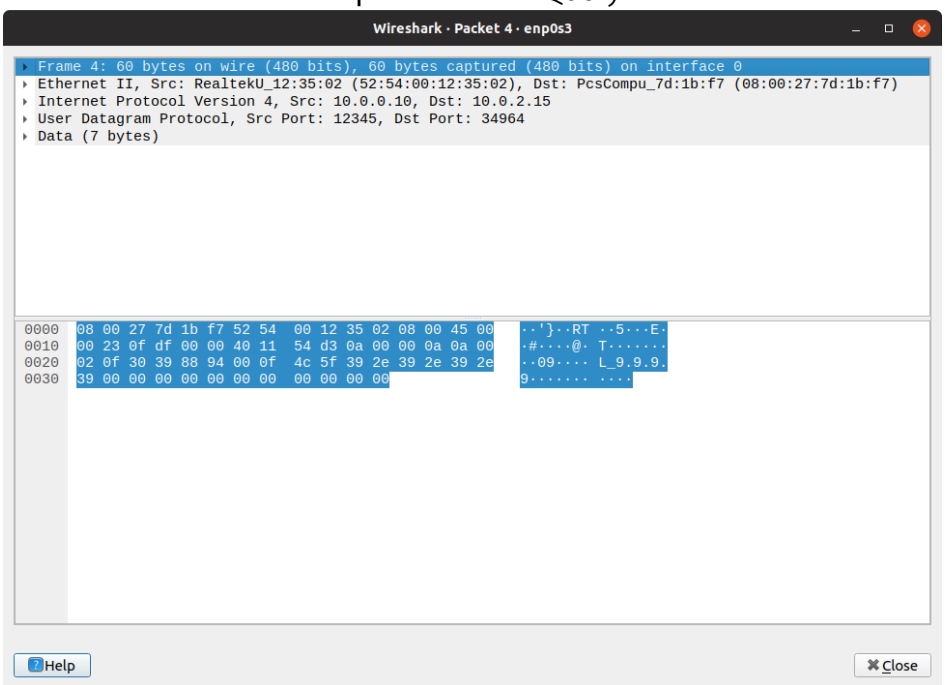
מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

לאחר שכבת הלינק נוצר לנו הFrame שמורכב מכל השכבות והוא זה שמועבר בסופו של דבר אל היעד שלנו.



נתבונן בחבילה שהיא התגובה לQuery שהחבילה הקודמת שלחה:



ניתן לראות שעיקר הדברים די דומה, החבילה מורכבת מהמידע (7 בייטים של המחרוזת של האייפי 9.9.9.9), היא נשלחת לפורט של הקליינט (34964) מהפורט של השרת (12345), כנ"ל לגבי כתובות האייפי והMAC.

אפשר בעצם להבין שסידור החבילה במודל הTCP/IP יוצר מעין היררכיה שמסדרת את המעבר של החבילות בין המקור ליעד, כך שזה מהווה סטנדרט מסוים בתהליך התקשורת בין השרת ללקוח.

תהליך הQuery and Answer הוא פשוט כמו שהוא נשמע, הלקוח שולח שאלה לשרת מסוים דרך חבילה, והשרת עונה לו באותה הצורה.