תרגיל 2:

חלק א:

הרצנו את קוד השרת והתחלנו להאזין.

לאחר מכן הרצנו את קוד הלקוח. ניתן לראות ב ווירשרק כי התבצעה לחיצת שלושת הידיים.
ניתן לראות כי התקבל "syn,syn ack, ack ".

תחילה הלקוח שולח ' syn' ו-sequence number שווה 0, לאחר מכן השרת מחזיר syn,ack,
כאשר ה-ack number שווה 1, ולבסוף הלקוח מחזיר ack עם sequence number שווה 1 ו-
ack number שווה 1.

tcp.port == 6542 Expression... +

| No. | Time | Source | Destination | Prot | Length | Info |
|---|---|---|---|---|---|---|
| 21 | 5.104981787 | 192.168.43.66 | 192.168.43.236 | TCP | 76 | 52618 → 6542 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM… |
| 22 | 5.111224470 | 192.168.43.236 | 192.168.43.66 | TCP | 76 | 6542 → 52618 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=146… |
| 23 | 5.111292774 | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 52618 → 6542 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=346770… |

כעת, הלקוח שולח הודעת באורך 15.

ניתן לראות כי השרת דבר ראשון מחזיר ack ששווה 16 בגלל שאורך ההודעה היה 15 +1 של
ה - ack מקודם.

מיד לאחר שליחת ה - ack השרת מחזיר את ההודעה של הלקוח באותיות גדולות.

כנהוג ב- TCP הלקוח גם שולח ack, וניתן לראות את ה- sequence number של הלקוח כעת
שווה ל-16, בגלל שהשרת שלח הודעה באורך 15.

| No. | Time | Source | Destination | Prot | Length | Info |
|---|---|---|---|---|---|---|
| 21 | 5.104981… | 192.168.43… | 192.168.43… | T… | 76 | 52618 → 6542 [SYN] Seq=0 Win=64240 Len=0 … |
| 22 | 5.111224… | 192.168.43… | 192.168.43… | T… | 76 | 6542 → 52618 [SYN, ACK] Seq=0 Ack=1 Win=6… |
| 23 | 5.111292… | 192.168.43… | 192.168.43… | T… | 68 | 52618 → 6542 [ACK] Seq=1 Ack=1 Win=64256 … |
| 74 | 25.57362… | 192.168.43… | 192.168.43… | T… | 83 | 52618 → 6542 [PSH, ACK] Seq=1 Ack=1 Win=6… |
| 75 | 25.63461… | 192.168.43… | 192.168.43… | T… | 68 | 6542 → 52618 [ACK] Seq=1 Ack=16 Win=65152… |
| 76 | 25.63778… | 192.168.43… | 192.168.43… | T… | 83 | 6542 → 52618 [PSH, ACK] Seq=1 Ack=16 Win=… |
| 77 | 25.63782… | 192.168.43… | 192.168.43… | T… | 68 | 52618 → 6542 [ACK] Seq=16 Ack=16 Win=642… |

```
▸ Frame 77: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.43.66, Dst: 192.168.43.236
▸ Transmission Control Protocol, Src Port: 52618, Dst Port: 6542, Seq: 16, Ack: 16, Len: 0
```

כעת הלקוח שולח את תעודת הזהות שלו.

השרת שולח לו גם את ה ack על ההודעה, כפי שמצופה מ - TCP, וביחד עם זאת שולח לו גם
את ה - ack המספר תעודת זהות "באותיות גדולות". ניתן לראות כי ה- ack number של
השרת הפך להיות 25 (כי תעודת זהות היא באורך 9).

הלקוח מחזיר לשרת ack על ההודעה, ולכן ניתן לראות כי גם ה-ack number שלו וגם ה-
sequence number

שלו הם כרגע 25.

```
No.  Time        Source        Destination   Prot Length Info
  21 5.104981… 192.168.43… 192.168.43… T…   76 52618 → 6542 [SYN] Seq=0 Win=64240 Len=0 …
  22 5.111224… 192.168.43… 192.168.43… T…   76 6542 → 52618 [SYN, ACK] Seq=0 Ack=1 Win=6…
  23 5.111292… 192.168.43… 192.168.43… T…   68 52618 → 6542 [ACK] Seq=1 Ack=1 Win=64256 …
  74 25.57362… 192.168.43… 192.168.43… T…   83 52618 → 6542 [PSH, ACK] Seq=1 Ack=1 Win=6…
  75 25.63461… 192.168.43… 192.168.43… T…   68 6542 → 52618 [ACK] Seq=1 Ack=16 Win=65152…
  76 25.63778… 192.168.43… 192.168.43… T…   83 6542 → 52618 [PSH, ACK] Seq=1 Ack=16 Win=…
  77 25.63782… 192.168.43… 192.168.43… T…   68 52618 → 6542 [ACK] Seq=16 Ack=16 Win=6425…
   1… 39.69211… 192.168.43… 192.168.43… T…   77 52618 → 6542 [PSH, ACK] Seq=16 Ack=16 W…
 117 39.70330… 192.168.43… 192.168.43… T…   77 6542 → 52618 [PSH, ACK] Seq=16 Ack=25 …
 118 39.70332… 192.168.43… 192.168.43… T…   68 52618 → 6542 [ACK] Seq=25 …k=25 Win=6425…

▸ Frame 116: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.43.66, Dst: 192.168.43.236
▾ Transmission Control Protocol, Src Port: 52618, Dst Port: 6542, Seq: 16, Ack: 16, Len: 9
    Source Port: 52618
    Destination Port: 6542
    [Stream index: 2]
    [TCP Segment Len: 9]
    Sequence number: 16    (relative sequence number)
    [Next sequence number: 25    (relative sequence number)]
    Acknowledgment number: 16    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 502

0000  00 04 00 01 00 06 e0 94  67 98 28 3a 00 00 08 00   ····· g·(:····
0010  45 00 00 3d 23 01 40 00  40 06 3f 3b c0 a8 2b 42   E··=#·@· @·?;··+B
0020  c0 a8 2b ec cd 8a 19 8e  65 42 6e b8 73 cc f1 a7   ··+····· eBn·s···
0030  80 18 01 f6 4d 97 00 00  01 01 08 0a ce b1 7e b8   ····M··· ·····~·
0040  64 88 6d 56 32 30 35 37  38 31 39 36 36            d·mV2057 81966
```

לבסוף הלקוח מתנתק וניתן לראות, לפי פורמט "TCP – FYN ACK, FYN ACK, ACK בין הלקוח לשרת. ניתן לראות כי דגל ה - FYN נדלק. כמו כן, השרת מחזיר ללקוח על הודעת הסיום, ack number שווה 26 ולכן בסוף גם הלקוח מחזיר לשרת sequence number שווה 26.

```
 118 39.70332… 192.168.43… 192.168.43… T…   68 52618 → 6542 [ACK] Seq=25 Ack=25 Win=64256.
 226 45.42287… 192.168.43… 192.168.43… T…   68 52618 → 6542 [FIN, ACK] Seq=25 Ack=25 Win=.
   2… 45.42916… 192.168.43… 192.168.43… T…   68 6542 → 52618 [FIN, ACK] Seq=25 Ack=26 …
 228 45.42932… 192.168.43… 192.168.43… T…   68 52618 → 6542 [ACK] Seq=26 Ack=26 …64256.

▸ Frame 227: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.43.236, Dst: 192.168.43.66
▾ Transmission Control Protocol, Src Port: 6542, Dst Port: 52618, Seq: 25, Ack: 26, Len: 0
    Source Port: 6542
    Destination Port: 52618
    [Stream index: 2]
    [TCP Segment Len: 0]
    Sequence number: 25    (relative sequence number)
    [Next sequence number: 25    (relative sequence number)]
    Acknowledgment number: 26    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
  ▾ Flags: 0x011 (FIN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
    ▸ .... .... ...1 = Fin: Set

0000  00 00 00 01 00 06 7c 67  a2 bd 54 f7 47 1b 08 00   ·····|g ··T·G···
0010  45 00 00 34 d2 af 40 00  40 06 8f 95 c0 a8 2b ec   E··4··@· @·····+·
0020  c0 a8 2b 42 19 8e cd 8a  73 cc f1 b0 65 42 6e c2   ··+B···· s···eBn·
```

חלק א

סעיף ב

: V1

הרצנו את קוד הלקוח והמשתמש. הלקוח שולח – 'Hello World' ומקבל את אותה הודעה מהשרת רק עם אותיות גדולות ולאחר מכן מתנתק.

קוד השרת הוא לקבל את ההודעה מהלקוח ולהחזיר אותה באותיות גדולות.

ניתן לראות בריצה את 'לחיצת שלושת הידיים, בין הלקוח לשרת. לאחר מכן ניתן לראות כי הלקוח שולח

הודעה, והשרת מחזיר לו ack. לאחר מכן השרת שולח את ההודעה של הלקוח רק באותיות גדולות

לבסוף הלקוח מתנתק.

| No. | Time | Source | Destination | Protoc | Length | Info |
|---|---|---|---|---|---|---|
| 485 | 27.936679… | 192.168.43.66 | 192.168.43.236 | TCP | 76 | 59770 → 4521 [SYN] Seq=0 Win=64240 Len=0 M |
| 486 | 27.939881… | 192.168.43.236 | 192.168.43.66 | TCP | 76 | 4521 → 59770 [SYN, ACK] Seq=0 Ack=1 Win=65 |
| 487 | 27.939910… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59770 → 4521 [ACK] Seq=1 Ack=1 Win=64256 L |
| 488 | 27.939963… | 192.168.43.66 | 192.168.43.236 | TCP | 81 | 59770 → 4521 [PSH, ACK] Seq=1 Ack=1 Win=64 |
| 489 | 27.944359… | 192.168.43.236 | 192.168.43.66 | TCP | 68 | 4521 → 59770 [ACK] Seq=1 Ack=14 Win=65152 |
| 490 | 27.944412… | 192.168.43.236 | 192.168.43.66 | TCP | 81 | 4521 → 59770 [PSH, ACK] Seq=1 Ack=14 Win=6 |
| 491 | 27.944448… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59770 → 4521 [ACK] Seq=14 Ack=14 Win=64256 |
| 492 | 27.944692… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59770 → 4521 [FIN, ACK] Seq=14 Ack=14 Win= |
| 493 | 27.949002… | 192.168.43.236 | 192.168.43.66 | TCP | 68 | 4521 → 59770 [FIN, ACK] Seq=14 Ack=15 Win= |
| 494 | 27.949142… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59770 → 4521 [ACK] Seq=15 Ack=15 Win=64256 |

: V2

הרצנו את קוד הלקוח והמשתמש.

קוד הלקוח הפעם הוא לשלוח Hello, Hello World לקבל הודעה מהשרת ולהתנתק.

קוד השרת הוא לקבל להאזין ולהחזיר הודעה באותיות גדולות לפי מה שקיבל.

ניתן לראות ב-Wireshark כי הלקוח שולח Hello, Hello World. השרת שולח ACK על ההודעה ולאחר מכן השרת שולח הודעה HELLO למשתמש. השרת שלח רק את ההודעה HELLO ולא את ההודעה המלאה, כי ה-BUFFER שלו הוא בגודל 5. הלקוח שולח ACK על ההודעה של השרת. לאחר מכן הלקוח מתנתק ושולח FYN,ACK. השרת שולח הפעם את שאר ההודעה HELLO WORLD . ההסבר לכך שהשרת יכול כעת לשלוח את ההודעה בגודל 10, זה כי ב-TCP יש אפשרות לשרת לשמור בצד את חלק מההודעה

ולשלוח אותה במלואה לאחר מכן.

יש לציין כי השרת מדפיס את ההודעות של הלקוח גם הוא בגודל של 5 כל אחת ולכן ההודעות מודפסות בחלקים.

ניתן לראות כי לאחר שהשרת שולח את שאר ההודעה אנו מקבלים RST, כי הלקוח כבר שלח התנתקות ולכן אין לו מי שיקבל את ההודעה הזו. לכן השרת מחזיר הודעה נוספת של FYN,ACK כדי שהשרת יבין שהחיבור הסתיים. בשלב זה השרת מבין שיש בעיה ולכן שולח RST וניתן גם לראות זאת בדגל של RESET שדולק עם 1.

לכן ההבדל בין V1 ל – V2 הוא שכאן ההודעה לא נשלחה במלואה בהתחלה, ולאחר מכן התבצעה התנתקות מצד הלקוח, מה שגרם לאחר מכן ללקוח לשלוח הודעת RST לשרת כי השרת שלח לו את שאר ההודעה בלי מי שיקבל אותה.

| No. | Time | Source | Destination | Protoc | Length | Info |
|---|---|---|---|---|---|---|
| 132 | 15.755781… | 192.168.43.66 | 192.168.43.236 | TCP | 76 | 59842 → 4521 [SYN] Seq=0 Win=64240 Len=0 MS |
| 133 | 15.778955… | 192.168.43.236 | 192.168.43.66 | TCP | 76 | 4521 → 59842 [SYN, ACK] Seq=0 Ack=1 Win=651 |
| 134 | 15.778989… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59842 → 4521 [ACK] Seq=1 Ack=1 Win=64256 Le |
| 135 | 15.779052… | 192.168.43.66 | 192.168.43.236 | TCP | 88 | 59842 → 4521 [PSH, ACK] Seq=1 Ack=1 Win=642 |
| 136 | 15.786628… | 192.168.43.236 | 192.168.43.66 | TCP | 68 | 4521 → 59842 [ACK] Seq=1 Ack=21 Win=65152 L |
| 137 | 15.786758… | 192.168.43.236 | 192.168.43.66 | TCP | 73 | 4521 → 59842 [PSH, ACK] Seq=1 Ack=21 Win=65 |
| 138 | 15.786805… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59842 → 4521 [ACK] Seq=21 Ack=6 Win=64256 L |
| 139 | 15.787069… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59842 → 4521 [FIN, ACK] Seq=21 Ack=6 Win=64 |
| 143 | 16.013336… | 192.168.43.236 | 192.168.43.66 | TCP | 83 | 4521 → 59842 [PSH, ACK] Seq=6 Ack=21 Win=65 |
| 144 | 16.013372… | 192.168.43.66 | 192.168.43.236 | TCP | 56 | 59842 → 4521 [RST] Seq=21 Win=0 Len=0 |

<u>:V3</u>

הרצנו את קוד הלקוח והמשתמש.

קוד הלקוח הוא לשלוח Hello World, לקבל הודעה מהשרת ולהדפיס אותה, לקבל עוד הודעה מהשרת ולהדפיס אותה ולסיים את החיבור.

קוד השרת הוא לקבל את ההודעה מהלקוח, להדפיס אותה אצלו, ולשלוח את ההודעה שהוא קיבל .באותיות גדולות 1000 פעמים.

ניתן לראות כי הלקוח שולח את ההודעה שלו, השרת אכן מדפיס אותה אצלו, וכעת שולח את ההודעה של הלקוח באותיות גדולות 1000 פעמים. בגלל שה – BUFFER של הלקוח הוא 1024, הוא מקבל רק חלק מההודעות של השרת, מחזיר עליהן ACK וכמובן מדפיס אותן. לאחר מכן ניתן להבין כי הלקוח מקבל את שאר ההודעות שלא נכנסו ב – BUFFER בפעם הראשונה. לכן הלקוח מדפיס אותן שוב (כפי שמתבקש בקוד) שולח עליהן ACK ביחד עם שליחת 'פין' כי קוד הלקוח נגמר.

השרת שולח ACK ללקוח אבל ממשיך לשלוח את שאר ההודעות כי הוא לא סיים לשלוח את 1000 HELLO WORLD פעמים בגלל שה - BUFFER של הלקוח לא אפשר זאת. המצב שנוצר הוא שהלקוח כבר סיים את החיבור. נוצר מצב שבו הלקוח שולח RESET לשרת, אבל השרת ממשיך לשלוח לו ACK - ים ובנוסף להמשך שליחת ההודעות שמחכות להישלח ללקוח.

לבסוף השרת שולח ללקוח FYN,ACK, אבל שוב הלקוח שולח לשרת RESET וכך התוכנית נגמרת.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 296 | 33.016408… | 192.168.43.66 | 192.168.43.236 | TCP | 76 | 59896 → 4521 | [SYN] Seq=0 Win=64240 Len=0 MSS= |
| 297 | 33.024444… | 192.168.43.236 | 192.168.43.66 | TCP | 76 | 4521 → 59896 | [SYN, ACK] Seq=0 Ack=1 Win=65160 |
| 298 | 33.024511… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59896 → 4521 | [ACK] Seq=1 Ack=1 Win=64256 Len= |
| 299 | 33.024566… | 192.168.43.66 | 192.168.43.236 | TCP | 81 | 59896 → 4521 | [PSH, ACK] Seq=1 Ack=1 Win=64256 |
| 300 | 33.030031… | 192.168.43.236 | 192.168.43.66 | TCP | 68 | 4521 → 59896 | [ACK] Seq=1 Ack=14 Win=65152 Len |
| 301 | 33.031329… | 192.168.43.236 | 192.168.43.66 | TCP | 1516 | 4521 → 59896 | [ACK] Seq=1 Ack=14 Win=65152 Len |
| 302 | 33.031344… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59896 → 4521 | [ACK] Seq=14 Ack=1449 Win=63104 |
| 303 | 33.031502… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 59896 → 4521 | [FIN, ACK] Seq=14 Ack=1449 Win=6 |
| 304 | 33.033012… | 192.168.43.236 | 192.168.43.66 | TCP | 1516 | 4521 → 59896 | [ACK] Seq=1449 Ack=14 Win=65152 |
| 305 | 33.033034… | 192.168.43.66 | 192.168.43.236 | TCP | 56 | 59896 → 4521 | [RST] Seq=14 Win=0 Len=0 |
| 306 | 33.033295… | 192.168.43.236 | 192.168.43.66 | TCP | 1516 | 4521 → 59896 | [ACK] Seq=2897 Ack=14 Win=65152 |
| 307 | 33.033306… | 192.168.43.66 | 192.168.43.236 | TCP | 56 | 59896 → 4521 | [RST] Seq=14 Win=0 Len=0 |
| 308 | 33.034537… | 192.168.43.236 | 192.168.43.66 | TCP | 8724 | 4521 → 59896 | [PSH, ACK] Seq=4345 Ack=14 Win=6 |
| 309 | 33.034629… | 192.168.43.66 | 192.168.43.236 | TCP | 56 | 59896 → 4521 | [RST] Seq=14 Win=0 Len=0 |

```
[Window size scaling factor: 128]
Checksum: 0x9298 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
0000  00 04 00 01 00 06 e0 94  67 98 28 3a 7e b5 08 00   ........ g·(:~···
0010  45 00 00 34 b8 e7 40 00  40 06 a9 5d c0 a8 2b 42   E··4··@· @··]··+B
0020  c0 a8 2b ec e9 f8 11 a9  b8 ef 71 bb 2a 01 7a 4d   ··+····· ··q·*·zM
0030  80 11 01 f5 92 98 00 00  01 01 08 0a d0 2a eb 92   ........ ·····*··
0040  65 d1 1d 85                                        e···
```

לכן ההבדל כאן בין קבצי הקוד הקודמים, הוא בעצם שכאן הלקוח מקבל מהשרת רק 2 הודעות ומדפיס אותן, אבל שוב יש הגבלה על גודל ההודעות שהלקוח יכול לקבל, מנגד השרת ממשיך לשלוח הודעות עד שהוא מסיים ושולרח FYN, על אף שהלקוח כבר ממזמן שלח FYN ACK להודעות השרת.

:V4

הרצנו את קוד הלקוח והשרת.

קוד הלקוח לשלוח Hello World 10 פעמים כל פעם, ככה 4 פעמים אחת אחרי השנייה. לאחר מכן לקבל הודעה אחת מהשרת ולנתק את החיבור.

קוד השרת הוא להמתין/לישון 5 שניות, לקבל הודעה מהלקוח, להדפיס אותה ולשלוח אותה חזרה באותיות גדולות.

בפועל, ככל הנראה בגלל שהשרת והלקוח על אותה רשת, הלקוח מספיק לשלוח הודעה ראשונה בגודל של 130, לקבל ACK מהשרת על כך, ואז לשלוח את שאר ההודעות וגם עליהן 'לקבל ACK מהשרת. לאחר מכן ניתן לראות כי השרת 'הולך לישון' ורק לאחר 5 שניות הוא 'מתעורר' ושולח חזרה ללקוח את ההודעות של הלקוח באותיות גדולות . הלקוח מחזיר לשרת ACK ושולח FYN ACK ומסיים את החיבור. השרת שולח ACK על כך, אבל הוא נכנס שוב למצב שינה ולכן רק אחרי 5 שניות השרת שולח FYN,ACK והלקוח מחזיר ACK ובכך נגמרת ההתחברות.

ההבדל בקוד זה לבין הקודים הוא ה"הירדמות" של השרת מה שמעכב אותו בשליחת סיום ההתקשרות. ניתן להבין או להסיק כי אם הלקוח והשרת לא היו על אותה רשת, או אם היה איזה שהוא **נתב** ביניהם, ייתכן והשרת לא היה מספיק לשלוח את ההודעה השנייה שעליו היה לשלוח ללקוח, היה הולך לישון, ורק לאחר מכן היה שולח את ההודעה השנייה.

| No. | Time | Source | Destination | Protoc | Length | Info |
|---|---|---|---|---|---|---|
| 57 | 6.0428895… | 192.168.43.66 | 192.168.43.236 | TCP | 76 | 57872 → 4521 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SAC |
| 60 | 6.0556336… | 192.168.43.236 | 192.168.43.66 | TCP | 76 | 4521 → 57872 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 M |
| 61 | 6.0556643… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 57872 → 4521 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval= |
| 62 | 6.0557042… | 192.168.43.66 | 192.168.43.236 | TCP | 198 | 57872 → 4521 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=130 |
| 63 | 6.0656653… | 192.168.43.236 | 192.168.43.66 | TCP | 68 | 4521 → 57872 [ACK] Seq=1 Ack=131 Win=65152 Len=0 TSva |
| 64 | 6.0657185… | 192.168.43.66 | 192.168.43.236 | TCP | 458 | 57872 → 4521 [PSH, ACK] Seq=131 Ack=1 Win=64256 Len=3 |
| 65 | 6.0720681… | 192.168.43.236 | 192.168.43.66 | TCP | 68 | 4521 → 57872 [ACK] Seq=1 Ack=521 Win=64768 Len=0 TSva |
| 105 | 11.228973… | 192.168.43.236 | 192.168.43.66 | TCP | 588 | 4521 → 57872 [PSH, ACK] Seq=1 Ack=521 Win=64768 Len=5 |
| 106 | 11.229034… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 57872 → 4521 [ACK] Seq=521 Ack=521 Win=63744 Len=0 TS |
| 107 | 11.229183… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 57872 → 4521 [FIN, ACK] Seq=521 Ack=521 Win=64128 Len |
| 114 | 11.340017… | 192.168.43.236 | 192.168.43.66 | TCP | 68 | 4521 → 57872 [ACK] Seq=521 Ack=522 Win=64768 Len=0 TS |
| 442 | 16.075993… | 192.168.43.236 | 192.168.43.66 | TCP | 68 | 4521 → 57872 [FIN, ACK] Seq=521 Ack=522 Win=64768 Len |
| 443 | 16.076045… | 192.168.43.66 | 192.168.43.236 | TCP | 68 | 57872 → 4521 [ACK] Seq=522 Ack=522 Win=64128 Len=0 TS |