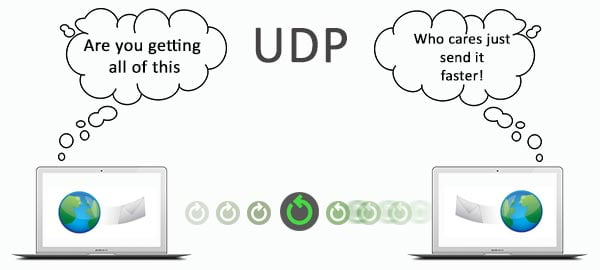
**UDP Time Server and Client application's**

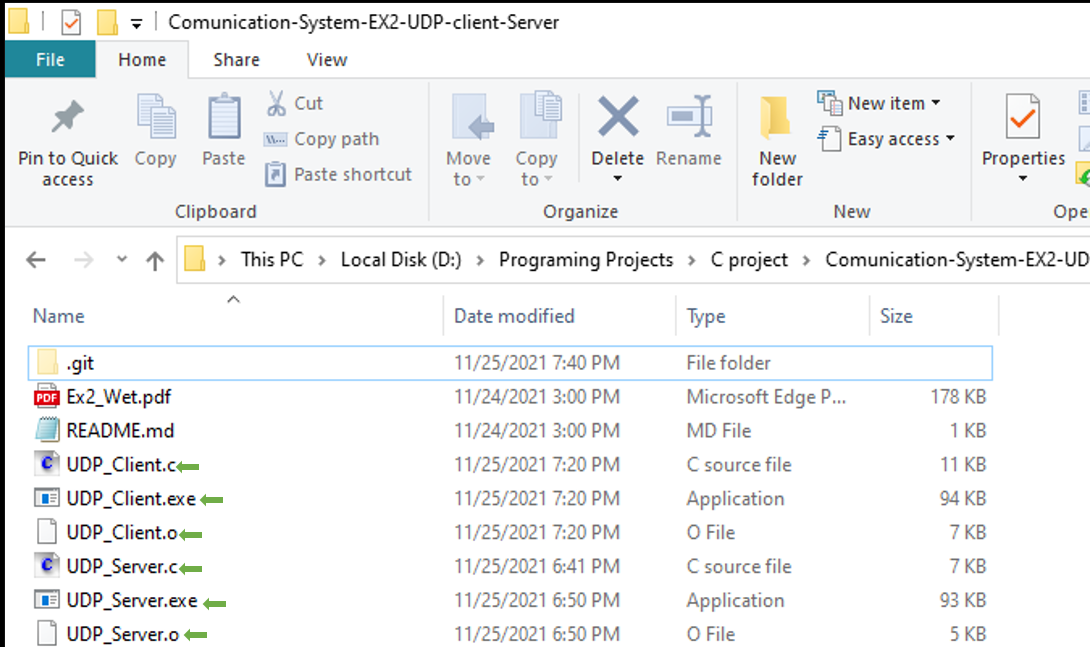
**–User Guide–**



**תוכן עניינים:**

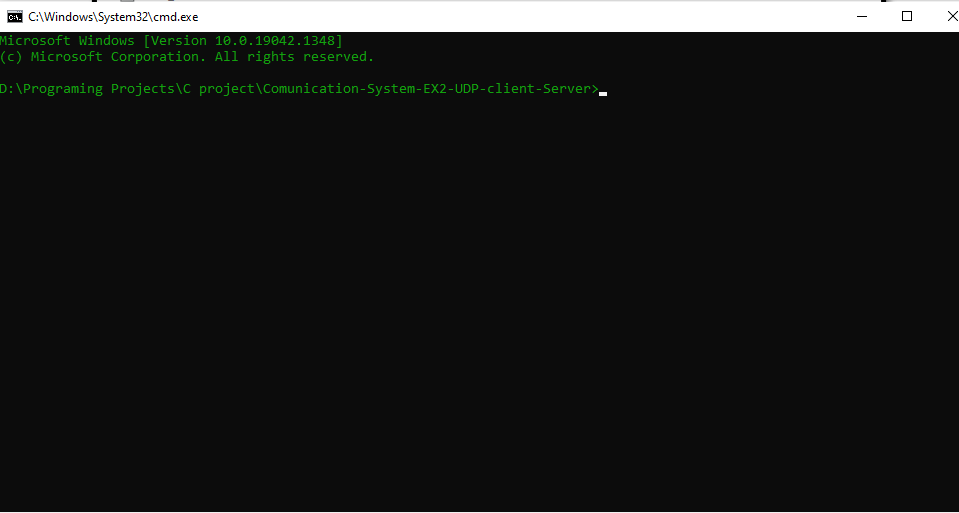
|  |  |  |
| --- | --- | --- |
| **נושא** | **תיאור ספציפי** | **עמודים** |
| **מדריך הפעלה** | שלב ראשון  Compile And Run | 3 |
| **מדריך הפעלה** | שלב שני  אתחול התוכניות | 4 |
| **מדריך הפעלה** | שלב שלישי  טיפול בבקשות הClient | ראו פירוט מטה |
| **מדריך הפעלה**  **בקשות Client** | **בקשות 0-1:**  Exit Client  Get Time | 5 |
| **מדריך הפעלה**  **בקשות Client** | **בקשות 2-3:**  Get Time Without Year  Get Time Since Epoch | 6 |
| **מדריך הפעלה**  **בקשות Client** | **בקשות 4-5:**  Measure Round Trip Time  Measure Client->Server Delay | 7-8 |
| **מדריך הפעלה**  **בקשות Client** | **בקשה 6:**  Get Day and Month | 9 |
| **תיעוד העברת ההודעות ב WIRESHARK** | תמונות של ההודעות שנשלחו/ התקבלו לכל אחד מהסעיפים הקודמים. (כולל הסבר קצר). | 10-12 |

**שלב ראשון -0 Compile And Run:**

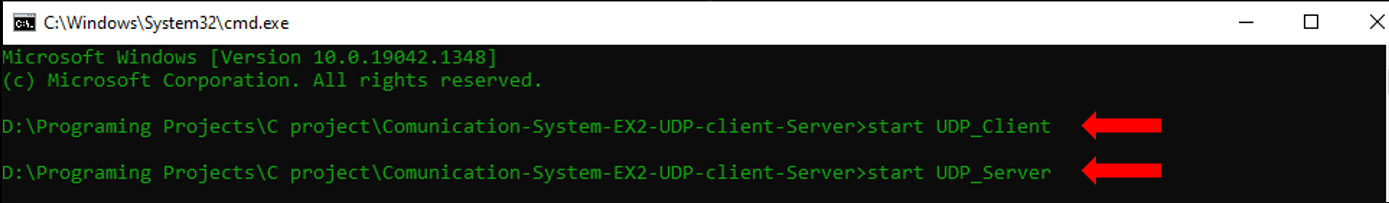
1. את התוכנית יש לקמפל ולהריץ בסביבת **Windows** , כך שלאחר הקומפילציה יופיעו הקבצים להלן:
2. ניתן להריץ את התוכניות **UDP\_Client** | | **UDP\_Server** או באמצעות ה **CMD** או באמצעות DoubleClick על הקבצי **exe**.

(**הערה**: רק הקבצים המסומנים בחץ בתמונה יופיעו לאחר קומפילציה של הקבצי C).

**הרצה דרך הCmd :**

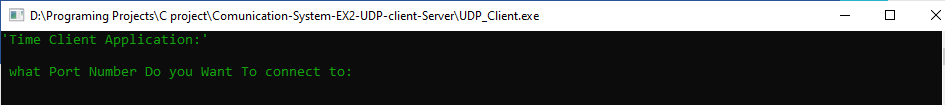
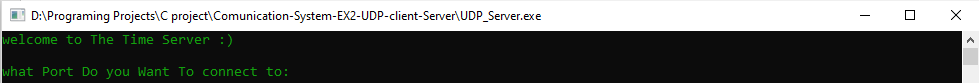
נוודא שאנחנו נמצאים בRepository הנכון – איפה שקבצי ה exe נמצאים.

ולאחר מכן נקליד את 2 הפקודות:

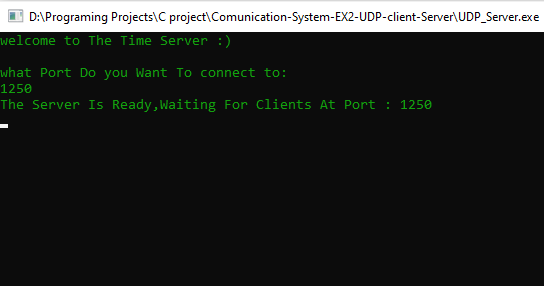
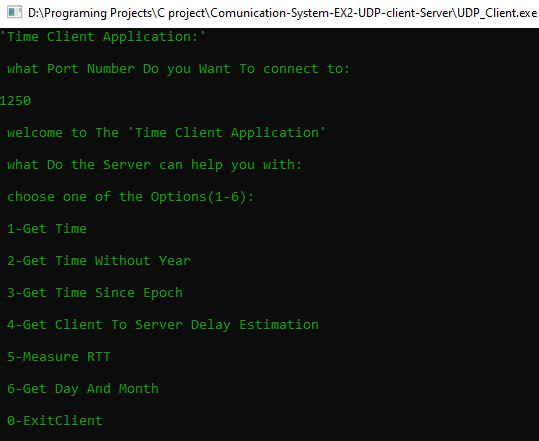


**לאחר סיום התהליך ייפתחו שני חלונות שבהם ירוצו 2 התוכניות השונות הServer וה Client.**

**שלב שני - אתחול התוכניות:**

1. בסיום השלב הקודם – יופיעו החלונות הבאים:
2. עבור אפליקציית **הClient** יופיע החלון:
3. עבור אפליקציית **הServer** יופיע החלון:
4. בשני החלונות הבאים נרצה להזין מספר **Port זהה** **ותקין** (מספר הגדול מ 1024, וקטן מ 65,535).

במידה והמספר לא תקין, התוכנית תצא בהודעת שגיאה.

1. במידה והמספר **תקין**, נקבל את המסכים הבאים של ממשק המשתמש:

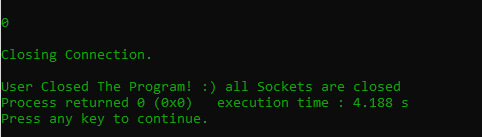
**Server** יודיע לנו שהוא מחכה לClients.

**Client** ייתן לנו לבחור בין 7 אפשרויות (על ידי הכנסה של מספר בין 0-6), הקלט של המשתמש נבדק לתקינות ובמידה והמשתמש הזין קלט לא תקין (מספר הגדול מ 6 או קטן מ 0, המערכת של הClient תצא בהודעת שגיאה).

\*\*מכיוון שממשק המשתמש שלנו ממומש בלולאה, הקשר בין הServer לClient מתקיים אך ורק לבקשה בודדת, **כלומר הקשר נסגר לאחר כל בקשה בודדת של הלקוח**- כך אנו יכולים להריץ מספר Client's שירוצו במקביל ולא יתפסו את ה Server לעצמם\*\*.

**שלב שלישי -טיפול בבקשות הClient :**

1. **בקשת Exit Client :**
2. כאשר המשתמש יקליד **0** – בקשה לסגירת הClient.
3. תופעל פונקציית הסגירה בתוכה ייסגר ה Socket שנפתח בתחילת התהליך.
4. תודפס למשתמש הודעה שהוא סגר את ה Socket כמו שצריך ושהתוכנית הסתיימה בהצלחה.



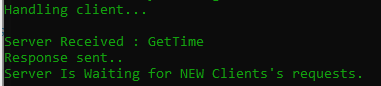
**Client**

ה **Server** לא מקבל הודעה במידה וה **Client** החליט לסגור את האפליקציה וימשיך לרוץ בלולאה ולחכות לבקשות מClients , ללא קשר לClient הספציפי שהתנתק.

1. **בקשת Get Time:**
2. כאשר המשתמש יקליד **1** - בקשה של **Get Time** תשלח לServer .
3. ה Server ייחשב את התאריך והשעה באמצעות ספריית Time.
4. הServer ימיר את התוצאה ל String ויישלח את התשובה ל Client .
5. ה Client ידפיס את ההודעה שקיבל מה Server (ה String של התאריך והשעה).

**ע"ג המסך נקבל את ההודעות הבאות:**

**Client**



**SERVER**

**לאחר שה Client קיבל את התשובות של ה Server**  :

ה Client יסגור את החיבור לשרת וימשיך בלולאה של ממשק המשתמש וידפיס לנו שוב את האפשרויות השונות.

הServer ימשיך בלולאה להאזין לבקשות נוספות מהClients .

**שלב שלישי - טיפול בבקשות הClient :**

1. **בקשת Get Time Without Year:**
2. כאשר המשתמש יקליד **2** - בקשה של **Get Time Without Year** תשלח לServer .
3. ה Server ייחשב את התאריך והשעה באמצעות ספריית Time.
4. הServer ימיר את התוצאה ל String **ללא העתקה של שדה השנה** ויישלח את התשובה ל Client.
5. ה Client ידפיס את ההודעה שקיבל מה Server (ה String של התאריך והשעה **ללא השנה**).

**Client**

**SERVER**



**לאחר שה Client קיבל את התשובות של ה Server**  :

ה Client יסגור את החיבור לשרת וימשיך בלולאה של ממשק המשתמש ידפיס לנו שוב את האפשרויות השונות.

הServer ימשיך בלולאה להאזין לבקשות נוספות מהClients .

1. **בקשת Get Time Since Epoch:**
2. כאשר המשתמש יקליד **3** - בקשה של **Get Time Since Epoch** תשלח לServer .
3. ה Server ייאתחל אובייקט מסוג Time, האתחול מחזיר לנו את מספר השניות מאז 1.1.1970.
4. הServer ימיר את התוצאה ל String יישלח את התשובה ל Client.
5. ה Client ידפיס את ההודעה שקיבל מה Server, מספר השניות מאז 1.1.1970.

**SERVER**

**Client**



**לאחר שה- Client קיבל את התשובות של ה- Server**  :

ה Client יסגור את החיבור לשרת וימשיך בלולאה של ממשק המשתמש וידפיס לנו שוב את האפשרויות השונות.

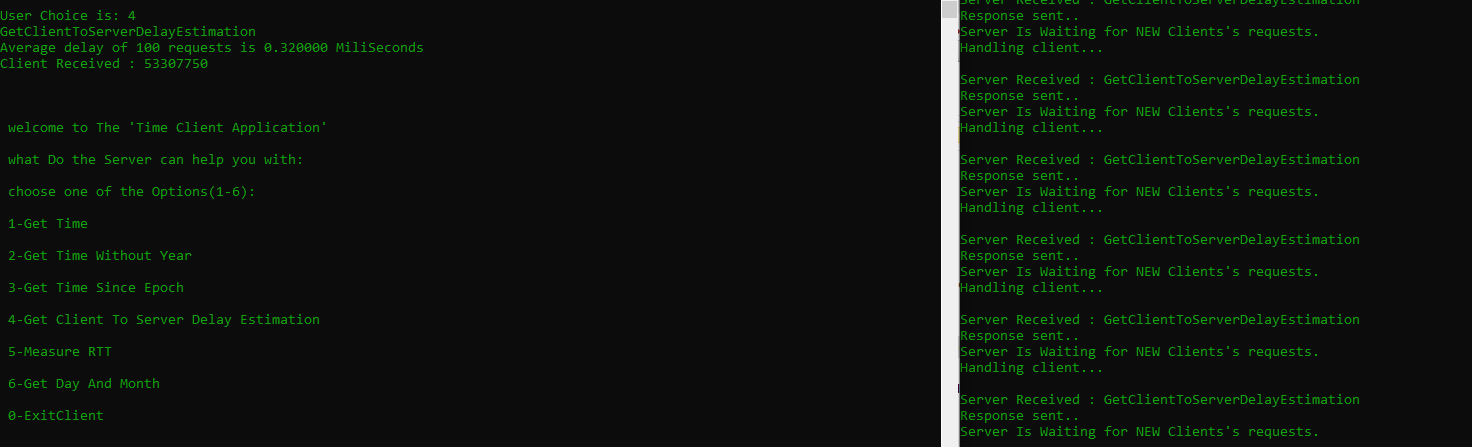
הServer ימשיך בלולאה להאזין לבקשות נוספות מהClients .

**שלב שלישי - טיפול בבקשות הClient :**

1. **בקשת Get Client To Server Delay Estimation:**
2. כאשר המשתמש יקליד **4** - בקשה של **Get Client To Server Delay Estimation** תשלח לServer .
3. ה **Client** יישמור את הזמן של רגע השליחה באמצעות הפונקציה ()**GetTickCount**, שמחזירה לנו במילי שניות את הזמן מאז שהמעבד אותחל.
4. ה **Server** יחזיר תשובה שתכיל את הזמן בו קיבל את החבילה.
5. ה **Client** ,ברגע קבלת התשובה, יישמור משתנה זמן נוסף.
6. ה **Client** ייחשב עבור כל השליחות והקבלות את ההפרש בזמן מרגע שליחת ההודעה עד רגע קבלת הזמן, מה Server – כלומר ייחשב את הClient To Server Delay הזמן מרגע שליחת הודעה עד לקבלתה בServer.
7. הClient שולח 100 בקשות רצופות כלומר חוזר על **SendTo** 100 פעמים לפני שהוא מגיע ללואה של הReceive שגם רצה 100 פעם.
8. בסוף אנו ממצעים את הזמנים שקיבלנו מכל השליחות והקבלות.

**Client**

**SERVER**



**לאחר שה Client קיבל את התשובות של ה Server**  :

ה Client יסגור את החיבור לשרת וימשיך בלולאה של ממשק המשתמש וידפיס לנו שוב את האפשרויות השונות.

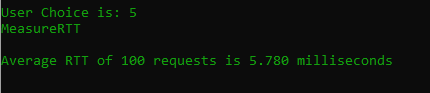
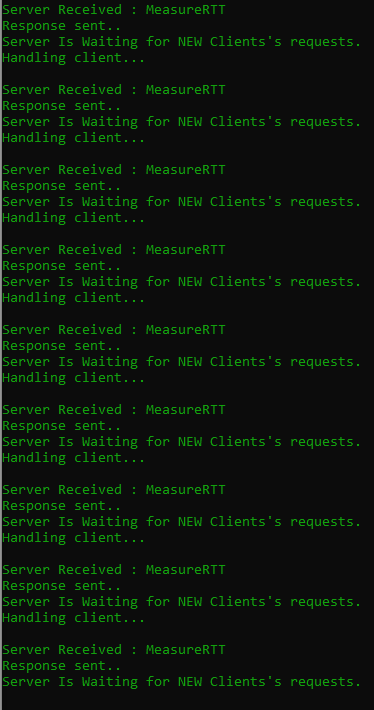
הServer ימשיך בלולאה להאזין לבקשות נוספות מהClients .

**שלב שלישי - טיפול בבקשות הClient :**

1. **בקשת :Measure RTT**
2. כאשר המשתמש יקליד **5** - בקשה של **MeasureRTT** תשלח לServer .
3. ה **Client** יישמור את הזמן של רגע השליחה באמצעות הפונקציה ()**GetTickCount**, שמחזירה לנו במילי שניות את הזמן מאז שהמעבד אותחל.
4. ה **Server** יחזיר תשובה " MRTT" שתהווה לנו דגל למדידת הזמן ה Receive.
5. ה **Client** ,ברגע קבלת התשובה, יישמור משתנה זמן נוסף באמצעות () **GetTickCount**.
6. ה **Client** ייחשב עבור כל איטרציה את ההפרש בזמן מרגע שליחת ההודעה עד רגע קבלת ה"דגל" מה Server – כלומר ייחשב את הRTT הזמן מרגע שליחת הודעה עד לקבלת תשובה.
7. הClient וה Server חזורים על התהליך (שמתואר בסעיפים א' - ה') - 100 איטרציות כשבכל איטרציה ההפרש בין זמן השליחה לזמן הקבלה נשמר במשתנה סכום.
8. בסוף 100 האיטרציות של התהליך: (שליחה -> קבלה -> חישוב), סכום 100 הזמנים שנמדדו מחולק ב 100 לצורך קבלת ה RTT הממוצע.

**SERVER**

**Client**



ה Server מעדכן בכל הודעה שהוא מקבל.

ושולח לClient את הדגל.

**לאחר שה Client קיבל את התשובות של ה Server**  :

ה Client יסגור את החיבור לשרת וימשיך בלולאה של ממשק המשתמש וידפיס לנו שוב את האפשרויות השונות.

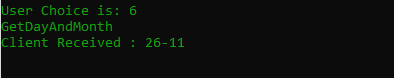
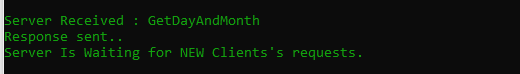
הServer ימשיך בלולאה להאזין לבקשות נוספות מהClients .

**שלב שלישי - טיפול בבקשות הClient :**

1. **בקשת Get Day And Month :**
2. כאשר המשתמש יקליד **6** - בקשה של Get Day And Month תשלח לServer .
3. ה Server ייחשב את התאריך והשעה באמצעות ספריית Time.
4. הServer ימיר את התוצאה ל String יישלח את התשובה ל Client.
5. ה Client ידפיס את ההודעה שקיבל מה Server, היום והחודש.

**SERVER**

**Client**

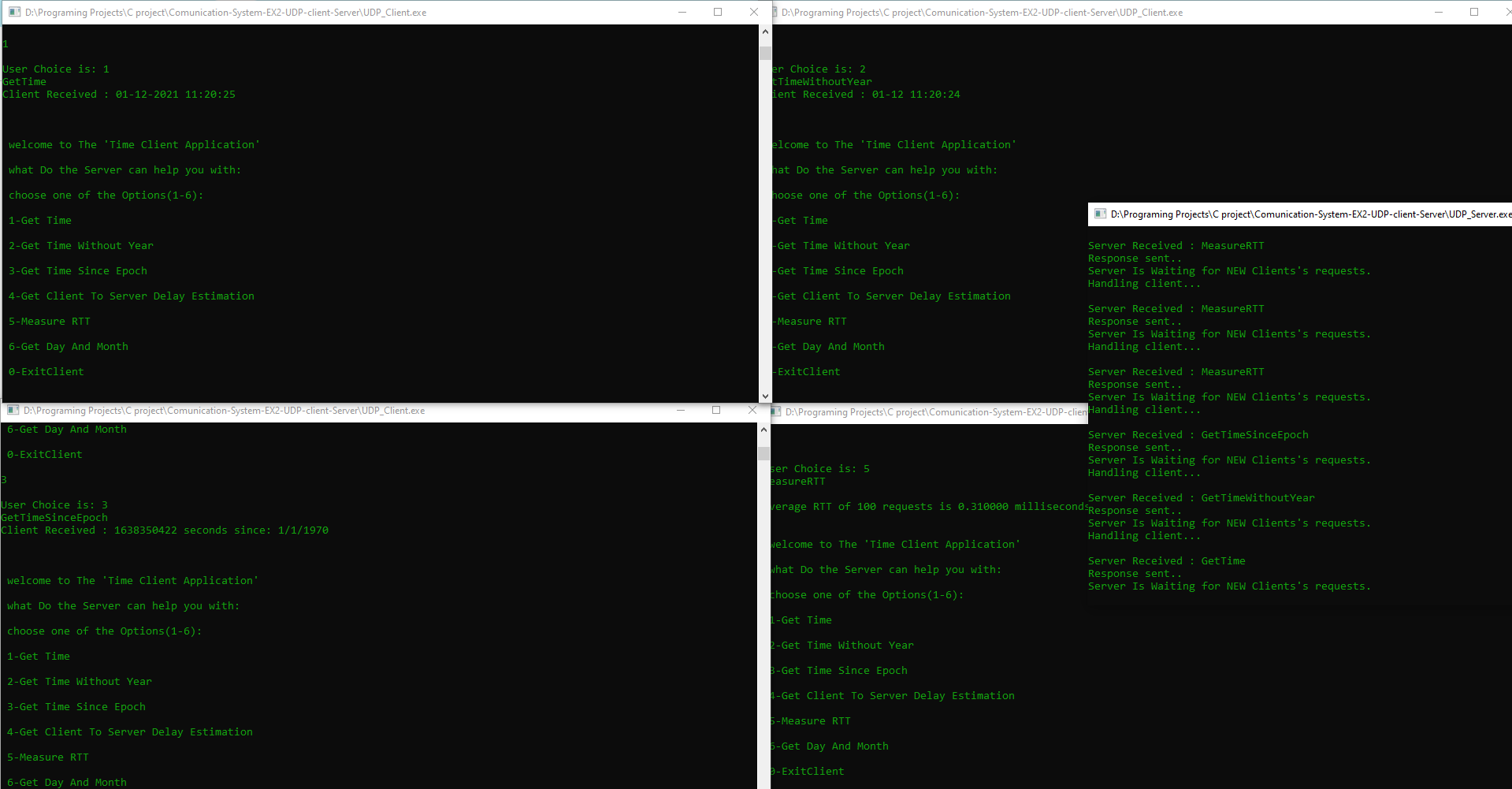


**לאחר שה Client קיבל את התשובות של ה Server**  :

ה Client יסגור את החיבור לשרת וימשיך בלולאה של ממשק המשתמש וידפיס לנו שוב את האפשרויות השונות.

הServer ימשיך בלולאה להאזין לבקשות נוספות מהClients .

**הרצה של 4 Clients במקביל :**



**SERVER**

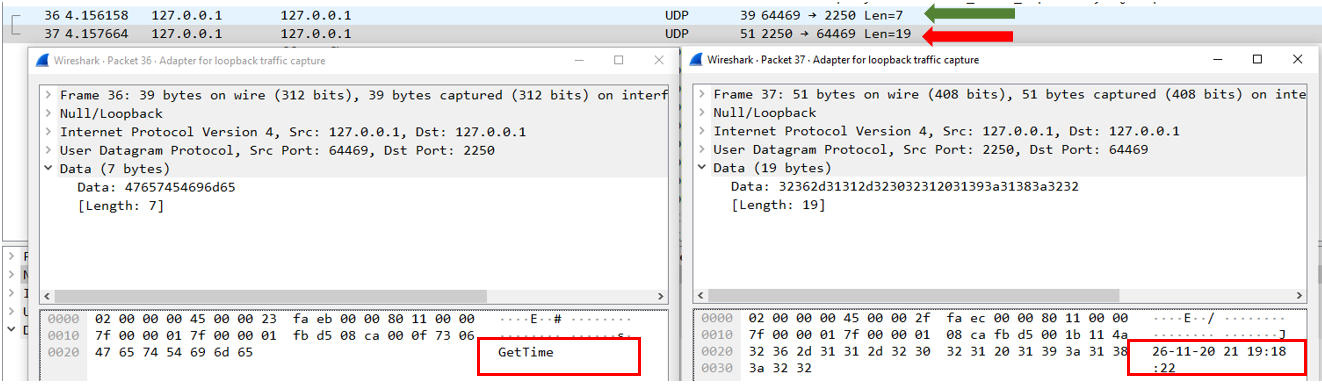
**Client 4**

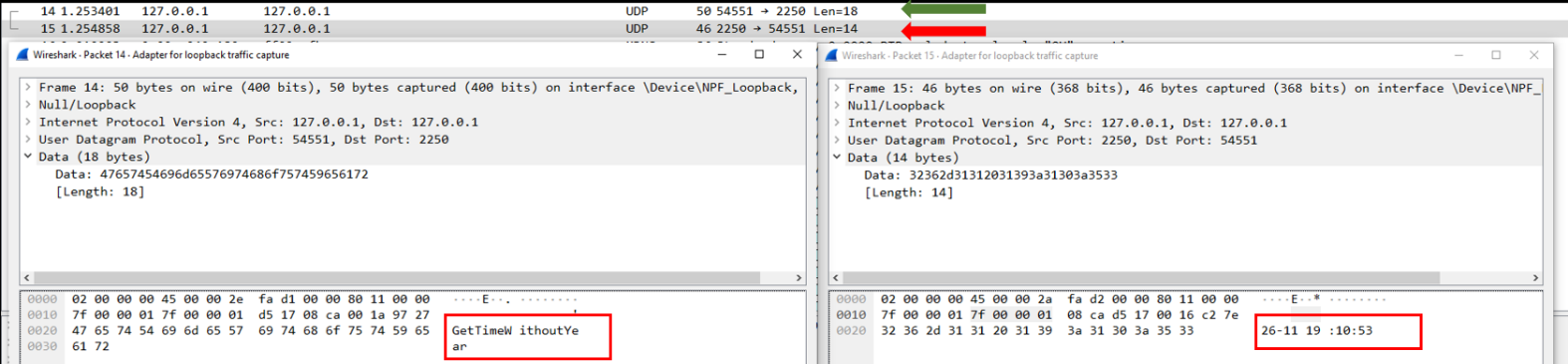
**Client 3**

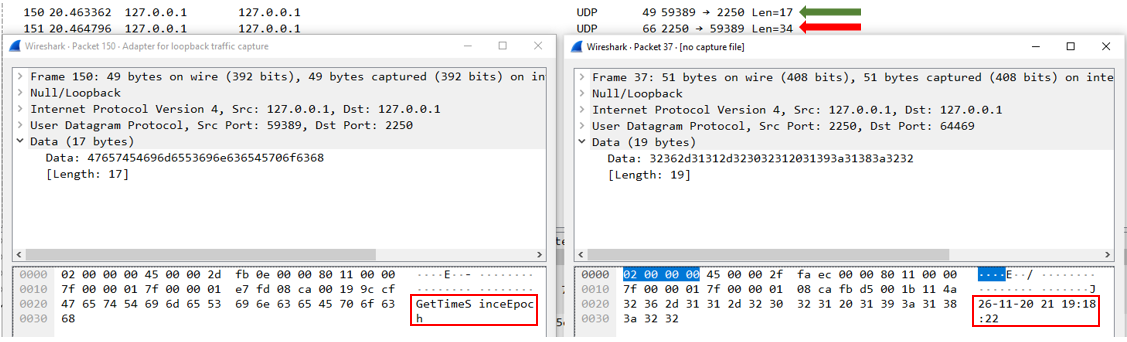
**Client 2**

**Client 1**

**תיעוד העברת ההודעות ב WIRESHARK:**

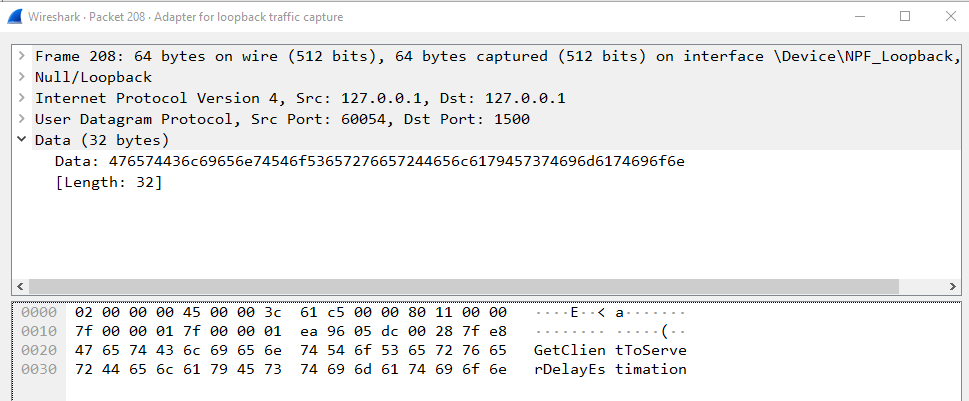
**בקשה 1 Get Time:**

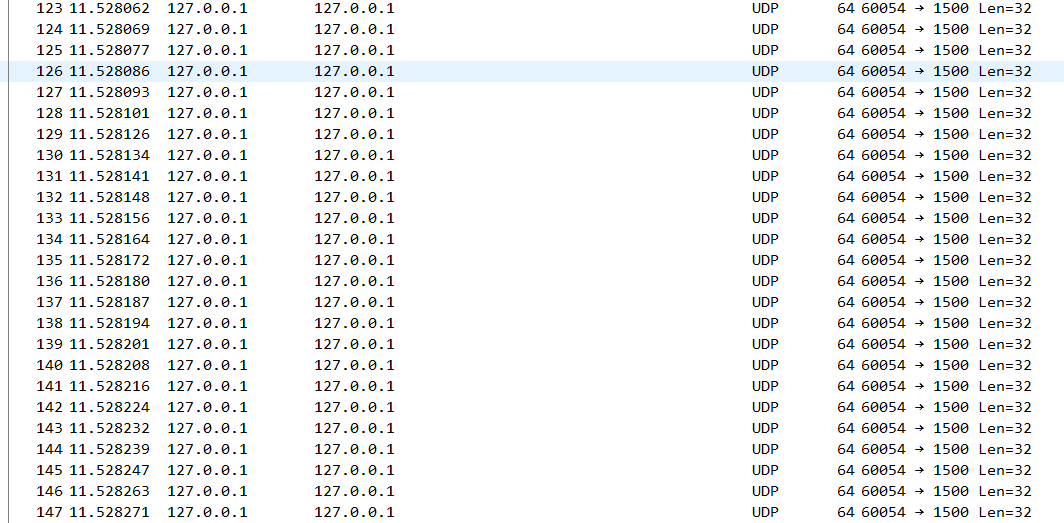
**בקשה 2 Get Time Without Year:**

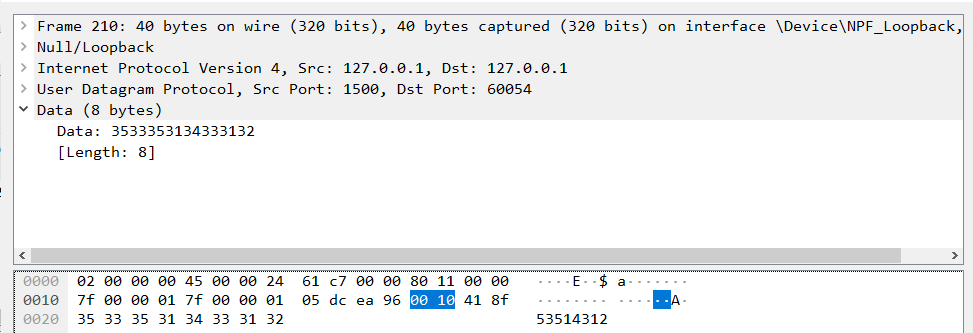
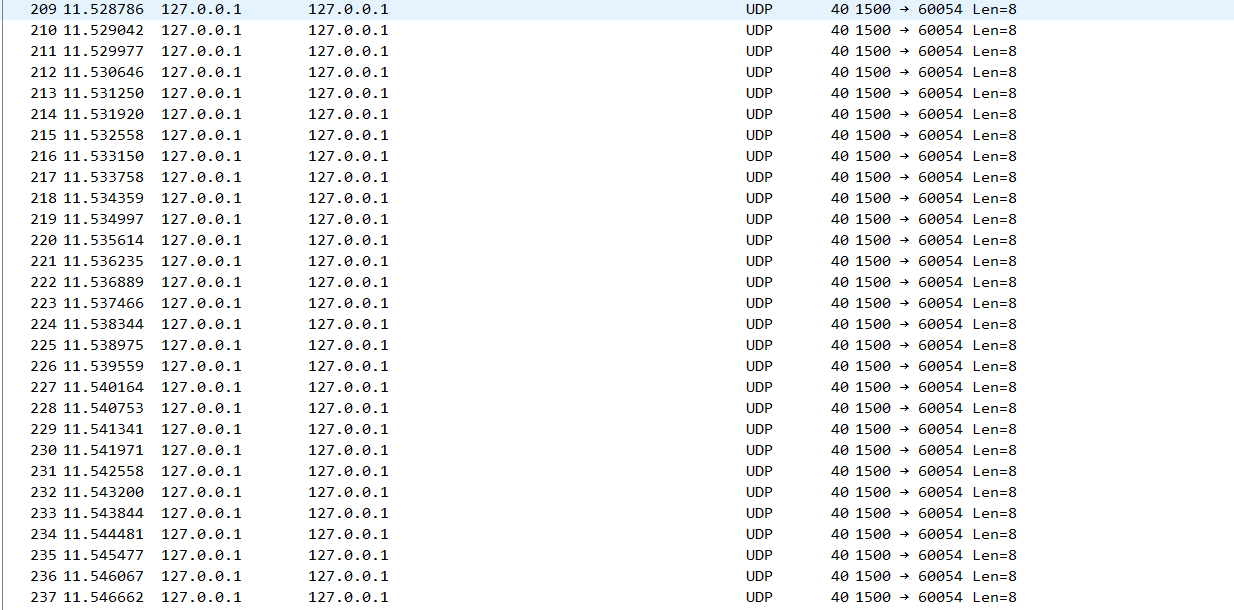
**בקשה 3 Get Time Since epoch:**

**תיעוד העברת ההודעות ב WIRESHARK:**

**בקשה 4 Get Client To Server Delay Estimation:**

**100 בקשות רצופות מה Client->Server :**



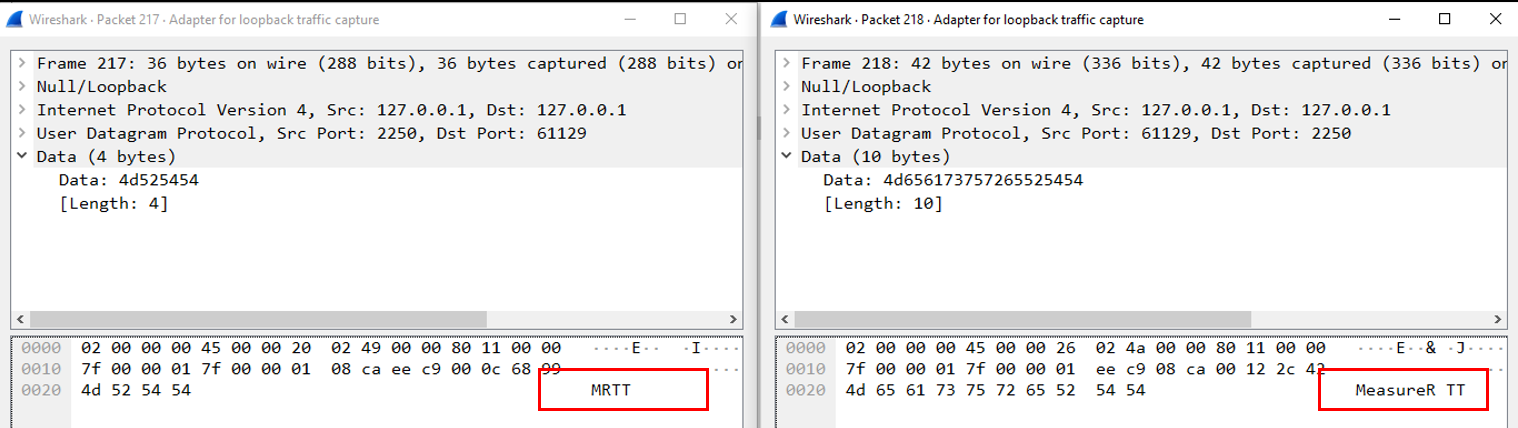
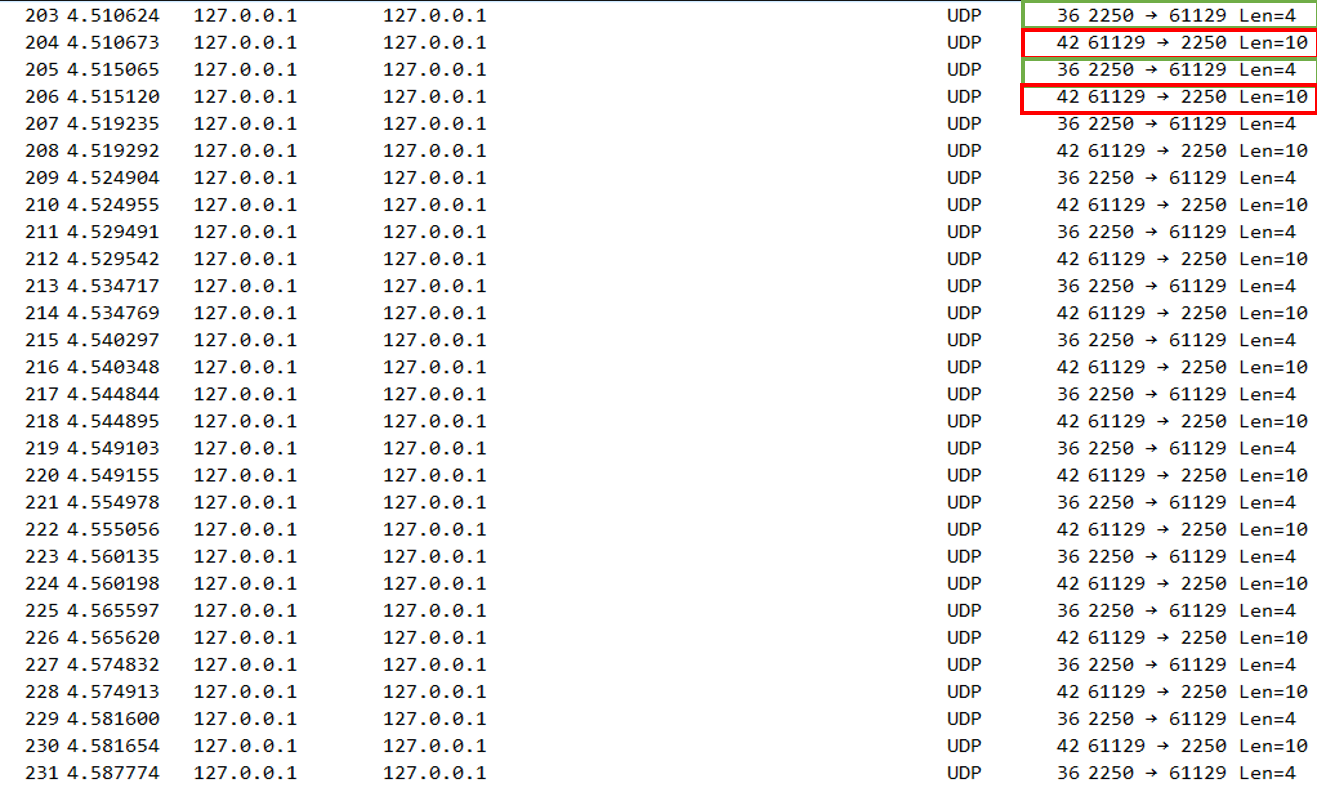
**100 קבלות רצופות מהServer->Client :**

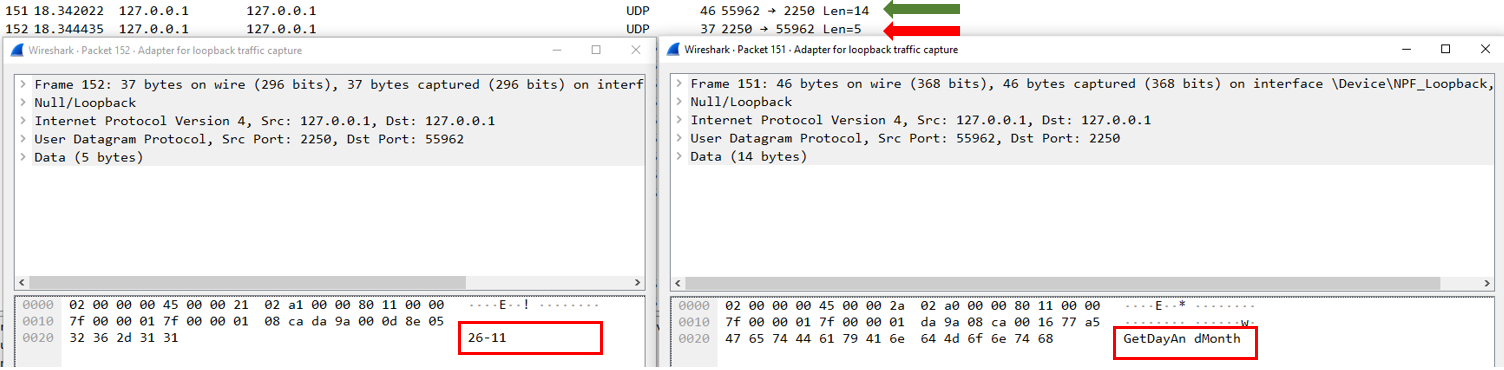
אנחנו מקבלים בכל פקטה את הזמן בו

הServer קיבל את ההודעה.

**בקשה 5 Measure Average RTT :**

100 הודעות נשלחות ו100 הודעות מתקבלות סה"כ 200 שורות של פאקטות מופיעות לנו ב WireShark:

****ניתן לראות שבמקרה של MRTT יש 100 איטרציות של **שליחה->קבלה** (**לא רציפות**).

**בקשה 6 Get Day And Month:**