הערה כללית: במקום לעשות בWIRESHARK הדפסה של החבילות, עשינו copy של הטקסט הנדרש, או screen-shot.

לתשובות 1-2:

199 5.297341 192.168.1.102 128.119.245.12 HTTP 104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)

Transmission Control Protocol, Src Port: health-polling (1161), Dst Port: http (80), Seq: 164041, Ack: 1, Len: 50

1. הIP של הלקוח הוא 192.168.1.102, והוא עובד בפורט 1161.
2. השרת נמצא בIP 128.119.245.12 בפורט 80 (שהוא הפורט הדיפולטיבי של HTTP).
3. אנו עובדים על הקובץ שהורדנו מהשרת שלהם, ולכן השאלה לא רלוונטית.
4. מס' חבילה: 1. הכותרת שלה בWIRESHARK:

1 0.000000 192.168.1.102 128.119.245.12 TCP 62 health-polling > http [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK\_PERM=1

רואים שזוהי חבילת SYN, ע"פ מה שכתוב בכותרת החבילה "SYN" בלי ACK, וזה נשלח מהלקוח אל השרת.

1. מס' חבילה: 2. הכותרת שלה בWIRESHARK:

2 0.023172 128.119.245.12 192.168.1.102 TCP 62 http > health-polling [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK\_PERM=1

ערך הACK-Field שלה הוא 1:

Acknowledgement number: 1 (relative ack number)

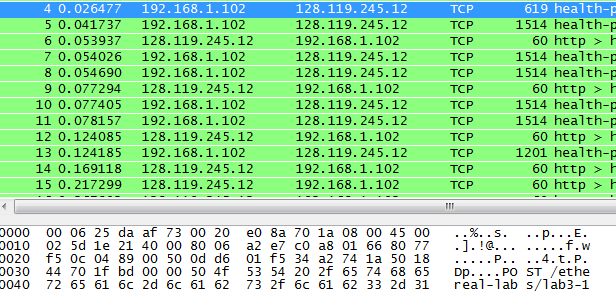
שדה זה מחושב ע"פ הSequence number של ההודעה האחרונה שהתקבלה, לפי הנוסחא:

Acknowledgement number = Sequence number + length-of-packet

(ההודעה שאנו שלחנו הכילה את השורה "Sequence number: 0 (relative sequence number)" ואורך הודעת SYN הוא 1 בלבד, ולכן הבייט הבא שצפוי להתקבל הוא 1).

רואים שזוהי חבילת SYN-ACK ע"פ מה שמופיע בכותרת החבילה "[SYN, ACK]", וזה נשלח מהשרת אל הלקוח.

1. חבילה מס' 4. צילום מסך:



רואים בנתונים של החבילה שמופיעות ברצף האותיות "POST", וזה מה שמאפיין את פקודת הPOST בפרוטוקול HTTP.

1. ראה טבלה:

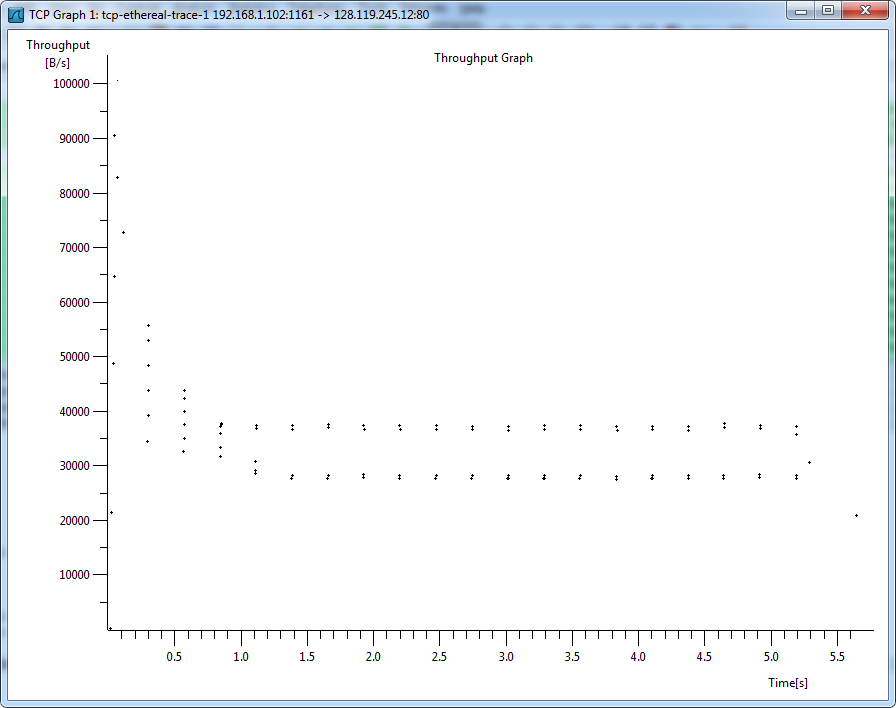
זיהינו את הACK ע"פ השוואה בין השדה "Acknowledgement number" בחבילת האישור לבין השדה "Next sequence number" בחבילה הנשלחת.

חישבנו את EstimatedRTT ע"פ הנוסחא המופיעה בשקפים של ההרצאות:

בהתחשב בהגדרה המופיעה שם:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| מס' חבילה | זמן שליחה | מס' חבילת ACK | זמן קבלת ACK | RTT | EstimatedRTT | אורך החבילה (Bytes) |
| 4 | 0.026477 | 6 | 0.053937 | 0.02746 | 0.02746 | 619 |
| 5 | 0.041737 | 9 | 0.077294 | 0.035557 | 0.028472 | 1514 |
| 6 | 0.053937 | חבילת ACK בלבד, שאין עליה אישור. | | | | 60 |
| 7 | 0.054026 | 12 | 0.124085 | 0.070059 | 0.03367 | 1514 |
| 8 | 0.054690 | 14 | 0.169118 | 0.114428 | 0.043765 | 1514 |
| 9 | 0.077294 | חבילת ACK בלבד, שאין עליה אישור. | | | | 60 |
| 10 | 0.077405 | 15 | 0.217299 | 0.139894 | 0.055781 | 1514 |
| 11 | 0.078157 | 16 | 0.267802 | 0.189645 | 0.072514 | 1514 |

1. בטבלה דלעיל. זיהינו את אורך החבילה ע"פ השדה length בכותרת של WIRESHARK.
2. בחבילה מספר 2 רואים את הwindow-size הנמוך ביותר, והוא: 5840. לא נראה שזה משפיע לרעה על השולח, כיון שהחבילות שלו הן בגודל מקסימלי של 1514 bytes (כפי שמוסבר בהערה מספר 4 בעמוד 6 של קובץ השאלות).
3. בדקנו את השדה "Sequence number" בכל הTRACE, וראינו שהמספרים המופיעים שם בצד הלקוח הם כולם סדרה עולה-ממש, כך שאין חבילה שנשלחה פעמיים. כמו"כ כל המספרים המופיעים בצד השרת הם ג"כ סדרה עולה-ממש, כך שגם השרת לא שלח חבילה אחת פעמיים.
4. עד לחבילה מספר 53 ראינו שבאופן קבוע השרת שולח ACK על כל חבילה שהתקבלה, ואח"כ הוא מתחיל לשלוח ACK על כל 2 חבילות שהתקבלו. לעיתים הוא שולח על 2 חבילות רצופות (לדוגמא: חבילות 61-62 מאשרות את חבילות 57-58).
5. להלן הגרף:



רואים שהthroughput בתחילה היה גבוה (כ90 KB/s), ובהמשך הוא התייצב באיזור נמוך יותר (כ30-40 KB/s).

1. לאורך כל הגרף רואים שהוא שולח 6 חבילות בכל פעם עד שהוא מקבל ACK, ואז שולח את 6 החבילות הבאות. ציפינו לראות עליה מעריכית בהתחלה ע"פ slow-start, ועליה ליניארית בהמשך ע"פ congestion avoidance, ואז שליחת כמה חבילות ללא קבלת ACK הגורמת לחזרה לslow-start וחוזר חלילה, אך את זה לא ראינו כלל.
2. אנו עובדים על הקובץ שהורדנו מהשרת שלהם, ולכן השאלה לא רלוונטית.