Lab Exercise 4: Exploring TCP

There are 7 labs during this course. For each student, the 5 best performing labs will contribute to your final lab mark.

Objectives:

- gain insights into the operation of TCP.
- get familiar with the ns-2 simulator (as preparation for the next two labs)

Prerequisites and Links:

- Week 4, 5 & 6 Lectures
- Relevant Parts of Chapter 3 of the textbook
- Introduction to Tools of the Trade (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17358)
- Basic understanding of Linux. A good resource is here (http://www.ee.surrey.ac.uk/Teaching/Unix/) but there are several other resources online:
- tcp-ethereal-trace-1 (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17307)
- Overview slides (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17316)
- simpleSim.tcl (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17308)
- Explanation of script simpleSim.tcl (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17315)

Marks: **10 marks.**

- This lab only requires submission of a lab report.
- Please attend the lab in your allocated lab time slot.
- This lab comprises of a number of exercises. Pl note that not all the exercises for this lab are marked. You have to submit a report containing answers for all questions in Exercise 1 and 2 only.
- We expect the students to go through as much of the lab exercises as they can at home and come to the lab for clarifying any doubts in procedure/specifications.

Deadline:

Midnight Friday 7th Sep 2018 . You can submit as many times as you wish before the deadline. A later submission will override the earlier submission, so make sure you submit the correct file. Do not leave until the last moment to submit, as there may be technical or communications error and you will not have time to rectify it.

Late Submission Penalty:

Late penalty will be applied as follows:

- 1 day after deadline: 20% reduction
- 2 days after deadline: 40% reduction
- 3 days after deadline: 60% reduction
- 4 or more days late: NOT accepted

Note that the above penalty is applied to your final mark. For example, if you submit your lab work 2 days late and your score on the lab is 8, then your final mark will be 8 - 3.2 (40% penalty) = 4.8.

Submission Instructions:

Submit a PDF document **Lab4.pdf** with answers to all questions for Exercise 1 & 2 only. **There is no need to create an archive.** Submit the pdf directly.

Original Work Only:

You are strongly encouraged to discuss the questions with other students in your lab. However, each student must submit his or her own work. You may need to refer to the material indicated above (particularly Tools of the Trade document) and also conduct your own research to answer the questions.

Exercise 1: Understanding TCP using Wireshark

For this particular experiment download the trace file: tcp-ethereal-trace-1 (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17307) .

The following indicate the steps for this experiment:

Step 1: Start Wireshark by typing *wireshark* at the command prompt.

Step 2: Load the trace file *tcp-ethereal-trace-1* by using the *File* pull down menu, choosing *Open* and selecting the appropriate trace file. This file captures the sequence of messages exchanged between a host and a remote server (gaia.cs.umass.edu). The host transfers a 150 KB text file, which contains the text of Lewis Carrol's *Alice's Adventure in Wonderland* to the server. Note that the file is being transferred from the host to the server using a HTTP POST message.

Step 3: Now filter out all non-TCP packets by typing "tcp" (without quotes) in the filter field towards the top of the Wireshark window. You should see a series of TCP segments between the host in MIT and gaia.cs.umass.edu. The first three segments of the trace consist of the initial three-way handshake containing the SYN, SYN ACK and ACK messages. You should see an HTTP POST message in the 4 th segment of the trace being sent from the host in MIT to gaia.cs.umass.edu (check the contents of the payload of this segment). You should observe that the text file is transmitted as multiple TCP segments (i.e. a single POST message has been split into several TCP segments) from the client to the server (gaia.cs.umass.edu). You should also see several TCP ACK segments been returned in the reverse direction.

IMPORTANT NOTE: Do the sequence numbers for the sender and receiver start from zero? The reason for this is that Wireshark by default scales down all real sequence numbers such that the first segment in the trace file always starts from 0. To turn off this feature, you have to click Edit->Preferences>Protocols->TCP (or Wireshark->Preferences->Protocols->TCP) and then disable the "Relative Sequence Numbers" option. Note that the answers in the solution set will reflect this change. If you conduct the experiment without this change, the sequence numbers that you observe will be different from the ones in the answers. Also, set the time shown in the 2nd column as the "Seconds since beginning of capture" under view->Time display format.

Question 1 . What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?

Question 2. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Ethereal window, looking for a segment with a "POST" within its DATA field.

Question 3. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the *EstimatedRTT* value (see relevant parts of Section 3.5 or lecture slides) after the receipt of each ACK? Assume that the initial value of *EstimatedRTT* is equal to the measured RTT (*SampleRTT*) for the first segment, and then is computed using the *EstimatedRTT* equation for all subsequent segments. Set alpha to 0.125.

Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the "listing of captured packets" window that is being sent from the client to the gaia.cs.umass.edu server. Then select: *Statistics->TCP Stream Graph>Round Trip Time Graph* . However, do not use this graph to answer the above question.

Question 4. What is the length of each of the first six TCP segments?

Question 5. What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

Question 6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

Question 7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).

Question 8. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Exercise 2: TCP Connection Management

Consider the following TCP transaction between a client (10.9.16.201) and a server (10.99.6.175).

No	Source IP	Destination IP	Protocol	Info
295	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [SYN] Seq=2818463618 win=8192 MSS=1460
296	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [SYN, ACK] Seq=1247095790 Ack=2818463619 win=262144 MSS=1460
297	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463619 Ack=1247095791 win=65535
298	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [PSH, ACK] Seq=2818463619 Ack=1247095791 win=65535
301	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095791 Ack=2818463652 win=262096
302	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [PSH, ACK] Seq=1247095791 Ack=2818463652 win=262144
303	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095831 win=65535
304	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535
305	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144
306	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535
308	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144

Answer the following questions:

Question 1 . What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?

Question 2. What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?

Question 3 . What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?

Question 4 . Who has done the active close? client or the server? how you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?

Question 5 . How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?

Exercise 3: Getting familiarised with ns-2 simulator (not marked, do not include in your report)

IMPORTANT NOTE: ns-2 and Nam are installed on all CSE lab machines. We do not recommend that you install ns-2 on your personal machines. This is not as straightforward as installing wireshark. Moreover, the provided scripts have ONLY been tested on CSE machines and may not work on other operating systems. We cannot offer support for running ns-2 natively on your local machines. It is possible to run ns-2 remotely via ssh.

ns-2 (http://www.isi.edu/nsnam/ns/) is a powerful simulator that provides substantial support for simulating TCP, routing and multicast protocols, among other things. It is capable of simulating the conditions that occur in wired or wireless networks. It is widely used in the research community and also in industry.

The simulator is written in C++. However, it uses OTcl as its command and configuration interface. In our lab exercises, we will use scripts written in OTcl. You will not be required to write any C++ code for any of the lab exercises. You will also not be required to write OTcl scripts from scratch. All scripts will be provided and at most ask you change a line or two in the scripts with proper instructions. To complete the lab exercises, you can safely assume ns-2 to be a black box. However, for those who are interested in finding out a bit more about ns-2, please refer to the following overview slides (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17316) which offers a good introduction.

We will also use a network animator tool: Nam (http://www.isi.edu/nsnam/nam/) . This allows us to visualise the topology and the transmission of packets during the experiments.

Illustrative Example: 2 nodes communicating directly over UDP

We will use this example to get acquainted with ns-2. The OTcl script for the first experiment is simpleSim.tcl (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17308) . It creates two nodes and simulates the sending of data packets from one node to the other over UDP. The full explanation of the script can be found here. (https://webcms3.cse.unsw.edu.au/COMP3331/18s2/resources/17315) **We strongly recommend** you read through it as it will be helpful for you for future exercises. It doesn't go over all details but rather gives you an overview of how to setup a topology, create traffic and run an experiment.

You can run the script by typing the following command:

```
$ ns simpleSim.tcl
```

This will start the nam tool and you will see the nam window (http://www.isi.edu/nsnam/ns/tutorial/nsbasic.html#nam) . When you click on the 'play' button in the window, you will see that after 0.5 seconds of simulated time, node 0 starts sending data packets to node 1. The transmission stops at t = 1.5 seconds and the simulation stops at time t = 2 seconds. You might want to slow nam down by using the 'Step' slider at the top right.

Once the experiment has concluded close nam. You will find an output trace file named 'out.tr' in your current directory. The format is as follows:

```
event time src dest PktType PktSize Flags Fid SrcSddr DestAddr SeqNum PktId
```

where:

- event:
  - r: receive at dest
  - +: enqueue
  - -: dequeue
  - d: drop
- ScrAddr and DestAddr :
  - node.port (e.g., 0.1 means node 0 , port 1)

Example trace:

```
r 0.519 0 1 cbr 500 ------- 0 0.0 1.0 1 1
+ 0.52  0 1 cbr 500 ------- 0 0.0 1.0 4 4
- 0.52  0 1 cbr 500 ------- 0 0.0 1.0 4 4
```

You may wish to examine out.tr to get a better understanding of what happens in the experiment.

We will use ns-2 in the next lab to run some experiments with TCP flows.

Resource created 4 months ago (Monday 16 July 2018, 02:50:39 PM), last modified 3 months ago (Monday 03 September 2018, 01:10:41 PM).

Comments

(/COMP3331/18s2/forums/search?forum_choice=resource/17355) (/COMP3331/18s2/forums/resource/17355)

Add a comment

Yiwei Han (/users/z3463398) 2 months ago (Fri Sep 07 2018 00:44:11 GMT+1000 (澳大利亚东部标准时间))

In exercise 2, is packet 308's sequence number correct? Shouldn't it be the same as the acknowledgment number of packet 306?

Reply

Ali Dorri (/users/z5095883) 2 months ago (Fri Sep 07 2018 08:35:11 GMT+1000 (澳大利亚东部标准时间))

Yes, that is correct. Actually this fact can help you in answering the question. Also check packets 304 and 305

Reply

Joshua Goodman (/users/z5117211) 2 months ago (Thu Sep 06 2018 21:36:53 GMT+1000 (澳大利亚东部标准时间))

Question 7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text)
What does this mean by "every other segment"? Is this asking for scenarios where literally only every second segment is acknowledged?

Reply

Nadeem Ahmed (/users/z3003139) 2 months ago (Thu Sep 06 2018 22:20:11 GMT+1000 (澳大利亚东部标准时间))

Yes. In Delayed Ack mechanism, TCP receiver waits for the arrival of another segment before issuing a single ACK acknowledging two segments.

Reply

Oliver Wolff (/users/z5115209) 2 months ago (Thu Sep 06 2018 21:18:13 GMT+1000 (澳大利亚东部标准时间))

In exercise 2 question 4., it asks "Who has done the active close?"

Does this ask 'who initiated the close?'

Reply

Nadeem Ahmed (/users/z3003139) 2 months ago (Thu Sep 06 2018 22:22:15 GMT+1000 (澳大利亚东部标准时间))

Active close is performed by the side that issues the FIN first, without waiting for the arrival of a FIN from the other end. Recall the side doing Active close finally has to wait for 2 MSL wait state after issuing the final ACK.

Reply

Oliver Wolff (/users/z5115209) 2 months ago (Thu Sep 06 2018 22:48:22 GMT+1000 (澳大利亚东部标准时间))

What is a MSL wait state? I do not see that stated anywhere in the lecture notes.

Reply

Nadeem Ahmed (/users/z3003139) 2 months ago (Thu Sep 06 2018 23:11:29 GMT+1000 (澳大利亚东部标准时间))

Week 6 slides 15, 16 and 17. It is mentioned under TIME_WAIT state. 2 x Max Segment Lifetime.

Reply

Xavier Poon (/users/z5110093) 2 months ago (Thu Sep 06 2018 17:09:54 GMT+1000 (澳大利亚东部标准时间))

do we have to put screenshots in our answers?

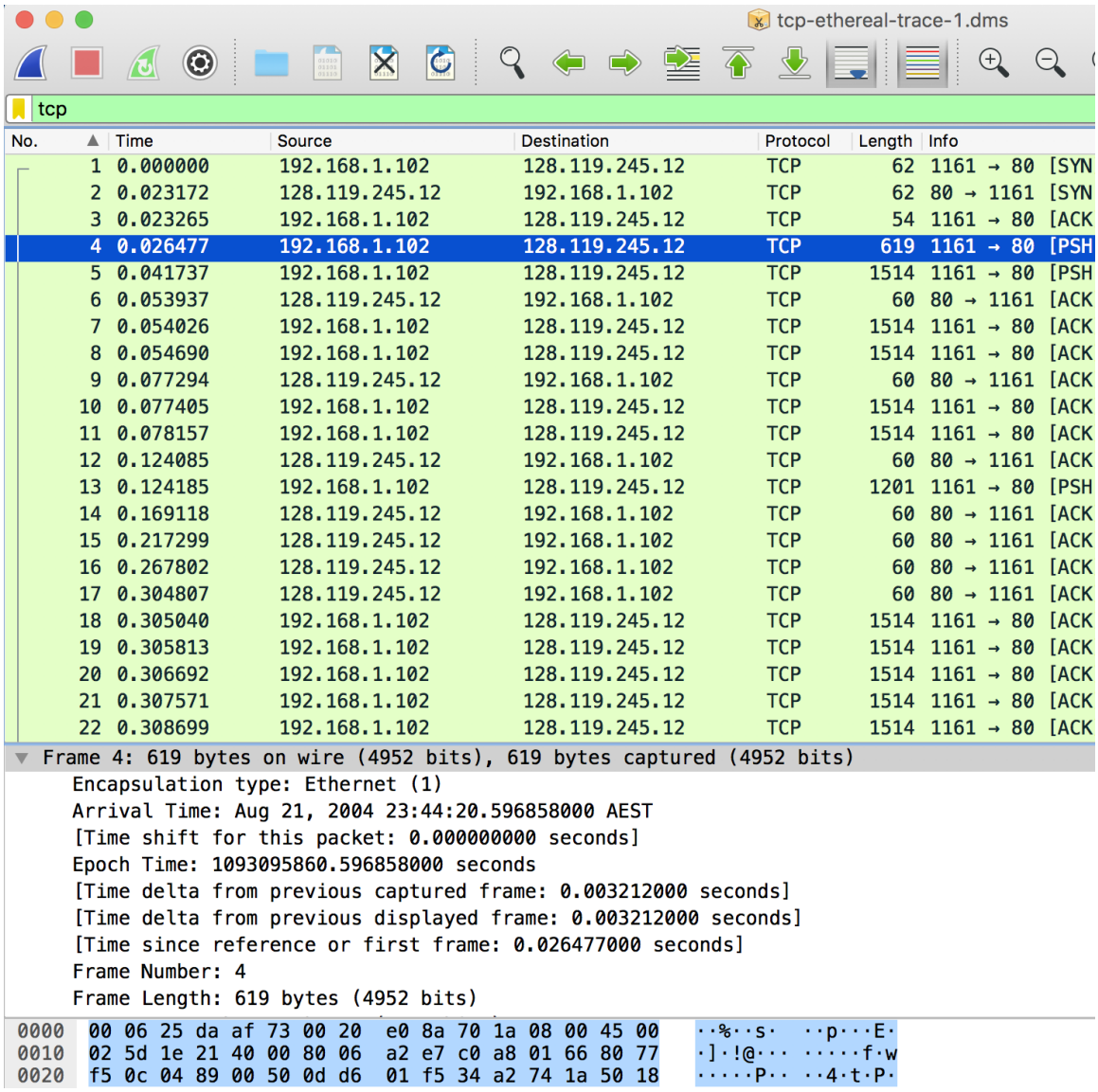
Reply

Nadeem Ahmed (/users/z3003139) 2 months ago (Thu Sep 06 2018 22:34:14 GMT+1000 (澳大利亚东部标准时间))

Not required.

Reply

Wenxin Wang (/users/z3454684) 3 months ago (Tue Sep 04 2018 22:22:04 GMT+1000 (澳大利亚东部标准时间)), last modified 3 months ago (Tue Sep 04 2018 22:34:27 GMT+1000 (澳大利亚东部标准时间))



I followed all the instruction to set Wireshark. and I got the 4th segment like this. It didn't show that POST mentioned in the instruction. Am I on the right track? or something wrong?

Reply

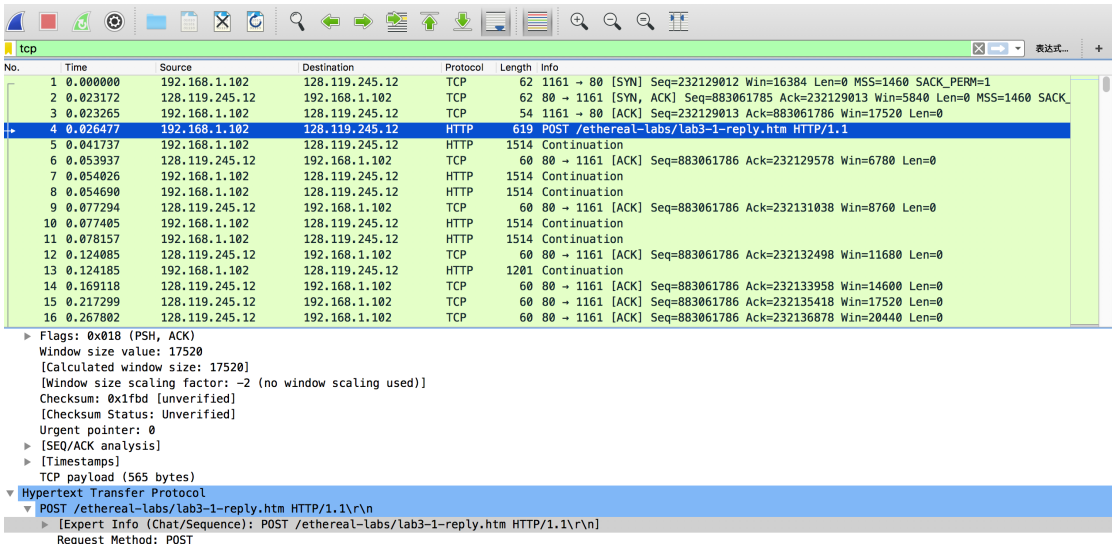
Ali Dorri (/users/z5095883) 3 months ago (Wed Sep 05 2018 09:03:52 GMT+1000 (澳大利亚东部标准时间))

You need to check the content of the packet

Reply

Bojing Fu (/users/z5142286) 3 months ago (Tue Sep 04 2018 23:01:30 GMT+1000 (澳大利亚东部标准时间))

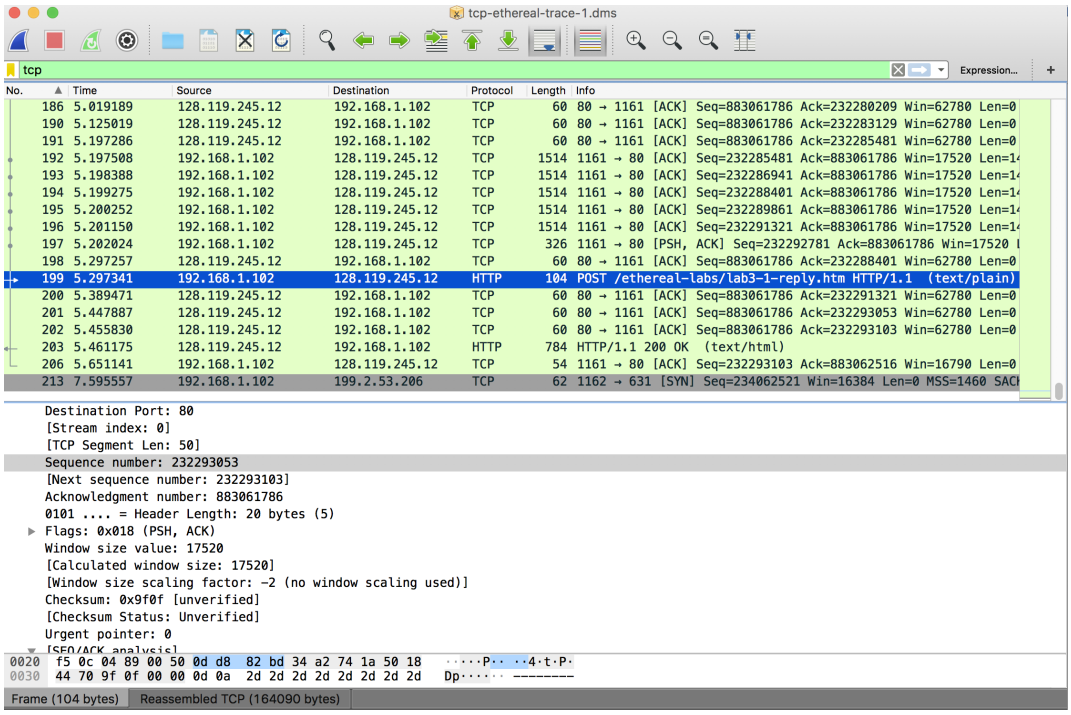
My wireshark shows like this, my No4 segment shows that it is a HTTP(POST) request instead of TCP.



Reply

Wenxin Wang (/users/z3454684) 3 months ago (Tue Sep 04 2018 23:04:39 GMT+1000 (澳大利亚东部标准时间))

For Q3 in Ex1, I use the No199 showing in blue to answer the question so I got different answer with you. I thing we are something different of the Wireshark sorting. This does make some different to answer Q2 and Q3.



Reply

Ali Dorri (/users/z5095883) 3 months ago (Wed Sep 05 2018 09:05:59 GMT+1000 (澳大利亚东部标准时间))

That is not the correct one to use. Check my comment above.

Reply

Samuel Hernandez Romero (/users/z5022909) 3 months ago (Tue Sep 04 2018 22:52:32 GMT+1000 (澳大利亚东部标准时间))

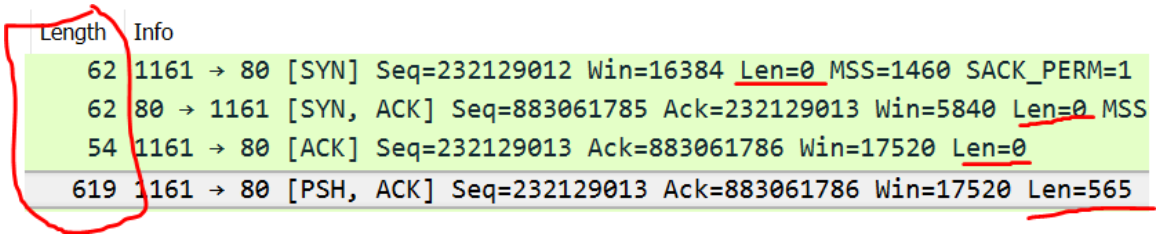
Can you see 'Dp....PO ST /ethe' in the next line (0030) ?

Reply

Ramal Ratnayake (/users/z5085255) 3 months ago (Tue Sep 04 2018 11:22:37 GMT+1000 (澳大利亚东部标准时间))

In exercise 1 q4, it asks for the size of each packet. Could someone clarify the difference between the two fields in Wireshark (marked on picture below)?

Thanks



Reply

Ramal Ratnayake (/users/z5085255) 3 months ago (Tue Sep 04 2018 17:26:56 GMT+1000 (澳大利亚东部标准时间))

Thank you to both of you! Makes sense now

Reply

Ali Dorri (/users/z5095883) 3 months ago (Tue Sep 04 2018 16:44:53 GMT+1000 (澳大利亚东部标准时间))

As the LIC mentioned, the left side length is the total size of the packet including the headers and the right one is the size of the data.

Reply

Nadeem Ahmed (/users/z3003139) 3 months ago (Tue Sep 04 2018 14:15:43 GMT+1000 (澳大利亚东部标准时间))

Information is not complete from this screenshot. Look at the length at various layers of the TCP/IP stack. Len=0 means zero payload for TCP (only TCP headers. How much is the header length? you have to look at details by clicking the TCP layer). Similarly seems like the IP header is using options in header as well. Do some accounting and your total length of frame shown on left should match up with all headers added together.

Reply

Amir Harambasic (/users/z5165168) 3 months ago (Tue Sep 04 2018 02:43:35 GMT+1000 (澳大利亚东部标准时间))

For Part 1 *Question 4.* What is the length of each of the first six TCP segments?

Does this refer to the first 6 segments overall from both sides or the first 6 segments after post as from the previous question.

Reply

Nadeem Ahmed (/users/z3003139) 3 months ago (Tue Sep 04 2018 07:15:40 GMT+1000 (澳大利亚东部标准时间))

First 6 Segments from Question 3.

Reply

Jialun Li (/users/z5172023) 3 months ago (Sun Sep 02 2018 13:19:54 GMT+1000 (澳大利亚东部标准时间)), last modified 3 months ago (Sun Sep 02 2018 14:43:25 GMT+1000 (澳大利亚东部标准时间))

[deleted]

Reply

Ali Dorri (/users/z5095883) 3 months ago (Sun Sep 02 2018 14:20:36 GMT+1000 (澳大利亚东部标准时间))

To find the segment that contains POST, check the content of each segment, You will see a POST in the message content.

Reply

Abanob Tawfik (/users/z5075490) 3 months ago (Sat Sep 01 2018 21:13:36 GMT+1000 (澳大利亚东部标准时间))

for question 3, do we consider the acks that the server is sending back to the client as part of the segment? or is it purely the psh ack segments?

Reply

Ali Dorri (/users/z5095883) 3 months ago (Sat Sep 01 2018 22:13:49 GMT+1000 (澳大利亚东部标准时间))

Can you clarify which question you exactly mean? q3 in exercise 1 or 2?

Reply

Abanob Tawfik (/users/z5075490) 3 months ago (Sat Sep 01 2018 22:29:37 GMT+1000 (澳大利亚东部标准时间))

in exercise 1

Reply

Ali Dorri (/users/z5095883) 3 months ago (Sat Sep 01 2018 23:29:22 GMT+1000 (澳大利亚东部标准时间))

To answer the question you need to check the time of the ACK **packets** sent from the server to the client.

Reply

Dongdong Chen (/users/z5189803) 3 months ago (Sat Sep 01 2018 15:39:34 GMT+1000 (澳大利亚东部标准时间))

Dear Ali,

To turn off this feature, you have to click Edit->Preferences>Protocols->TCP (or Wireshark->Preferences->Protocols->TCP) and then disable the “Relative Sequence Numbers” option. Note that the answers in the solution set will reflect this change.

Does that mean that we have to turn off this option, using the real sequence numbers to answer those questions?

Reply

Ali Dorri (/users/z5095883) 3 months ago (Sat Sep 01 2018 19:16:24 GMT+1000 (澳大利亚东部标准时间))

Yes, in other words it must not start with 0.

Reply

Dongdong Chen (/users/z5189803) 3 months ago (Sat Sep 01 2018 20:35:51 GMT+1000 (澳大利亚东部标准时间))

Thank you.

Reply

Mohamed Al Mouiee (/users/z5114185) 3 months ago (Sat Sep 01 2018 17:45:09 GMT+1000 (澳大利亚东部标准时间))

Do not use relative sequence numbers, so uncheck the "relative sequence numbers" box, if it's on by default.

Reply

Dongdong Chen (/users/z5189803) 3 months ago (Sat Sep 01 2018 20:36:01 GMT+1000 (澳大利亚东部标准时间))

Thank you.

Reply

Load More Comments