Started on	Friday, 30 October 2020, 10:12
State	Finished
Completed on	Friday, 30 October 2020, 14:30
Time taken	4 hours 17 mins
Feedback	Your overall score for this part is between 90% and 100%. The exact grade as well as details for each question will be made available as soon as the submission period for the quiz closes.

Question **1**

Complete

Marked out of 1.00

Lab3 Part2 Question 1:

How many messages do you receive when you send the command

```
CMD short:0
```

? 15

How many messages do you receive when you send the command

```
CMD short:1
```

? 15

Watch the terms used in the question.

We ask you to count PMU messages. What is a PMU message ?

Question **2**

Complete

Marked out of 1.00

Lab3 Part2 Question 2:

In Wireshark, how many packets FROM the server with a PAYLOAD length STRICTLY positive do you observe:

With the command

`CMD_short:1`

? 15

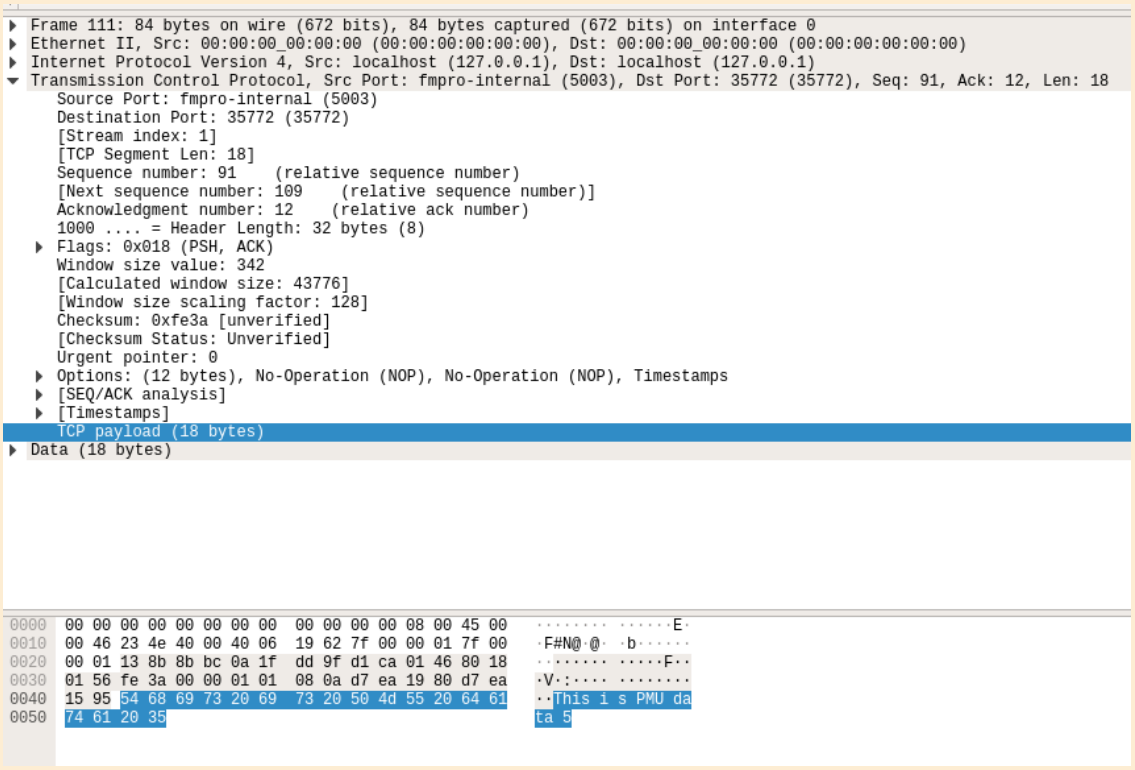
With the command

`CMD_short:0`

? 11

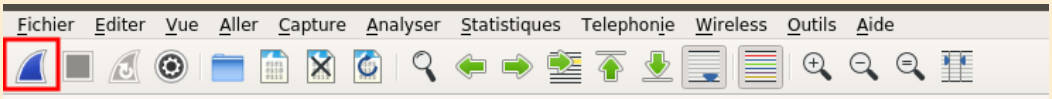
We specifically ask you to consider ONLY the packets (=TCP segments) that have a non-empty payload.

In Wireshark, once you select a packet, in the packet details, expand the tree corresponding to the TCP protocol, and look for the payload field to get its length.

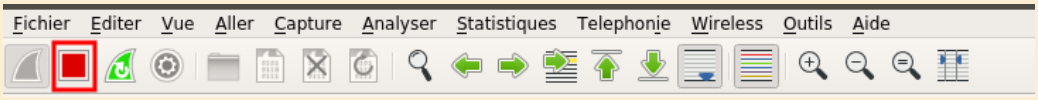


You have lot of packets displayed in Wireshark and you don't really know where are the packets you're looking for ? Here are several tips for you:

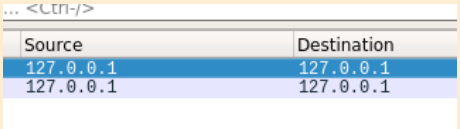
TIP1: Try to launch the Wireshark capture JUST BEFORE you launch the client scrip.



And to stop it JUST AFTER the client script is completed (with CMD_sort:0, it's very fast!)

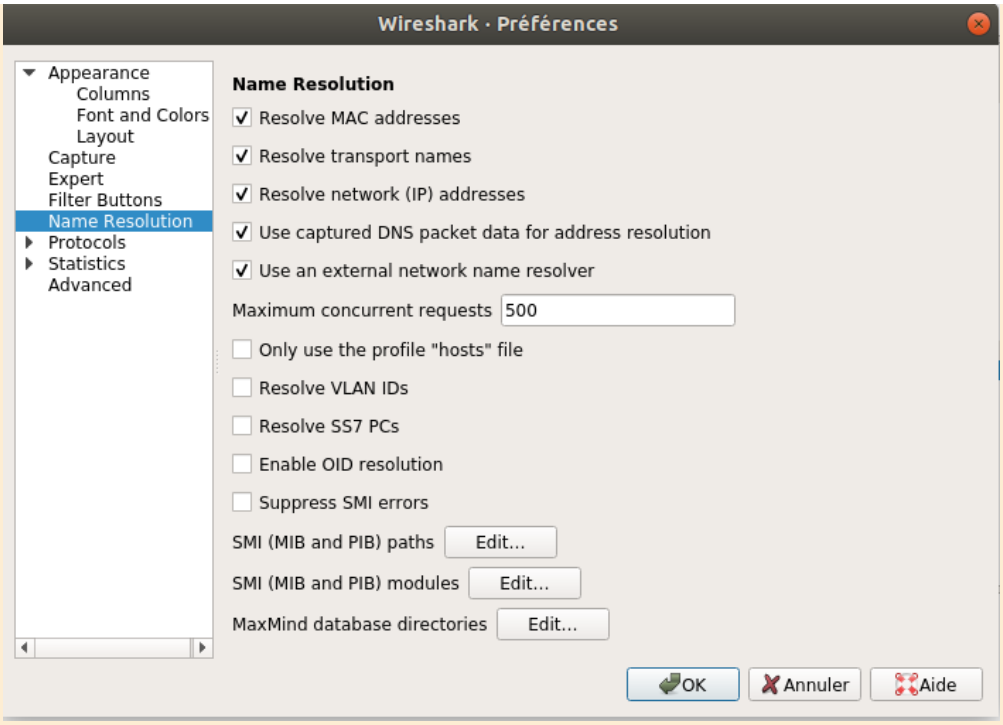


TIP2: In the list of packets, IPs displayed in the "Source"/"Destination" columns are difficult to interpret (learning by heart all the IPv4 addresses is not mandatory for networking students).



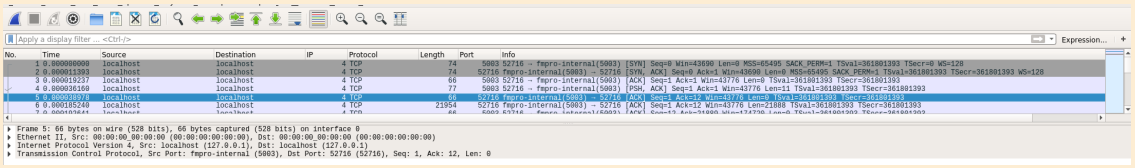
We can ask Wireshark to resolve for us the IP addresses into (domain) names.

Open the preferences window (Edit>Preferences). In the "Name Resolution" section, make sure the boxes are checked as follows:

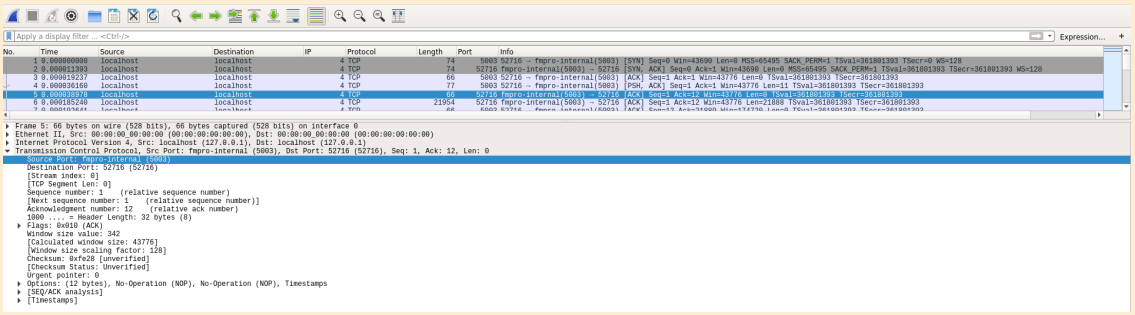


And now the IPs should be resolved, which will help you identify traffic coming from tcpip.epfl.ch (you might need to restart the capture).

TIP3: Filters ! Filters in Wireshark are a powerful tool that you need to master. However, you are not required to learn by heart all the names of the filters. So here is a tip for you to identify the filter you need. Let's assume we want to display only the packets sent FROM the port 5003. The first step is to find and select one of these packets.

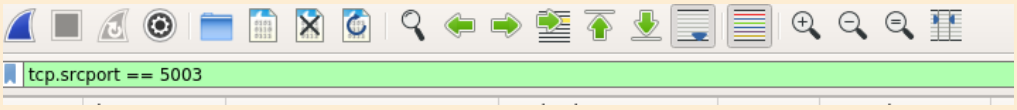


In the above capture, we found and selected one of the packets sent by the server. Then, if we want to filter through the port number, we know that port numbers are located in the TCP header, so we expand the TCP field in the packet inspector.



In the above capture, we identified the "source port" field. Now, use a right click, select "Prepare a filter" and "Selected". In the case you already had a filter prepared in the filter field, you can choose to combine the current filter using one of the option "...and selected", "...or selected", etc.

This will create for you a filter in the filter field. Notice the green highlighting that indicates a correct format. Your cursor will also be placed in the filter field, and with the cursor IN the filter field, you need to press ENTER to apply the filter.



Question **3**

Complete

Marked out of 3.00

Lab3 Part2 Question 3:

The goal of this question is to explain the observations made in the two previous answers.

With

`CMD_short:1`

, you receive exactly 1 message per packet .

With

`CMD_short:0`

, you receive sometimes more than 1 message per packet .

If we take a look at the server code, we find the following line which is executed whatever the `CMD_short` command it receives:

```
c.send(message.encode())
```

where:

- `message` is one of the PMU message you identified in Question 1
- `c` is a socket provided by the `accept()` call when your PDC client connects to the PMU. See part 1.2 as well as the lecture.

The `send()` call made by the server's application:

- asks the server OS to create a TCP segment containing `message` and to send it via the connection to the client
- puts `message` in a queue and asks the server OS to send the message (via TCP), but without specifying how
- puts `message` in a queue but asks the server OS to wait a bit before sending the message (via TCP)

. The TCP segment is created by the server OS when the server OS decides it, according to some internal algorithm .

Usually, the chosen algorithm is the Nagle 's algorithm.

Check the lecture:

TCP-IP 2018: The Transport Layer: TCP and UDP (part 3)

Question **4**

Complete

Marked out of 2.00

Lab3 Part2 Question 4:

From a programmer's perspective, how can your application be sure it received all the messages from the PMU server ?

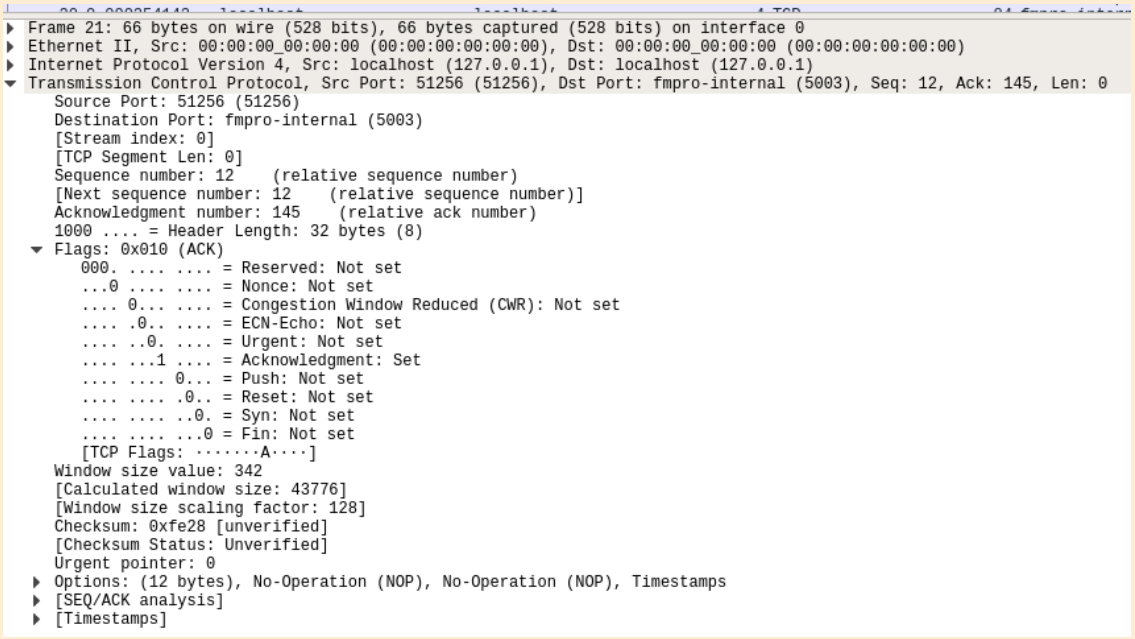
- Your application should catch the `SocketNoMoreDataException` exception. When raised, it means that the socket has no more data to read
- Your application should wait 30 seconds. If more than 30 seconds are elapsed since the last byte has been received, it means the server has closed the connection and has no more data to send
- Your application should test the value returned by `recv()`. When this value has a `False` boolean representation, it means that the connection has been orderly closed by the remote end and there is no more data to be read.
- Your application should wait for the user to terminate the program (`CTRL+C`).
- Your application can never be sure to have received all the data, packets might have been lost in the network.

From a network engineer's perspective, which of the following flags is set on the TCP header when the sender informs that it closes the connection and has nothing more to send ?

Answer: Fin

For the first part of the question, refer to Question6 in part 1.2 of the lab.

For the second part of the question, in Wireshark, you can see the flags by extending the corresponding field in the packet details:



Question **5**

Complete

Marked out of 1.00

Lab3 Part2 Question 5:

In Wireshark, how may packets with a NON EMPTY payload have you seen coming from the PMU TCP server ?

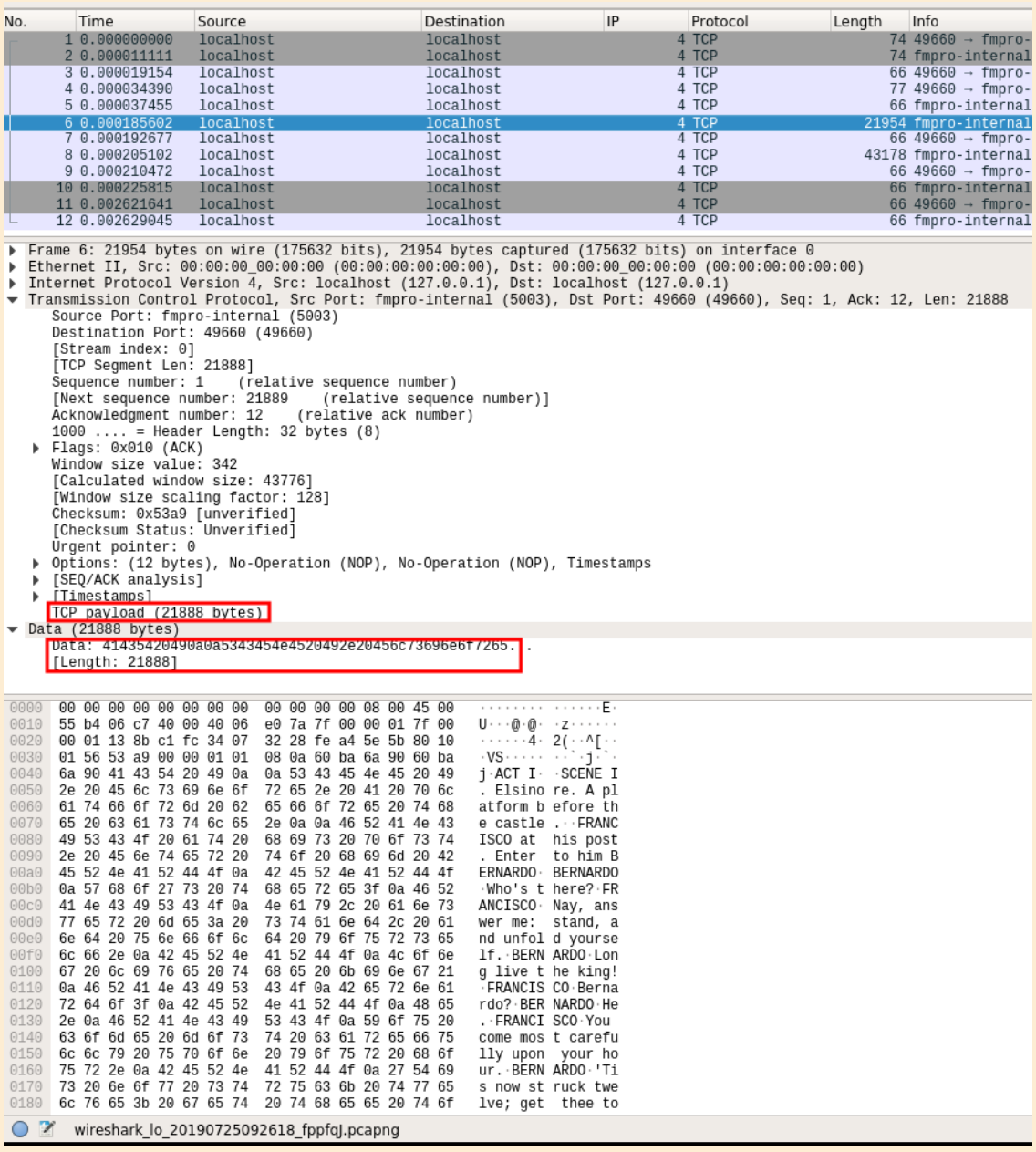
Answer: 48

See help of Question2.

Here is an additional tip when using Wireshark:

You potentially have many packets with a non-empty payload and you don't want to count all of them ? In this case, we can use Wireshark filters !

First, let's have a look at one packet with a non empty payload, we extend the TCP and the "Data" tree in the packet detail field:

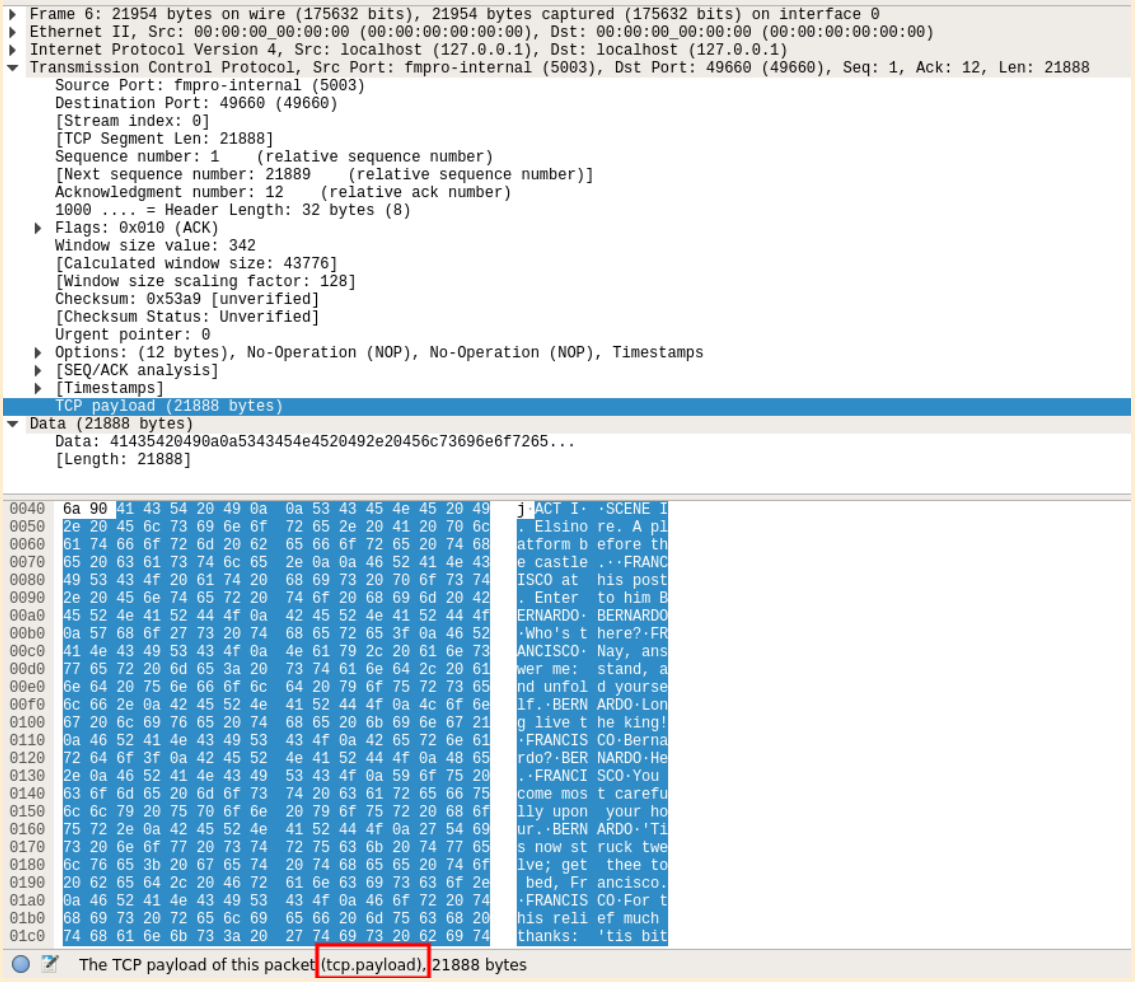


Three fields are of interest in terms of payload (see red boxes above):

- The "TCP payload" field in the section "TCP"
- The "Data" field in the section "Data"
- The "Length" field in the section "Data". Note that the "Length" field is shown in "[]" brackets.

Brackets indicate a value computed by Wireshark but not directly readable in the packet. It usually concerns length (Wireshark counts the bytes for you), sequence numbers, Acknowledgment analysis, etc.

Now let's retrieve the name of the three fields (rather than their DISPLAY name). If you select one of the field, its name will be prompted at the bottom of the Wireshark window:



Now our goal is to create a DISPLAY FILTER that displays only the packets with a non empty payload, using one of the three options.

What to write in the display filter and which of the three fields to use ?

TCP Payload and Data fields hold the (binary) value of the payload while Data Length equals their length.

At first glance, Data Length could be seen as easier to use, and our display filter would look like "We want the data length to be strictly positive", ie:

```
data.len>0
```

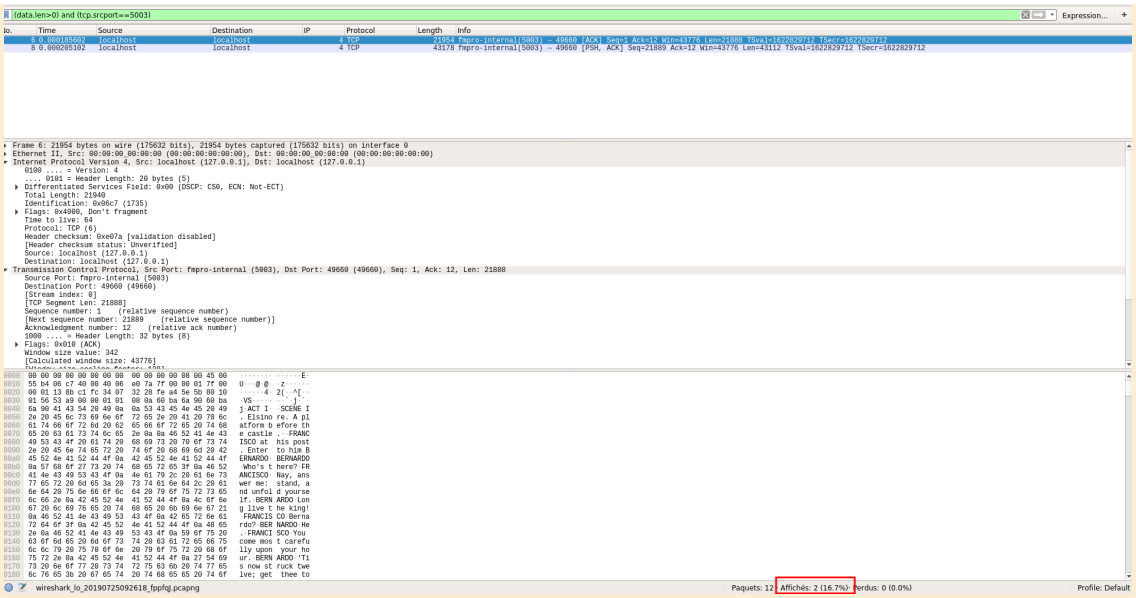
However Wireshark manages booleans the same way a programming language does. Specially, anything that is different to "null" or "0" or "non-existent" is considered as "True". It means that

tcp.payload
is also a valid display filter and will display all packets with an existing TCP payload.

Don't forget to combine one of the possible filters with a filter selecting the traffic coming FROM the PMU. In Wireshark, display filters can be combined with boolean operators.

Once the display filter is green, is means the filter can be accepted (otherwise you have a synthax issue, check the field names with the above method). Don't forget to press ENTER with your cursor in the display filter field to apply the filter.

Only the packets that match the filter will be displayed, you can see their count on the bottom right corner of the window:



Question **6**

Complete

Marked out of 2.00

Lab3 Part2 Question 6:

How many times was the

`recv()`

call invoked at the client side ? 20

Is the number of packets you see in Wireshark the same as the number of

`recv()`

invocations at your client ?

- Yes
- No

What can you change IN YOUR CLIENT CODE to change the number of

`recv()`

invocations ? Answer with a short sentence:

the buffer size argument passed to the recv() function

Refer to Question 5 of Part1.2.

For the last question, we are mostly expecting 2 keywords that are the name of the value you can change.

Question **7**
Complete
Marked out of 3.00

Lab3 Part2 Question 7:

This question will summarize your findings of part 2. Complete the following text

TCP is a stream-oriented protocol.

From the programmer's perspective, TCP provides a service which is analogous to:

- A post system distributing post cards. Each postcard contains fields that the sender is required to provide: a date, an opening "Dear...", a message of known maximum size (the size of a postcard), a closing form that encodes the relationship between the sender and the recipient and a signature. The post office also adds the date and distribute the postcard to the recipient.
- A garden hose without any hole: anything that enters the hose will exit it on the remote end in the same order, irrespective of whether the hose is transporting water, fertilizer, pesticides or soda.

Who decides how the data transported by TCP must be formatted and interpreted ?

The transport layer .

In which of the following applications is a stream-oriented transport protocol a good solution ?

- [x] Downloading a large file using FTP (File Transfert Protocol)
- [ ] Sending homemade control messages of a small size from a PMU to a PDC, each one spaced by 1second
- [x] Loading a web page containing one unique large image using HTTP (HyperText Transfert Protocol)

For the last part of the question, consider the following:

What are HTTP, FTP protocols ? What is their purpose ?

When you create homemade control messages, do you use HTTP or FTP or any equivalent protocol ?

You were asked to prompt one PMU message per line, did you achieve it ? Was is easy ?

[Graded] Lab3 - Part 1.2 | Jump to... | [Graded] Lab3 - Part 3

/