| | |
|---|---|
| **Started on** | Wednesday, 11 November 2020, 15:03 |
| **State** | Finished |
| **Completed on** | Wednesday, 11 November 2020, 15:06 |
| **Time taken** | 2 mins 58 secs |
| **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 2.00

Flag question

Lab3 Part6 Question 1:

How many packets with a TCP payload STRICTLY positive do you receive from the server ?

4

How many messages did you receive ?

10

How many time was the function recv() called ? (or the event on_message called)

11

Which of TCP or Websocket required for you the most complex handling of messages in order to print one message per line ?

◉ TCP ○ Websocket

First three questions : no specific hint

Last question: Try to submit your Websocket script to the submission system and tell us which of Part2 or Part6 took you more time to pass the tests of the submission system ?

Question **2**

Complete

Marked out of 1.00

Flag question

Lab3 Part6 Question 2:

With CMD_floodme,

How many packets with a TCP payload STRICTLY positive do you receive from the server ?

17

How many time was the function recv() called ? (or the event on_message called)

1

NA

Question **3**

Complete

Marked out of 2.00

Flag question

Lab3 Part6 Question 3:

Let's conclude !

TCP is a stream-oriented transport-layer protocol.

Websocket is a message-oriented application-layer protocol.

UDP is a datagram-oriented transport-layer protocol.

Question **4**

Complete

Marked out of 2.00

Flag question

Lab3 Part6 Question 4:

While the notions of datagram-oriented and message-oriented might be close, there are some subtle yet fundamental differences, let's review them:

After successful graduation with outstanding grades in the TCP/IP lecture, you quickly become chief system architect for a project within a big Internet operator.

The system you are in charge is based on the Internet protocol. For each of the following independent situations, we provide a requirement for that system. For each situation, choose the solution that best fits the requirement with the minimal effort.

**Situation A:**

Requirement: The system exchanges periodic messages of control data, of size 2500 bytes.

Which solution do you choose ?

- (•) A solution based on WebSockets
- ( ) A solution based on UDP

**Situation B:**

Requirement: In the system, power, energy and computational capabilities are extremely limited.

Which solution do you choose ?

- ( ) A solution based on WebSockets
- (•) A solution based on UDP

**Situation C:**

Sometimes, real-time systems are classified into two categories: soft real-time or hard real-time.

In a **soft real-time system**, delayed information has still a positive added value, but this value is decreasing

with time.

Example: The news system. Imagine you learn that in 2018, a big company has bought the rights to create a TV series based on "The Foundation" books by I.Asimov. If your receive the information six months late, that piece of news has still a positive added value for you (you learned something). But if you receive the piece of news 10 years late (that is, after the TV series has been produced and after you watched it), then the information has lost most of its value, but in any case the value is never negative.

In a **hard real-time system**, delayed information has null or negative added value.

Example: A fire alarm. Imagine you hear a fire alarm delayed by one day. Then this information is of no use and can even have a negative effect, spreading panic among people.

Requirement: The system must be **hard-real time**

Which solution do you choose ?

○ A solution based on WebSockets ◉ A solution based on UDP

**Situation D:**

Requirement: The system is to be operated on the public Internet. Congestion and middle-boxes must be expected

Which solution do you choose ?

◉ A solution based on WebSockets ○ A solution based on UDP

Hints:

**Situation A:** Note that 2500 is over the MTU of most link-layer technologies. Think what happens when you want to sent data of more than one MTU with each of the technologies.

**Situation B:** Encapsulation and headers ...

**Situation C:** In a hard real-time system, what should we do with lost data ?

**Situation D:** Consider how middle-boxes may react for each protocol, assuming congestion is expected.

Finish review

◄ [Graded] Lab3 - Part 5

Jump to...

Lab3 - Link to the scoring system ►

## Quiz navigation

1 2 3 4

Show one page at a time

Finish review

EPFL

Contact EPFL CH-1015 Lausanne +41 21 693 11 11