Started on	Wednesday, 11 August 2021, 2:25 PM
State	Finished
Completed on	Monday, 16 August 2021, 8:18 PM
Time taken	5 days 5 hours
Marks	23.00/23.00
Grade	**10.00** out of 10.00 (**100**%)

Information

When working through the following quiz please keep the following points in mind:

- You are asked to implement a number of Python functions. The default language is Python3.
- Occasionally you will have to use mathematical functions from the Python 'math' package. You will need to add suitable 'import' statements to your solutions and when using such a function you will have to prefix them with the package name, e.g. to call the function 'floor' you will have to write 'math.floor' in your code.
- In many questions it is asked that you enter numerical answers in certain units. For example, we may ask you to enter your answer in milliseconds. If your answer is 0.050 seconds then you will have to enter '50', as this is the equivalent value in milliseconds.

Question **1**

Correct

Mark 1.00 out of 1.00

In the Internet protocol stack the TCP protocol is in layer 4 (transport layer), the IP protocol is in layer 3 (Internet layer), and the ubiquitous Ethernet technology and protocol jointly covers layers 1 and 2 (Physical and Network Interface layers). We are given an Ethernet frame containing an IP packet, which in turn contains a TCP segment. In which sequence (from left to right) will the headers of TCP, IP and Ethernet show up in the resulting packet?

Select one:

- a. Ethernet, IP, TCP ✔
- b. TCP, IP, Ethernet
- c. IP, Ethernet, TCP
- d. Ethernet, TCP, IP
- e. IP, TCP, Ethernet

Your answer is correct.

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Suppose we are a protocol running at a layer N that is not the highest (application) layer. Most if not all protocols that are not running in the application layer have some field in their N-layer header that is meant for "payload multiplexing" or "protocol multiplexing", and which tells to which layer N+1-protocol (or application) the payload of the N-layer packet belongs. In the Internet, a typical protocol configuration consists of the TCP protocol (running in the transport layer) embedding its packets into the IP protocol (running on the network layer), which in turn can embed its packets into an Ethernet frame. The protocol multiplexing fields in the different headers are called:

- Ethernet: 'Type/Length' field
- IP: 'Protocol' field
- TCP: 'DstPort' and 'SrcPort'

Find out (either by Googling or by using corresponding COSC 264 slidesets ["Local Area Networks and Ethernet" and "IPv4 and Related Protocols"]) what the required values in the Ethernet and IP protocol multiplexing fields are for a TCP/IP/Ethernet packet. We are using IP version 4 here.

Select one:

- a. Ethernet = 0x0800, IP = 0x06 ✔
- b. Ethernet = 0x4711, IP = 0x97
- c. Ethernet = 0x1000, IP = 0x10
- d. Ethernet = 0x497E, IP = 0x05
- e. Ethernet = 0x0806, IP = 0x01

Your answer is correct.

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Consider a large network made up of routers and end hosts and focus on two particular end hosts that can reach each other via three intermediate routers. Suppose we use the TCP/IP reference model. Which layers do the routers need to minimally implement?

Select one:

- a. Physical, Network Interface, Internet ✔
- b. Physical, Network Interface, Internet, Transport
- c. Physical
- d. Physical, Network Interface, Internet, Transport, Application
- e. Physical, Network Interface

Your answer is correct.

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Same setup as in the previous question. Suppose that the Physical and Network Interface layers work perfectly on all links in the network, i.e. they have excellent facilities for transmission error detection / correction and never lose a packet during transmission. Suppose we want to transmit a large file reliably and in the right sequence between the two considered end hosts. Indicate whether or not the following statement is true: "we already account for all possible errors and do not need additional error detection / correction mechanisms on the transport layer".

Select one:

- True
- False ✔

Packets can also get lost while being in the memory of routers and not just when they are in transit over a link, hence additional mechanisms are needed on the transport layer. So the statement is false.

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

In which layer of the OSI reference model would you put the TCP protocol?

Select one:

- a. Link
- b. Representation
- c. Network
- d. Transport ✔
- e. Physical
- f. Session
- g. Application

Your answer is correct.

Correct

Marks for this submission: 1.00/1.00.

Information

Suppose we have captured the following, 60 bytes long packet from an Ethernet interface (e.g. with the wireshark tool):

```
0000  00 0c 29 ae 8e 88 00 50  56 e0 07 01 08 00 45 00
```

```
0010  00 2c 9c 3f 00 00 80 06  57 35 5b bd 59 58 c0 a8
```

```
0020  d1 99 00 50 cc b9 02 bf  5c fa 4f bb eb 82 60 12
```

```
0030  fa f0 ed cc 00 00 02 04  05 b4 00 00
```

In this output, the first column just counts bytes (in hexadecimal) and does not belong to the actual packet data. The next sixteen columns are the relevant ones, they contain the packet data, each entry is one byte given as a hexadecimal number. You can ignore the last two bytes.

This packet represents a TCP segment encapsulated in an IP packet, which in turn is encapsulated into an Ethernet frame. All the protocol headers have their minimum possible size (note that some protocols, e.g. TCP or IP, allow for extensions or options, which can make the headers a bit longer, but this does not happen here).

For the following questions you will need to peek forward to the "Local Area Networks and Ethernet" slideset (which gives the Ethernet header format) and the "IPv4 and related protocols" slideset, which shows the IPv4 header format. **Important:** Note that the Ethernet header included in the 60 bytes does not include the Ethernet preamble and SFD header fields.

Question **6**

Correct

Mark 1.00 out of 1.00

Find out the length of the Ethernet header here. Peek forward into the section on Ethernet in the "Local Area Networks and Ethernet" slideset and note that that the captured data does not contain the preamble and SOF fields of the Ethernet header. In your response, give the value (in hex, e.g. "0xAB") of the first byte immediately following the Ethernet header.

Answer: 0x45 ✔

The Ethernet header is 14 bytes long (6 B DstAddr, 6 B SrcAddr, 2 B Type/Length)

Correct

Marks for this submission: 1.00/1.00.

Question **7**

Correct

Mark 1.00 out of 1.00

Which protocol header starts with the byte you have identified in the previous problem?

Select one:

- a. Ethernet
- b. IP ✔
- c. TCP

Your answer is correct.

Correct

Marks for this submission: 1.00/1.00.

Question **8**

Correct

Mark 1.00 out of 1.00

Now consult the packet format for the IPv4 header by peeking ahead into the slideset "IP and Related Protocols" (particularly the section on the IPv4 packet format). What is the value in the version field? Please give it as a decimal number.

Answer: 4 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **9**

Correct

Mark 1.00 out of 1.00

What is the minimum length of the IP header in bytes (without any options)? Give a number.

Answer: 20 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **10**

Correct

Mark 1.00 out of 1.00

What is the value in the 'ProtocolType' field in the IP header? Give the value in hexadecimal (e.g. "0xAB").

Answer: 0x06 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **11**

Correct

Mark 1.00 out of 1.00

The IP source address field is a 32-bit field. One usually specifies IP addresses in the "dotted-decimal notation", in which the value of each byte of the 32-bit address is given separately as a decimal number, from the highest to the lowest value and separated by dots. An example IP address is "130.149.49.77". Please give the value of the IP source address field in dotted-decimal notation. Note that all 16- and 32-bit fields in the IPv4 header are given in the Big Endian format.

Answer: 91.189.89.88 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **12**

Correct

Mark 1.00 out of 1.00

Give the value of the IP destination address field in dotted-decimal notation.

Answer: 192.168.209.153 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **13**

Correct

Mark 1.00 out of 1.00

Give the value of the first byte of the TCP header as a hexadecimal number (e.g. "0xAB").

Answer: 0x00 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **14**

Correct

Mark 1.00 out of 1.00

Now consult Wikipedia to find out the precise format of the TCP header. What is the value of the 'SourcePort' field? Please give it as a decimal number. Again, recall that 16- and 32-bit header fields are given in Big-Endian order.

Answer: 80

Correct

Marks for this submission: 1.00/1.00.

Question **15**

Correct

Mark 1.00 out of 1.00

What is the value in the TCP 'DestinationPort' field? Please give it as a decimal number.

Answer: 52409

Correct

Marks for this submission: 1.00/1.00.

Information

Fragmentation and Reassembly is a standard protocol mechanism which is included in several real-world protocols, for example the ubiquitous IP protocol in version 4.

Let us consider the general case, where F+R is carried out by some layer N in a protocol stack, which is neither the lowest nor the highest layer. The layer N implementation at the transmitter station accepts a large data block for transmission from layer N+1, let us say it consists of *S* bytes, and wants to transmit that to the peer layer-N entity in the receiver. To do so, layer N has to use the services of layer N-1 to transmit the data to the receiver. However, layer N-1 accepts from layer N only packets up to a certain maximum size, say *M* bytes, which here we assume is substantially smaller than the size of the message handed over by layer N+1 (i.e. $M < S$).

Therefore, layer N breaks up the N+1 data into smaller chunks, and forms a separate layer N packet from each of these chunks by adding the layer N-header (and trailer) -- we call these packets the fragment packets or simply fragments. We assume that the total size of the layer-N header and trailer is *O* bytes. The size of the resulting fragments generated by layer N must not exceed *M* to be compatible with the service offered by layer N-1.

After having broken down the N+1 data into a number of layer-N fragments, the fragments (which, as a reminder, are complete layer N packets!) are handed over to layer N-1 for transmission and eventually sent over the network to the receiver. The difficulty now is that the network can loose or re-order the fragments, i.e. they may not arrive at the receiver in the sequence they have been sent.

The goal is to construct a protocol mechanism at layer N that allows the receiving layer N+1 entity to receive the message generated by layer N+1 at the transmitter in unmodified form, i.e. as one large piece of data without alterations. The operation of fragmentation and reassembly at layer N shall remain transparent to layer N+1.

To achieve this, the layer-N entity at the receiver will collect the layer-N fragment packets in a buffer until all fragments have been received or until a timer expires. To allow the receiver to store the fragments in the right order, the transmitter numbers the fragments, i.e. there is a special field in the layer-N header which contains information about the position of the present fragment with respect to the set of all fragments (for example the 'FragmentOffset' field in the IPv4 header). When the receiver receives all the fragments before the timer expires, it can extract the data out of the fragments, put it together into the right sequence and hand over the received block of *S* bytes of layer N+1-data to its own layer N+1 implementation. When the timer expires at the receiver before it has received all fragments, it will assume that one or more fragments have been lost in the network and will drop the segments it has received to free the buffer.

Beyond this the F+R mechanism needs a further header field in the layer-N header: it can well happen that the layer N+1 at the transmitter hands over large pieces of data in quick succession, and for each of these the layer-N entity at the transmitter would generate several numbered fragments. Because of delays and re-ordering in the underlying network it may well happen that fragments belonging to the two different layer N+1 messages arrive in any random order. To allow the layer N-entity at the receiver to tell apart fragments belonging to different layer N+1 messages, the layer-N header contains a field for identifying the (layer N+1) message as such, for example a sequence number field which the layer N entity increases always after receiving a new layer N+1 message.

Question **16**

Correct

Mark 1.00 out of 1.00

Let *S* be the message size generated by layer N+1, *M* be the maximum packet size for a packet generated by layer N (and handed over to layer N-1), and *O* the size of the per-packet overhead of a layer N packet. All the sizes are given in bytes. Please find a general expression for the number of fragments generated, which involves only the parameters *S*, *M*, and *O* and one of the functions ceil, floor. Please implement this function in Python.

For example:

Test	Result
`print (number_fragments(10000, 100, 1000))`	12

Answer: (penalty regime: 10, 20, ... %)

Reset answer

```
import math
def number_fragments (messageSize_bytes, overheadPerPacket_bytes, maximumNPacketSize_bytes):
    s = messageSize_bytes
    o = overheadPerPacket_bytes
    m = maximumNPacketSize_bytes

    total_packages = math.ceil(s/(m-o)) #rounds a number up to the next largest integer.
    total_O = total_packages * o

    return total_packages
```

	Test	Expected	Got	
✔	`print (number_fragments(10000, 100, 1000))`	12	12	✔
✔	`print (number_fragments(12000, 80, 1200))`	11	11	✔
✔	`print (number_fragments(12000, 200, 1000))`	15	15	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **17**

Correct

Mark 1.00 out of 1.00

Mark whether the following statement is true: all fragments have the same size.

Select one:

- ○ True
- ◉ False ✔

Correct, as the last fragment can be smaller

Correct

Marks for this submission: 1.00/1.00.

Question **18**

Correct

Mark 1.00 out of 1.00

Find a general expression for the size of the last fragment which involves only the parameters *S*, *M*, and *O* and possibly one of the functions ceil, floor. Implement it as a Python function.

For example:

Test	Result
`print (last_fragment_size(10000, 100, 1000))`	200

Answer: (penalty regime: 10, 20, ... %)

Reset answer

```
import math

def last_fragment_size (messageSize_bytes, overheadPerPacket_bytes, maximumNPacketSize_bytes):
    s = messageSize_bytes
    o = overheadPerPacket_bytes
    m = maximumNPacketSize_bytes

    total_fragments = math.floor(s/(m-o)) #rounds a number down to the next largest integer.
    last_packet_size = s - ((total_fragments) * (m-o))

    return last_packet_size + o
```

	Test	Expected	Got	
✔	`print (last_fragment_size(10000, 100, 1000))`	200	200	✔
✔	`print (last_fragment_size(11345, 100, 1000))`	645	645	✔
✔	`print (last_fragment_size(13545, 120, 1500))`	1245	1245	✔
✔	`print (last_fragment_size(17755, 180, 1500))`	775	775	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **19**

Correct

Mark 1.00 out of 1.00

Consider the case of IP packets encapsulated into Ethernet packets. The IP header has a size of 20 bytes (there is no IP trailer), and the maximum IP packet size that can be transferred over Ethernet is 1500 bytes. Now suppose that a higher layer protocol hands over 1500 bytes of data to the IP protocol. How many fragments will be generated?

Select one:

- a. Two ✔
- b. Three
- c. One

Your answer is correct.

Correct

Marks for this submission: 1.00/1.00.

Question **20**

Correct

Mark 1.00 out of 1.00

What will be the size (in bytes) of the last fragment in the previous question? Please give a decimal value.

Answer: 40.0 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **21**

Correct

Mark 1.00 out of 1.00

Similar setup as in the previous question, but now the higher layers hand over a message of 10,000 bytes size. What is the number of generated fragments?

Answer: 7 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **22**

Correct

Mark 1.00 out of 1.00

Similar setup as in previous question, but now the higher layers hand over a message of 10,000 bytes size. What is the size of the last fragment in bytes?

Answer: 1140 ✔

Correct

Marks for this submission: 1.00/1.00.

Question **23**

Correct

Mark 1.00 out of 1.00

And what is the size of the first fragment in bytes?

Answer: 1500 ✔

Correct

Marks for this submission: 1.00/1.00.

◄ Superquiz: Packet processing with Python (Practice copy)

Jump to...

Quiz: Layering and reference models, Basic protocol mechanisms (Practice copy) ►