Dashboard / My courses / COSC264 / Mid-term test 2021 / Mid-term test 2021

Started on	Monday, 20 September 2021, 7:01 PM
State	Finished
Completed on	Monday, 20 September 2021, 8:31 PM
Time taken	1 hour 30 mins

Question **1**
Correct
Marked out of 2.00

Which layer of the OSI reference model is responsible for translating between different representations of a data type?

Select one:
- a. Physical layer
- b. Network layer
- c. Transport layer
- d. Representation layer
- e. Session layer
- f. Application layer
- g. Link layer

Your answer is correct.

Question **2**
Complete
Marked out of 4.00

The IPv4 protocol supports a fragmentation-and-reassembly mechanism. One of the rules is that reassembly is **only** carried out in the final destination and not in an intermediate router. Please justify this rule. Why does it make sense?

Reassembly is only carried out at the destination so it can:
- Allocates buffer large enough for the whole message
- Starts a timer to know when a message has expired (when not all the fragments arrive in time

Reassembly is not carried out in intermediate routers as:
- Then you are wasting space to allocate details (such as headers and flags) in only partial reassembly of the message as fragmentation/reassembly creates significant overhead and complexity to the receiver.
- Won't know when a fragment never arrives in time to deallocate space in the buffer.

Question **3**

Correct

Marked out of 2.00

The IPv4 protocol:

Uses 64-bit wide addresses	False
Does not provide reliable service	True
Guarantees that datagrams are received in the same order as they are sent.	False
Uses acknowledgements	False
Requires an initial connection setup before any data can be sent.	False

Your answer is correct.

Question **4**

Not answered

Marked out of 5.00

The minimum payload has to be no less than 46 bytes in Ethernet. Please explain why Ethernet requires a minimum payload.

Question **5**

Correct

Marked out of 6.00

We consider a fragmentation and reassembly mechanism between a transmitter and receiver. Suppose that the maximum frame size is *F* bytes in total (for example, for Ethernet we would have *F=1,500* bytes), out of which there are *O* bytes overhead. The protocol entity at the transmitter gets a message of *M* bytes length from the higher layers and wants to break these down into fragments. You can assume that the maximum message size is $M_{max} = 65,535$ bytes.

To manage the fragmentation-and-reassembly process, each fragment includes an 'offset' field in its header, which indicates the byte index of the first byte of the fragment payload within the overall message -- the first fragment would contain the value 0 here.

Please write a Python function which takes as parameters the values for the maximum frame size *F*, the per-frame overhead *O*, and the overall message size *M*, and which returns a list containing the (value of the) offset field of the first fragment, the offset field of the second fragment, and so on.

For example:

Test	Result
`print(fragmentOffsets(1500,40,3000)==[0,1460,2920])`	`True`

Answer: (penalty regime: 0, 10, 20, ... %)

Reset answer

```
import math
def fragmentOffsets (fragmentSize_bytes, overheadSize_bytes, messageSize_bytes):
    """Please write a Python function which takes as parameters the values for the m
    F  =  fragmentSize_bytes
    O  =  overheadSize_bytes
    M  =  messageSize_bytes

    result = []

    payload = F - O

    frag_n = math.ceil(M/payload)

    index = 0
    for i in range(0, frag_n):
        result.append(index)
        index += payload

    return result
```

	Test	Expected	Got	
✔	`print(fragmentOffsets(1500,40,3000)==[0,1460,2920])`	`True`	`True`	✔

Passed all tests! ✔

Question **6**

Correct

Marked out of 5.00

Consider a fixed output link of a router. To cope with transient overload situations, the output link has a buffer or queue, so that packets arriving to the output while another packet is being transmitted over the same output do not need to be dropped. The queue is organised as a FIFO (first-in-first-out) queue.

When a new packet arrives, it is checked whether the output link is currently transmitting a packet:

- If not, the transmission of the new packet is started immediately and the waiting time of the packet is zero.
- Otherwise, the packet is sent to the end of the queue. Once the output has finished transmitting the previous packet, it will take the head-of-queue packet out of the queue and start its transmission. The waiting time of the newly arriving packet is the sum of the time required to finish the transmission of the packet currently in service when the new packet arrives, plus the transmission times of all packets ahead in the queue.

All arriving packets have the same size of *L* bits, and the data transmission rate on the outgoing link is *R* bits per second. Write a Python function which calculates the waiting time of a newly arriving packet. This function takes as input:

- The parameters *L* and *R*.
- A flag which tells whether there is another packet currently being transmitted (True) or not (False)
- The number *N* of other packets already stored in the queue (valid only if the flag is True)

You can assume that the arrival time of the new packet is random and that on average about half of the currently transmitted packet has already finished transmission.

For example:

Test	Result
`print(abs(queueingDelay(1000,1000000,True,0)-0.0005)<0.00001)`	True
`print(abs(queueingDelay(1000,1000000,False,0)-0.0000)<0.00001)`	True

Answer: (penalty regime: 0, 10, 20, ... %)

Reset answer

```python
def queueingDelay (packetSize_bits, dataRate_bps, flagCurrentTransmission, numberIn(
    L     =  packetSize_bits
    R     =  dataRate_bps
    flag =  flagCurrentTransmission
    N     =  numberInQueue

    result = 0

    if flag == True:
        result = ((L/R) * N) + ((L/R)/2)
    #else:
        #return N * (L/R)

    return result
```

	Test	Expected	Got	
✔	`print(abs(queueingDelay(1000,1000000,True,0)-0.0005)<0.00001)`	True	True	✔
✔	`print(abs(queueingDelay(1000,1000000,False,0)-0.0000)<0.00001)`	True	True	✔

Passed all tests! ✔

Question **7**

Correct

Marked out of 2.00

Which of the following statements are true about Differential Manchester encoding?

- [x] a. Mid-bit transition is only for clocking.
- [x] b. Transition at start of a bit period represents zero.
- [x] c. No transition at start of a bit period represents one.
- [ ] d. None of the above

Your answer is correct.

Question **8**

Correct

Marked out of 5.00

Write a Python function which takes a 32-bit IPv4 address and converts this into its dotted-decimal string representation.

For example:

Test	Result
`print(IPToString(0x20304050))`	`32.48.64.80`

Answer: (penalty regime: 0, 10, 20, ... %)

Reset answer

```python
# Q8
def IPToString (addr):

    first = (addr & 0xFF000000) >> 24
    sec = (addr & 0x00FF0000) >> 16
    third = (addr & 0x0000FF00) >> 8
    fourth = (addr & 0x000000FF)

    result = str(first) + "." + str(sec) + "." + str(third) + "." + str(fourth)
    return(result)
```

	Test	Expected	Got	
✔	`print(IPToString(0x20304050))`	`32.48.64.80`	`32.48.64.80`	✔

Passed all tests! ✔

Question **9**

Correct

Marked out of 1.00

Which of the following bits are represented by the symbols in the figure (assuming Manchester encoding)?

- a. 01001100101
- b. 10110011100
- c. 01001100111
- d. 10110100011

Your answer is correct.

Question **10**

Complete

Marked out of 4.00

Suppose that you build a simple client-server application with Python (using Socket calls). The server echoes back whatever message you send from the client side. Is it possible for you to implement your client program in a different language (e.g., C) to communicate with the same server program in Python? If possible, why?

Yes.

Everyone uses the same program, no differentiation between client and server as everybody is client and everybody is server.

The server is ready to accept service requests from any client and respond to them

Question **11**

Complete

Marked out of 4.00

People say "IP over everything, everything over IP". How do you understand it?

I understand it as the model in The Hourglass Model for the Internet Protocol Stack where IP sit on top of Ethernet, WLAN, PPP, and SONET, while everything else is on top of IP.

Basic IP service is datagram delivery so we can ensure seamlessly integrated data moving all over the globe for best effort.

Question **12**

Complete

Marked out of 4.00

In the OSI reference model error control is carried out on both the link layer (with a per-hop scope) and on the transport layer (end-to-end scope). Suppose we have a "perfect" transport-layer error control, i.e. all errors are detected and repaired reliably. Please argue why it may still be useful to additionally have link-layer error control. What can go wrong if no link layer transmission is protected by error control?

- The sending side sends segments are divided into multiple packets at the network layer and each packet on multiple frames at the link level.
- Each segment travels the network (divided as frames and packets) and is recomposed only at the receiving side.
- Between the sending and receiving side could be a lot of intermediate routers. During that transit, there could be problems as: One or more frames are discarded, One or more packets get lost, Packets lose their original order. That is what will go wrong if no link layer transmission is protected by error control and useful to have additional link-layer error control

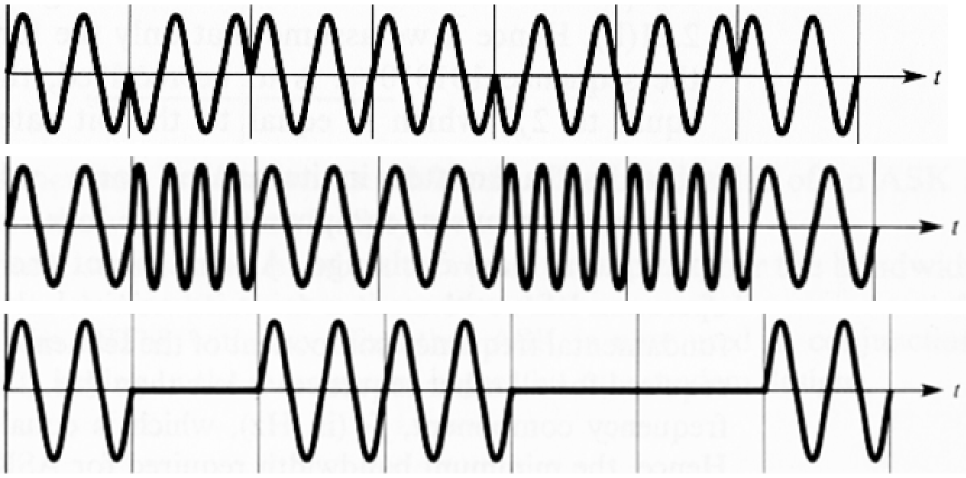
- A malfunctioning router modifies the data in a packet
- These problems will pass undetected through the routers until they get to the error control of the transport layer on the receiving side.

Question **13**

Correct

Marked out of 2.00

Please match the waveforms with its corresponding modulation schemes.



PSK

FSK

ASK

Your answer is correct.

Question **14**

Incorrect

Marked out of 4.00

Suppose we have a system with three stations wishing to transmit data. There are three time slots available, each time slot large enough for a packet transmission. Each of the three stations picks one of the three time slots randomly with uniform distribution.

What is the probability that all three stations pick distinct time slots and hence enjoy a successful transmission without any collision? Please give the result to three digits after the decimal point without any rounding.

Answer: 0.333

Question **15**

Correct

Marked out of 2.00

Please calculate the transmission delay for a packet of length L=1,500 bytes over a link with a data rate of R=5 Mbps. Please give it in seconds.

Answer: 0.0024

Question **16**

Correct

Marked out of 4.00

Below a forwarding table of an IP router is shown. Note that in some cases several entries match the same destination address. In such a case, the most specific entry is chosen, i.e. the entry with the most one-bits in the network mask. Please work out the forwarding decisions made by a router for different destination addresses.

Destination Network / Netmask	**Outgoing interface**
130.2.0.0 / 16	eth0
130.2.10.0 / 24	eth1
177.4.12.0 / 24	eth2
145.6.0.0 / 16	eth0
132.16.12.0 / 24	directly attached
0.0.0.0/0	eth3

130.2.11.10	Forward to eth0
130.2.10.11	Forward to eth1
132.16.12.2	Deliver to directly attached station
130.3.4.5	Forward to eth3
145.6.7.8	Forward to eth0

Your answer is correct.

Question **17**

Complete

Marked out of 4.00

Please explain the operation of the ARP protocol. What is it good for and how does it operate?

Address Resolution Protocol good for:

- ARP determines MAC address for given IP address
- ARP is specified in RFC 826
- ARP is not restricted to Ethernet, but in general is geared towards LANs with broadcast capabilities
- ARP is dynamic:
  - The MAC address for a given IP address does not need to be statically configured, ARP allows to determine this on-the-fly
  - Advantage: nodes can be moved or equipped with new network adapters without any re-configuration

Operates by:

- Each station maintains an ARP Cache, which stores the mappings from IP to MAC addresses that the station currently knows about

Question **18**

Correct

Marked out of 2.00

Which layer of the OSI reference model carries out modulation and demodulation?

Select one:

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

Your answer is correct.

Question **19**

Incorrect

Marked out of 6.00

We consider **packet switching**.

We are given a system with a number of N+2 stations A, R_1, R_2, ..., R_N, B such that end host A is connected to the first router R_1, the first router R_1 is connected to router R_2, and so on, and the last router R_N is connected to the other end host B (i.e. all the stations form a chain). Hence, there are N routers.

Now suppose that A has a message of size M bits which is an integer multiple of the maximum packet user data size of S bits. Station A prepares M/S packets and sends them back-to-back, without any gap. An individual packet has total size $S+O$ bits, where O is the number of overhead bits per packet. All the routers can process incoming packets without a gap: if a router has finished processing one packet (which takes P seconds processing time) and the next packet has been completely received at this time, processing of this next packet can start immediately and we have a kind of "pipelining effect". Note that all routers can process incoming packets and transmit outgoing packets at the same time. All the links in the system can transmit at a data rate of R bits per second, and the propagation delay on each link is T seconds.

Suppose station A starts transmission of the first packet at time 0. Find a general expression for the time by which station B will have received all M/S packets (use the simplification that M/S is an integer and there are hence no slack packets). You do not need to consider any processing times at station B. Implement your expression as a Python function.

For example:

Test	Result
`print(abs(packetSwitching(3, 10000, 1000, 100, 0.001, 1000000, 0.02)-0.0973)<0.0001)`	`True`

Answer: (penalty regime: 0, 10, 20, ... %)

Reset answer

```python
def packetSwitching (numberRouters, messageSize_b, userDataSize_b, overheadSize_b, pr
    N  =  numberRouters
    M  =  messageSize_b
    S  =  userDataSize_b
    O  =  overheadSize_b
    P  =  processingTime_s
    R  =  dataRate_bps
    T  =  propagationDelay_s
```

	Test	Expected	Got	

✖	`print(abs(packetSwitching(3, 10000, 1000, 100, 0.001, 1000000, 0.02)-0.0973)<0.0001)`	`True`	`***Error***`
			`Traceback (most recent call last):`
			`File "__tester__.python3", line 21, in <module>`
			`print(abs(packetSwitching(3, 10000, 1000, 100, 0.001, 1000000, 0.02)-0.0973)<0.0001)`
			`TypeError: unsupported operand type(s) for -: 'NoneType' and 'float'`
				✖

Testing was aborted due to error.
Your code must pass all tests to earn any marks. Try again.

Show differences

Question **20**

Not answered

Marked out of 4.00

Explain the difference between analog and digital transmission and argue in which of these an error-free transmission is possible.

Question **21**

Not answered

Marked out of 3.00

Please explain the operation of TDMA (Time Division Multiple Access).

Question **22**

Not answered

Marked out of 1.00

Which of the following statements are true? (In regarding to a typical communications system.)

- [ ] a. The goal of channel coding is to correct random errors through introducing redundancy.
- [ ] b. It is optimal to separate source coding and channel coding under all conditions.
- [ ] c. The goal of source coding is to reduce redundancy.
- [ ] d. The decoded data at the receiver side after source decoding may be different from the original data before source coding.

Your answer is incorrect.

Question **23**

Not answered

Marked out of 2.00

Suppose that a sender and receiver communicate over datagram sockets. The sender calls write() three times: the first time with 10 bytes of data, the second and third time with 20 bytes of data each. All the data is successfully transferred to the receiver. The receiver will call recvfrom() on this socket and attempts to read 1024 bytes into a buffer. How many bytes will the recvfrom() call return?

Select one:

- a. 5 Bytes
- b. 1023 Bytes
- c. 1024 Bytes
- d. 10 Bytes
- e. 30 Bytes

Your answer is incorrect.

Question **24**

Not answered

Marked out of 1.00

In baseband transmission schemes such as NRZ or on-off keying, we use two voltage levels. In ***M***-ary transmission, we can use multiple voltage levels to represent a group of bits. How many bits does one voltage level represent if we use 16 voltage levels in an ***M***-ary transmission scheme?

- a. 2
- b. 8
- c. 4
- d. 16

Your answer is incorrect.

Question **25**

Not answered

Marked out of 2.00

Please calculate the propagation delay for a signal traveling a distance of 15,000 km, assuming a speed of light of C=200,000 km/s. Please give your answer in seconds.

Answer:

Question **26**

Not answered

Marked out of 4.00

Please explain the operation of the nonpersistent-CSMA protocol. Also discuss the advantages and disadvantages of choosing the probability distribution for the random backoff time such that the expected backoff time is comparatively small related to the size of a packet.

Question **27**

Correct

Marked out of 2.00

Suppose a station has opened a datagram socket using the socket() function and as its next step wants to send some data over this socket. Which function will the station have to use?

Select one:

- a. send()
- b. write()
- c. sendto()

Your answer is correct.

Question **28**

Correct

Marked out of 2.00

Which socket type will you need to use for reliable and in-sequence data transfer of a large block of bytes?

Select one:

- a. Raw socket (SOCK_RAW)
- b. Stream socket (SOCK_STREAM)
- c. Sequenced-packet socket (SOCK_SEQPACKET)
- d. Datagram socket (SOCK_DGRAM)

Your answer is correct.

Question **29**

Not answered

Marked out of 3.00

Please explain the basic operation of a bridge and how bridges learn which station is in which of the coupled local area networks.

Question **30**

Correct

Marked out of 2.00

Which conversion function will you have to use to convert a 32 bit integer value from network representation to host representation?

Select one:

- a. ntohs()
- b. htonl()
- c. ntohl()
- d. htons()
- e. inet_aton()

Your answer is correct.

Question **31**

Not answered

Marked out of 5.00

The maximum packet size of a specific link is called Maximum Transmission Unit (**MTU**). The **path MTU** is defined as the smallest MTU of all links along a path between source and destination. How can a sender determine the path MTU when it wants to send a packet to a destination?

Question **32**

Correct

Marked out of 1.00

How many bits is represented by a QAM-256 symbol?

- a. 8
- b. 256
- c. 4
- d. 6

Your answer is correct.