

Question Group A

[2 + 4 + 2 + 2 marks]

Question Group B

1. Identify the missing step in this code.

```
import RPi.GPIO as GPIO
GPIO.setup(13, GPIO.OUT)
GPIO.output(13, GPIO.HIGH)
```

2. What is TCP sliding window mechanism? How should it be adapted for resource-constrained IoT applications?
3. List any two characteristics of wireless sensor networks which make them different from regular networks.
4. Device X emits a signal of speed q towards device Y at $t=5s$ and receives a copy of it back at $5.3s$. It is known that Y takes $0.1s$ to process the signal before echoing it back. How far away are the two devices?

1. Abdullah uses the following code for GPIO cleanup in a Raspberry Pi program. Identify the problem in his approach.

```
import RPi.GPIO as GPIO
def main():
    # program goes here
    GPIO.cleanup()
```

2. What is the TCP delayed ACK feature, and how does it impact IoT applications?
3. Discuss why security is an important challenge for WSNs.
4. Device X emits two different kinds of signals, at exactly same time, towards device Y, which receives them at $t=6s$ and $6.5s$. How far away are the two devices if the speed of signal travel is 300 and $350m/s$.

A

1

Setting the pin numbering mode – bcm or board numbering

2

By default, TCP sends out N segments immediately, and then starts waiting. N is defined by congestion-window or receive-window, whichever is smaller. Once receives gets a cumulative ACK of m segments ($m < N$), it sends out another m segments. So effectively, N packets are always in-flight (sent, waiting for acknowledgement). Purpose of this mechanism is to increase the throughput as high as possible.

IoT devices do not have enough memory to maintain a window of size N . Also, throughput is not a concern in many IoT applications. Therefore, the sender uses a window size of 1 , which makes it a stop-and-wait methodology.

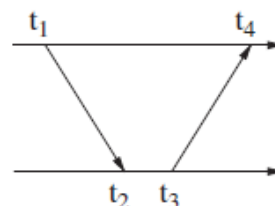
3

Answers will vary, see L1p, slides 6-12.

4

$t_1 = 5, t_4 = 5.3, (t_3 - t_2) = 0.1$

$$\text{distance} = \frac{(5.3 - 5) - 0.1}{2} q = 0.1 q$$



B

1

Problem is that cleanup() call will never run if the program crashes half way though due to an exception.

2

TCP ACKs are not immediately sent, receiver waits 200ms for another segment to arrive so that it can send a cumulative ACK for both. If no other segment arrives within 200ms, the first segment is ACKed. Purpose of this scheme is to halve the number of ACKs sent in a busy TCP connection.

In IoT, most senders operate in stop-and-wait fashion, by not sending the next segment unless previous ACK has arrived. Delayed ACKs will significantly slow down the data rate in such scenario.

3

Sensor networks often monitor critical infrastructure or carry sensitive information, making them desirable targets for attacks.

OR

Conventional security techniques are often not feasible due to their computational, communication, and storage requirements.

4

Refer to figure, the wait time ($t_3 - t_1$) is given to be zero. Distance is simply the difference of speeds multiply difference in arrival time

$$\text{Distance} = (350 - 300) \times (6.5 - 6)$$

