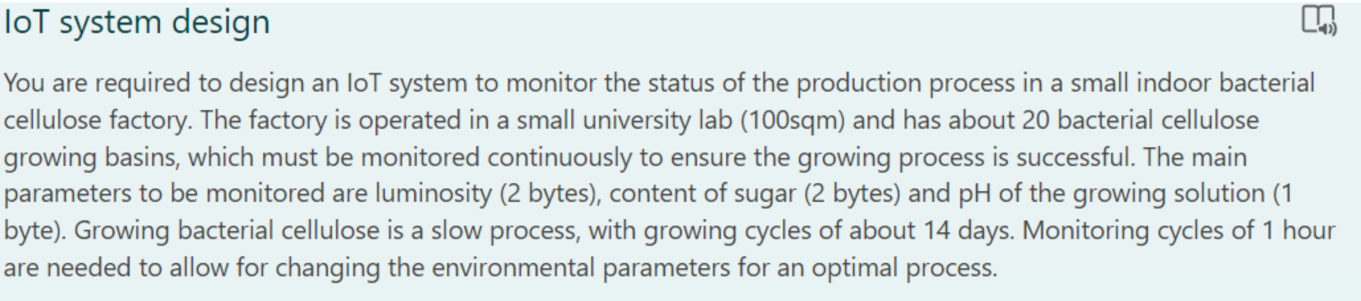


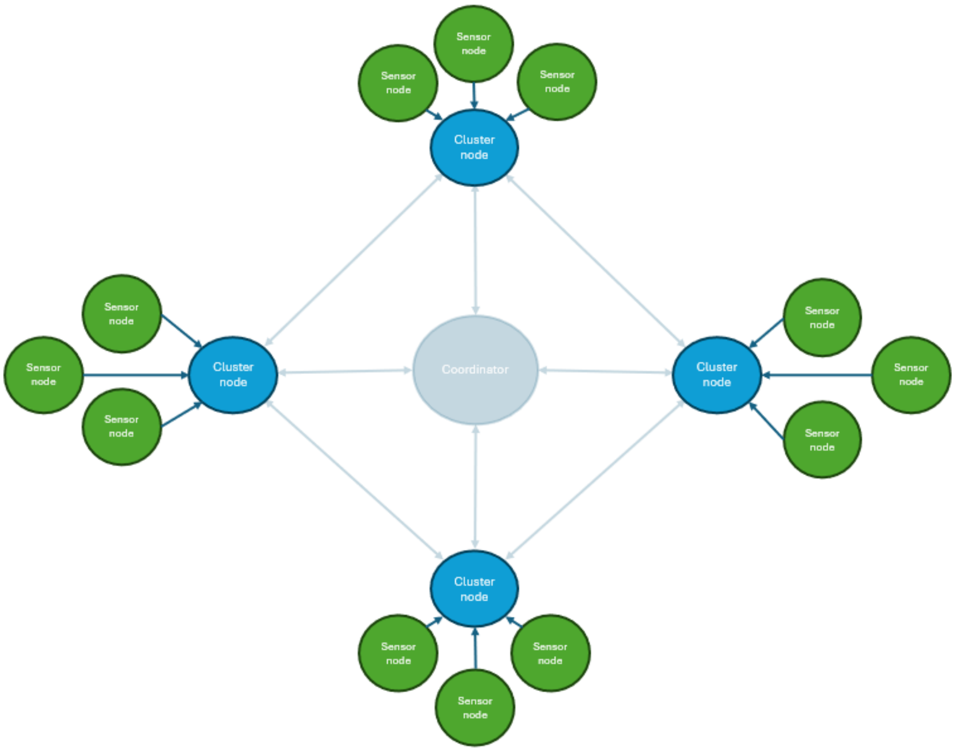
IoT system design

You are required to design an IoT system to monitor the status of the production process in a small indoor bacterial cellulose factory. The factory is operated in a small university lab (100sqm) and has about 20 bacterial cellulose growing basins, which must be monitored continuously to ensure the growing process is successful. The main parameters to be monitored are luminosity (2 bytes), content of sugar (2 bytes) and pH of the growing solution (1 byte). Growing bacterial cellulose is a slow process, with growing cycles of about 14 days. Monitoring cycles of 1 hour are needed to allow for changing the environmental parameters for an optimal process.

Q1 ~ propose an overall design for the system, mainly focusing on the communication technology to be used. Motivate your choice

Measuring the states of each cellulose growing basins is a business-critical activity; guaranty performance, reliability error avoidance and recovery, must be our main goals.
Consequently a battery-powered device is not advised to achieve sufficient reliability, even if it were possible to guarantee an autonomy of 14 days (or more) a non-swift battery replacement can delay required reactions to critical states causing the loss of 1 growing basin.
Moreover, the direct power supply makes avoiding duty cycles (with deep sleep periods) possible, allowing researchers to require additional data asynchronously. Since growing bacterial cellulose, as described, is a slow process, as default each basin will communicate its state once an hour, consequently allowing researchers to request additional data can be appropriate.

Peripheral network (very short range):



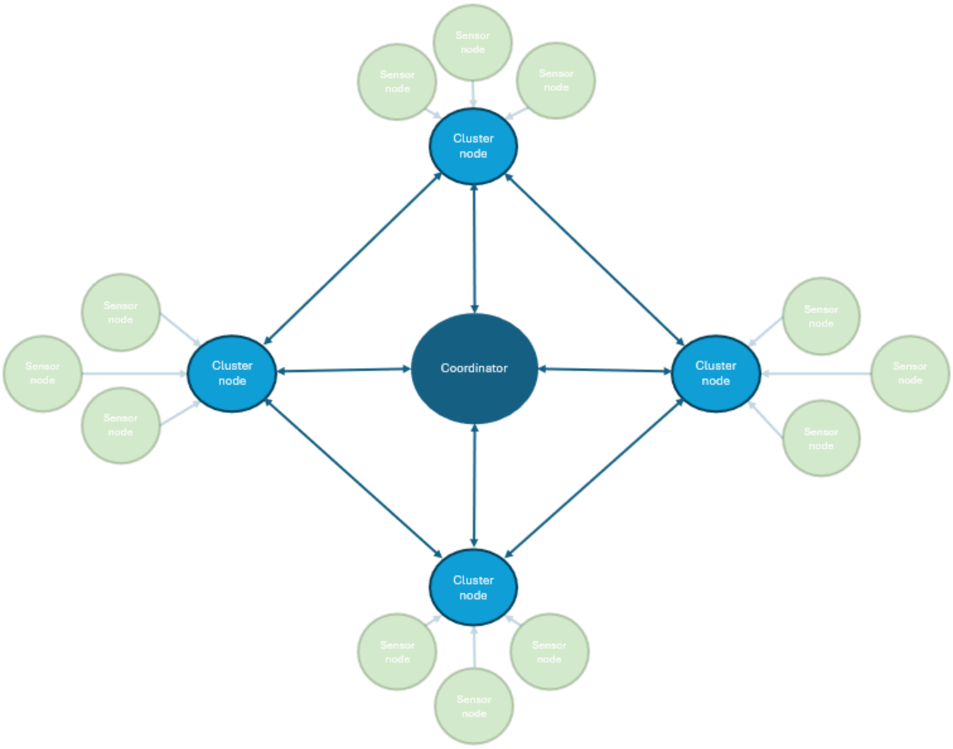
The peripheral network is characterized by a very short distance between actors (sensor and cluster nodes), so it can be easily implemented through wired protocols.
Each basin measure device comprises 3 independent sensor nodes (to enable error avoidance and simplify maintenance) connected to a cluster node through PoE (power over ethernet). the sensor nodes respectively measure the luminosity, sugar, and pH of the growing solution, they are programmed to perform 3 data measurements and send it to the cluster node as soon as they receive power.

The cluster node act as a CoAP server, it's indipendently connected to each senor node throught ethernet connectivity (no possible collisions in this stage, ACK is not required) and it also provide power supply (turning on the device only when it require data). Its role is to aggregate senor data into a single message (each sensor perform multiple reading to mitigate measurements errors, and send one CoAP post request for each reading) and send it to the coordinator.

> **note:** This configuration allows sensor redundancy, if we attach multiple identical sensors to a cluster node, if (after raising power up) the main sensor does not respond (providing data) the cluster will ask data to the backup sensor.

> **note:** In this configuration the cluster node is a single point of failure, it must be highly reliable.

Main network (short range):



Due to the longer distances the main network needs to be wireless, based on Zigbee (frequency-band: 2.4GHz, bit rate: 250kb/s). Zigbee has a theoretical max distance range of 100 m, and so for our implementation is required to use mesh topology to extend the max reachable distance (all the cluster nodes are full-function devices and can contribute to the mesh creation).
As already said the implementation focus is on reliability, so the Zigbee network must run in Beacon Enabled mod to avoid collisions (with CAP time equal to zero), once an hour the coordinator will send a beacon and will receive updated data from each of the 20 cluster nodes; each cluster node will provide all the data in a single message with a fixed data length of 5 Bytes (the whole messages stays in a single packet with the standard size of 128 Bytes), the CFP time can be computed as follows:

$$T_{CFP} = 20 \cdot (\frac{128 \cdot 8}{250 \cdot 10^3}) \simeq 0,082s$$

> **note:** cluster nodes aren't power constricted and can take the Zigbee antenna powered during sleeping phases to receive asynchronous data requests from the coordinator.

> **note:** the coordinator can run Zigbee2MQTT open protocol to achieve cloud independence, publishing each data coming from a cluster nod to its specific MQTT topic.

Q2 ~ write the pseudocode of the firmware that should be run by the monitoring device installed on each basin

Pseudocode for sensor nodes:

```
void setup() {
  for(int i = 0; i<3; i++){
      sensorData = doSensorRead();
      CoAPPost(sensorDat);
  }
}
```

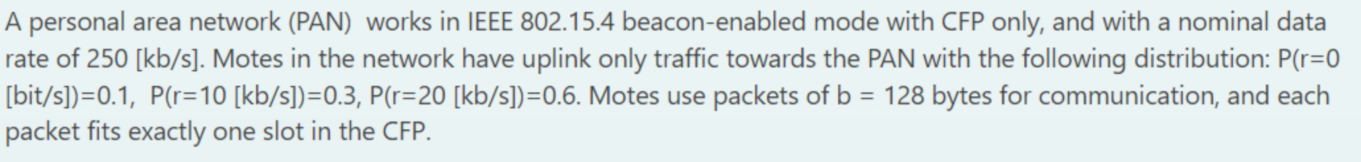
Pseudocode for clusterNode nodes (with redundancy):

```
void onBeaconRecv(beaconMsg){
  aggregatedData = Struct{
      float luminosity;
      float sugar;
      float ph;
  }

  for(sensor in luminositySensorList){
      try{
          aggregatedData.luminosity = average(sensor.getMeasure()); //provide power to the sensor and wait for data returned as a list, raise an exception if the sensor does not respond
          break;
      }catch{
          //communicate sensor failure
      }
  }

  //same for other sensor types

  sendZigbeeMSG(aggregatedData);
}
```

Q3 ~ as an add-on, you are required to install a VGA camera (640x480 pixels, 8 bits per pixel) to monitor the status of the growing process. Is the solution proposed at the previous points still valid? If not, propose an alternative solution.

The peripheral network can be maintained identically, by simply adding the VGA camera as a new sensor node. However the same can't be said for the main network, in fact ZigBee is not designed for high traffic and continuous communication, in this case, WiFi is a better way to go.
Alternativelly we can keep the previous network unchanged plugging the 20 cameras directly to the coordinator with a dedicate network wired (ethernet), or wirless (WiFi).

> **note:** in the second case, particular attention is required to avoid interference, WiFi and Zigbee work on the same frequency-band (2.4 Ghz), channels must be seated properly.



A personal area network (PAN) works in IEEE 802.15.4 beacon-enabled mode with CFP only, and with a nominal data rate of 250 [kb/s]. Motes in the network have uplink only traffic towards the PAN with the following distribution: P(r=0 [bit/s])=0.1, P(r=10 [kb/s])=0.3, P(r=20 [kb/s])=0.6. Motes use packets of b = 128 bytes for communication, and each packet fits exactly one slot in the CFP.

Q4

$$BI = \frac{L}{r} = \frac{128 \cdot 8}{10 \cdot 10^3} = 0,1024s$$

Q5

$$T_{slot} = \frac{L}{R} = \frac{128 \cdot *8}{250 \cdot 10^3} = 4,096 \cdot 10^{-3}s$$

Q6

$$T_{active} = T_{CFP} = 30$$

Q7

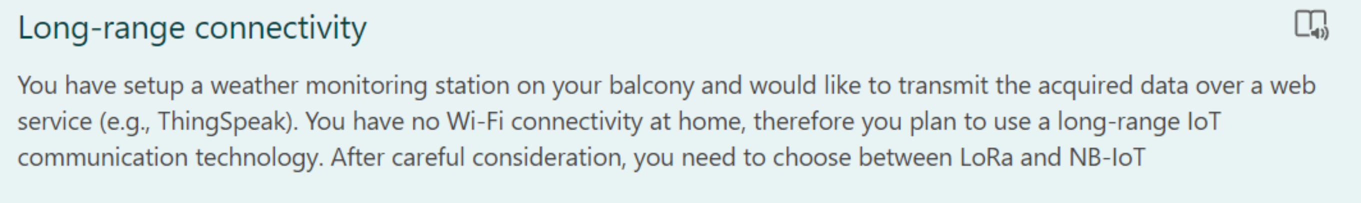
$$N_{active-slots} = N_{slots} \cdot (0,3 + 0,6) = \left(\frac{T_{CFP}}{T_{slot}} - 1\right) \cdot 0,9 = 5,85 \simeq 6$$

Q8

$$N_{active-slots} = N_{slots} \cdot 0,1 = \left(\frac{T_{CFP}}{T_{slot}} - 1\right) \cdot 0,1 = 0,65 \simeq 1$$

Q9

$$N_{slots} = \left(\frac{T_{CFP}}{T_{slot}} - 1\right) = 6,5 \simeq 7$$



Long-range connectivity

You have setup a weather monitoring station on your balcony and would like to transmit the acquired data over a web service (e.g., ThingSpeak). You have no Wi-Fi connectivity at home, therefore you plan to use a long-range IoT communication technology. After careful consideration, you need to choose between LoRa and NB-IoT