Problem solutions 4

Alberto Defendi - alberto.defendi@helsinki.fi

February 14, 2022

1 Solutions

Exercise 1:

Algorithm 1: Monitoring

```
Input: knowNodesList: list of known nodes
Function Discovery(knowNodesList):
    stop = 0;
    while stop ≠ 1 do
        connection = broadcastAdvertise("device-name",
         "message") ;
        if connection.replies() ≠ ∅ then
            connectNodes(nearestNode.getKnownNodesList(),
             connectedNodesList);
        end if
    end while
Function KeepAlive(knownNodesList):
    counter = 0;
    while counter is 5 do
        response = HBRQ(knownNodeList);
        if response = HBRS then
            counter = 0;
        end if
        else
            counter = counter + 1;
        end if
    end while
```

The network has to be delay tolerant, since the animals tend to move and the communication with the infrastructure might be postponed, if the infrastructure is not available.

Exercise 2:

I chose application 3 and 4.

3

1. - Cellular, since the car has an onboard computer that exchanges data of at least 20 MB/s for downloading the maps, and the advantage of a good battery capacity. LTE, 5G or 6G may be used.
   - The car is connected to wireless radios and exchanges data with the cloud.
   - Wide network range since the LTE radios may be distant.
2. I would use HTTPS since the data is exchanged via a cellular connection. This exposes data to other users, and possible attacks such as man-in-the middle.
3. Data can be uploaded to the server while the car is connected to a WiFi station at home. Here the car can download maps of the frequently visited areas, update the police database and to push sensor's data to the server, if it was not done while the app was functioning. This makes possible to save the limited connection rate, and save the battery of the car. It can be considered to evaluate the car's condition offline using the car's computer to reduce the delays with the server, since the sensors re-calibration and the issues are operating in real-time. This doesn't increase the production costs excessively, since the on-board computer has enough computational power, if it's able to show maps.

4

- For the end devices we use ZigBee, since it has a wide range, and sufficient data-rate. For the gateway and the server, we use a cellular connection, since the amount of data can reach a rate of some MB/s.
- Mesh network for end devices and client-server for the gateway and cloud.
- The range should be between 10-100m, since the sensor can be located distantly.

1. Zigbee for the edge devices and HTTPS for the gateway-server.

2. The edges must offload data the earliest as possible, since the sensor's control units are low-end devices, with power, memory, and energy constraints. Consequently, we can't do any data analysis on edges. Furthermore, we have the advantage that the IoT can be connected to a high-speed connection with the cloud. The gateway could reorganize the data, labeling it with each node name before sending it.

Exercise 3:
We chose article [MZT18].

(i) BLE because the device is closer than 10m to the gateway and has low power consumption.

(ii) For edge-fog is a PAN, and an area of 10m suffices. The smartphone-cloud are at distance of several Km.

(iii) From fig. 6, W-Air consist in a sensor that communicates with a smartphone using Bluetooth. The smartphone exchanges data with the cloud using wireless or LAN connection. In a simplified way, the sensor collects the data from the sensors, the smartphone performs some calibrations on the data, and the cloud is responsible for the data analysis. This model could be viewed as a FC architecture, where the sensor is the edge, the smartphone as the fog, and the server as the cloud.

(iv) COAP because the device has low memory, it needs to continuously exchange the measurements with a smartphone. It fits here because the sensor should only report data to the smartphone in a publish/subscribe way (where the smartphone is the central entity), and a connection-less protocol like UDP works since loosing some measurements may not be problematic. For the smartphone and the server, HTTPS can be used with request/response.

References

[MZT18] Balz Maag, Zimu Zhou, and Lothar Thiele. "W-Air: Enabling Personal Air Pollution Monitoring on Wearables". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.1 (Mar. 2018). DOI: 10.1145/3191756. URL: https://doi.org/10.1145/3191756.