Mini-Project #1

Note: Min-Project #1 Idea is built on original idea of Prof. Dr. Oliver Hahm (oliver.hahm@fb2.fra-uas.de) Frankfurt University of Applied Sciences — Faculty 2: Computer Science and Engineering with his explicit permission.

Context:

The goal of the Internet of Things is to effortlessly incorporate billions of *Smart Objects* into existing Internet systems. Smart Objects came into existence with the advent of small, affordable computers equipped with energy-efficient micro-controllers, low-power radio transceivers, sensors, and actuators that interact with the physical world, often relying on batteries for power. These systems facilitate a link between the physical and digital realms. To analyze collected data, operate actuators, or handle devices, a cloud-based backend is commonly employed.

Procedure:

Throughout the project, participants will collaborate in teams to create a software solution enabling the secure and energy-efficient transmission of sensor data from Smart Objects to a cloud backend. Additionally, the collected data will be visualized through the cloud. To accomplish this objective, participants are tasked with developing firmware for Smart Objects using the open source operating system RIOT. The software will undergo evaluation in an IoT testbed (FIT-IoT Testbed), operating on real IoT hardware and an IPv6 connection. For collaboration and version control, teams will employ git.

Technologies

Please use the following tools, protocols, and programming languages:

- C/C++/RUST (ANSI C is recommended because RIOT is primarily written in this)
- RIOT (https://www.riot-os.org)
- FIT IoT-Lab (https://www.iot-lab.info)
- Protocols of the TCP/IP-Suite
- AWS IoT Cloud/Google Cloud IoT Core/Microsoft® Azure IoT (You can use any of these three cloud solutions, or you can set up your own at CSC Cloud Cloud services for Research - Services for Research - CSC Company Site)
    - Please use credits wisely so that you can finish the project within free credits granted by AWS/Google/Azure.
    - If you set your own cloud on CSC, then you do not need to worry about credit limits. You can login to the CSC cloud with your University of Oulu email.

Essentially, in this mini-project, you need to:

- Create a firmware based on RIOT (https://riot-os.org).

- The firmware should periodically read sensor data and send it towards an IoT Cloud provider.
- You need to program nodes on FIT-IoT Testbed with your firmware.
- Sending data to the cloud requires
  - a border router
  - multi-hop forwarding towards the border router
- Create a driver for an emulated sensor

Your total accumulated points from mini-project #1 will be based on how efficient your execution (end-to-end IoT pipeline) is:

- How efficient is your sensing layer?
- How efficient is your networking layer?
- How resilient is your design in case of data loss during packet collisions and other events in the wireless network?
- How efficient is your data management layer?
- How reliable is your data? Can you verify the ground truth, in case calibration points of sensors change over the time?

To get better understanding about factors which may affect design of your end-to-end IoT pipeline, you may try to solve the following two questions (assignments from the last year), however you are not expected to submit the answer of these questions during this year:

1. Consider an air quality sensor that integrates a MOX gas sensor, a LSP particulate matter sensor, and environmental sensors measuring temperature and relative humidity. The MOX sensor requires heating to take measurements. Heating the sensor from sleep state requires 5 seconds, and it takes up to 10 seconds for the sensor to cool down. Both the values of the gas sensors and the values of the particulate matter sensor are corrected using a calibration pipeline that uses external humidity and temperature as input.s, during which heat is released into the sensor.
   a. What would be the optimal duty cycling period for these sensors and why?

Note: Higher duty cycle-> Higher energy consumption and IoT nodes have energy constraints.

2. Consider the task mentioned above, wherein an air quality sensor that integrates a MOX gas sensor, an LSP particulate matter sensor, and environmental sensors measuring temperature and relative humidity. The MOX sensor requires heating to take measurements. Heating the sensor from sleep state requires 5 seconds, and it takes up to 10 seconds for the sensor to cool down. Both the values of the gas sensors and the values of the particulate matter sensor are corrected using a calibration pipeline that uses external humidity and temperature as input. Correcting the measurements requires operating a microcontroller for 1 second period, during which heat is released into the sensor. Consider also that the number of bits transmitted per second is 2 bits for heating, 1 bit for cooling and 5 bits for correcting the measurements. What would be the optimal bandwidth for this application and why? Assume that in the communication channel, 1% of packets collide and need to be retransmitted during the night, and 3% the rest of the time. What is the throughput of the channel?