IoT / WI joint projects

This document describes projects that students attending both the Internet of Things (IoT) and Wireless Internet (WI) courses can choose to implement.

Main rules:
- Projects can be taken individually or in groups of maximum 2 students.
- Projects are evaluated a maximum of 4 points for both the IoT and WI courses. That is, with a single project you get a bonus of 4 points for the IoT exam and a bonus of 4 points for the WI exam.
- The evaluation takes into account the quality of the solution, cleanliness of the written code and technical depth of the final report.
- The hard deadline for selecting projects is 19/05/2022. After this deadline, students who did not choose a joint project CANNOT take it.
- The hard deadline for joint project delivery is 01/09/2022. After this deadline, no bonus points will be given.

Project 1: Finding Dory

Dory, is a regal blue tang that last year joined Marlin the clownfish looking for Nemo. One year after Dory and Marlin rescue Nemo, Dory lives with them at the Great Barrier Reef. But one day, due to her memory loss problems, she gets lost. Luckily, Marlin and Nemo have already prepared a system to find the forgetful Dory. They placed 6 special anchors in the four corners of the Great Barrier Reef and installed a hacked operative system on Dory's phone. Marlin and Nemo mapped all of the Great Barrier Reef with the anchors, creating a fingerprint dataset composed of the Received Signal Strength Indicator (RSSI), a smartphone in some locations of the Great Barrier Reef.

Now, they only need to capture the probe requests coming from Dory's smartphone and map them into her fingerprint dataset. For this task, they need you. Would you help Nemo and Marlin in this adventure?

In this project, you are requested to find the coordinates (X,Y) of the position of Dory using the dataset of RSSI values from the six sniffer-anchors. The dataset is, however, hidden and fragmented inside MQTT publishes/subscriptions and CoAP requests packets. Every correct MQTT and CoAP request will give you a piece of the dataset, and, in the end, you will be able to reconstruct it and eventually find Dory's coordinates. The dataset is a fingerprint dataset, and for each (X,Y) coordinate, it contains an array of RSSI values corresponding to the values captured by the anchors in that position. Once found the RSSIs emitted by Dory's smartphone, you have to determine his location by comparing the RSSI measurement with the database's entries.

To help you, you should implement the following points:

1. Find <u>all</u> the fragments of the dataset. The more pieces you find, the easier it will be to find Dory's positions.
2. Those fragments are hidden inside:
    a. The payload of the publications of MQTT topics
    b. CoAP GET/POST/PUT/DELETE resources response
    c. CoAP resource observe
    d. HTTP resources

    ??

3. Reconstruct the dataset and find Dory
    a. If needed, filter, clean, remove outliers, etc…
    b. Compare Dory's RSSI with the ones in the dataset
    c. Obtain the (X,Y) position of Dory

<u>Useful resources:</u>

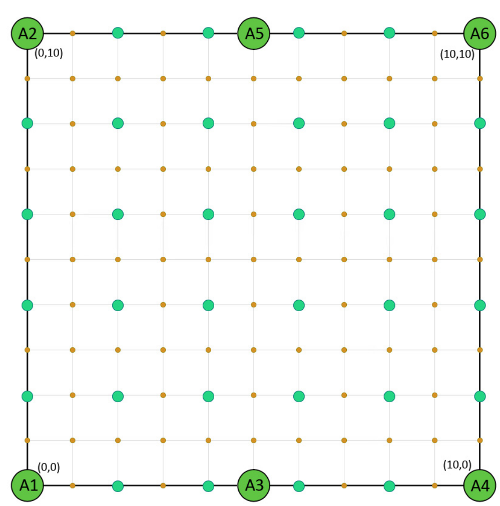
- MQTT and CoAP server address: **131.175.120.117**
  <u>Important! https://www.ict.help.polimi.it/network-vpn-global-protect/?lang=en install and use the PoliMi VPN for getting resources from the server</u>

- Anchors' coordinates:
    - A1: (0,0)
    - A2: (0, 10)
    - A3: (5, 0)
    - A4: (10, 0)
    - A5: (5,10)
    - A6: (10,10)

- An example of a frame is (each position has 5 measurements):

X,Y Position	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**
(6,4)	[-83, -43, -67, -57, -98]	[-83, -43, -67, -57, -98]	[-83, -43, -67, -57, -98]	[-83, -43, -67, -57, -98]	[-83, -43, -67, -57, -98]	[-83, -43, -67, -57, -98]

- In the dataset, you can find small green dots  (only even positions). However, Dory can be both in even and odd positions.



You are free to use your favourites tools (i.e. the ones seen during laboratories), your favourite programming language, editor, etc.

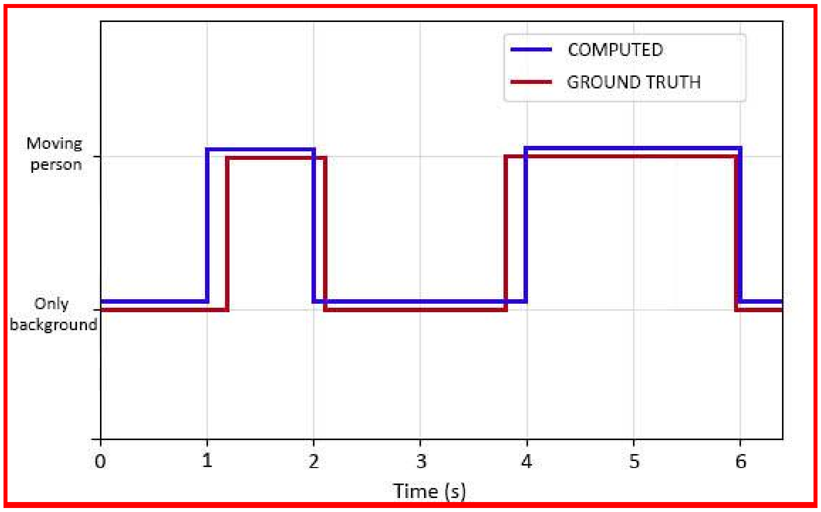
<u>Final report</u>
The final report must include the code, the fingerprint dataset, a report that explains the approach and, of course, Dory's coordinates.

Project 3: Spy your mate

Have you ever thought about what happened in the background during a webcam call or in your smart surveillance camera? Can the network traffic reveal some useful information even if encrypted?

In this project, you are requested to analyse the traffic generated by a webcam call and, from such traffic, identify whether a person or only the background is recorded in the video. You have to approach this problem from a Machine Learning (ML) perspective.
To help you, you may follow the following approach:

1) Start a video call with your project's mate with your favourite program: skype, teams, zoom, etc. If you're doing the project alone, you can call yourself with two different devices.
2) Capture the network traffic with Wireshark and filtering all the video call packets (watch out for inbound and outbound video traffic)
3) During the call, have frames with the person in the camera and frames with only the background. Be extra careful in this process: here, you create your ground-truth labels (the real information provided by direct observation). Stay at least 1 minute in each position, and then switch it.
4) In order to record the ground truth events, you have to use Node-RED. Create a simple dashboard that allows you to store the ground-truth value (e.g. 1 a for person, 0 for no-person). Save everything in a csv file for further ML analysis.
5) Import your data (Wireshark capture + ground truth) in your favourite ML environment (pure Python, R, Matlab, Jupyter notebook, etc). It may be useful to convert your Wireshark file into a csv.
6) Play with some ML recognizer of your choice (at least two). Your final goal is to identify for each time interval whether the person was in the video or not.
7) Report your inferences in a Node-RED chart as in Figure 1 (down here).



Machine Learning tips and tricks:

- Aggregate wireshark packets with a time window of T seconds (T is of your choice and may be 100ms < T < 900ms). Be consistent with the time interval used in the ground truth process!
- For each time window, some interesting features to extract might be:
    - The interval between packets in ms (you can even split it in inbound and outbound)
    - Packet number (same as above)
    - Packet size (same as above)
    - Inbound/outbound rate
    - Etc.
- For each feature, you can extract average, standard deviation, median, etc..
- Eventually, you'll end up with a dataset like this (but with more features!):

Time Interval	Avg pkts interval	Inbound pkts	Outbound pkts	Avg pkt size	Person
0ms-500ms	0.384 s	37	3748	23237 byte	0
500ms-1min	0.5655 s	1239	9837	8232827 byte	1

Final report

The final report must include the code, the node-red export, the input dataframe (Wireshark capture + ground truth), discussion of the results and the final graph done with Node-RED. In the discussion, you are strongly encouraged to use graphics to explain your results (e.g. confusion matrix, learning curve, etc.).