Project Report

Riccardo Mencucci Francesco Pallotta

June 30, 2022

Searching the Dataset

MQTT

We used python with the paho mqtt library to contact the server and retrieve data. The script is located at `mqtt/capture.py` relative to the repository directory. The script first establishes a connection to the server, then subscribes to all topics with the `#` metacharacter and the listen for exactly 30 minutes. The result of the capture is then saved to the `data.json` file.

We then used the `data-processing.ipynb` notebook to clean and filter the captured values, resulting in 13 valid unique observations saved in the `mqtt_coords.txt` file.

Then we extracted manually other observation which we used to interact with Coap, namely those with the following topic and payload:

Topic	Payload
`coap/post/mixed/`	`?problem=memory`
`coap/post/mixed/`	`go to the Doctor of the BarrierReef`
`coap/lies`	`resources can be hidden, find all of them and you'll get a treasure`
`coap/hidden`	`find the HiddenTreasure in the BarrierReef`
`coap/resource`	`/root/BarrierReef/FishLocator?user=Dory`
`anemone/in/the/barrier/reef`	`/root/BarrierReef/Anemone?owner=Marlin`
`great/barrier/reef/with/post`	`/root/PostMe6?search=entry_post`
`other/coap/resource`	`/root/BarrierReef/Apps?fingerprint=True`
`other/coap/resource`	`&gps=False`
`other/coap/resource`	`wait for this A LOT!`

COAP

To retrieve COAP data we used mainly two tools: Firefox with Copper running on the course virtual machine, and tzolov/coap-shell. With the latter we executed a `discover` request which exposed several resources listed in the `coap/resources.txt` file relative to the repository main directory.

Then for every resource we performed a request with every possible method type, with special care for resources which needed an `observe` request.