

# Computing in Communication Networks

Assignment #4: Vehicular Scenario  
Vu Anh Minh Le

June 11, 2022

## 1 Running with docker

### 1.1 Running the server

Pull the docker server by the command: **docker pull minhval0307/unitn\_gpstracking\_server:latest**  
Then run the command: **docker run -it --rm -d -p 8080:80 --name web minhval0307/unitn\_gpstracking\_server**  
On a web browser, access: <http://localhost:8080/>

To stop the container, run: **docker stop web**

### 1.2 Running the clients

There are two kinds of clients, say CARS and DRONES.

To run CARS client, run the following commands:

**docker pull minhval0307/publisher\_cars:latest**  
**docker run -it --rm --name cars minhval0307/publisher\_cars**

To run DRONES client, run the following commands:

**docker pull minhval0307/publisher\_drones:latest**  
**docker run -it --rm --name drones minhval0307/publisher\_drones**

To stop CARS and DRONES clients, run:

**docker stop cars drones**

## 2 Running directly with application files

Access <https://github.com/minhval/ComputingInComNet.git> to download the application source for both server and clients.

Inside folder server/src, run **index.html** (by double clicking on it) to start the tracking page.

Inside folder clients/cars, run command **python3 publisher\_cars.py** to start CARS client.

Inside folder clients/drones, run command **python3 publisher\_drones.py** to start DRONES client.

Note: packages **numpy** and **paho-mqtt** must be installed before starting the clients.

## 3 Images on the run

Here are several images presenting the result after running the server and the clients.

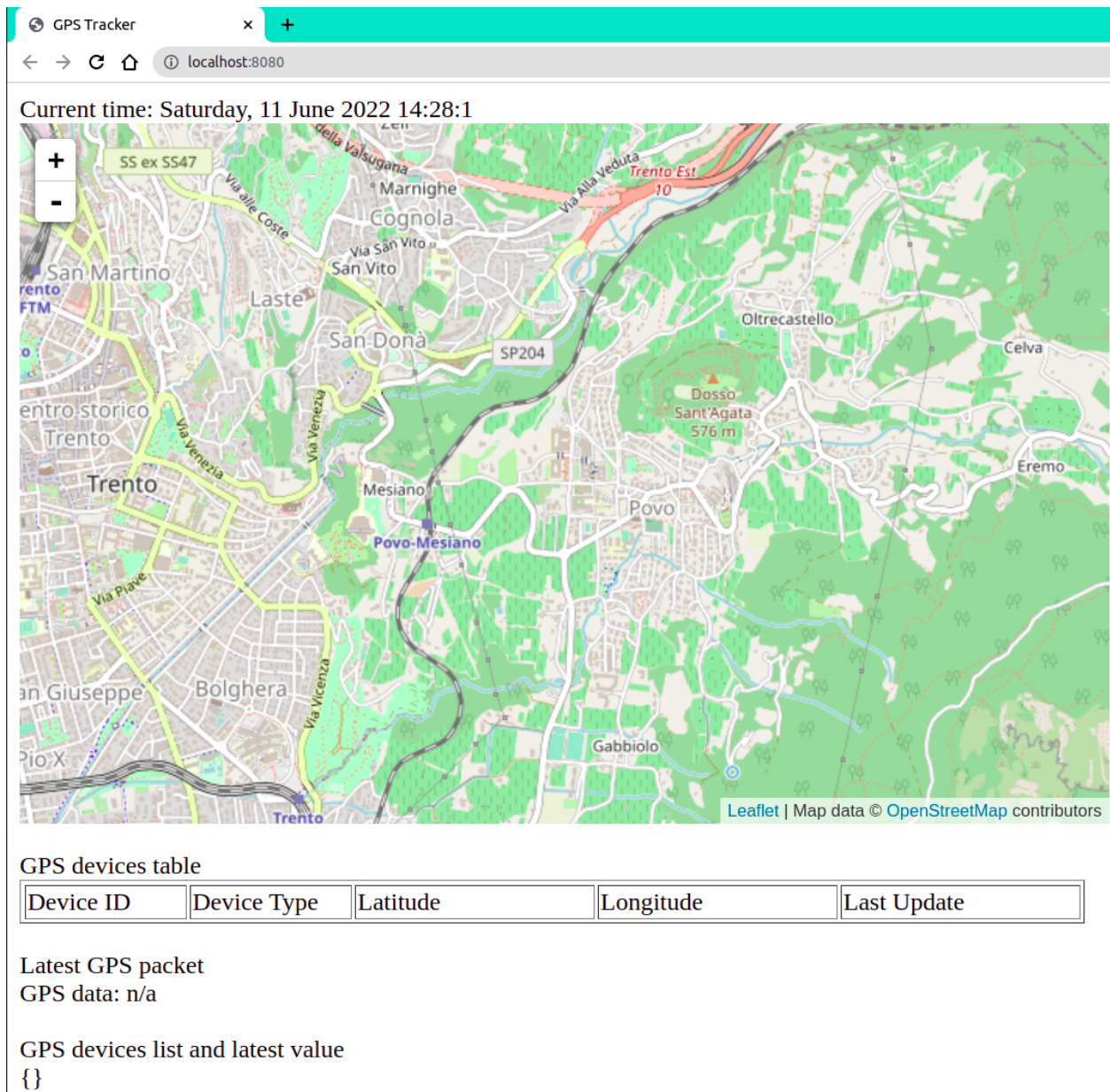


Figure 1: *localhost:8080* with no data received.

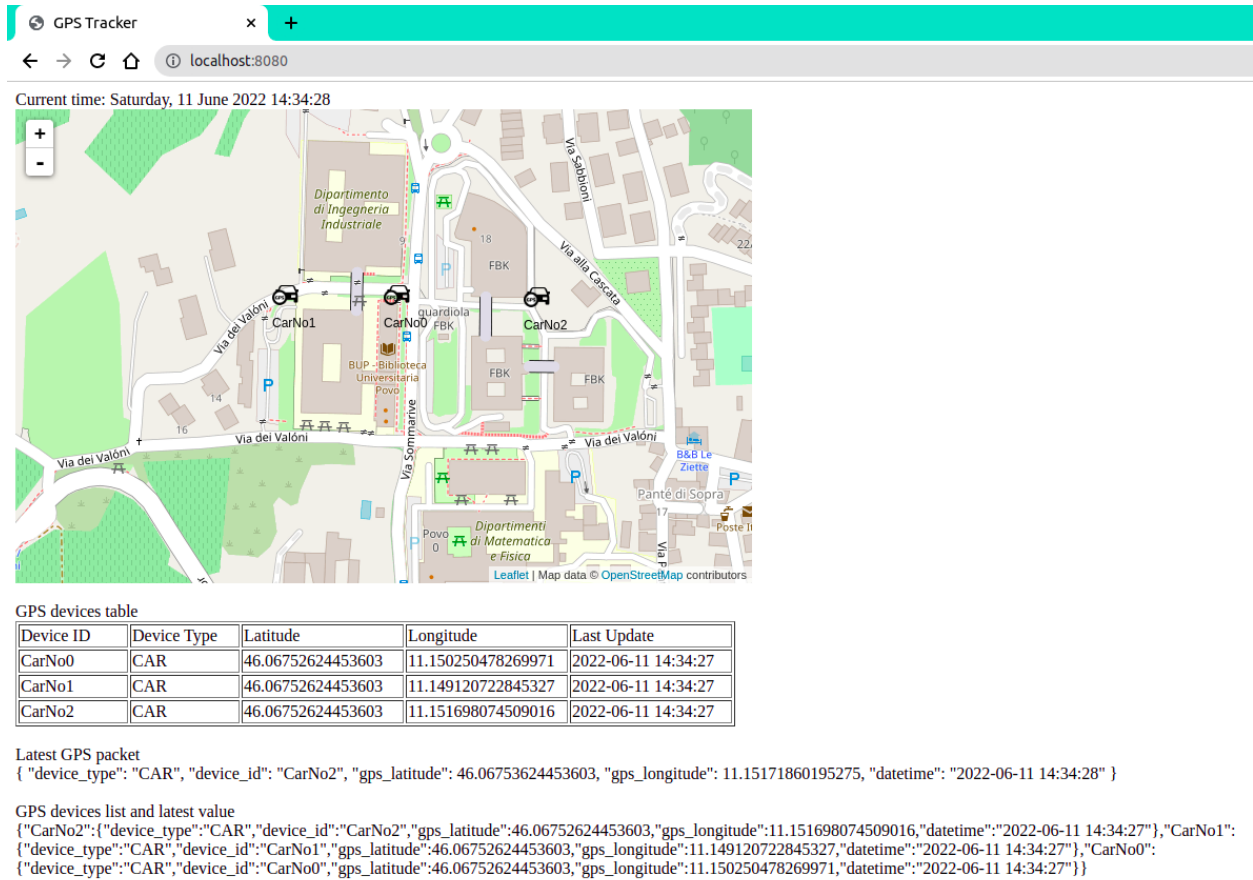


Figure 2: *localhost:8080* with CARS data after a while.

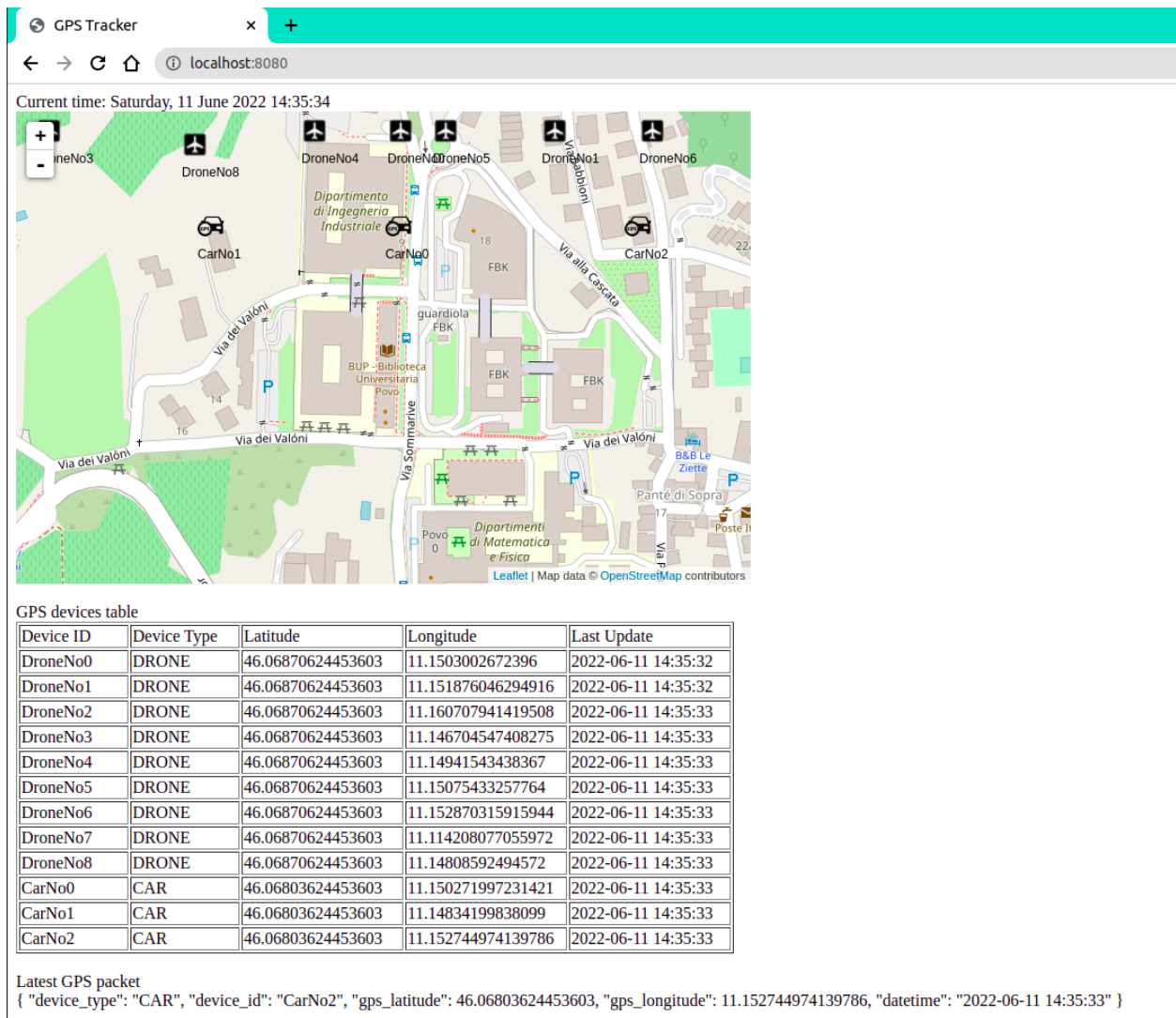


Figure 3: *localhost:8080* with CARS and DRONES data after a while.