SAPIENZA
UNIVERSITÀ DI ROMA

Autonomous Networking a.y. 22-23

Homework 2: Assignment

Professor: Gaia Maselli - gaia.maselli@uniroma1.it
TA: Giulio Attenni - giulio.attenni@uniroma1.it
TA: Flavio Giorgi - flavio.giorgi@uniroma1.it

December 7, 2022

1 Introduction

Homework 2 will be about state of the art Q-Learning routing protocols in networks of drones. In this section we are going through different contingencies, in Section 2 instead we give the problem formulation and finally in Section 3 we give technical implementation details.

1.1 To do

The homework consists in the following parts:

- Implementation of a Q-Learning based routing protocol chosen among a lists of state of the art protocols
- Simulation campaign to get data
- Data plots
- PDF Report

The first task is to properly read and understand the paper describing the protocol. Then you have to implement and test the routing protocol on the simulator. Finally, you should launch a proper simulation campaign (see Section 3.3), testing also geographic, random and your homework 1 q-learning solution (if you have more than one use the best). Once you get all the data you have to elaborate on them and do plots about the metrics of interest (see Section 3.4). As in homework 1, you have to write a report where you describe what you have done and all the results you got. To write the report use **Overleaf**. You have to download the file *template.zip*, upload it into Overleaf editor and use it to write your report. Please, do not modify the template structure! Once you have done the report you can export the manuscript as PDF.

1.2 Homework Delivery

To deliver the homework one of the participants of the group have to put together in a zip file named HW2_GROUPNAME.zip with: (1) The file *q_learning_routing.py* (2) the pdf report (3) other .py files, if there are any, containing other solutions. You can upload the zip file on google classroom in two different dates:

- 15/01/2023
- 05/02/2023

The discussion about both the homework will be held approximately 1 week after the delivery deadline.

You have to upload the zip file on Google classroom before the deadlines namely **15/01/2023 at 23:59** and **05/02/2023 at 23:59**. **Be Aware: the deadlines will not be extended for any reason!**

1.3 Evaluation

In homework 2 (as in homework 1) we won't take too much into account absolute results. It is not important if the protocol performs worse than other solutions (apart from random baseline). We are going to evaluate mainly code and report.

2 Problem Formulation

In homework 2 you have to reproduce a state-of-the-art routing protocol. Since each protocol has different peculiarities you can modify the simulator to implement characteristic protocol features. The setting is always the same: there are many drones in the area of interest that are following pre-defined tours and sometimes generate packets. Each change within the simulator MUST BE feasible in real scenarios. If you have doubts please send an email to us. You can (and should) use the simulator you used to do homework 1. To evaluate your protocol we provide some useful **metrics of interest**:

- mean number of relays
- packet mean delivery time
- packet mean delivery ratio

It is desirable for a good protocol to maximize the number of packets delivered to the depot on time while minimizing delivery time.

3 Implementation Details

3.1 How to implement the ql-base routing protocol

In order write the ql-based routing protocol you have to implement the method *relay_selection* in the *QLearningRouting* class that you can find in *src.routing_algorithms.q_learning_routing.py*. To select

```
for nd in "5" "10" "15" "20" "25" "30";
do
    for alg in "GEO" "RND" "QL";
    do
        echo "run: ${alg} - ndrones ${nd} "
        python -m src.experiments.experiment_ndrones -nd ${nd} -i_s 0 -e_s 1 -alg ${alg} &
        python -m src.experiments.experiment_ndrones -nd ${nd} -i_s 1 -e_s 2 -alg ${alg} &
        python -m src.experiments.experiment_ndrones -nd ${nd} -i_s 2 -e_s 3 -alg ${alg} &
    done;
done;
wait
```

Figure 1: bash script content

that particular routing algorithm during the simulation you can go in *src.utilities.config.py* and set ROUTING_ALGORITHM = RoutingAlgorithm.QL.

3.2 How to implement more than a single solution

If you want to add more solutions you can create other routing protocols classes. To do that create a new file in *src.routing_algorithms*, and name it as you want. Then within that file you have to write a new class, for instance *NewMethodRouting*, and you must implement the relay_selection method. **IMPORTANT: if you want to implement a RL-based protocol you need to include the feedback method, you can use as a reference the feedback function in QLearningRouting**. Finally you need to add the class into the src.utilities.config.py file, importing the class and adding to the RoutingAlgorithm(enum) class an entry, for example NMR = NewMethodRouting. Finally you can select the new protocol using ROUTING _ALGORITHM = RoutingAlgorithm.NMR

3.3 How to run multiple simulations

Homework 1 asks you to make an experimental study about the routing protocols you implemented. In order to do that you must conduct a proper simulation campaign. In src.experiments there is a bash script called run_ndrones.sh that allows you to launch multiple simulation. In Figure 1 you can see the script content. The first for loop iterates over the number of drones used during the simulation. The inner for loop instead iterates all over the algorithms so **if you add other solutions remember to add them using the same acronyms you used in the src.utilities.config.py file**. Finally you can see the options -i_s and -e_s, that are respectively the starting and the ending simulation seeds. All of you should use the same seeds $\in [0, 30]$. Anyway, you can modify the seeds ranges because launch too many simulation at the same time will deplete all your PC resources. Thus you can choose to decrease the ranges for example -i_s = 0, -e_s = 5 and run all 30 seeds in different sessions. Results from the simulation are saved in JSON files under the data.evaluation_tests folder.

3.4 How to plot data

Once you have done all the simulation you must aggregate all the data from all the JSON files and get averages and stds values for each one of the metric of interest. You can do that implementing the method *compute_data_avg_std* in *src.plots.data.data_elaboration.py*. You can use any external library such as Pandas or Numpy to do that. Once you have properly implemented that method you have to implement a proper plotting script. In *src.plots.plot_data.py* there is an example that you can use as a guideline. You can make data elaboration through the *plot_data.py* script (recommended) or you can make data elaboration, save the results, and then make the plots: it is up to you. **NOTE:** the independent variable in all the plots have to be the number of drones.

3.5 How can I run bash script on Windows?

Go to start, and type *windows power shell* and open it. Then using cd command place the working directory in /your/path/to/DroNETworkSimulator and run .\src\experiments\run_ndrones.sh