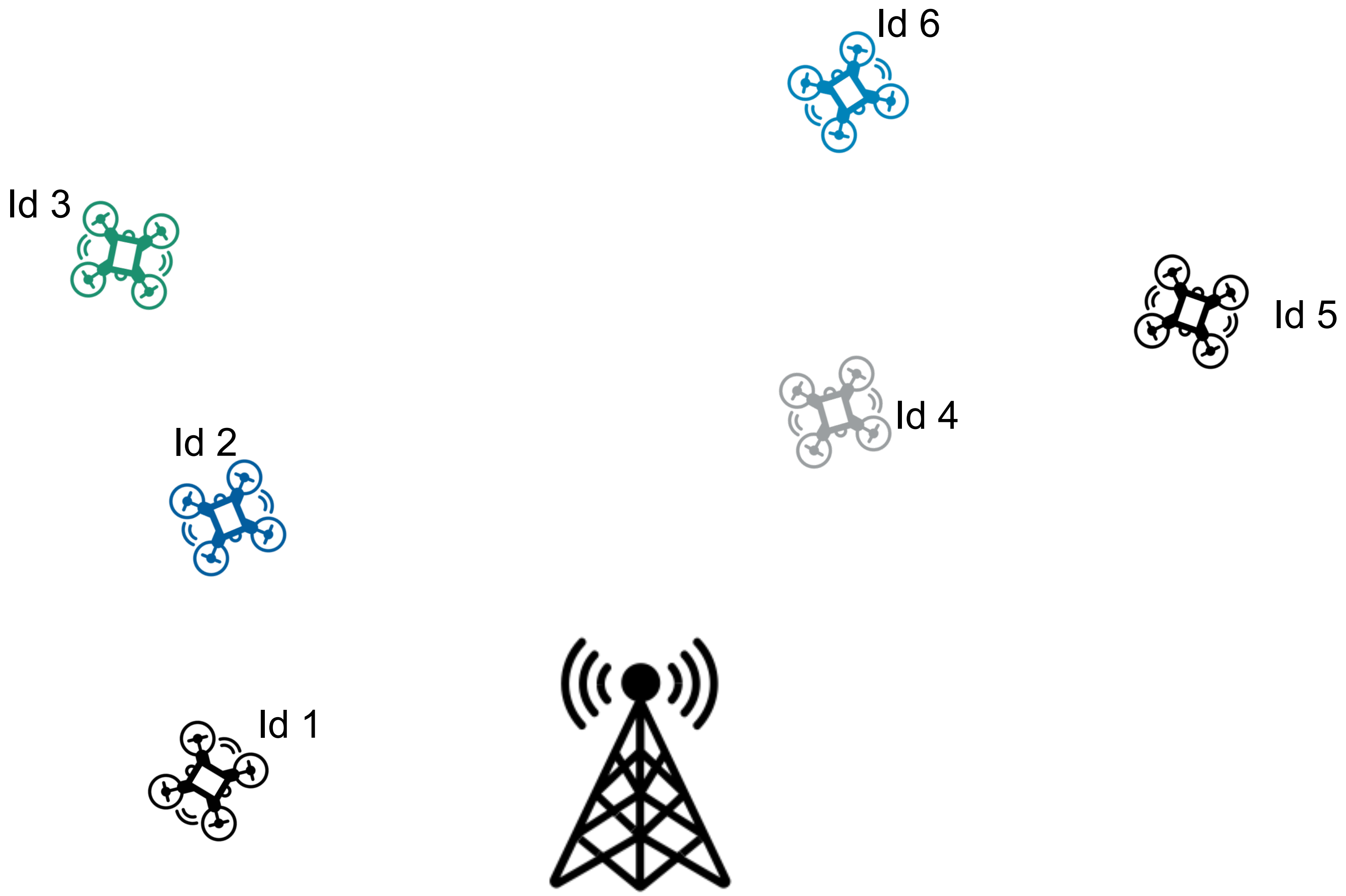
SAPIENZA
UNIVERSITÀ DI ROMA

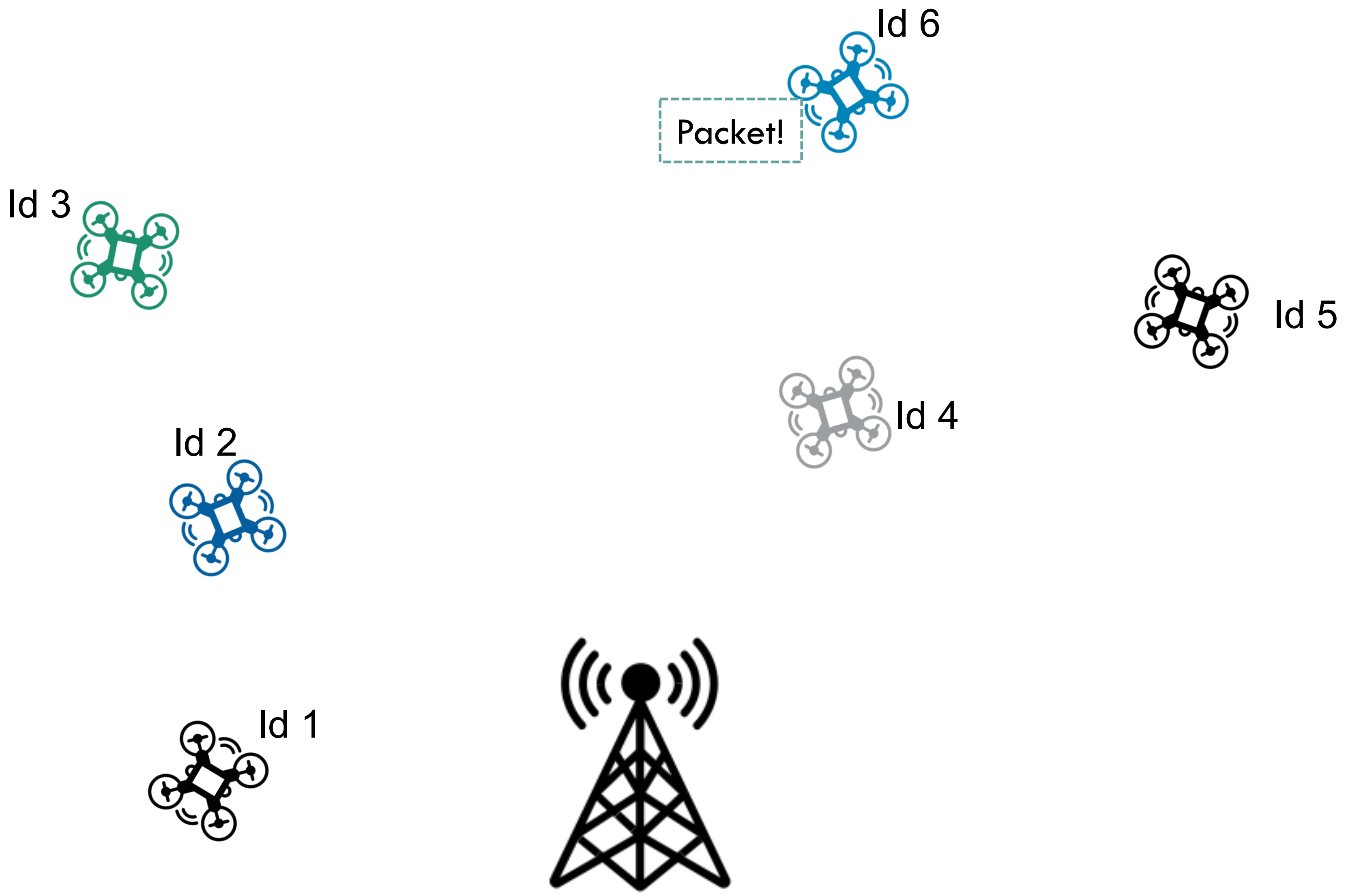
DroNET Simulator

Course: *Autonomous Networking - Prof. Gaia Maselli (A.A. 2021-2022)*
Speaker: *Dr. Andrea Coletta - 03-11-2021*

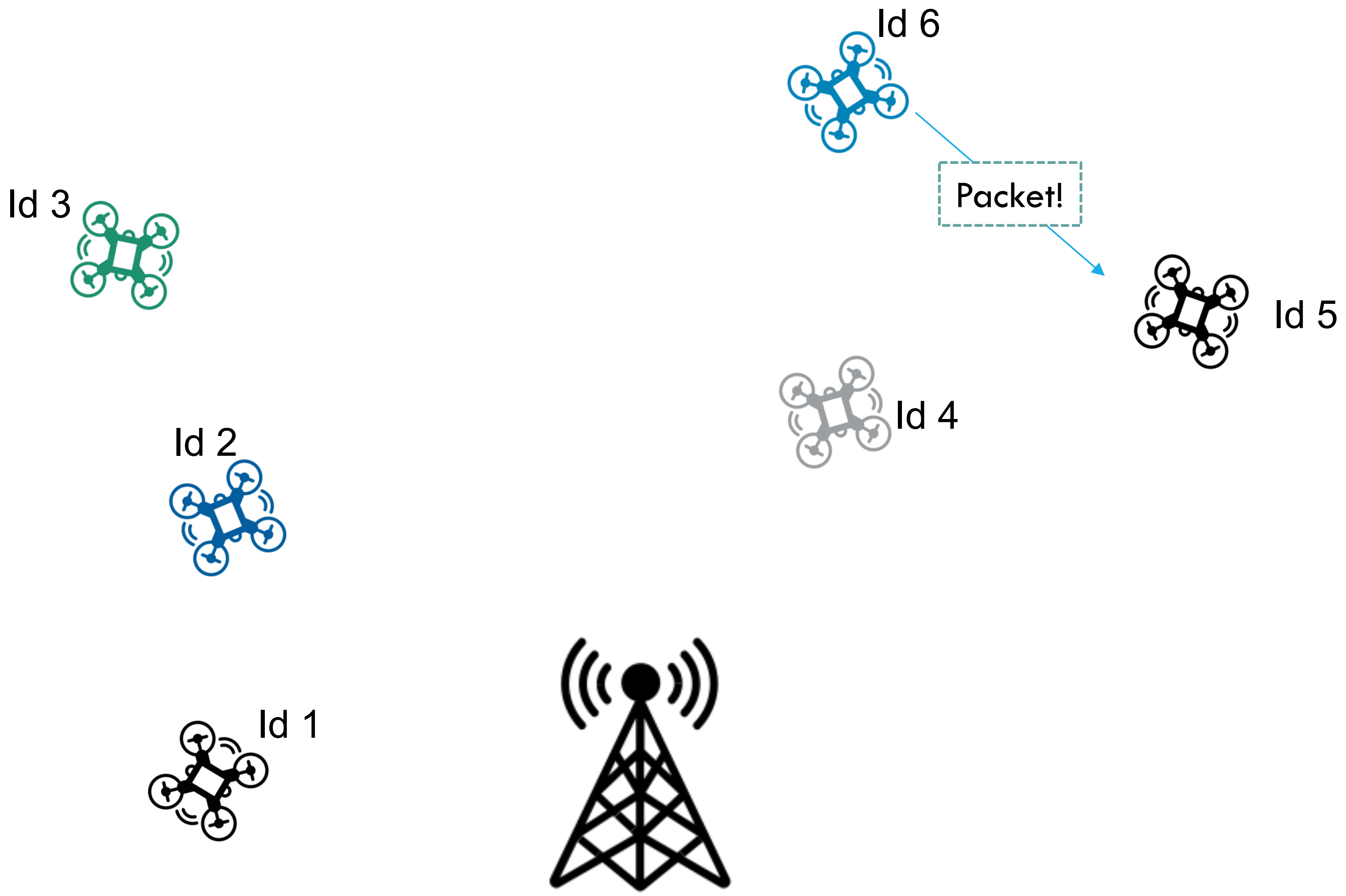
HOMEWORK 1 - ROUTING



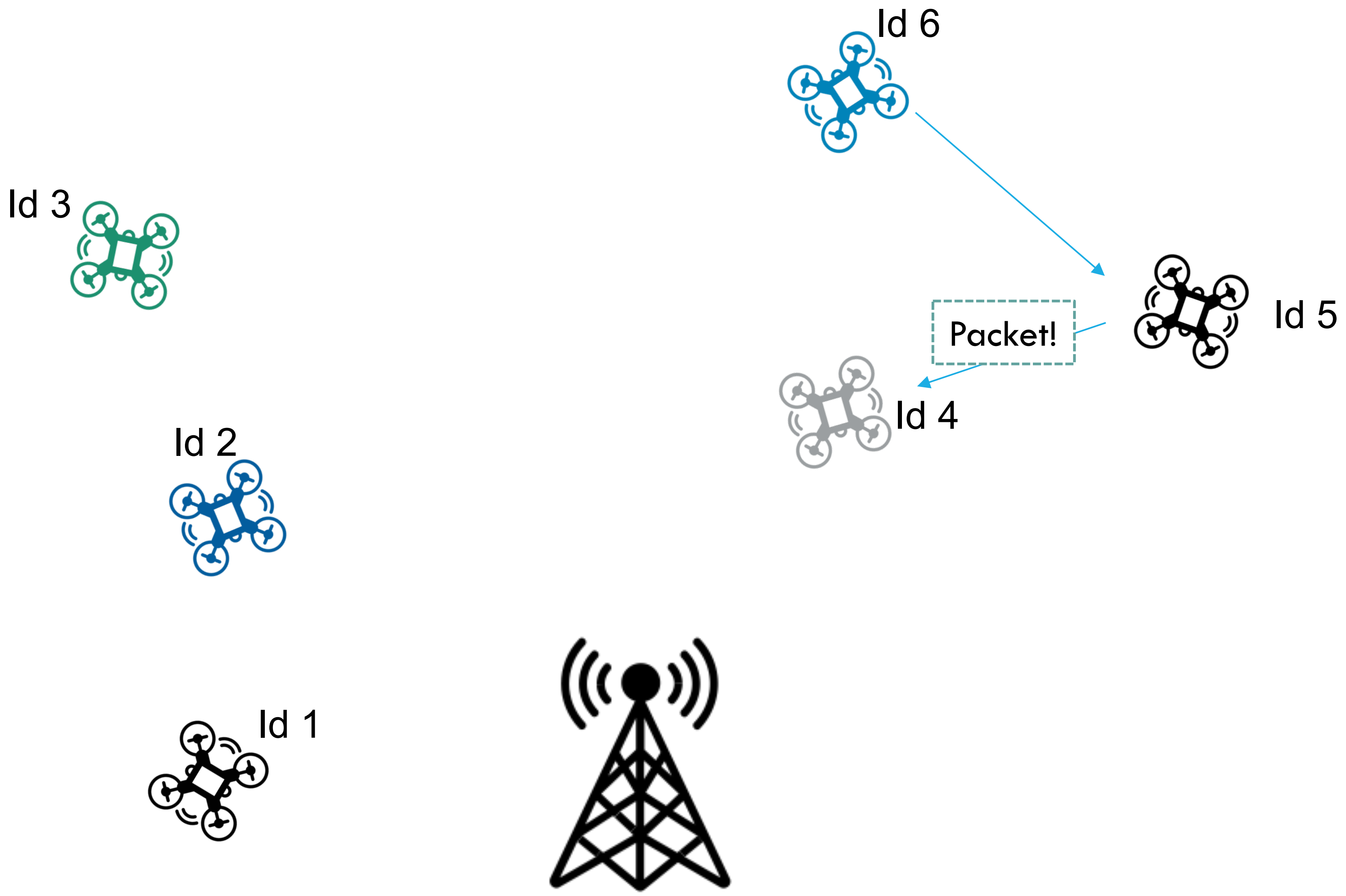
HOMEWORK 1 - ROUTING



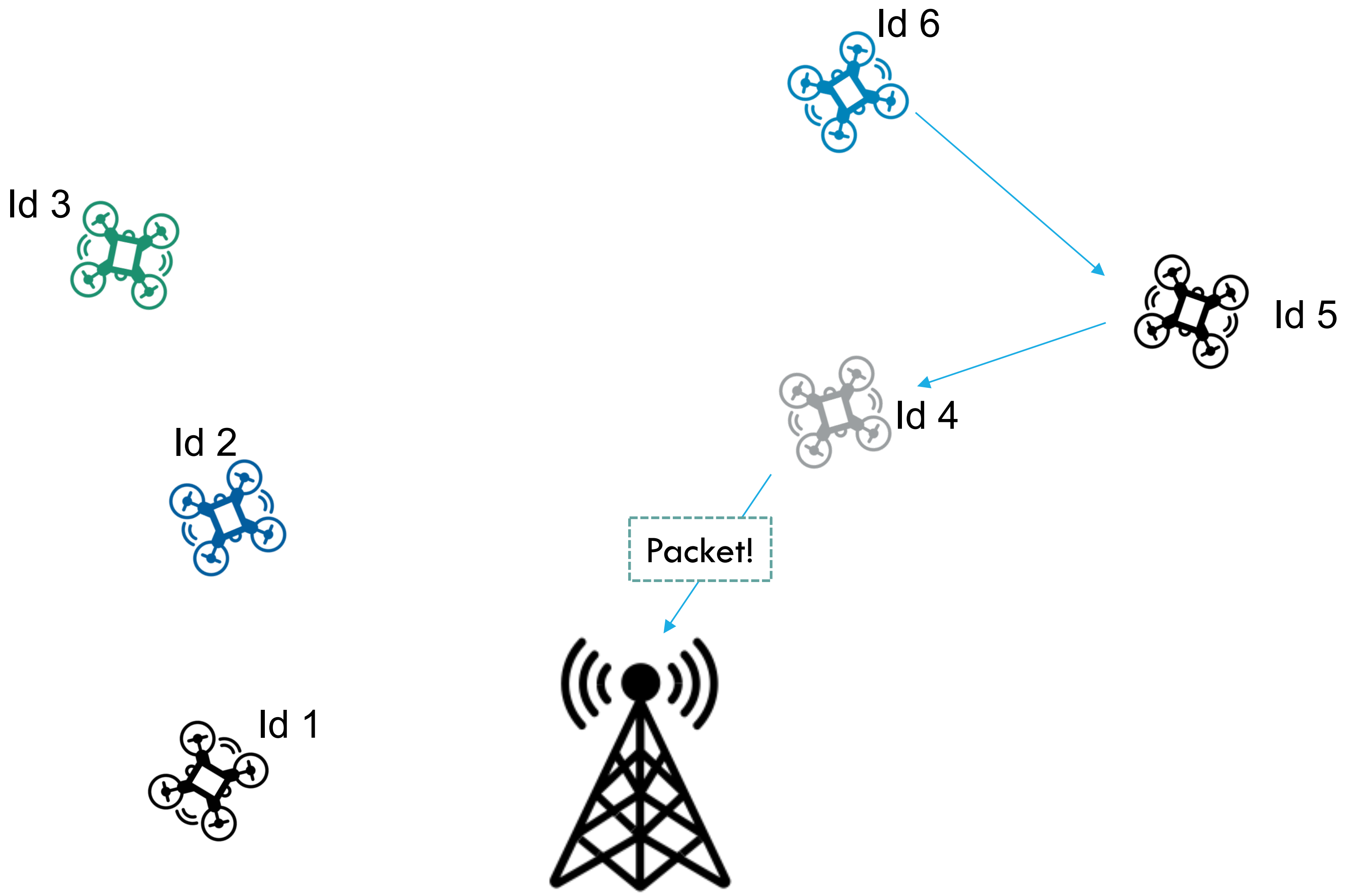
HOMEWORK 1 - ROUTING



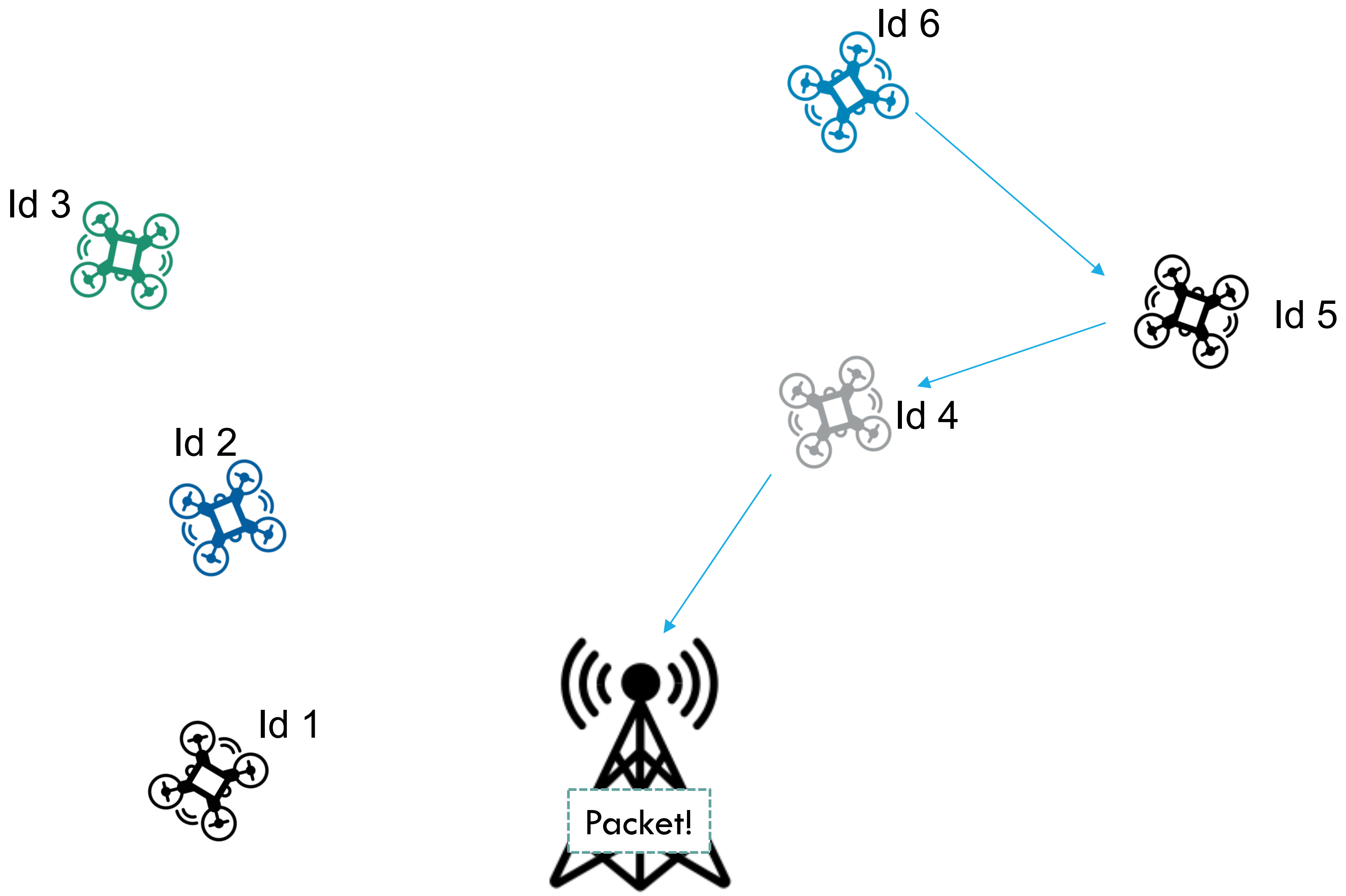
HOMEWORK 1 - ROUTING



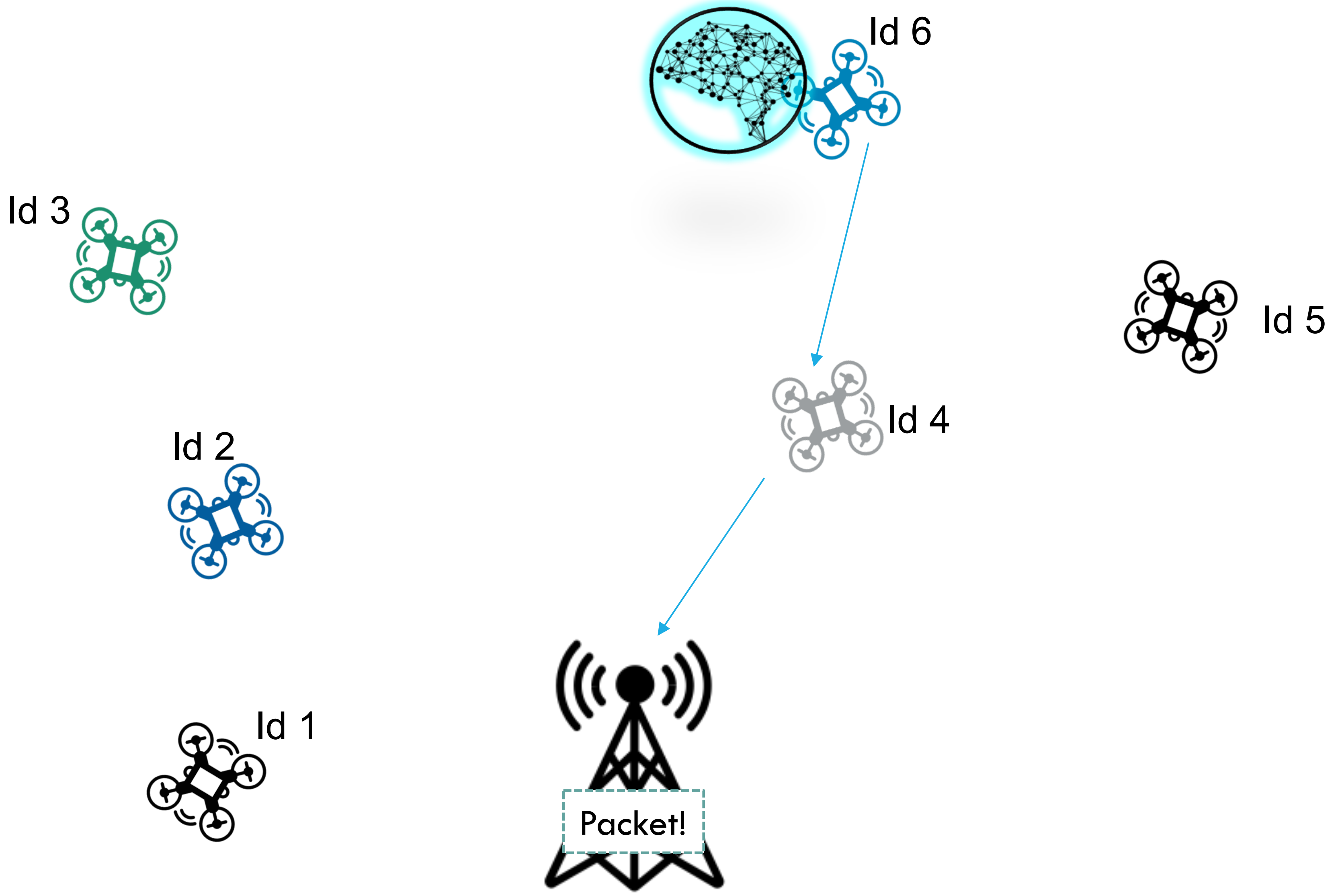
HOMEWORK 1 - ROUTING



HOMEWORK 1 - ROUTING

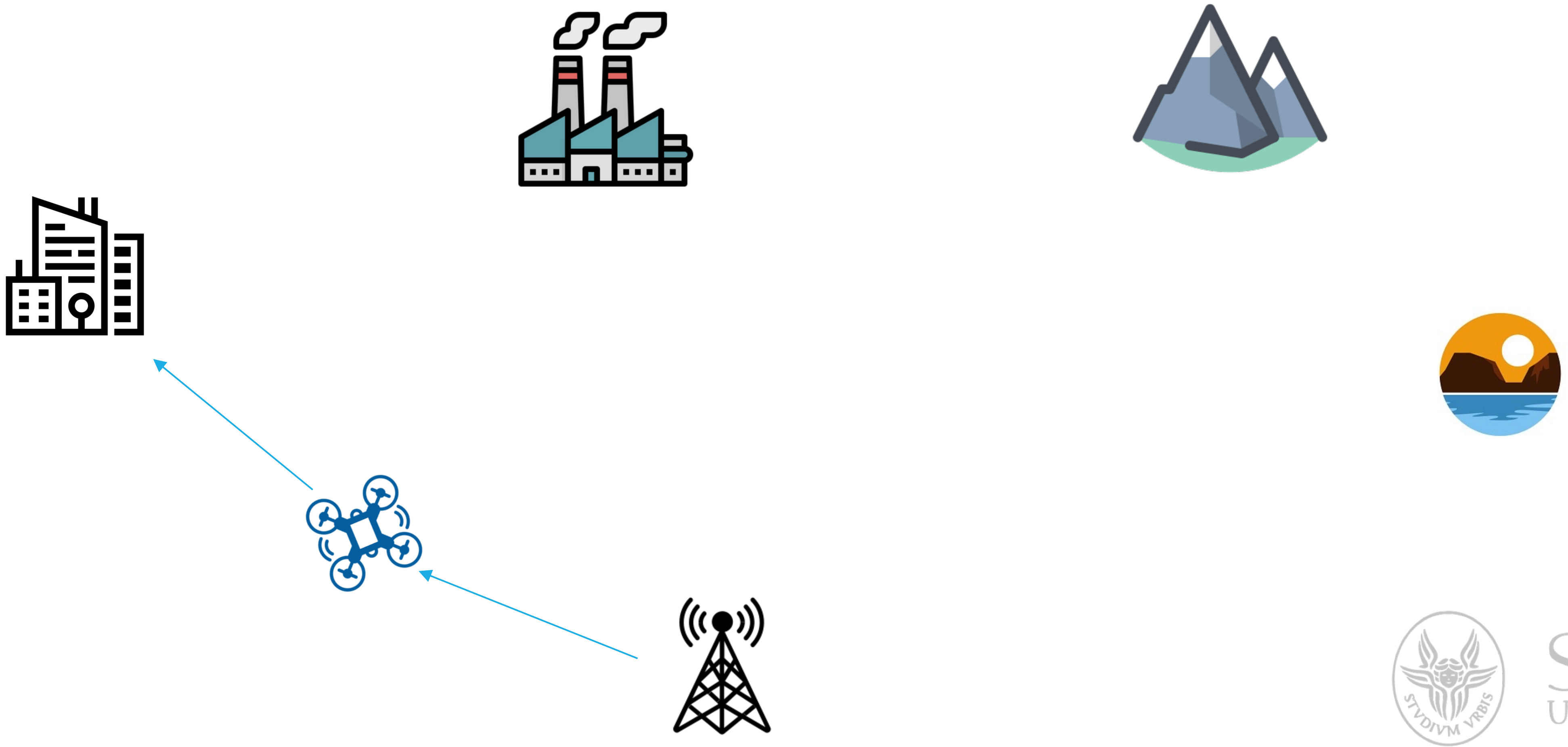


HOMEWORK 1 – REINFORCEMENT LEARNING ROUTING



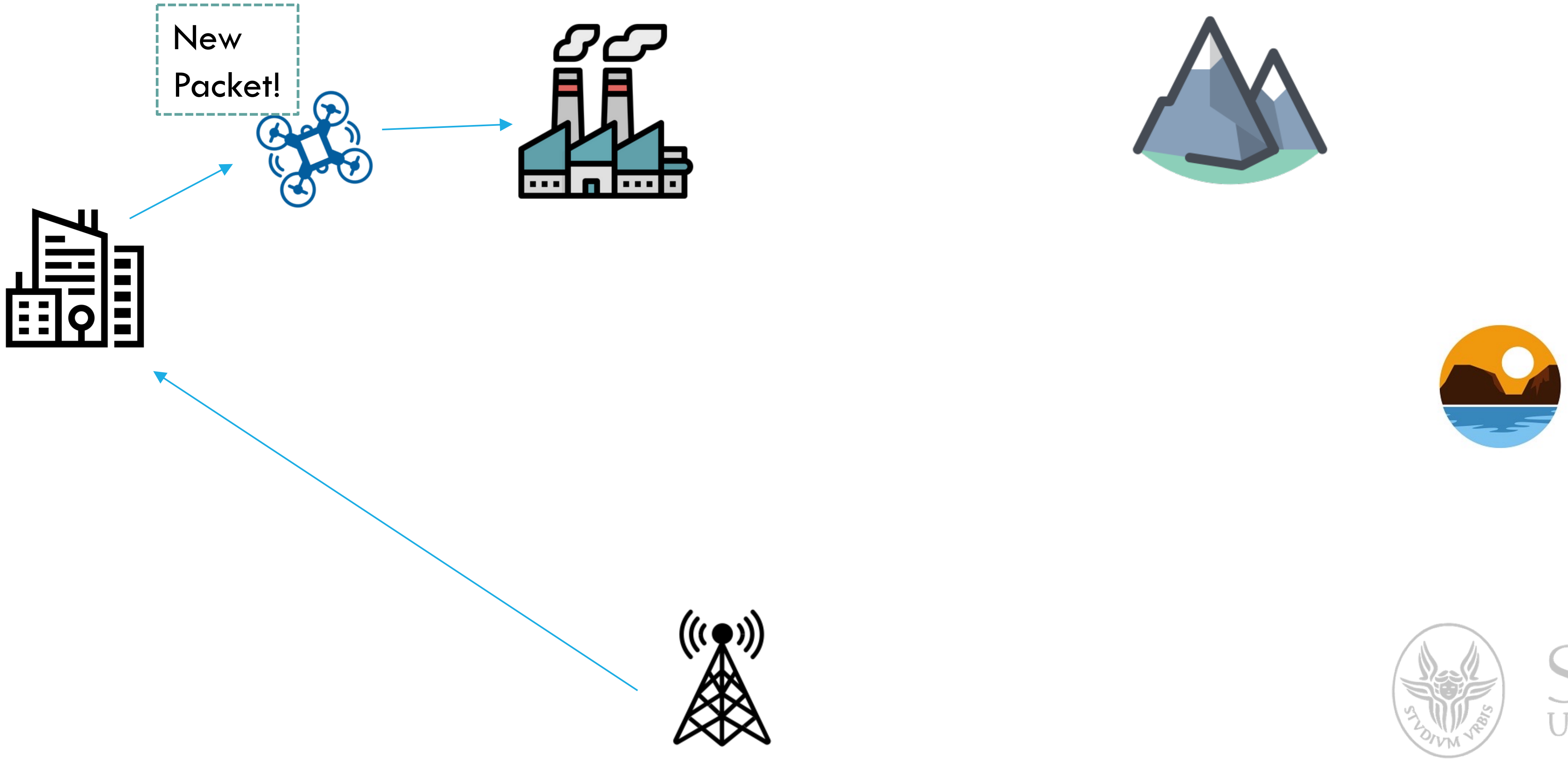
HOMEWORK 1 – PATROLLING SCENARIO

Drones continuously explore an area of interest to detect and monitor the area.



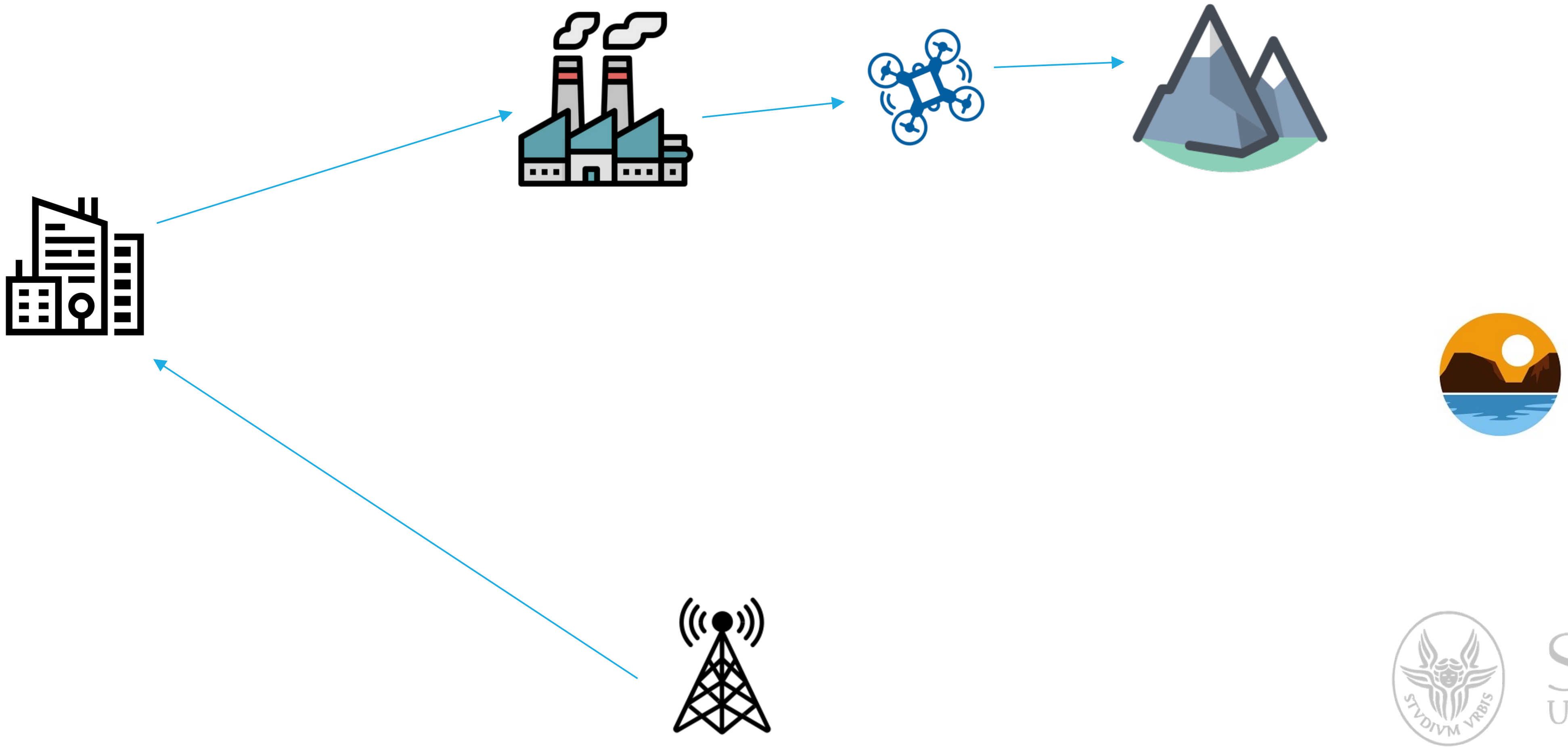
HOMEWORK 1 – PATROLLING SCENARIO

Drones continuously explore an area of interest to detect and monitor the area.



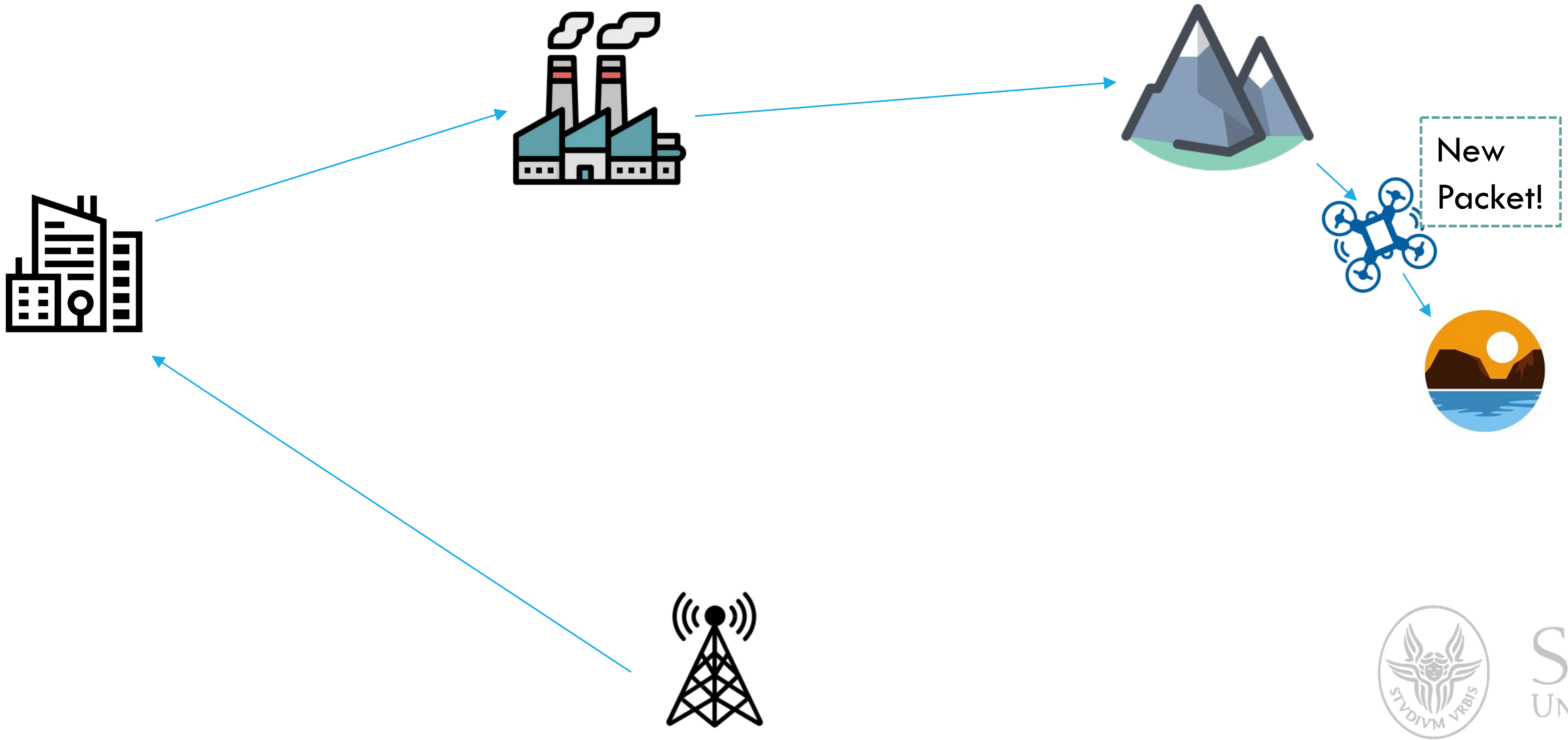
HOMEWORK 1 – PATROLLING SCENARIO

Drones continuously explore an area of interest to detect and monitor the area.



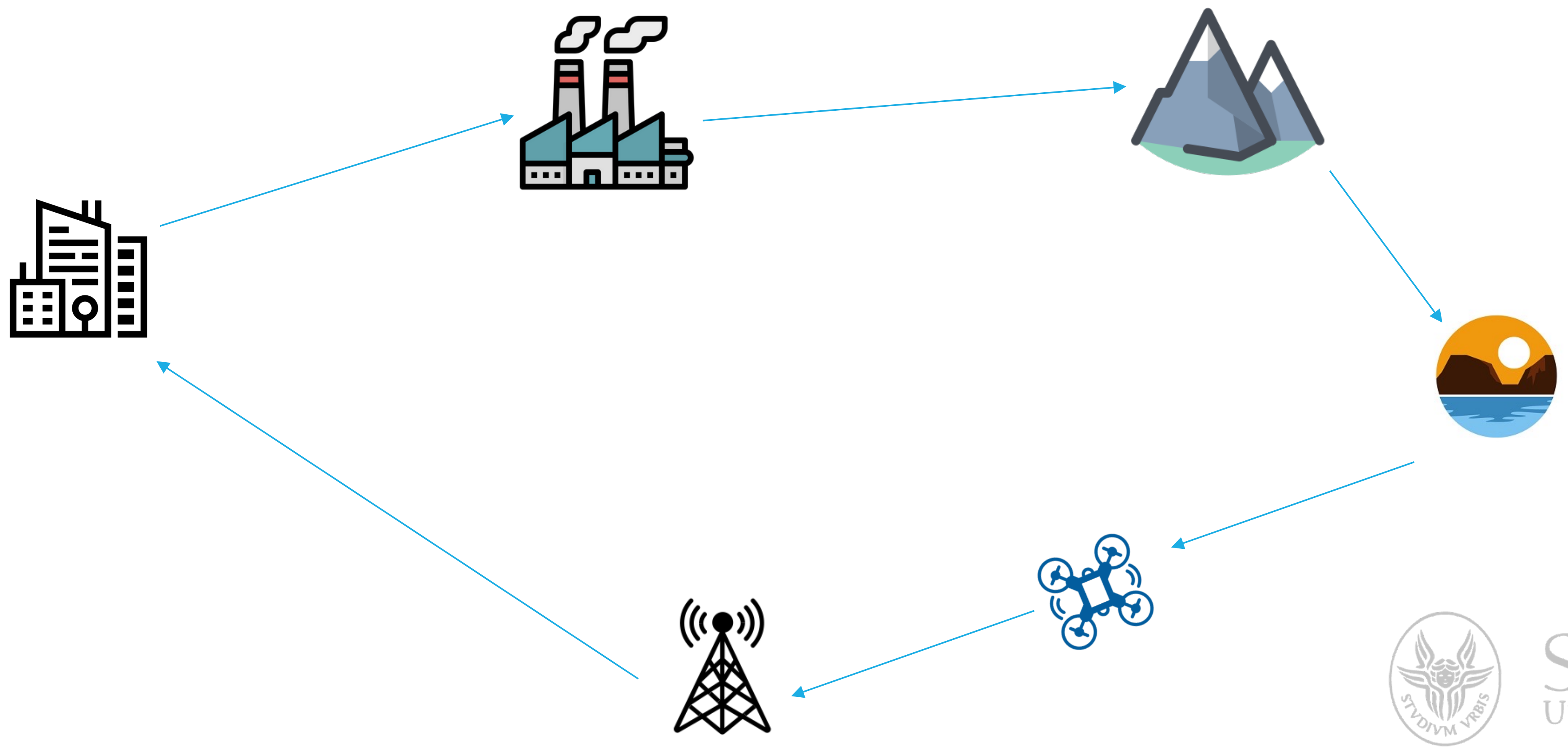
HOMEWORK 1 – PATROLLING SCENARIO

Drones continuously explore an area of interest to detect and monitor the area.



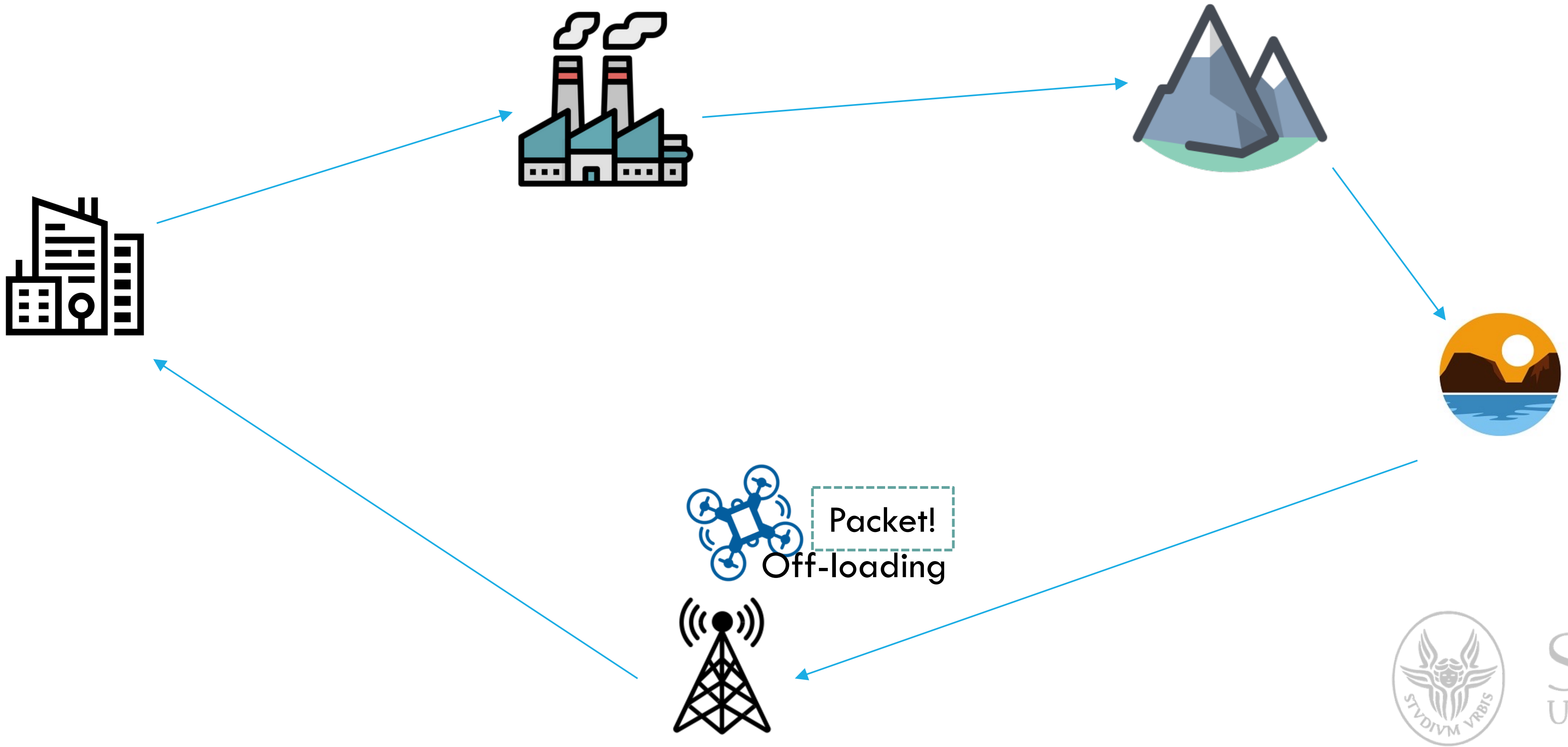
HOMEWORK 1 – PATROLLING SCENARIO

Drones continuously explore an area of interest to detect and monitor the area.



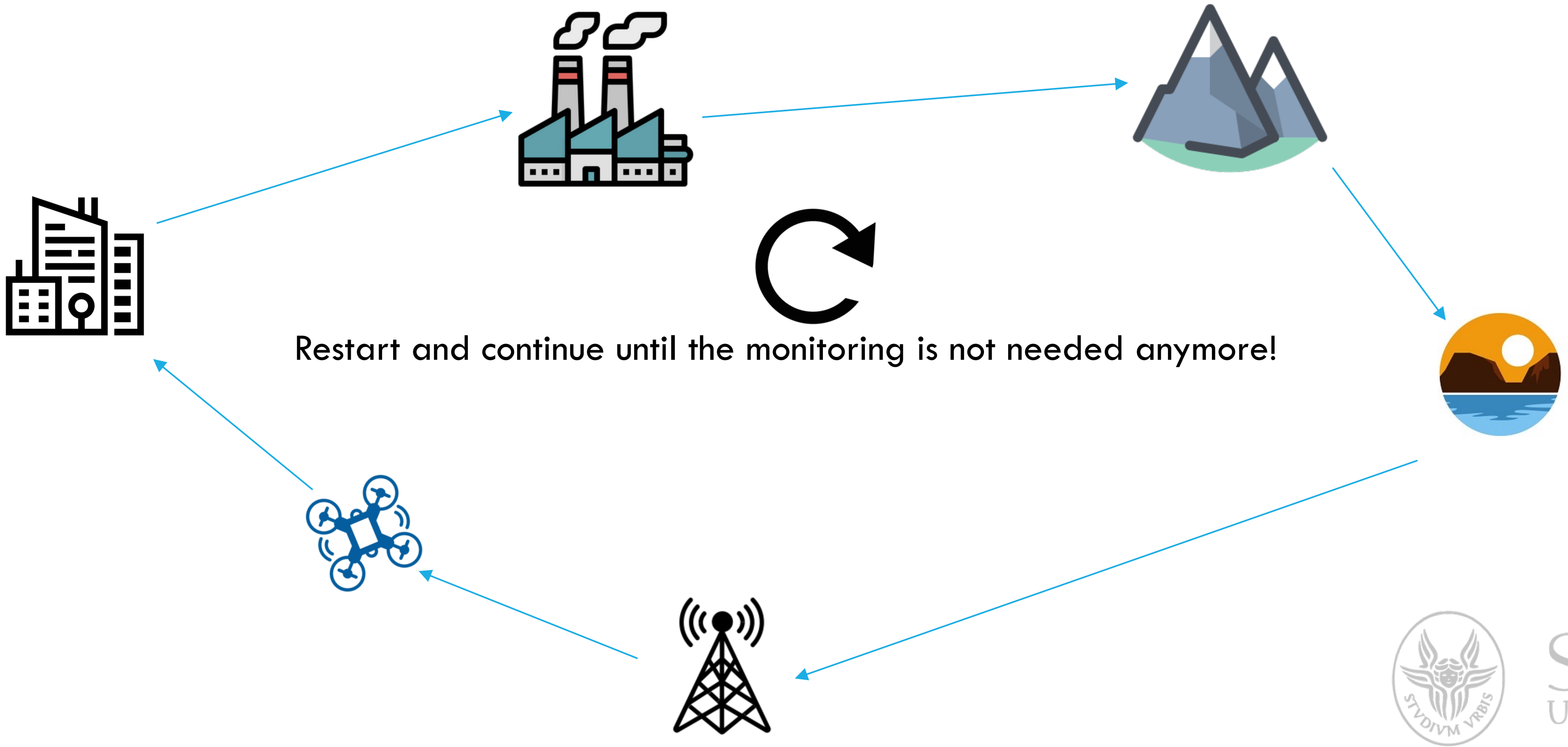
HOMEWORK 1 – PATROLLING SCENARIO

Drones continuously explore an area of interest to detect and monitor the area.



HOMEWORK 1 – PATROLLING SCENARIO

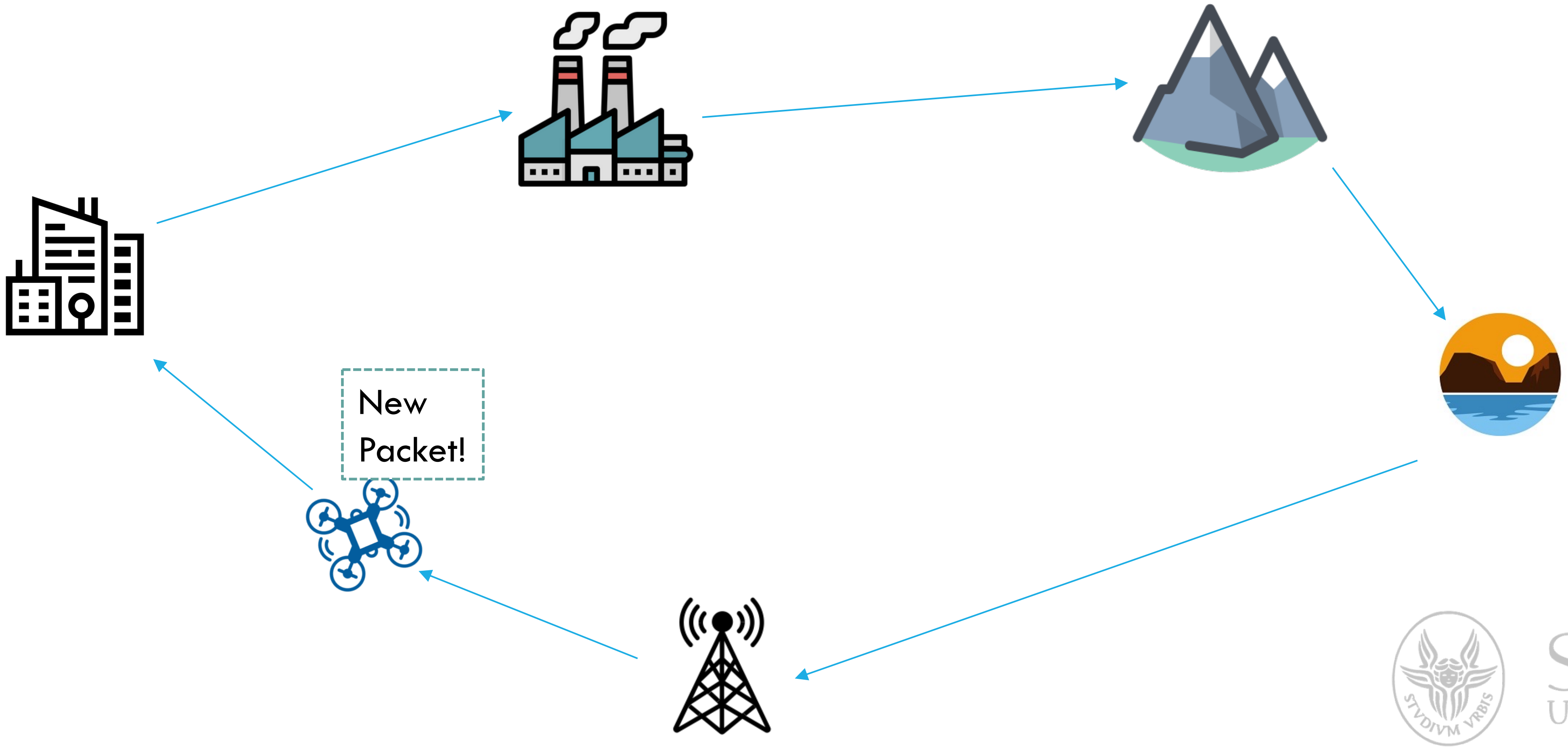
Drones continuously explore an area of interest to detect and monitor the area.


Restart and continue until the monitoring is not needed anymore!

HOMEWORK 1 – PATROLLING SCENARIO

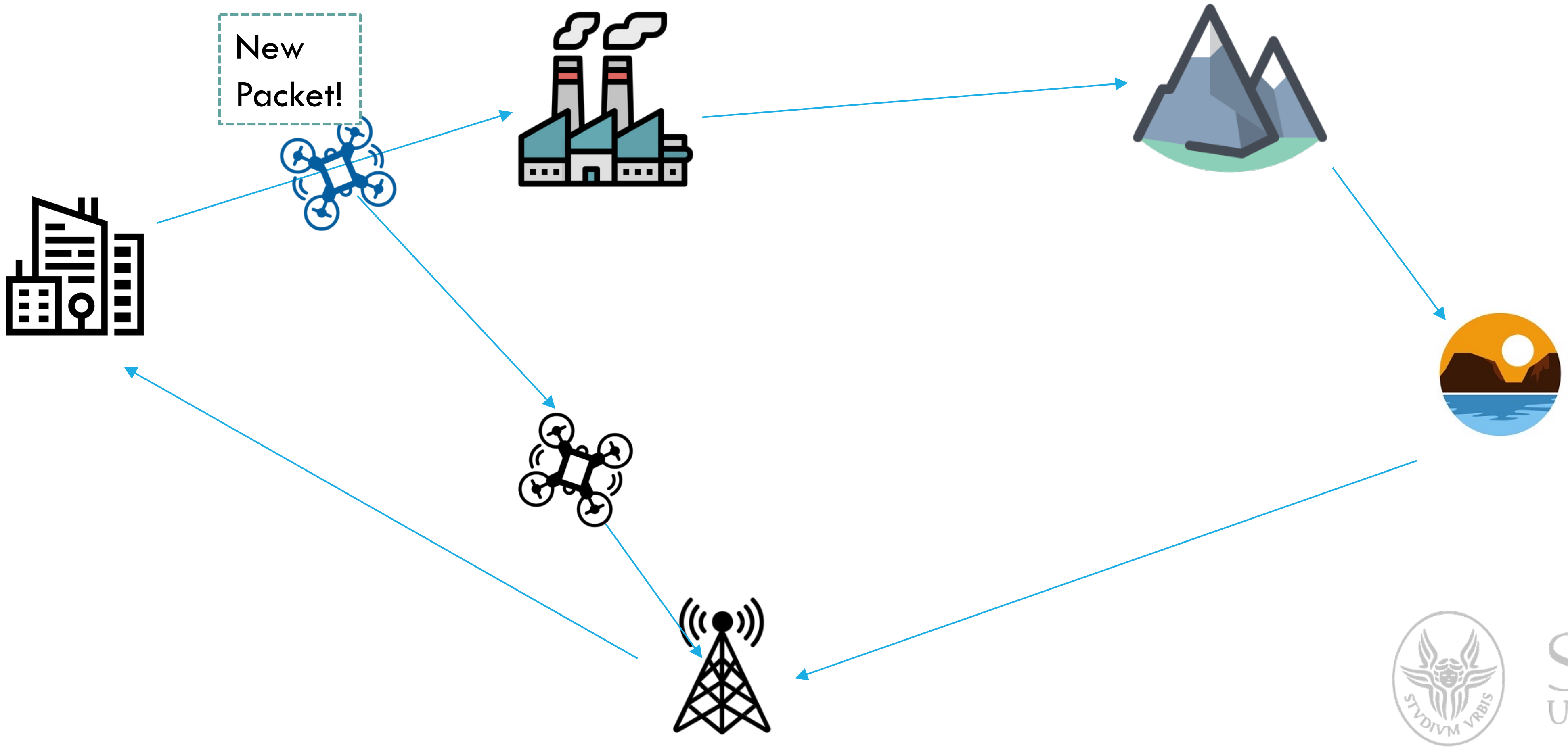
We have problems with this approach?

A packet has to wait until the drone arrives at the depot!!



HOMEWORK 1 – PATROLLING SCENARIO

Idea: Using a squad of drones and a routing protocol we can improve the delivery!!!

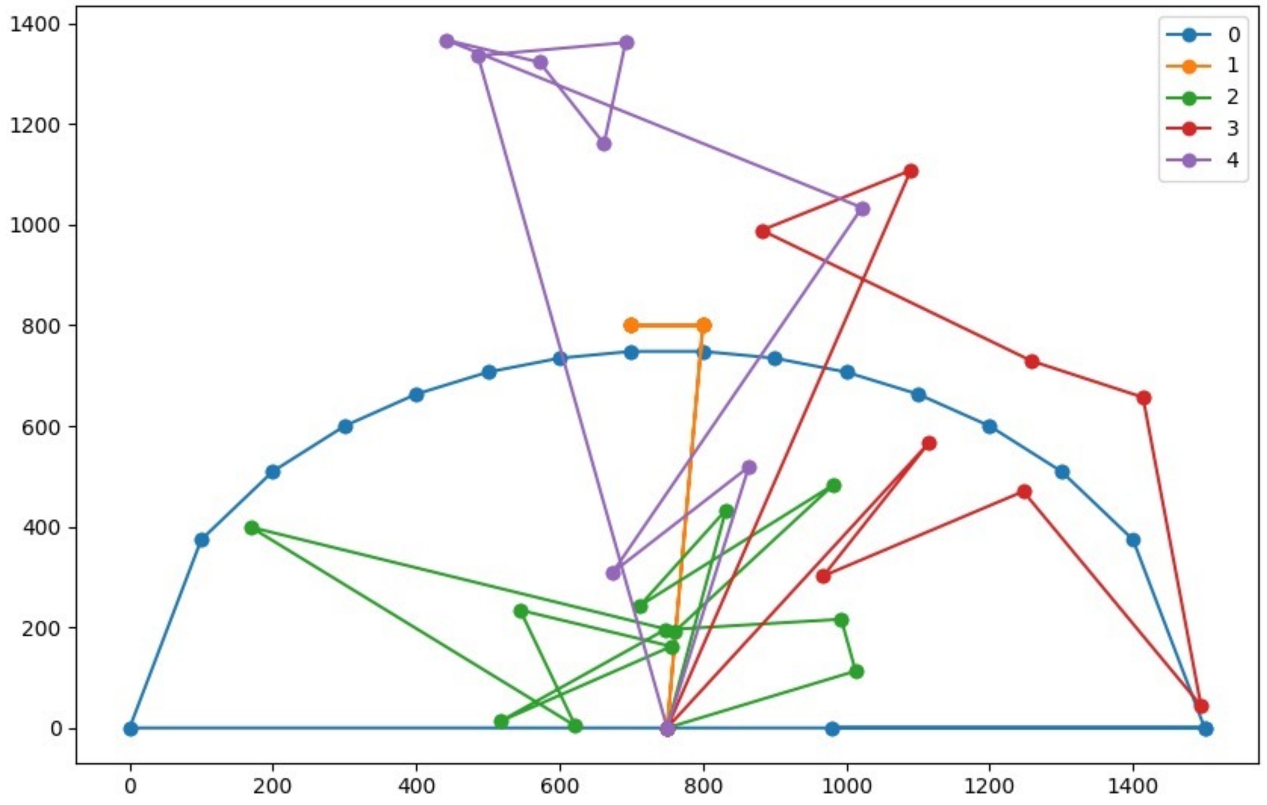


HOMEWORK 1 – PATROLLING SCENARIO

Idea: Using a squad of drones and a routing protocol we can improve the delivery!!!

We have a squad of N-drones (N >= 5):

- 3 Drones work as ferries, they move back-and-forth to the depot. To deliver data.
- N-3 drones are performing monitoring tasks.



Drone in [0,1,2] do not collect data, but they can physically move packets to the destination!

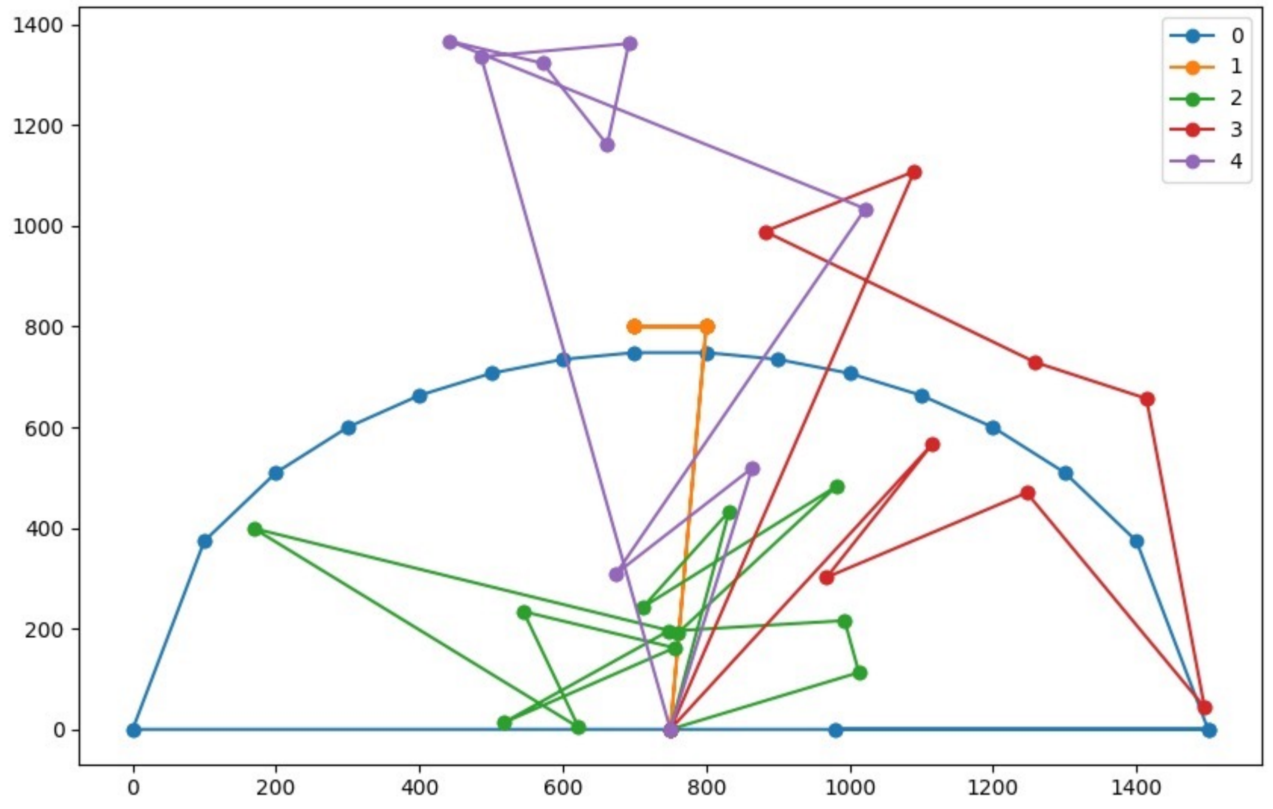
The drone routing can:

- Store the data and wait to arrive at the depot
- Send the data to a neighbor drone, which may arrive at the depot before him.

HOMEWORK 1 – YOUR TASK

Idea: Using a squad of drones and a routing protocol we can improve the delivery!!!

Each drone can take decision: keep or transmit the packet!
When a drone receives a packet, it can carry that packet to the depot or send again to another drone.



GOAL:

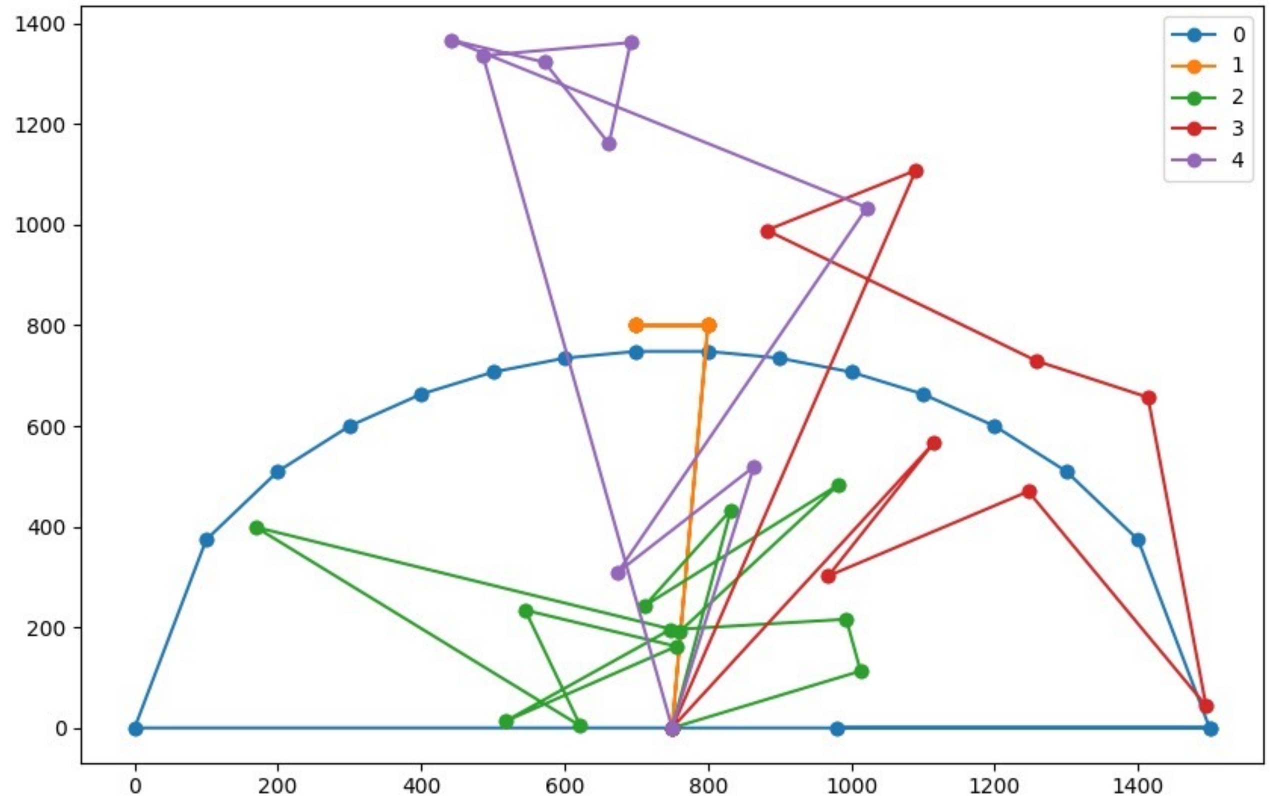
Create a **Reinforcement Learning** Routing Protocol for drones.

To decide whether keep or send the packet.

HOMEWORK 1 – ASSUMPTIONS

Idea: Using a squad of drones and a routing protocol we can improve the delivery!!!

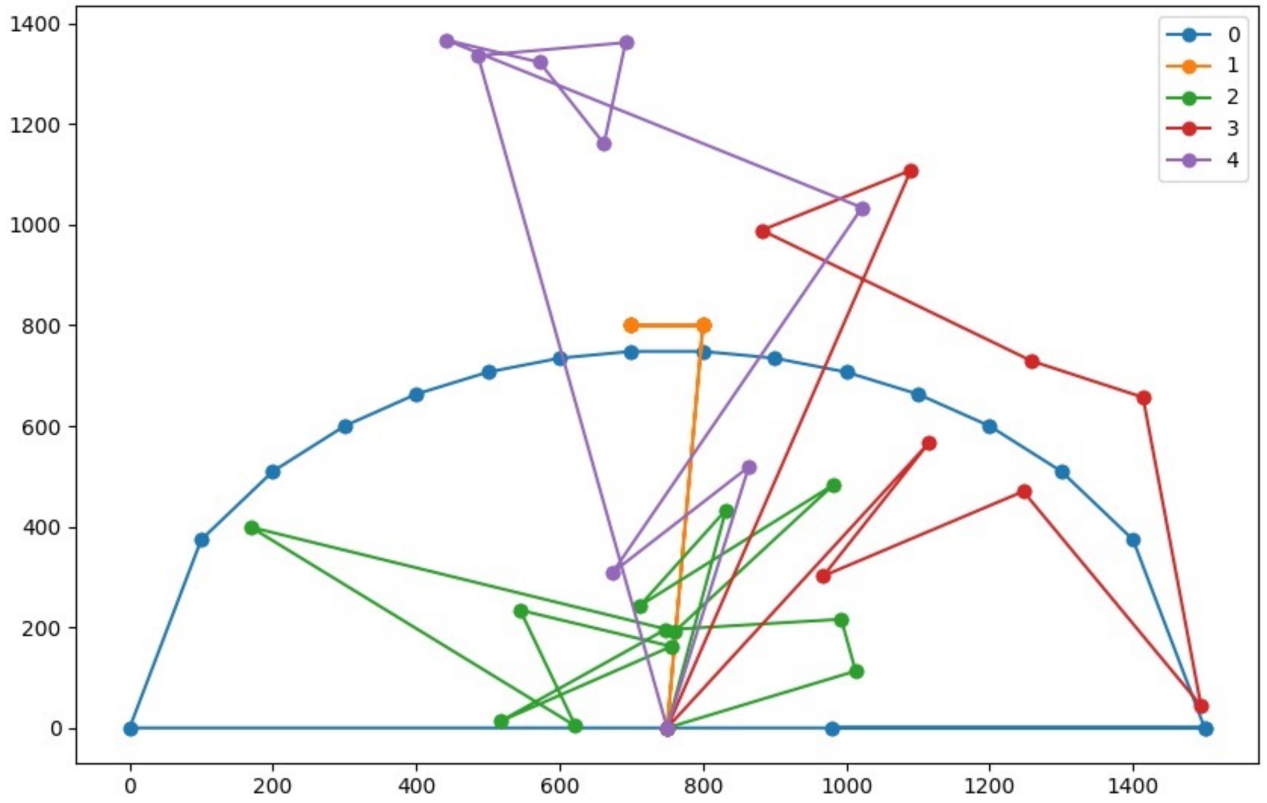
The patrolling mission repeat with the same trajectories until the simulation ends.
Note: You can't assume to know your trajectory, but you know your history and that it may be repeated (with small changes).
The packets have a limited time to live!



HOMEWORK 1 – GOAL

You have to delivery as much as possible packets to the depot as primary task.
As secondary task you have to reduce the latency of the packets!

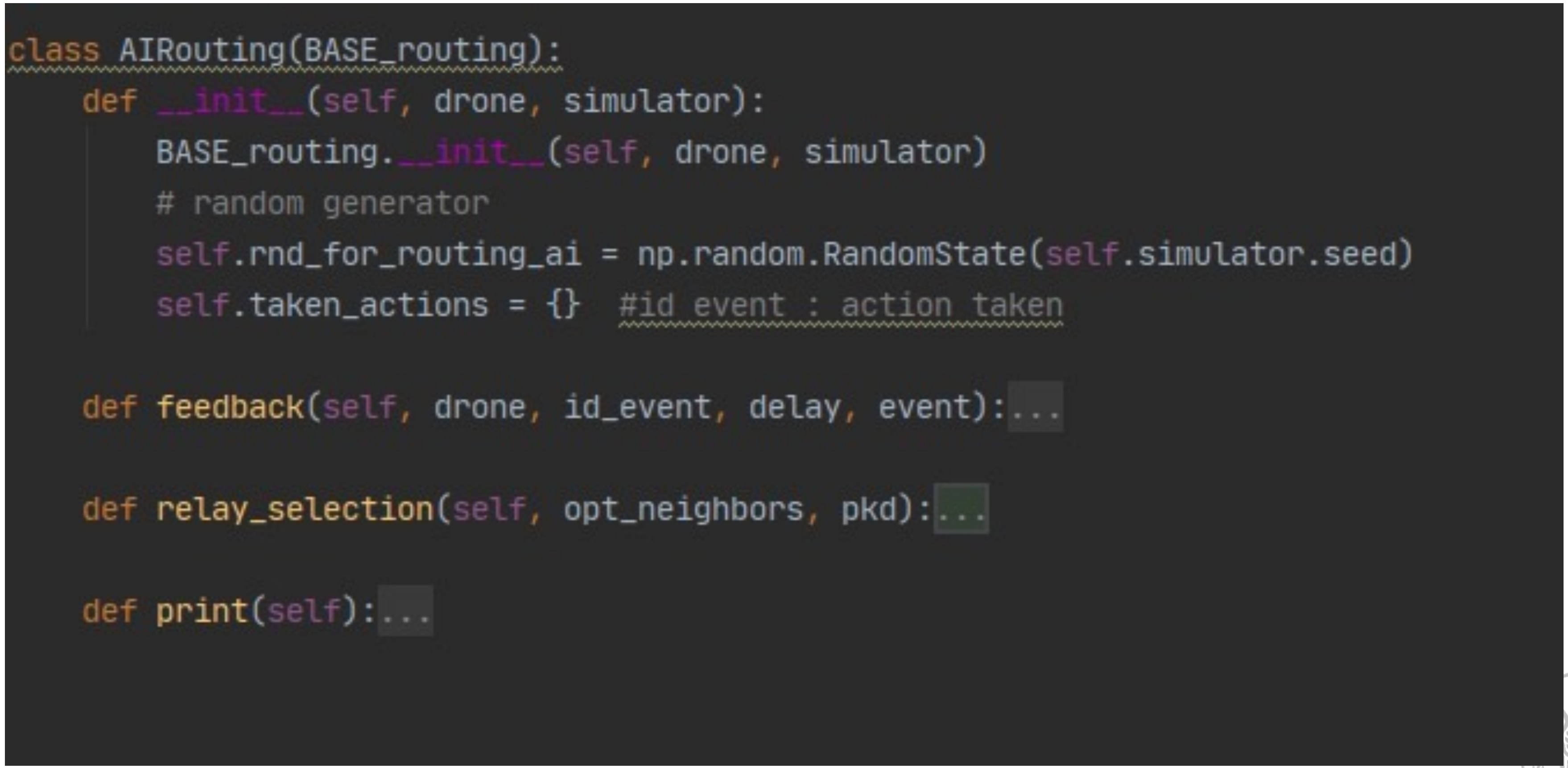
The final score/goal of your approach is to minimize: $1.5 \cdot |expired\_packets| \cdot ttl + \sum_{pck \in delivered} delivery\_time$



ttl -> the maximum time to live of a packet in seconds

HOMEWORK 1 – HOW

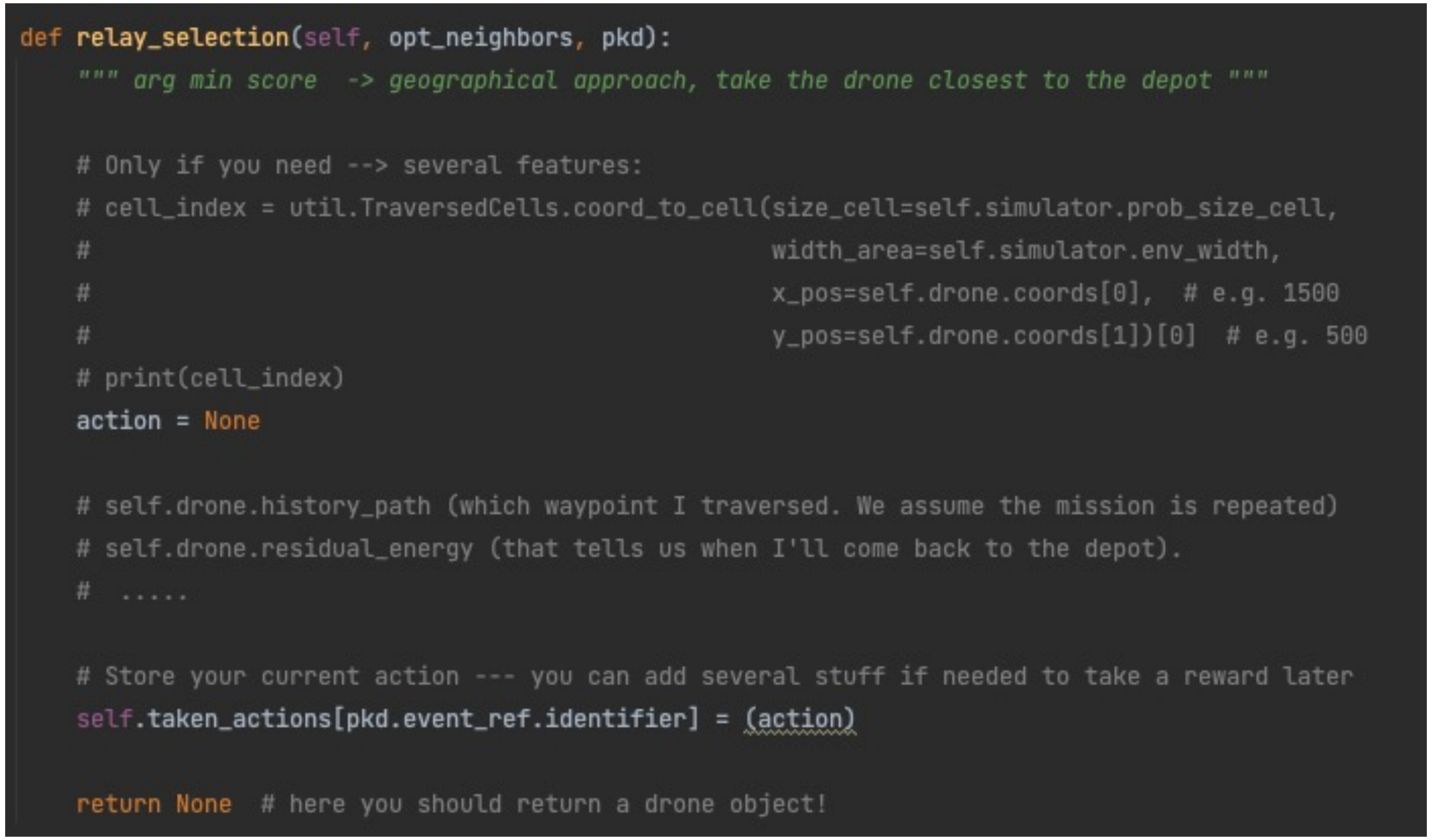
You can implement the Class below, using a Reinforcement Learning approach.
We suggest you start with a simple bandit approach.



```
class AIRouting(BASE_routing):
    def __init__(self, drone, simulator):
        BASE_routing.__init__(self, drone, simulator)
        # random generator
        self.rnd_for_routing_ai = np.random.RandomState(self.simulator.seed)
        self.taken_actions = {}  #id event : action taken

    def feedback(self, drone, id_event, delay, event):...

    def relay_selection(self, opt_neighbors, pkd):...

    def print(self):...
```

HOMEWORK 1 – HOW

The main method is the same:



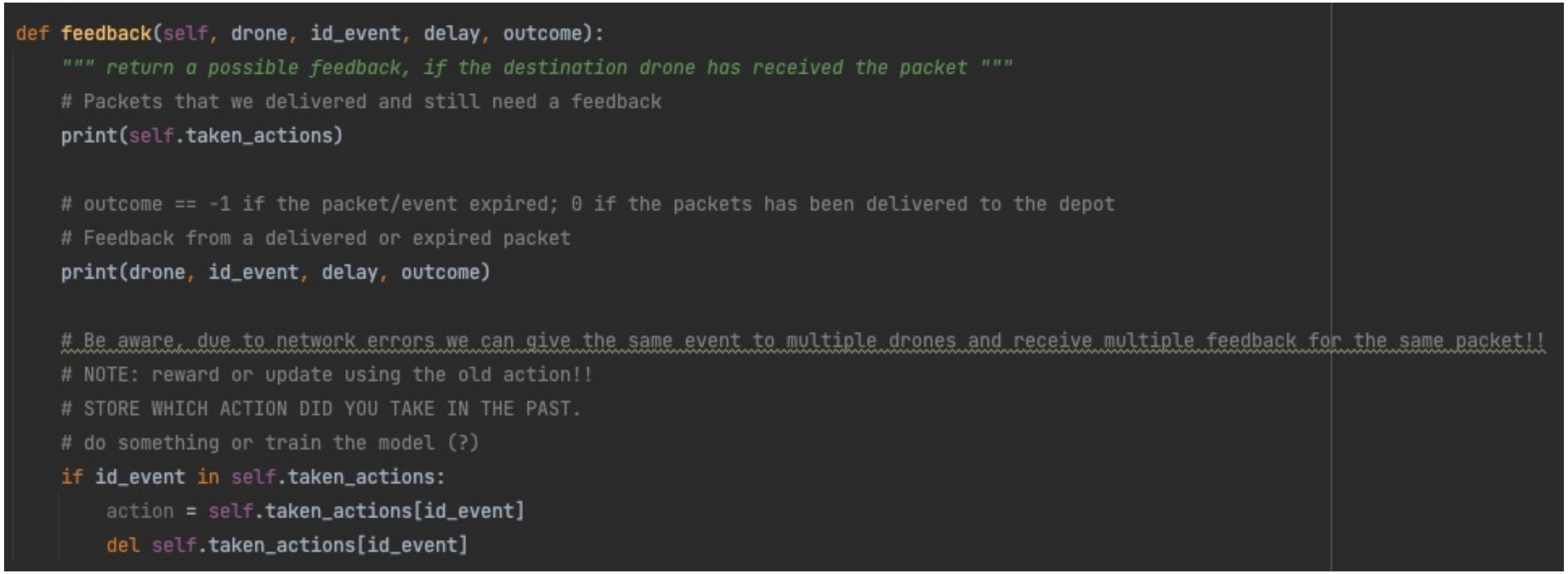
```
def relay_selection(self, opt_neighbors, pkd):
    """ arg min score  -> geographical approach, take the drone closest to the depot """

    # Only if you need --> several features:
    # cell_index = util.TraversedCells.coord_to_cell(size_cell=self.simulator.prob_size_cell,
    #                                                 width_area=self.simulator.env_width,
    #                                                 x_pos=self.drone.coords[0],  # e.g. 1500
    #                                                 y_pos=self.drone.coords[1])[0]  # e.g. 500
    # print(cell_index)
    action = None

    # self.drone.history_path (which waypoint I traversed. We assume the mission is repeated)
    # self.drone.residual_energy (that tells us when I'll come back to the depot).
    #  .....

    # Store your current action --- you can add several stuff if needed to take a reward later
    self.taken_actions[pkd.event_ref.identifier] = (action)

    return None  # here you should return a drone object!
```

Notice that, you can store the state and the action (or what you prefer) in the self.taken_action which is useful to update your model according the feedback!

The feedback is delayed (when the packet arrives at the depot or expires) you will be notified.

HOMEWORK 1 – HOW

The feedback is notified using a control channel (long-range radio with low bit rate) to each drone.



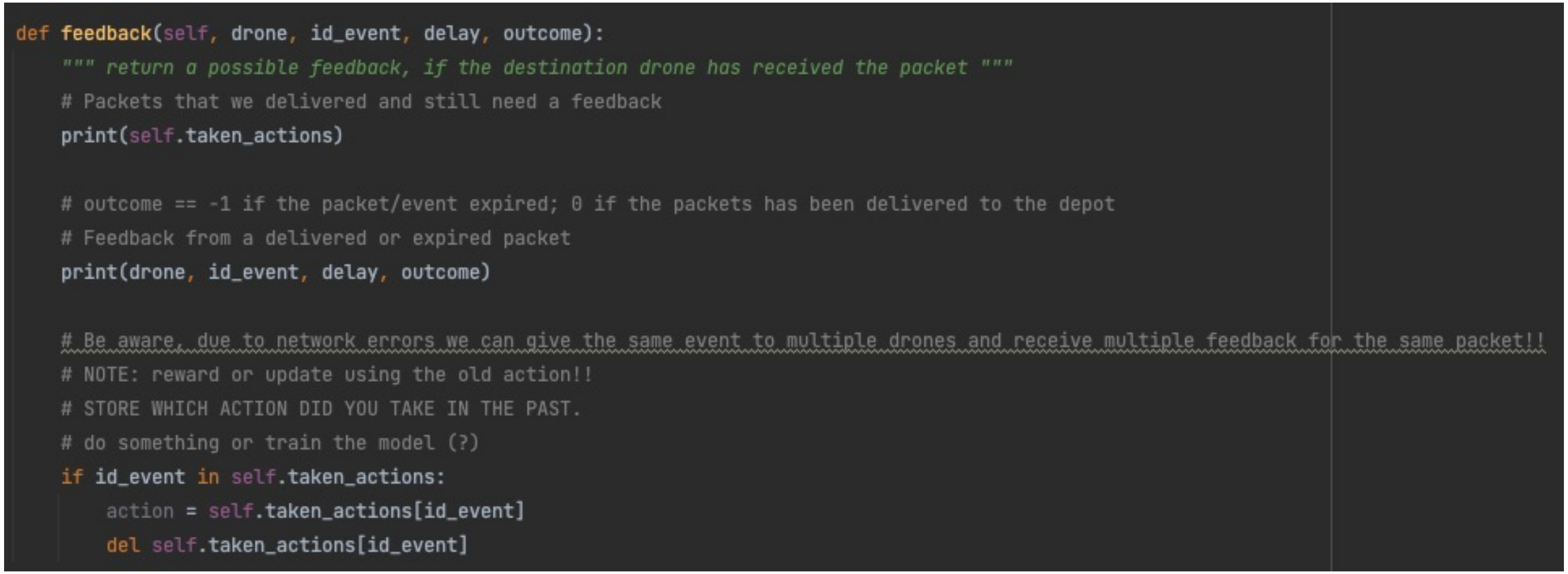
```
def feedback(self, drone, id_event, delay, outcome):
    """ return a possible feedback, if the destination drone has received the packet """
    # Packets that we delivered and still need a feedback
    print(self.taken_actions)

    # outcome == -1 if the packet/event expired; 0 if the packets has been delivered to the depot
    # Feedback from a delivered or expired packet
    print(drone, id_event, delay, outcome)

    # Be aware, due to network errors we can give the same event to multiple drones and receive multiple feedback for the same packet!!
    # NOTE: reward or update using the old action!!
    # STORE WHICH ACTION DID YOU TAKE IN THE PAST.
    # do something or train the model (?)
    if id_event in self.taken_actions:
        action = self.taken_actions[id_event]
        del self.taken_actions[id_event]
```

It contains: the drone that delivered the packet, the identifier of the event; the delay of the packet (time elapsed from the beginning of the event); the outcome.

The outcome is 0 if the event has been delivered to the depot; -1 if the event has expired before.

HOMEWORK 1 – HOW

The feedback is notified using a control channel (long-range radio with low bit rate) to each drone.



```
def feedback(self, drone, id_event, delay, outcome):
    """ return a possible feedback, if the destination drone has received the packet """
    # Packets that we delivered and still need a feedback
    print(self.taken_actions)

    # outcome == -1 if the packet/event expired; 0 if the packets has been delivered to the depot
    # Feedback from a delivered or expired packet
    print(drone, id_event, delay, outcome)

    # Be aware, due to network errors we can give the same event to multiple drones and receive multiple feedback for the same packet!!
    # NOTE: reward or update using the old action!!
    # STORE WHICH ACTION DID YOU TAKE IN THE PAST.
    # do something or train the model (?)
    if id_event in self.taken_actions:
        action = self.taken_actions[id_event]
        del self.taken_actions[id_event]
```

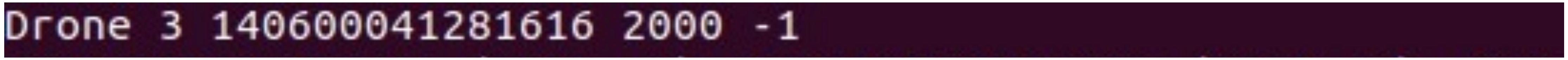
Using the **id_event** and **self.taken_actions** you can understand which action and state have caused that result!

Notice that, due to communication errors, an event can be delivered multiple times!!

HOMEWORK 1 – HOW

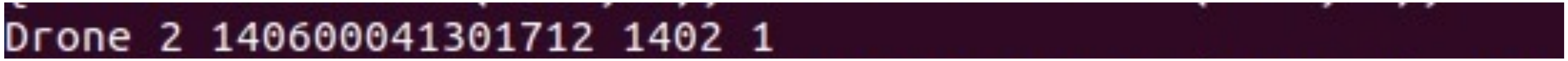
Example of a feedback:

Drone, the identifier of the event, the delay of the packet, the outcome (expired).



```
Drone 3 140600041281616 2000 -1
```

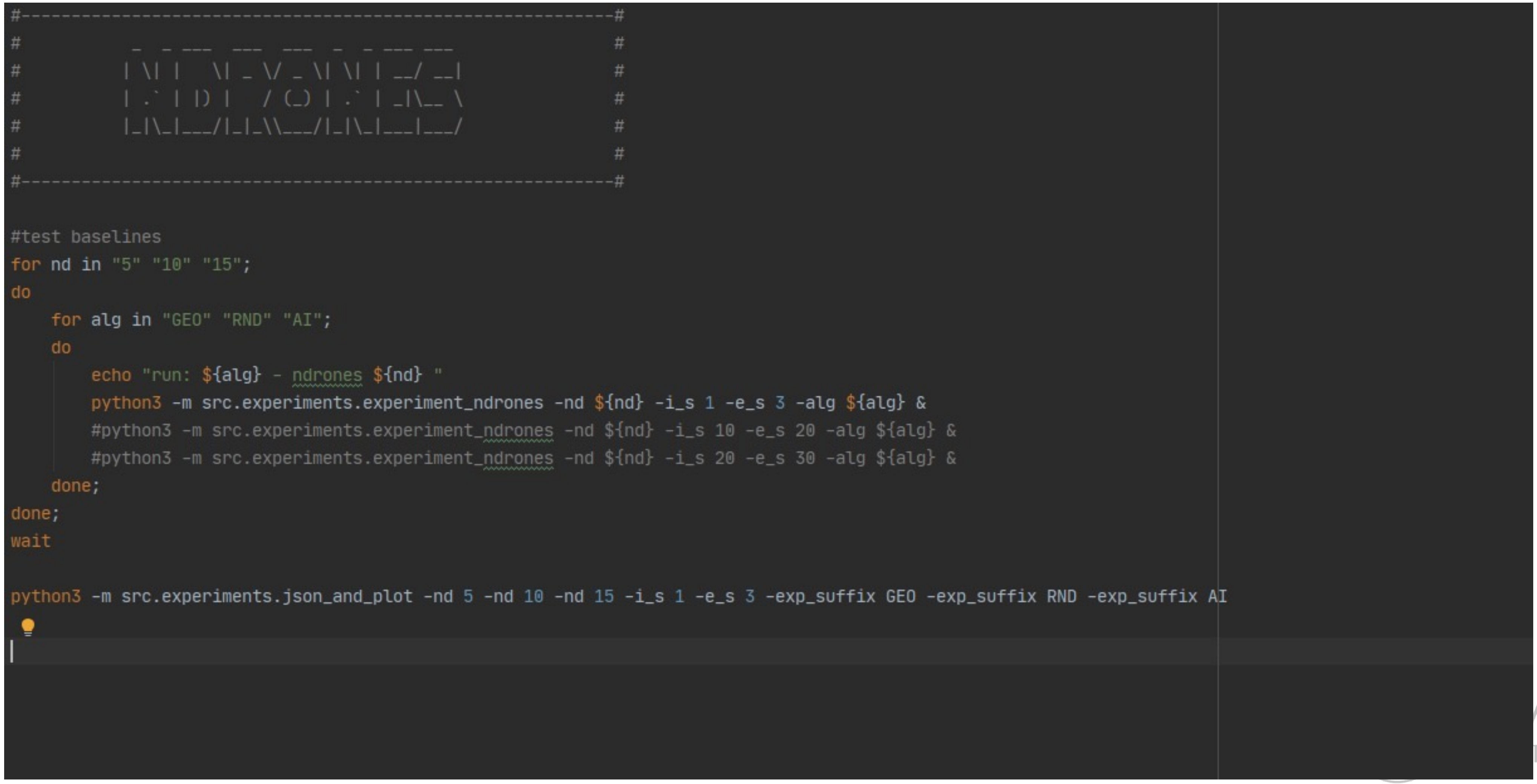
Drone, the identifier of the event, the delay of the packet, the outcome (delivered).



```
Drone 2 140600041301712 1402 1
```

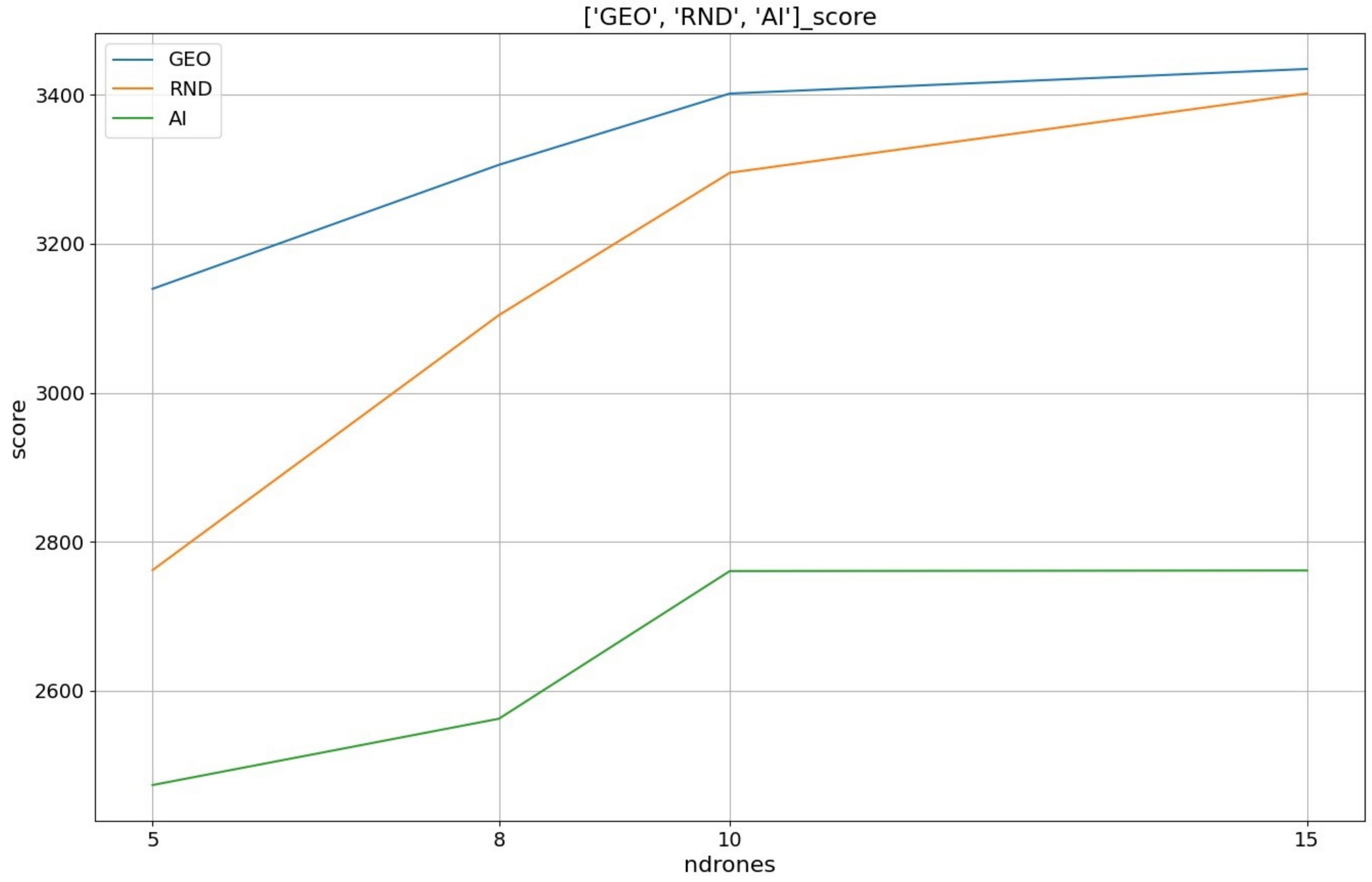
HOMEWORK 1 – RUN MULTIPLE SIMULATIONS

As for the first homework:

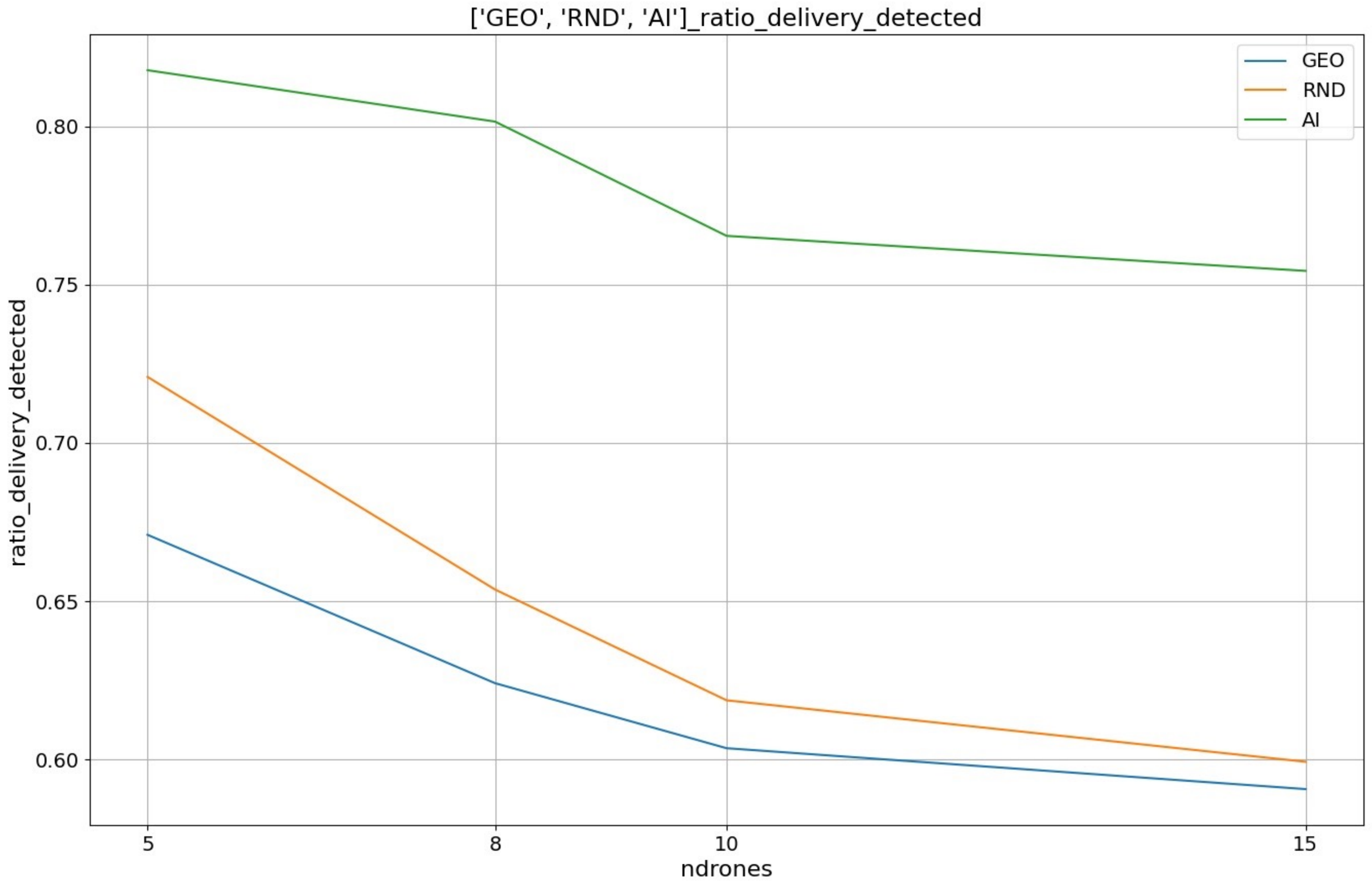


```
#----------------------------------------------------------#
#            _  _ ___  ___  ___  _  _ ___ ___              #
#           | \| |   \| _ \/ _ \| \| | __/ __|             #
#           | .` | |) |   / (_) | .` | _|\__ \             #
#           |_|\_|___/|_|_\\___/|_|\_|___|___/             #
#                                                          #
#----------------------------------------------------------#


#test baselines
for nd in "5" "10" "15";
do
    for alg in "GEO" "RND" "AI";
    do
        echo "run: ${alg} - ndrones ${nd} "
        python3 -m src.experiments.experiment_ndrones -nd ${nd} -i_s 1 -e_s 3 -alg ${alg} &
        #python3 -m src.experiments.experiment_ndrones -nd ${nd} -i_s 10 -e_s 20 -alg ${alg} &
        #python3 -m src.experiments.experiment_ndrones -nd ${nd} -i_s 20 -e_s 30 -alg ${alg} &
    done;
done;
wait

python3 -m src.experiments.json_and_plot -nd 5 -nd 10 -nd 15 -i_s 1 -e_s 3 -exp_suffix GEO -exp_suffix RND -exp_suffix AI
```

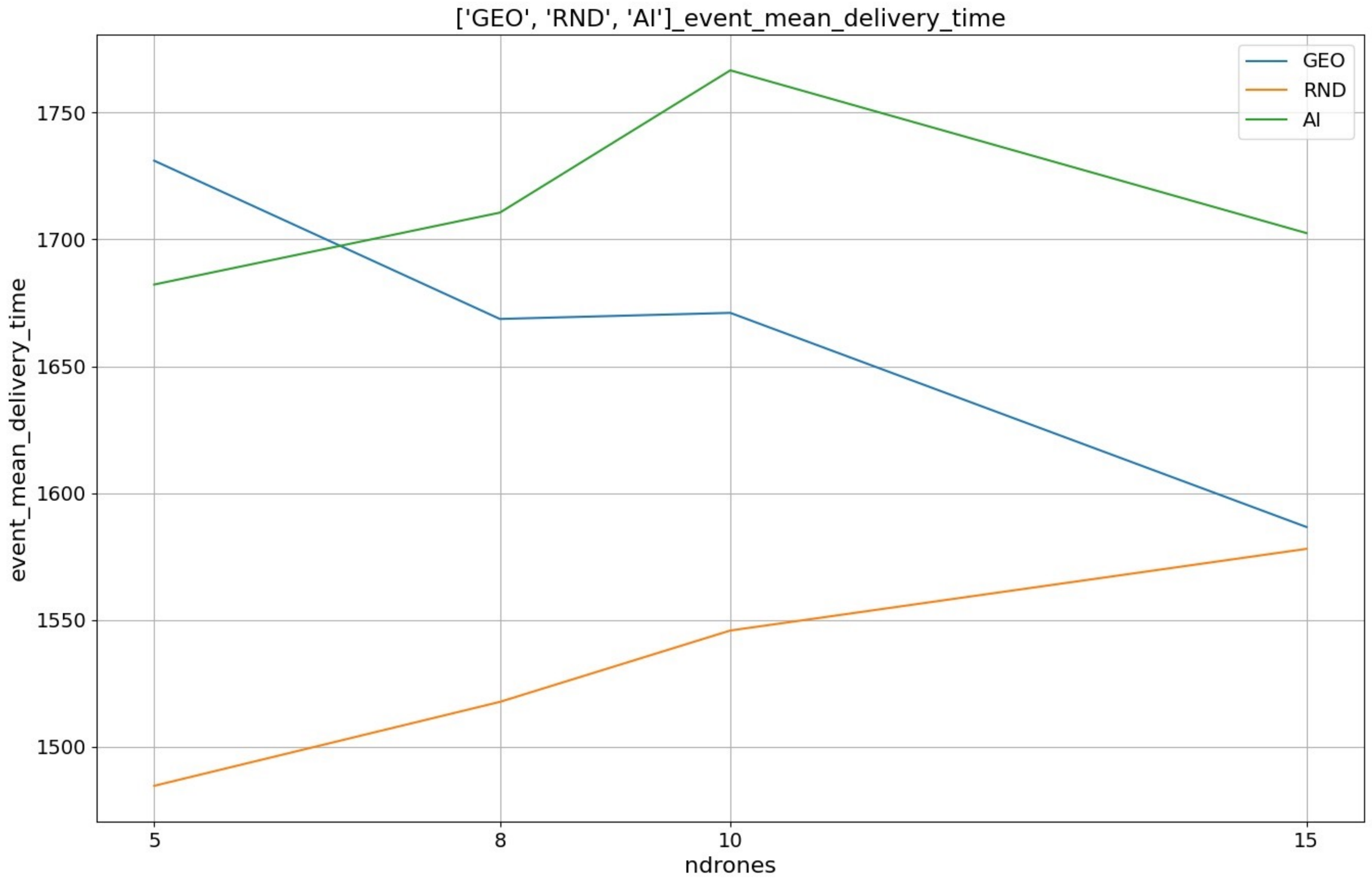
HOMEWORK 1 – RESULTS



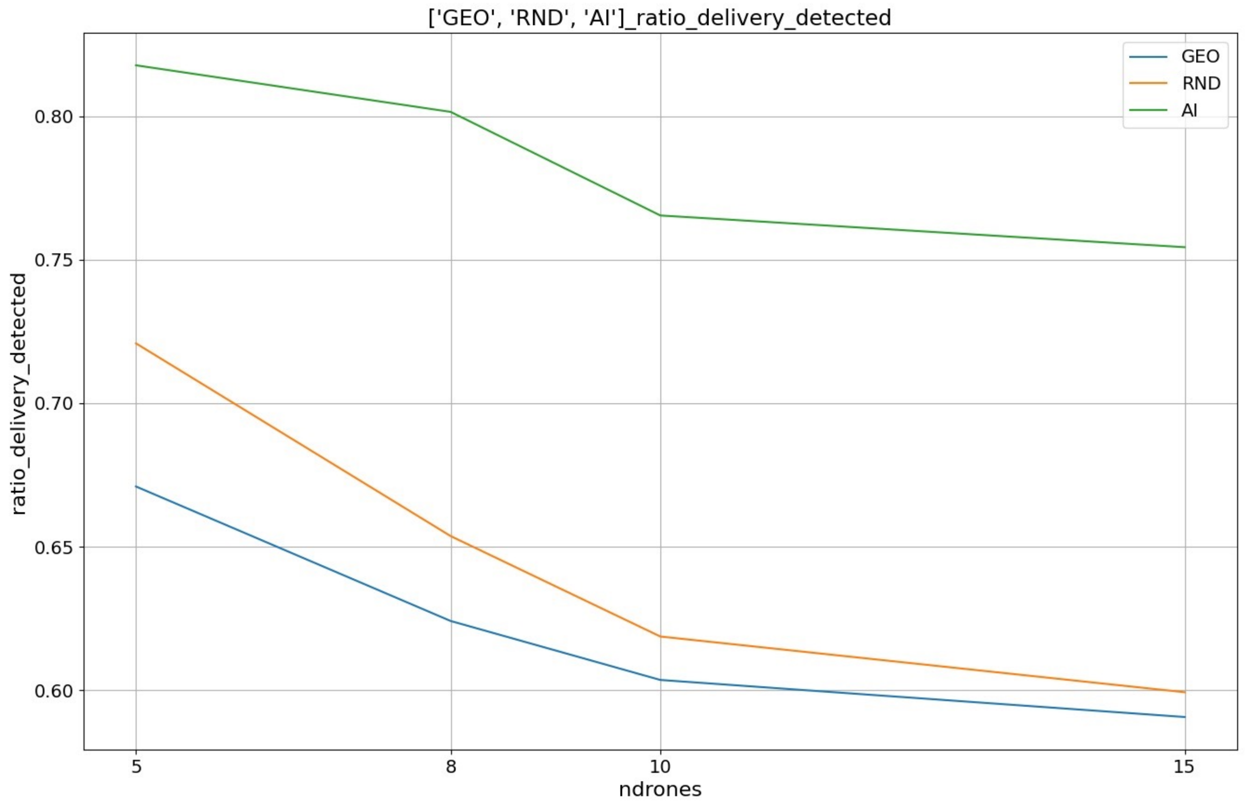
HOMEWORK 1 – RESULTS

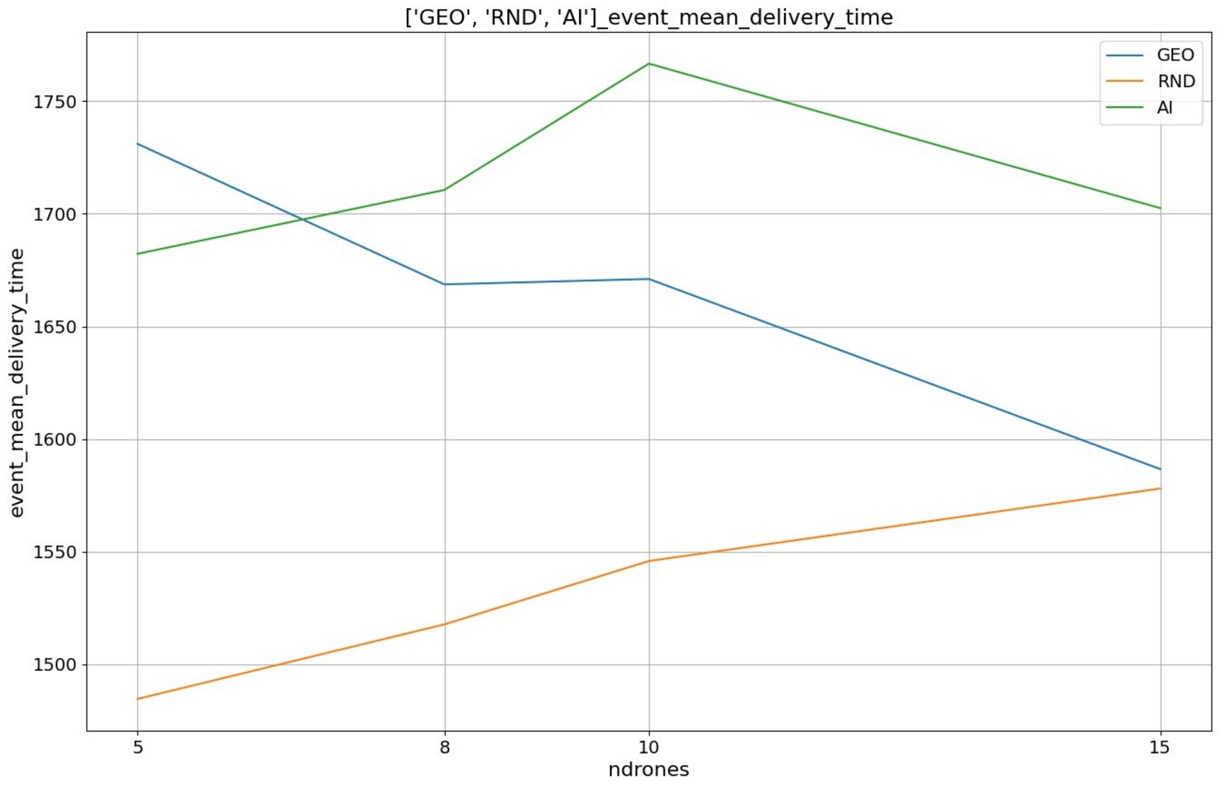


HOMEWORK 1 – RESULTS



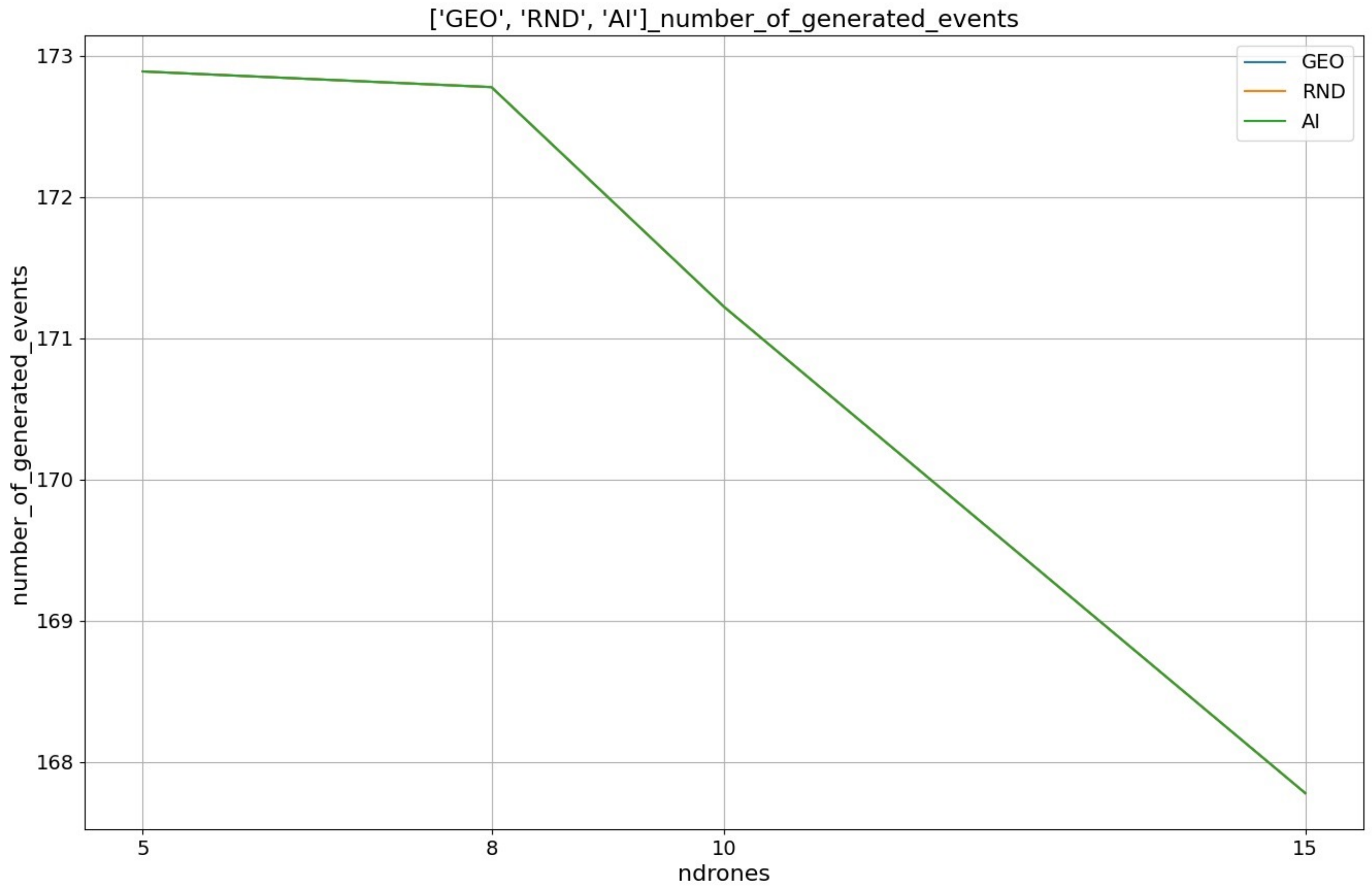
HOMEWORK 1 – RESULTS





HOMEWORK 2 – RESULTS

Only for check, this chart!



HOMEWORK 1!!

Deadline: 23:59:59 - 17/11/2021 (firm!)

How : in group of **3**

Submission: report + code

Score: up to 31 (30 cum laude)

Interaction lecture: 12:00 - 15/11/2021

FAQ: Common questions will be answered on google drive, link on the homework post

Evaluation: approach; report; algorithm score (% of delivered packets).

Challenge: rank of top 10 algorithms.

Note: Convergence time is really important here!

HOMEWORK 1 - GRADE

Homework 1: up to 31 (30 cum laude)

If grade ≥ 18 then 2 options:

1. **Stop here:** final grade is scaled to 18 (oral discussion is required[1])
2. **Homework 2** (see next slide)

Note:

[1] – grade can eventually increase up to 21

Suspected cases of plagiarism will be contacted to validate (or invalidate) the overall exam.

HOMEWORK 2 - GRADE

Homework 2: up to 31 (30 cum laude)

Final homeworks grade: (grade homework 1 + grade homework 2) / 2

If final homeworks grade ≥ 18 then 2 options:

1. **Stop here**: final grade is round(Final homework grade / 30 * 24) - (oral discussion is required[1])
2. **Further integration** (exam / project)

Note:

[1] – grade can eventually increase up to 27.

Suspected cases of plagiarism will be contacted to validate (or invalidate) the overall exam.

HOMEWORK 1 - SUBMISSION

How:

- **email** (**subject=**[Autonomous Networking - A.A. 2021-2022] – HMW1)
- **Classroom**

Format:

A zip called **studentid1_surnmane1_ studentid2_surnmane2_studentid3_surnmane3.zip** with:

- a brief report, at most 1000 words (images, biography and final notes are not counted). The final notes should clarify which part was mainly done by whom (50 words for each student of the team). The final part is not included in the 1000 words, then max 1150 words.
- A unique src file with the algorithm. Create a new name for your proposal, "algorithmname" and add the first "studentid1" at the end of the name : "algorithmname_studentid1" .

E.g., group of students made by: "Black - id: 999" and "Donald – id: 01".

They create a new algorithm called "X_RP", then, the delivery will be:
Zip: 999_black_01_donald.zip
Inside the zip:
- ***999_black_01_donald_report.txt***
- ***999_black_01_donald_routing_protocol.py*** (which contains the algorithm class called "**X_RP_999**")

Currently the file algorithm is called "ai_routing.py" and the algorithm is called AIRouting.
Submit <u>only</u> this file but <u>change</u> the file name and the algorithm name (see example here).

HOMEWORK 1 - REPORT

The report should include :

- Main approach and used techniques.
- Analysis of the performance and convergence.
- All the need additional libraries!!

Possible ML/AI libraries to use (<u>if needed, not mandatory</u>):

- Sklearn
- Keras
- Tensorflow

Please use only recent versions. In case of doubts write us an email. Disable any GPU usage before submit the homework.

If your model requires an intensive training phase, send to us also the persistent model to use.

HOMEWORK 1 – END

Possible write questions here:

https://docs.google.com/spreadsheets/d/1PbrruWdEf2w3eAmIYkXpNnypV191EFsxh0FTPhOXIHQ/edit?usp=sharing

You can also send to me your question with an email, subject ***'[Autonomous Networking - A.A. 2021-2022] – question'***

CONTACTS

Andrea Coletta:

coletta@di.uniroma1.it