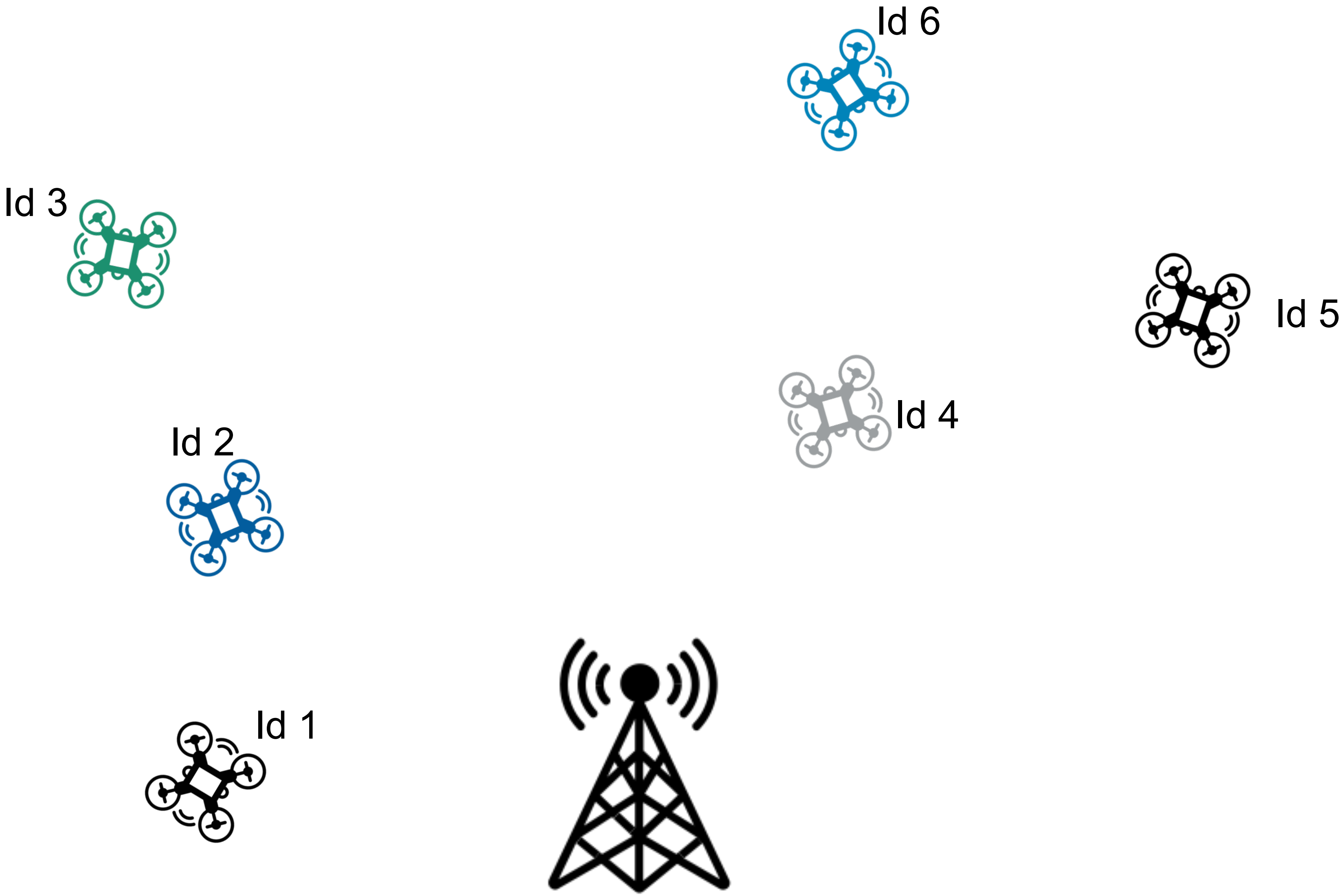
DroNET Simulator

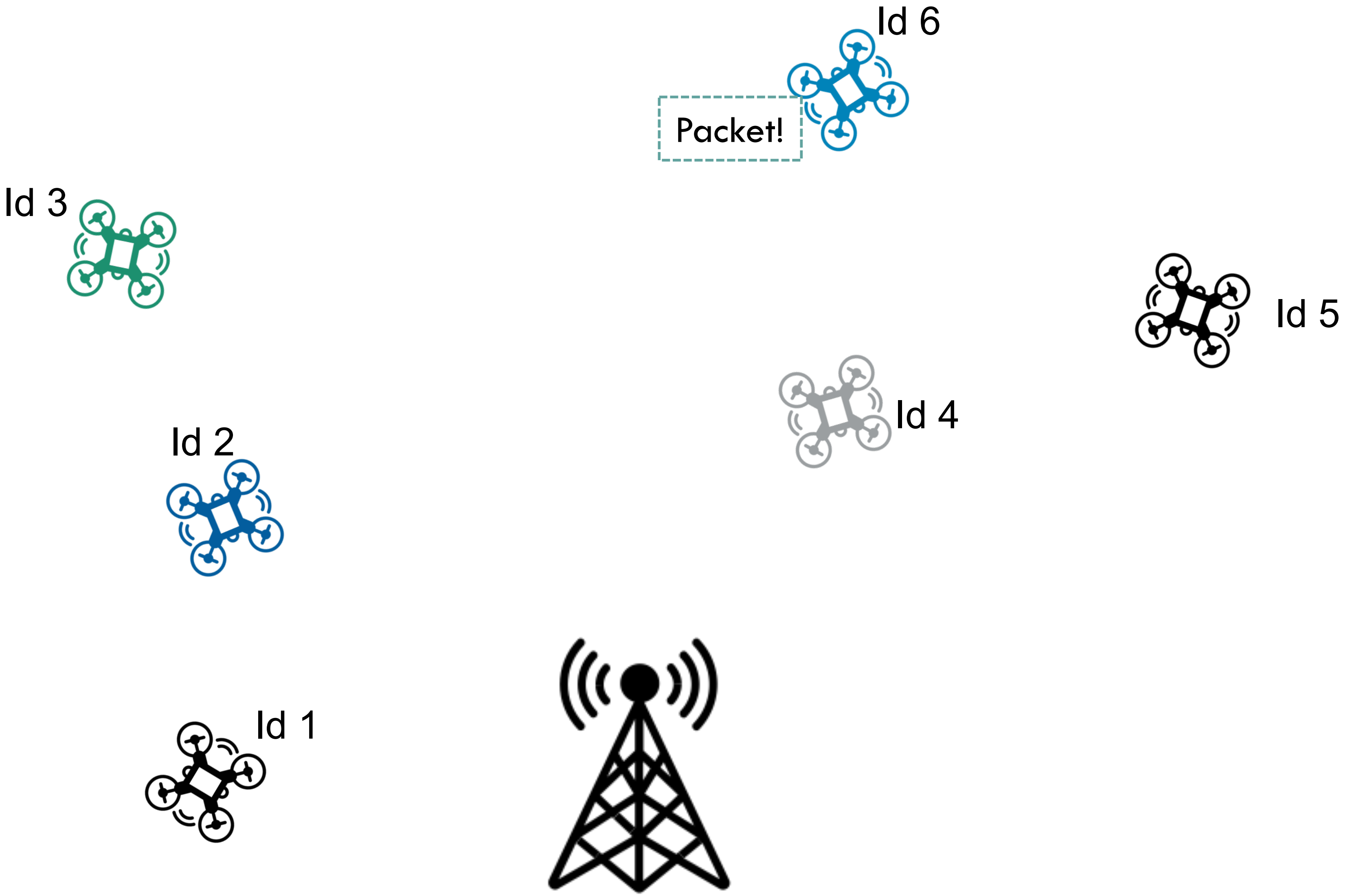
Course: *Autonomous Networking - Prof. Gaia Maselli (A.A. 2020-2021)*
Speaker: *Dr. Andrea Coletta - 15-12-2020*

SAPIENZA
UNIVERSITÀ DI ROMA

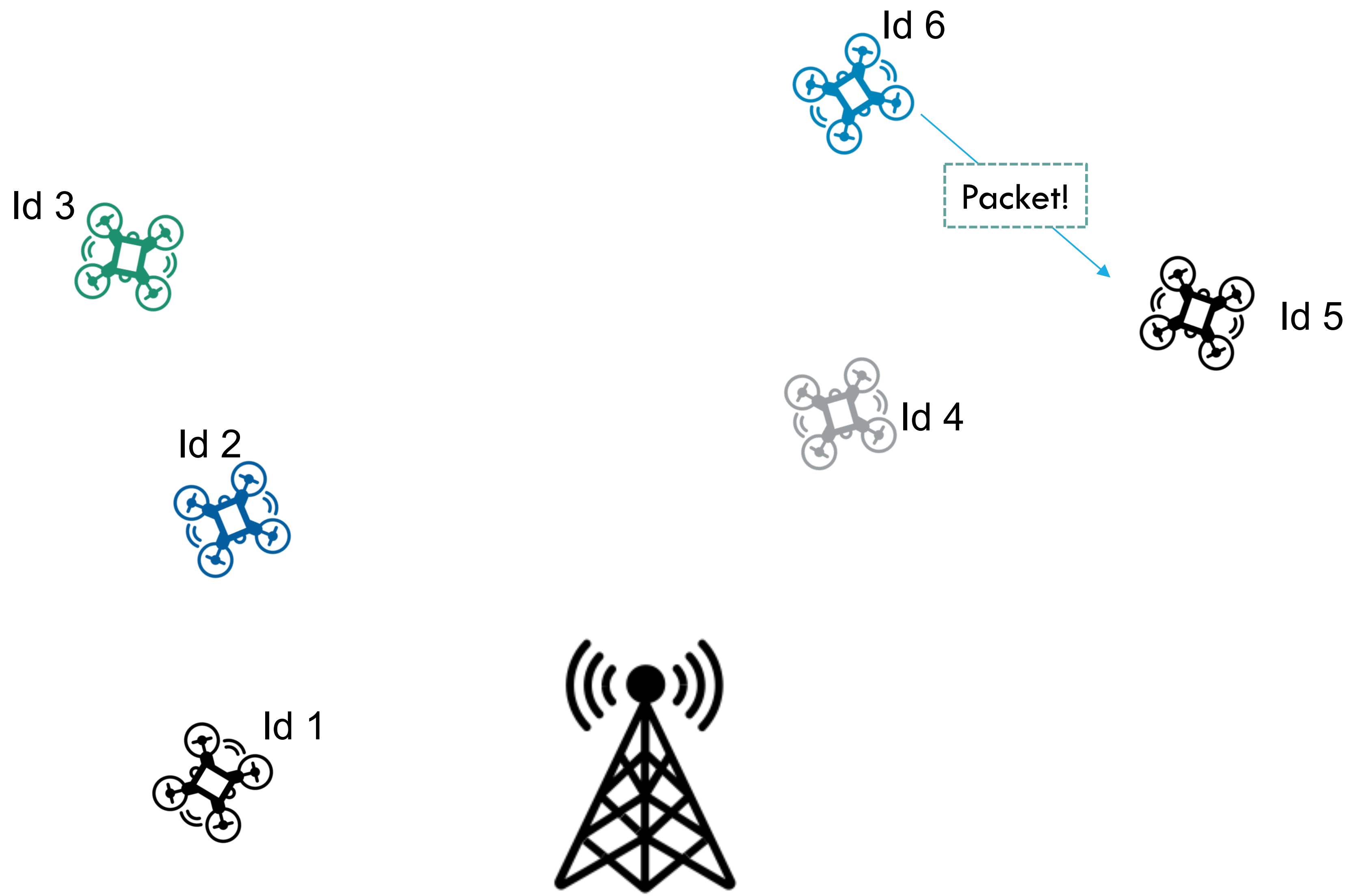
HOMEWORK 3 (FINAL) - ROUTING



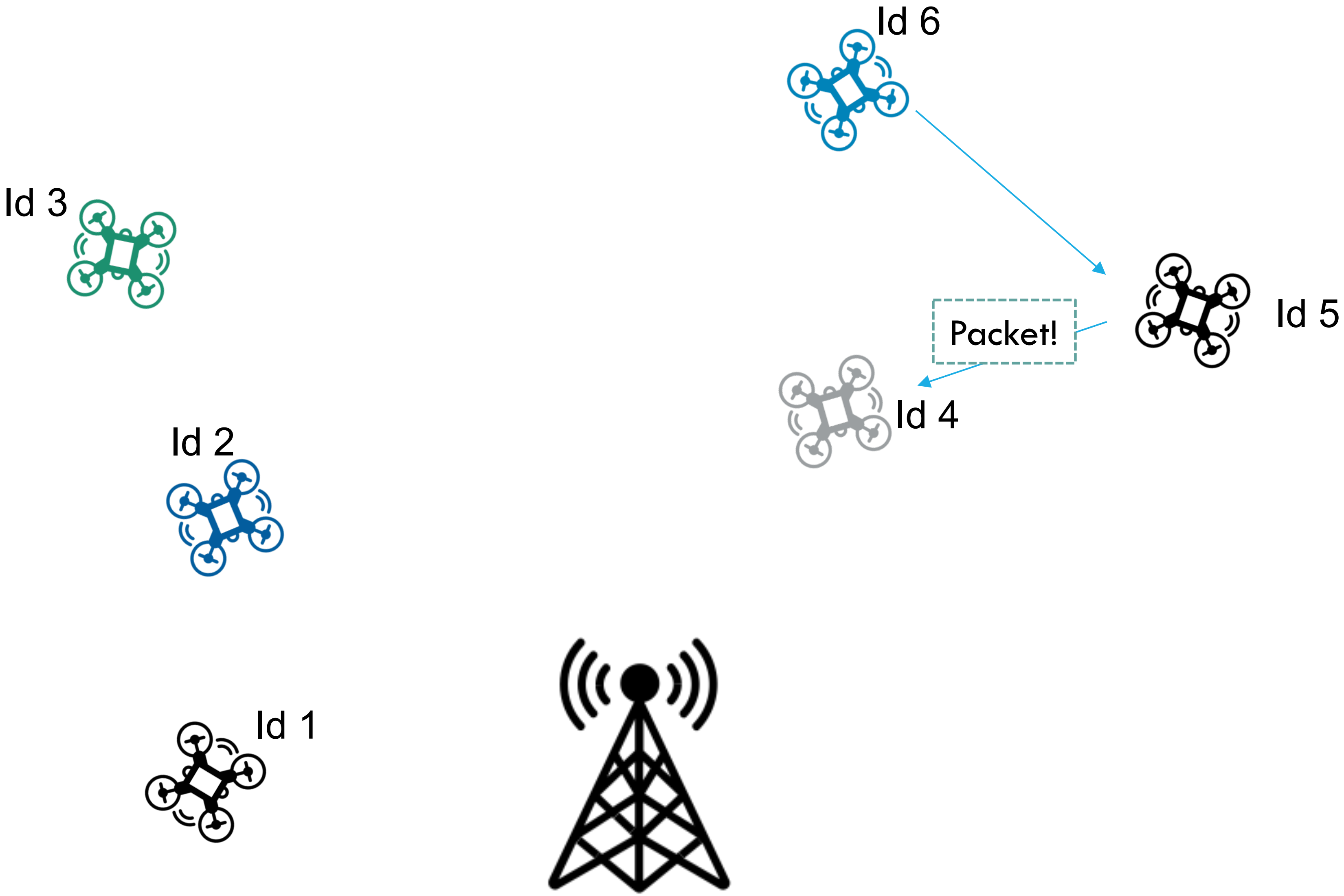
HOMEWORK 3 - ROUTING



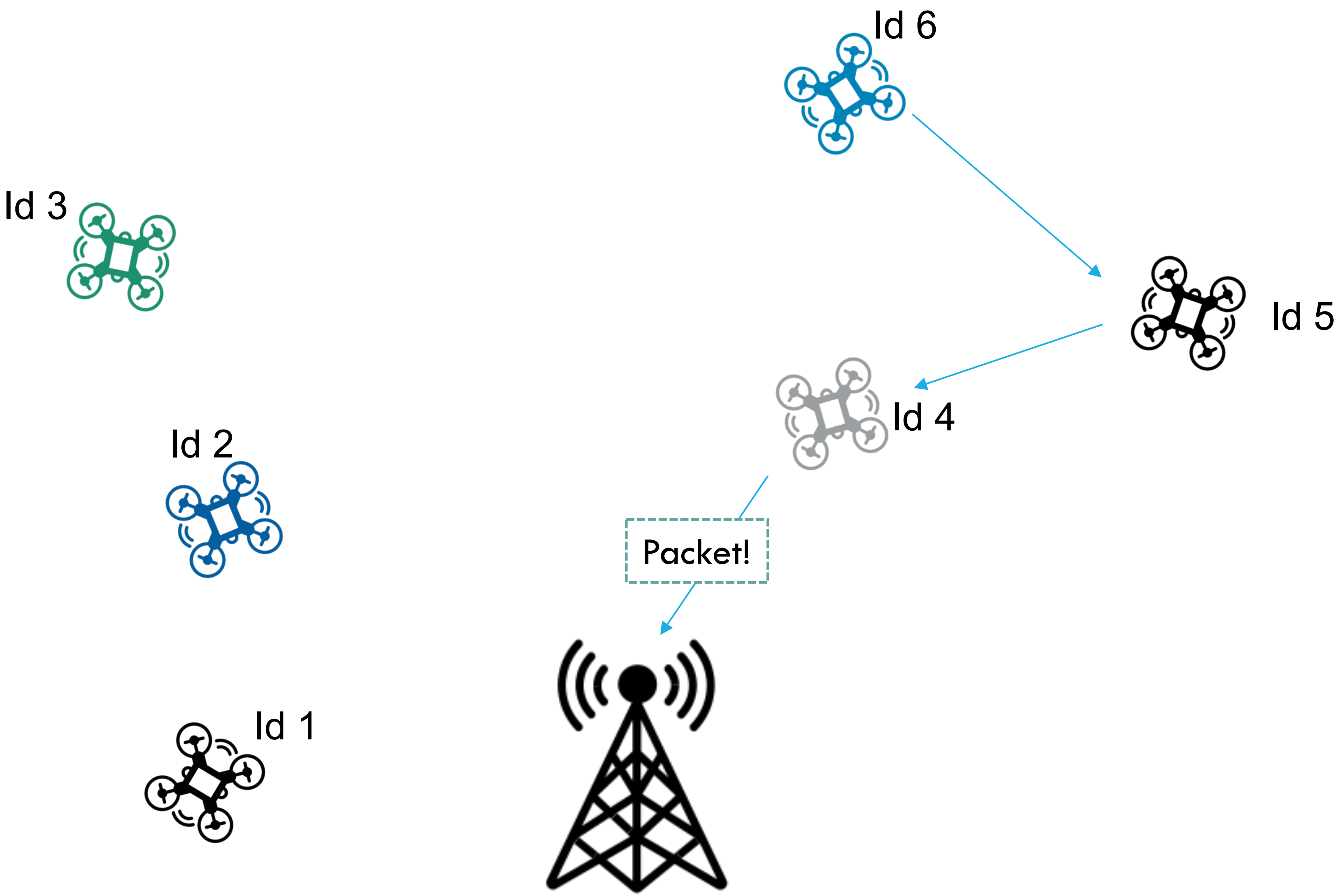
HOMEWORK 3 - ROUTING



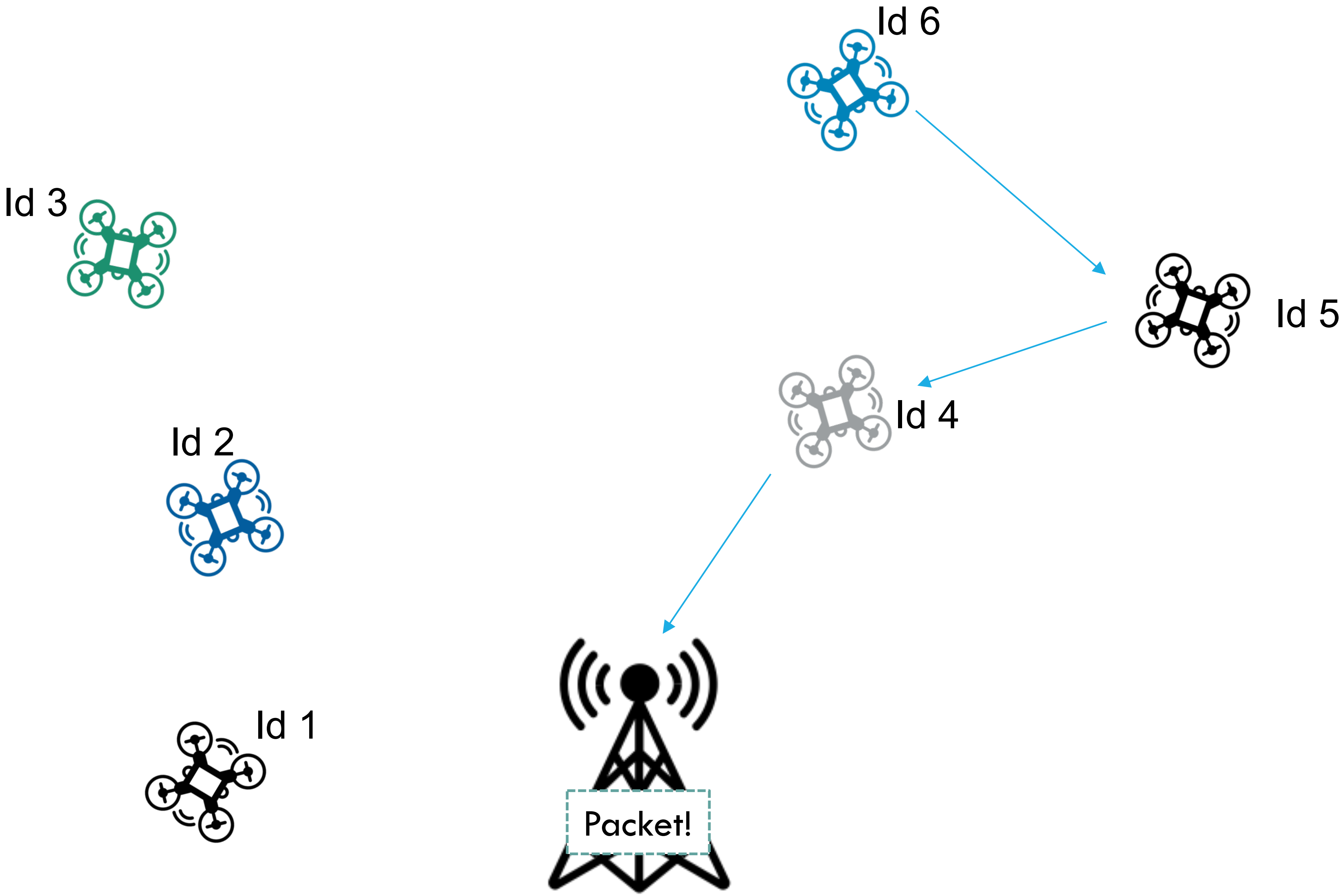
HOMEWORK 3 - ROUTING



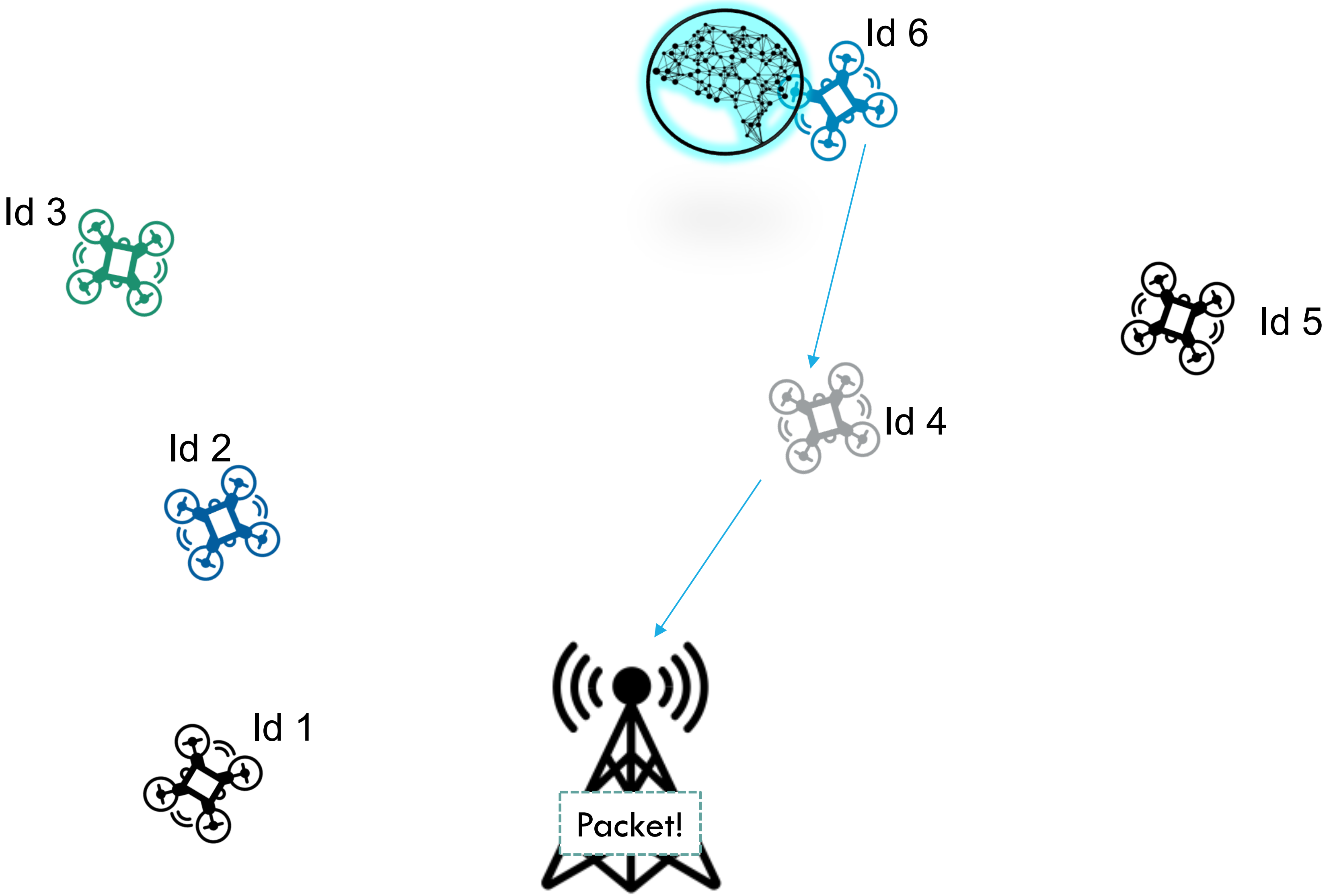
HOMEWORK 3 - ROUTING



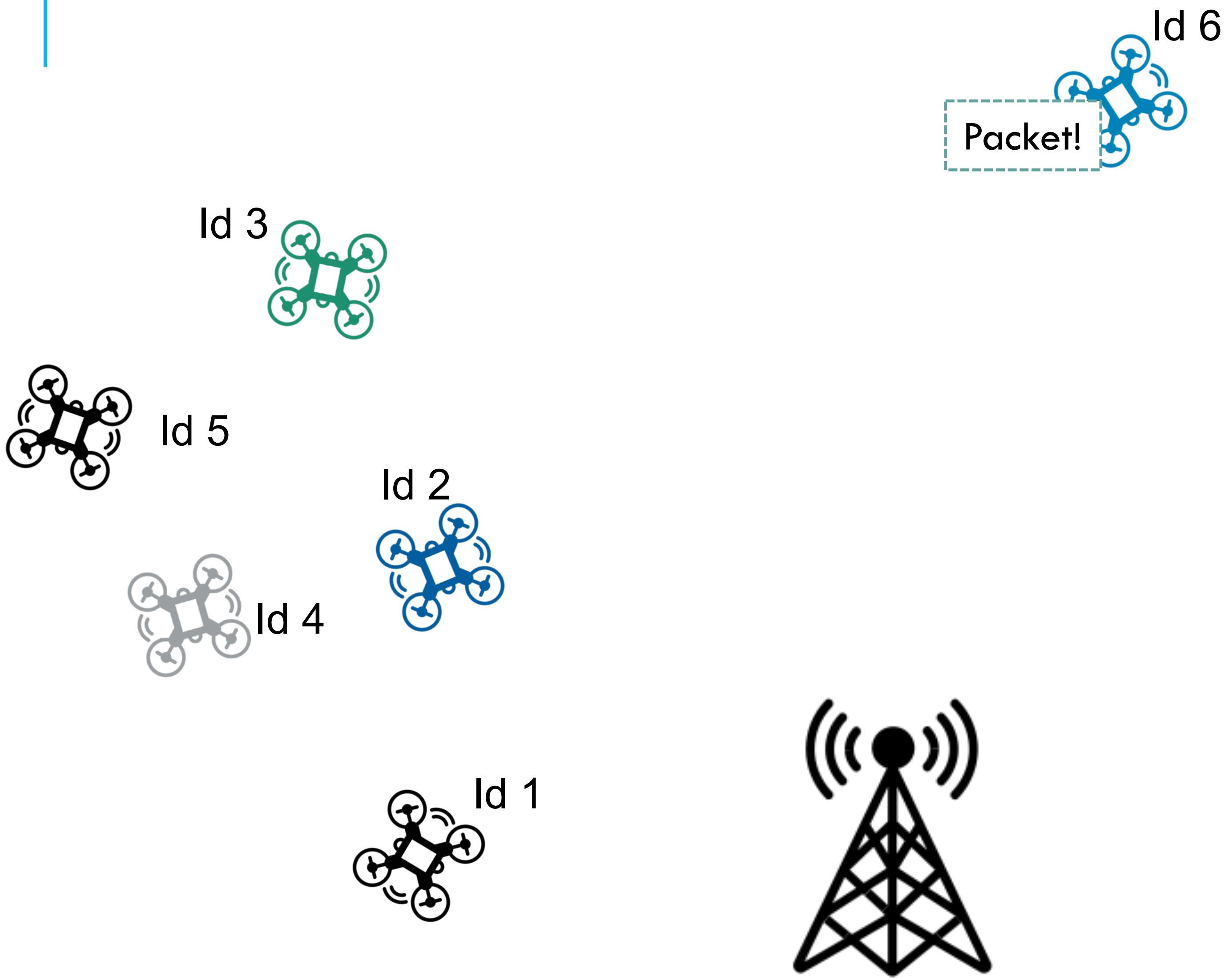
HOMEWORK 3 - ROUTING



HOMEWORK 3 – REINFORCEMENT LEARNING ROUTING

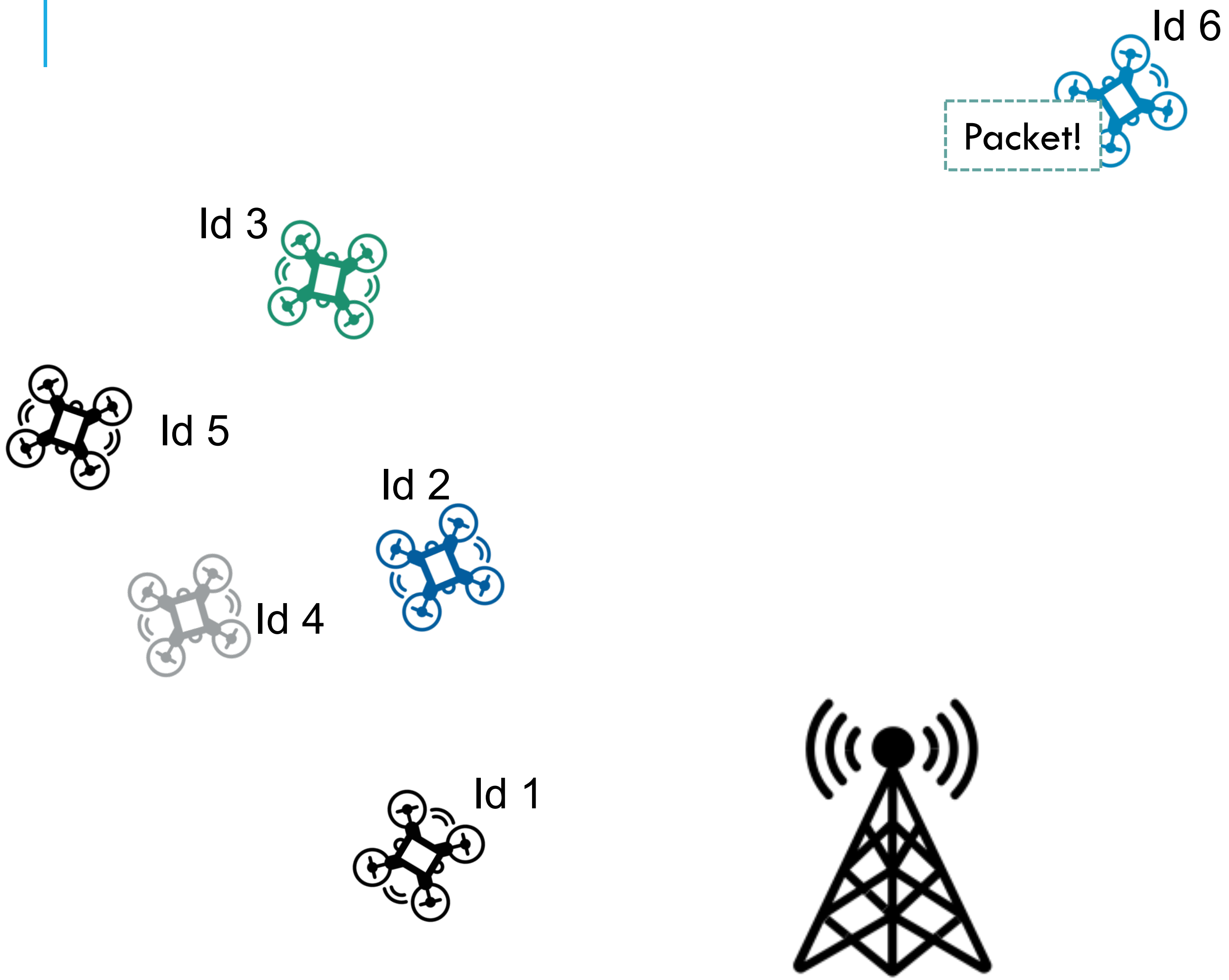


HOMEWORK 3 - ROUTING



What can I do to deliver the packet?

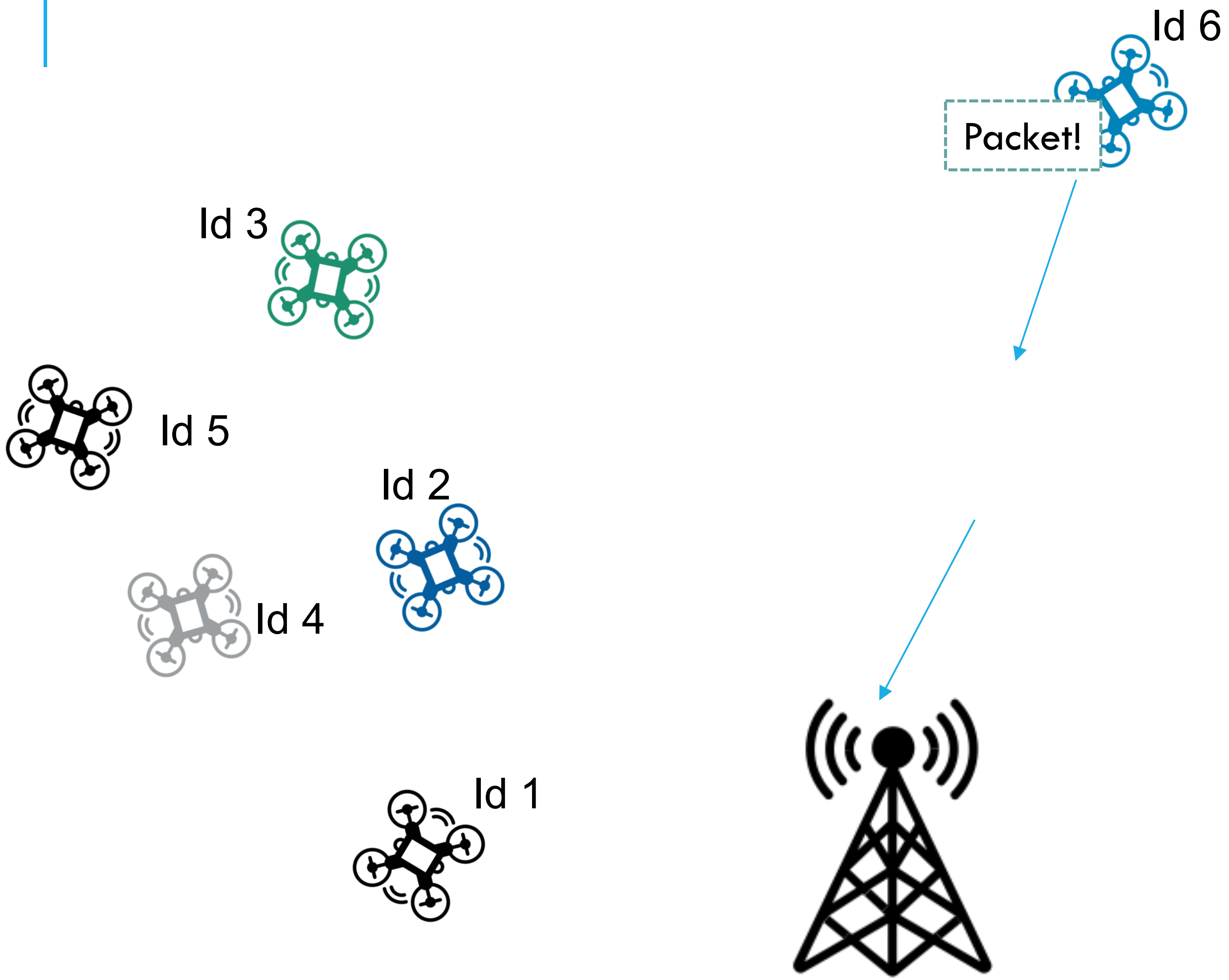
HOMEWORK 3 - ROUTING



What can I do to deliver the packet?

Physical delivery

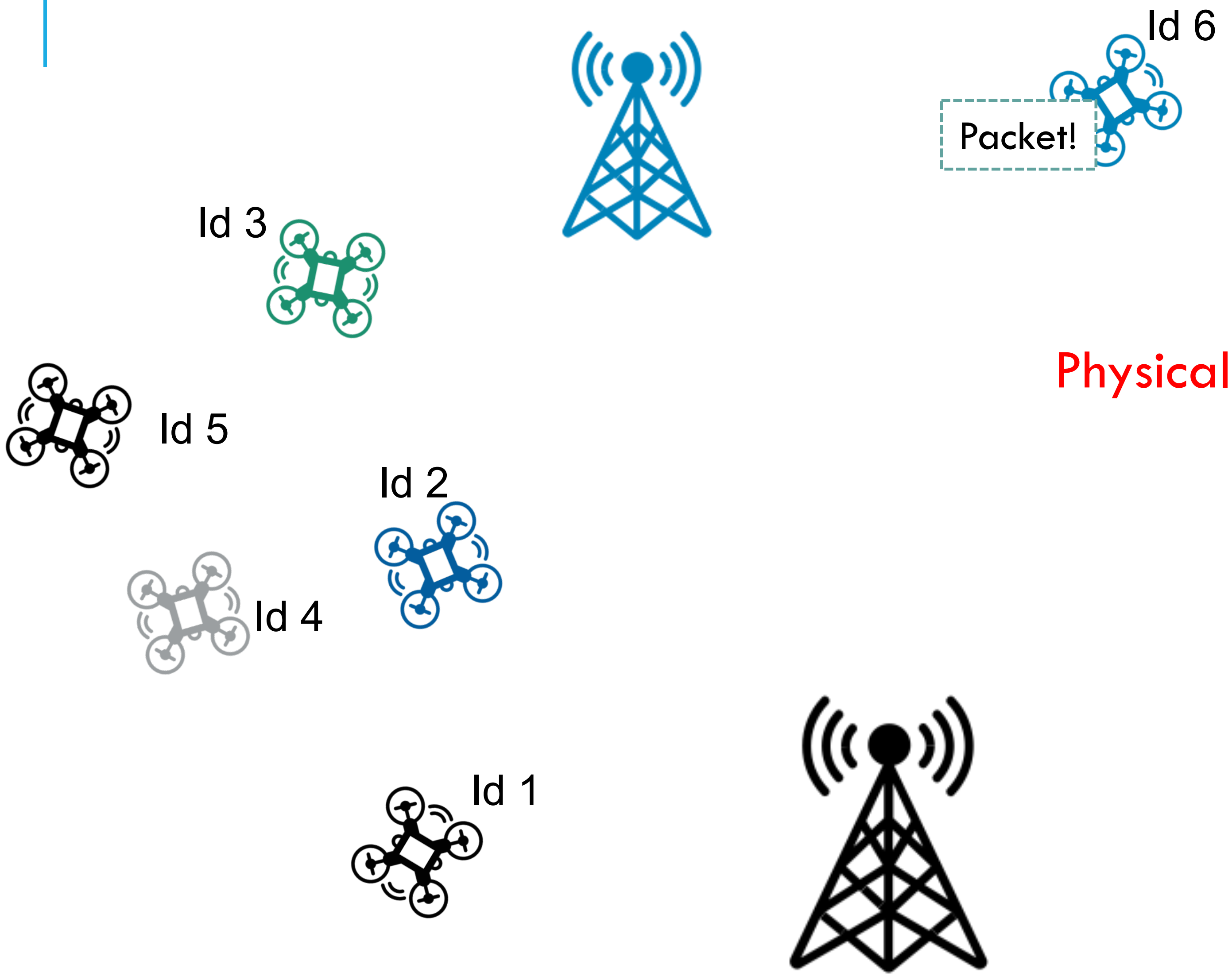
HOMEWORK 3 - ROUTING



What can I do to deliver the packet?

Wait for another drone

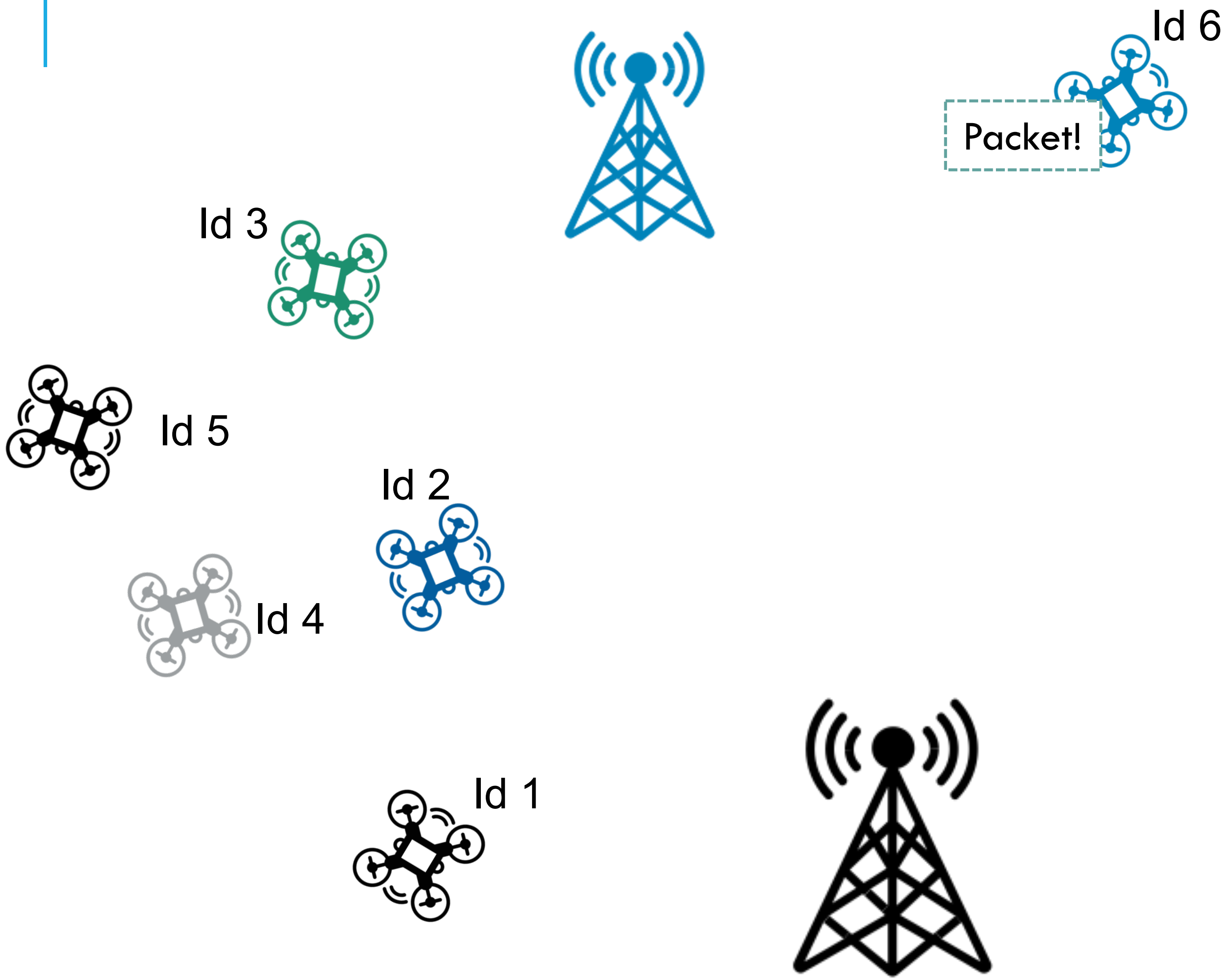
HOMEWORK 3 – ROUTING WITH 2 DEPOTS



What can I do to deliver the packet?

Physical delivery

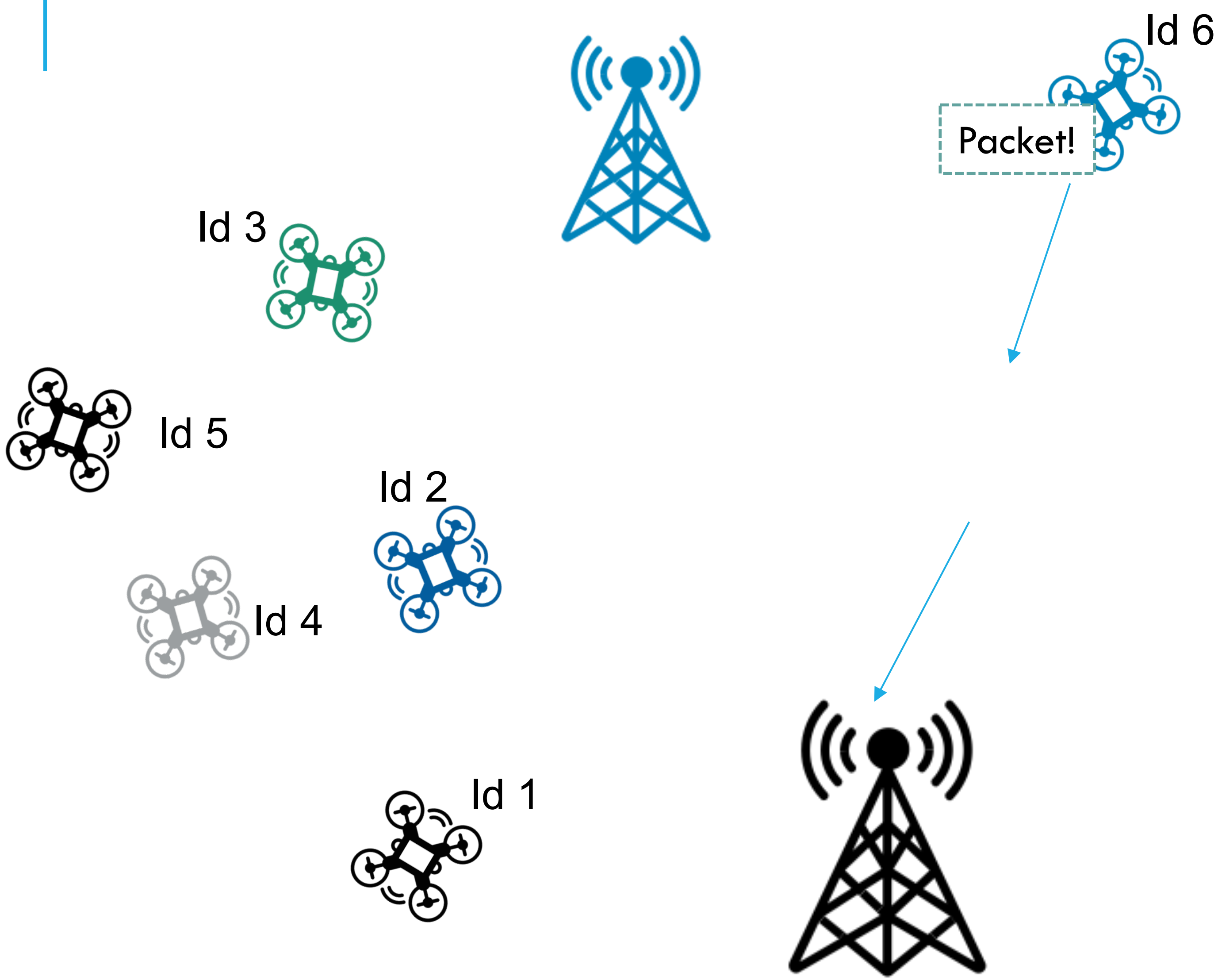
HOMEWORK 3 - ROUTING WITH 2 DEPOTS



What can I do to deliver the packet?

Physical delivery

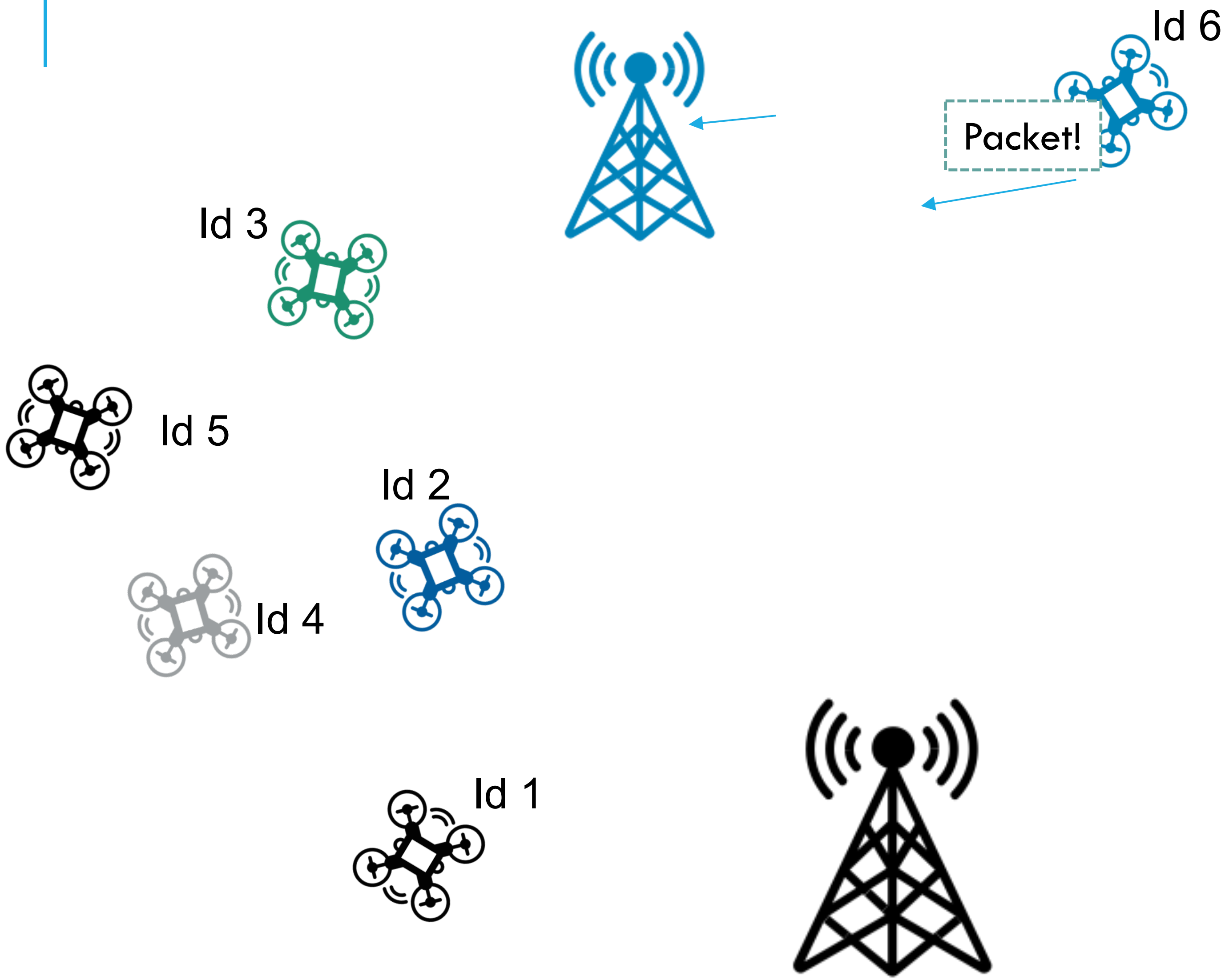
HOMEWORK 3 - ROUTING WITH 2 DEPOTS



What can I do to deliver the packet?

Wait for another drone

HOMEWORK 3 - ROUTING WITH 2 DEPOTS



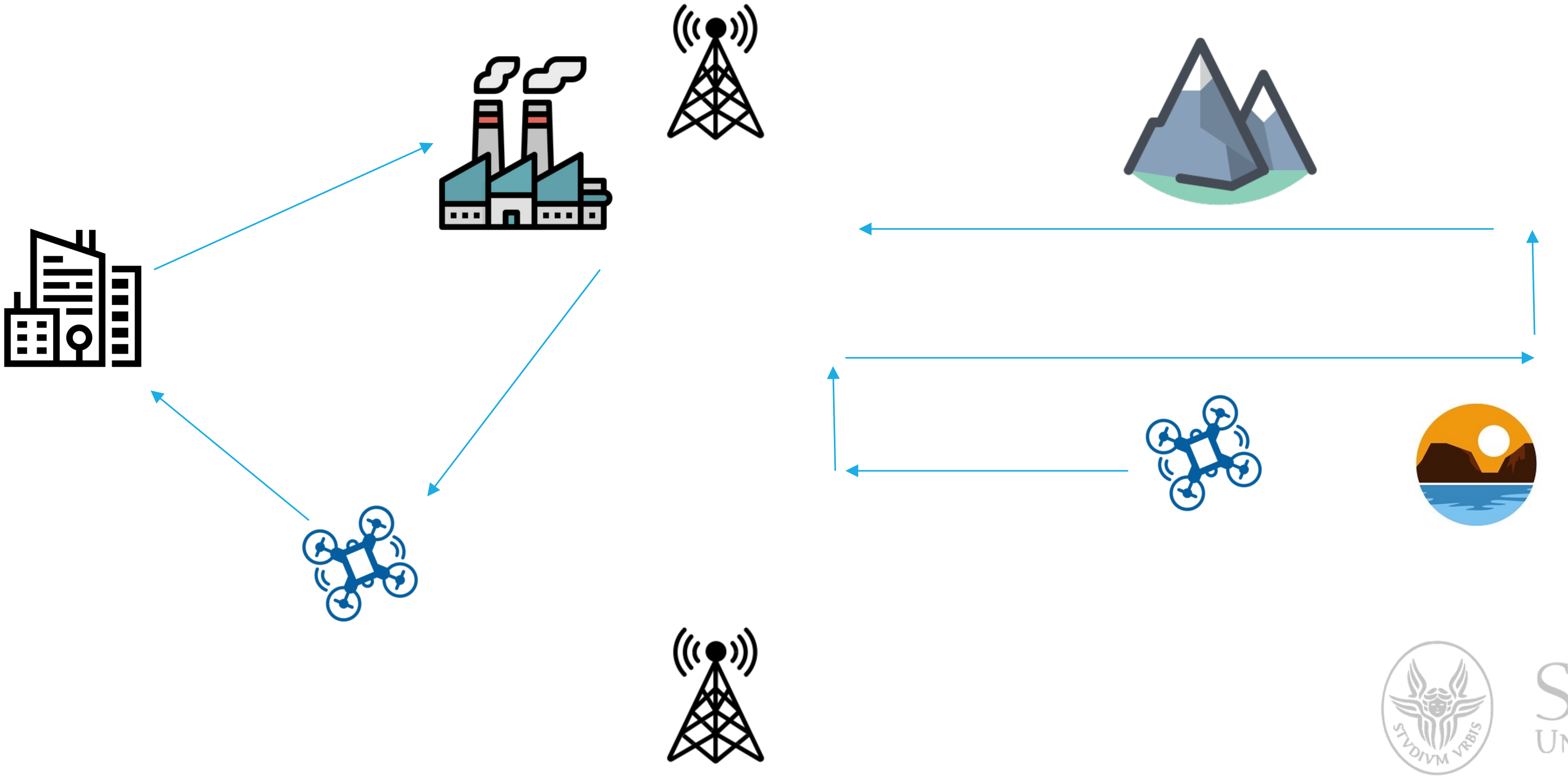
What can I do to deliver the packet?

Wait for another drone

HOMEWORK 3 – GENERAL SCENARIO

It's the SAME OF HOMEWORK 2! You can work on RANDOM OR SWEEP PATHS!

We have an area of interest to detect and monitor the area, no circle trajectories.

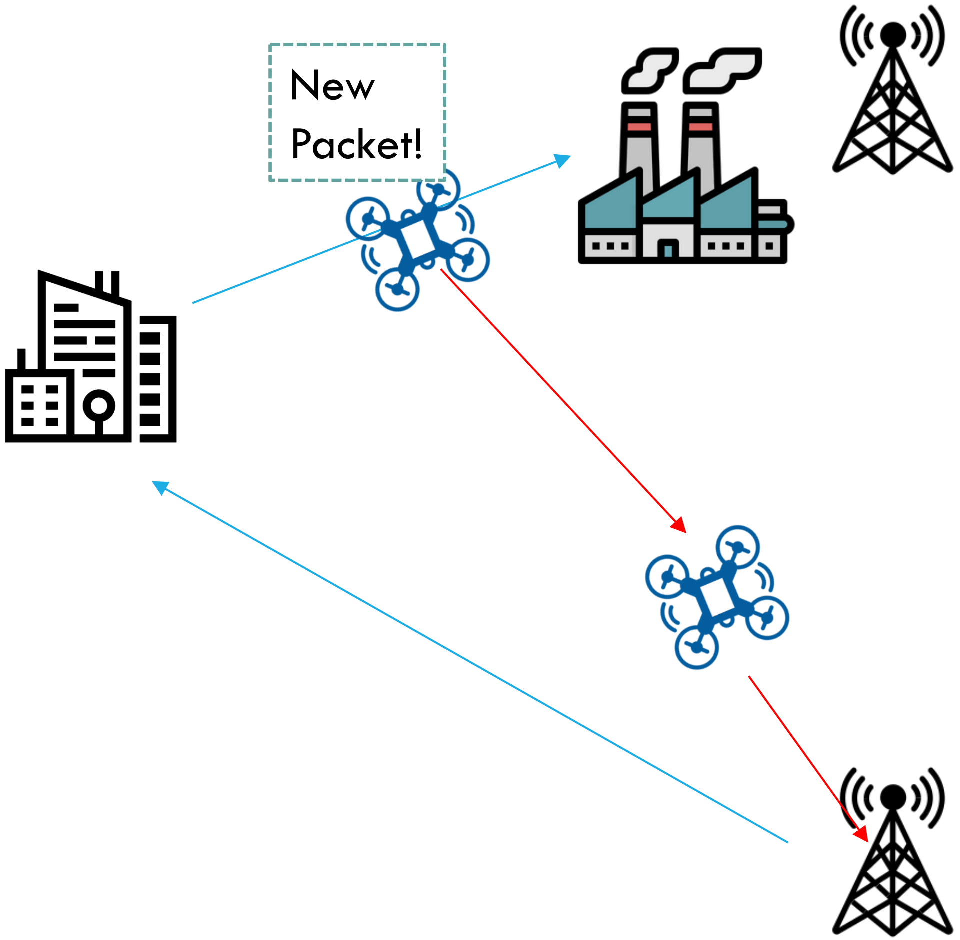


SAPIENZA
UNIVERSITÀ DI ROMA

HOMEWORK 3 – SCENARIO

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

We have a squad of N-drones (N >= 2), and NO ferries; and 2 DEPOTS



All the drones collect data!

Each drone may have a different speed.

During a physical delivery , you can't sense new data (you lost packets on the ground).

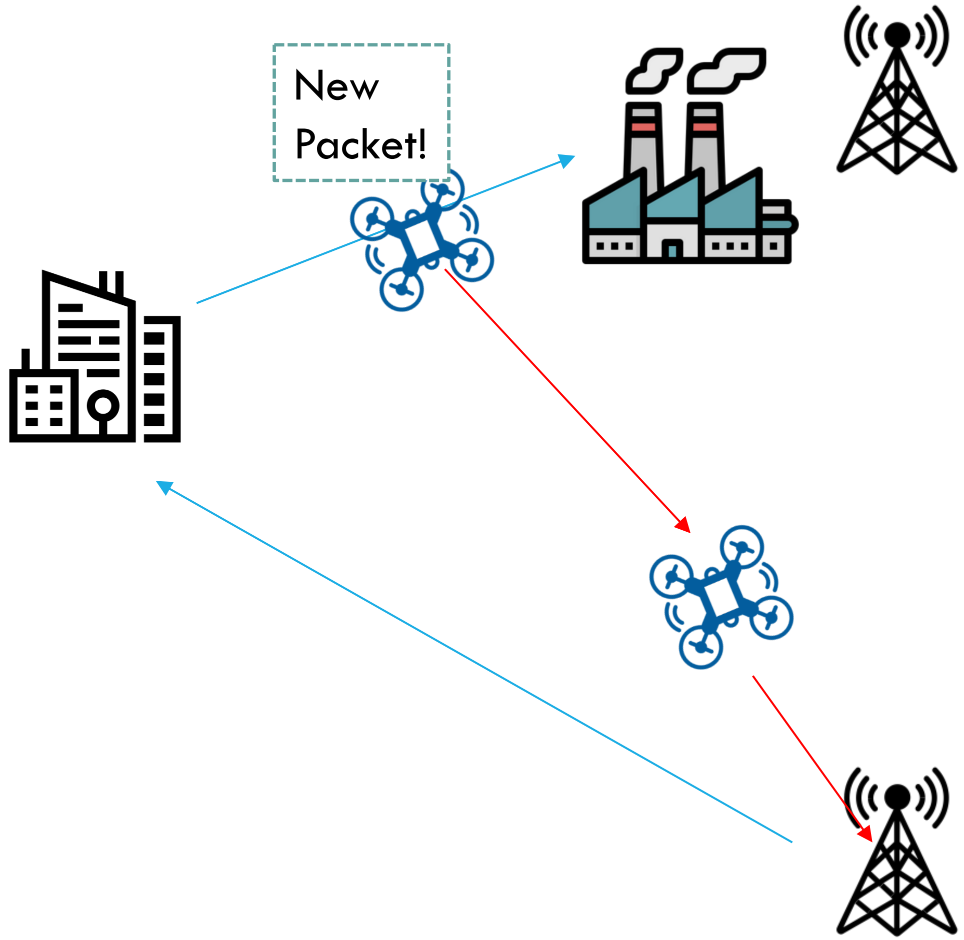
The drone routing can:

- Store the data and wait to arrive at one of depots
- Send the data to a neighbor drone, which may arrive at one of the depots before him.
- Move the drone physically to one of the depots

HOMEWORK 3 – SCENARIO

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

We have a squad of N-drones (N >= 2), and NO ferries; and 2 DEPOTS



GOAL:

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

To decide whether keep, send the packet, or move to one of the depots.

HOMEWORK 3 – ASSUMPTIONS

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

Use the hello packet as usually, and be aware that:

- The speed of each drone may be different
- The hello packet tells you if a drone is going to the depot, to physically deliver packets
- You can't assume to know your trajectories.
- We don't have anymore the energy residual !! We assume the mission is short respect to the drone batteries.
- Each drone has a unique "packet error rate", the radio onboard can be more or less powerfull depending on the drone, you don't know this info, but you can learn it.
- The packets have a limited time to live, now 4x respect hmw1.
- Depots have a bigger communication range (same of drones)

HOMEWORK 3 – GOAL

You have to delivery as much as possible packets to the depots as primary task.
As secondary task you have to reduce the latency of the packets, but also the energy spent to physical movement toward the depot!

We keep the score :

$$1.5 \cdot |expired\_packets| \cdot ttl + \sum_{pck \in delivered} delivery\_time$$

But we have two new chart about the time/energy spent to move toward the depot for a delivery.

- Time : seconds need to reach the depot, and come back to the mission
- Energy : time + fixed cost (around 4seconds) to reverse rotors and change direction.

HOMEWORK 3 – HOW

You can implement the Class below, using a Reinforcement Learning approach.
We suggest you start from homework 2 solution!!

```
class AIRouting(BASE_routing):
    def __init__(self, drone, simulator):...

    def feedback(self, drone, id_event, delay, outcome, depot_index=None):...

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


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

HOMEWORK 3 – 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!
```

Now you have 4 actions:

- MOVE to bottom depot (return -1)
- MOVE to top depot (return -2)
- KEEP packet (return None)
- SEND packet (return the drone)

Notice that, if you selected -1 or -2, you will move until your buffer is empty.
But you need still select a relay drone, if any, to empty your buffer.

HOMEWORK 3 – RUN MULTIPLE SIMULATIONS

As for the second homework, **or you can use "python3 src/experiments/run_exp.py"**

MGEO is a simple approach:

If Len(neighbors) == 0:

MOVE to closest depot;

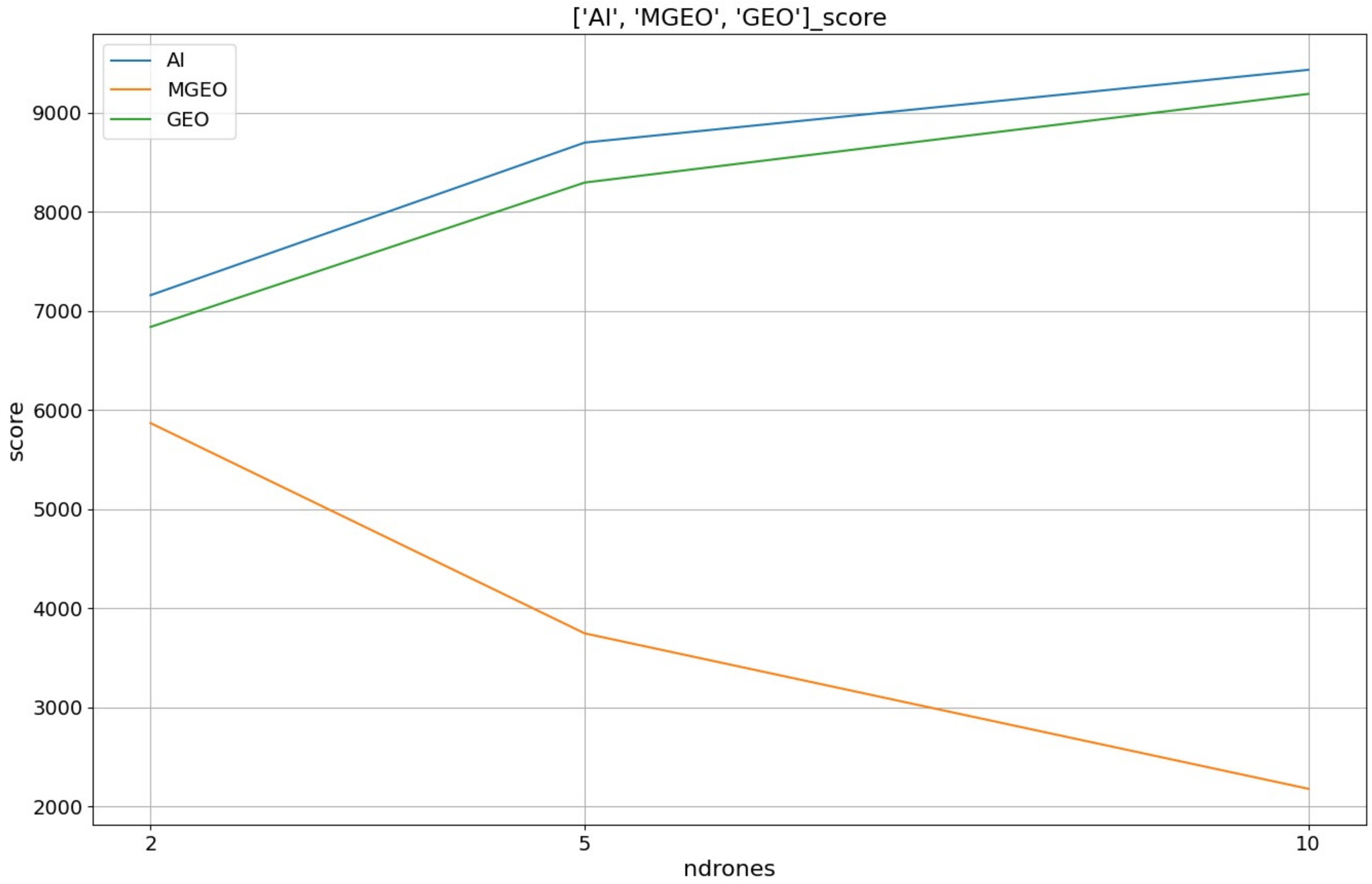
Else:

Geographic routing using the closest DEPOT

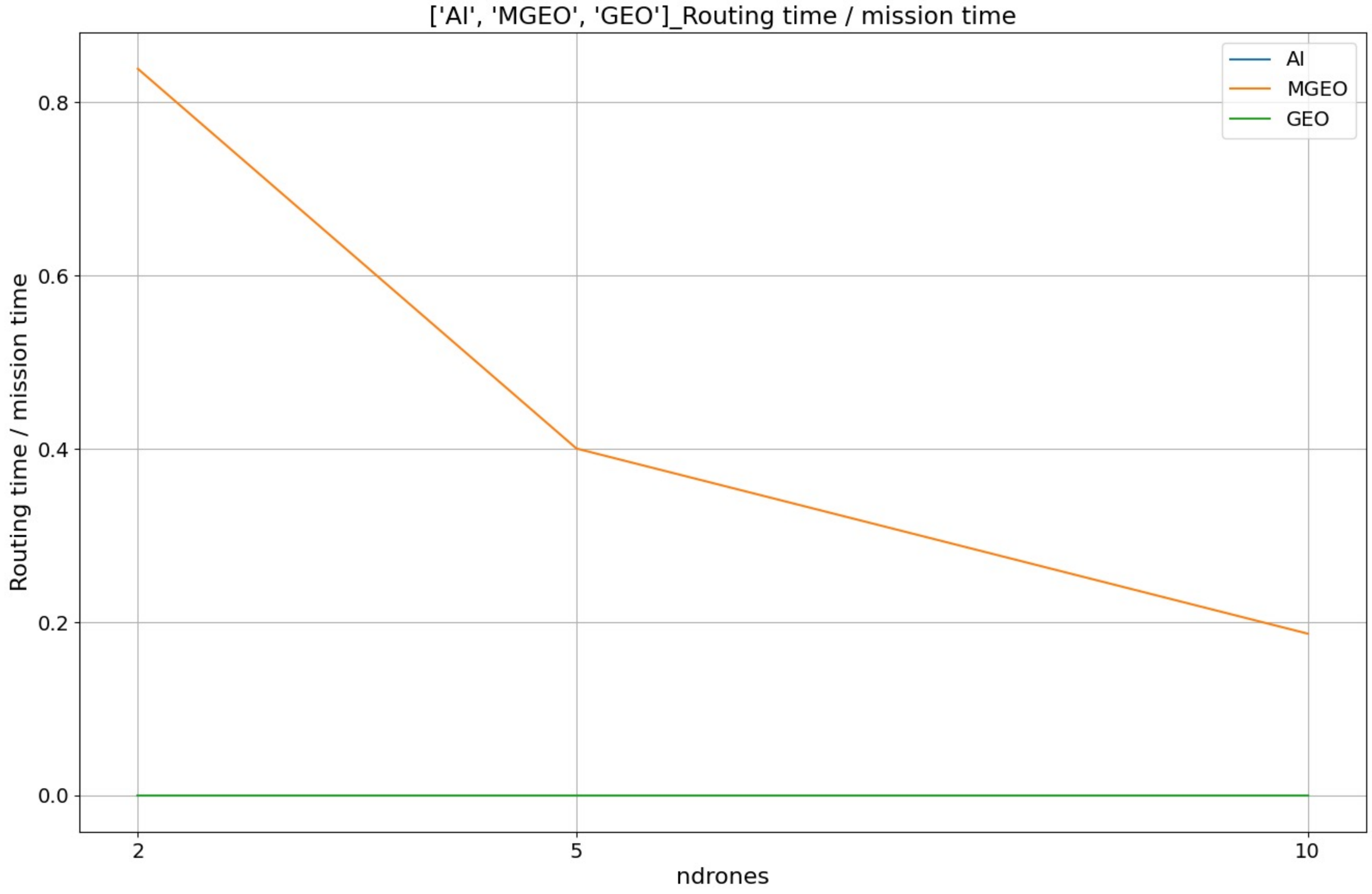
```
#test baselines
for nd in "2" "5" "10" "15" "20" "30" "40";
do
    for alg in "GEO" "RND" "AI" "MGEO";
    # if you experienced too much time to run experiments, remove "GEO" and "RND"
    do
        echo "run: ${alg} - ndrones ${nd} "
        python3 -m src.experiments.experiment_ndrones -nd ${nd} -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 2 -nd 5 -nd 10 -nd 15 -nd 20 -nd 30 -nd 40 -i_s 1
```

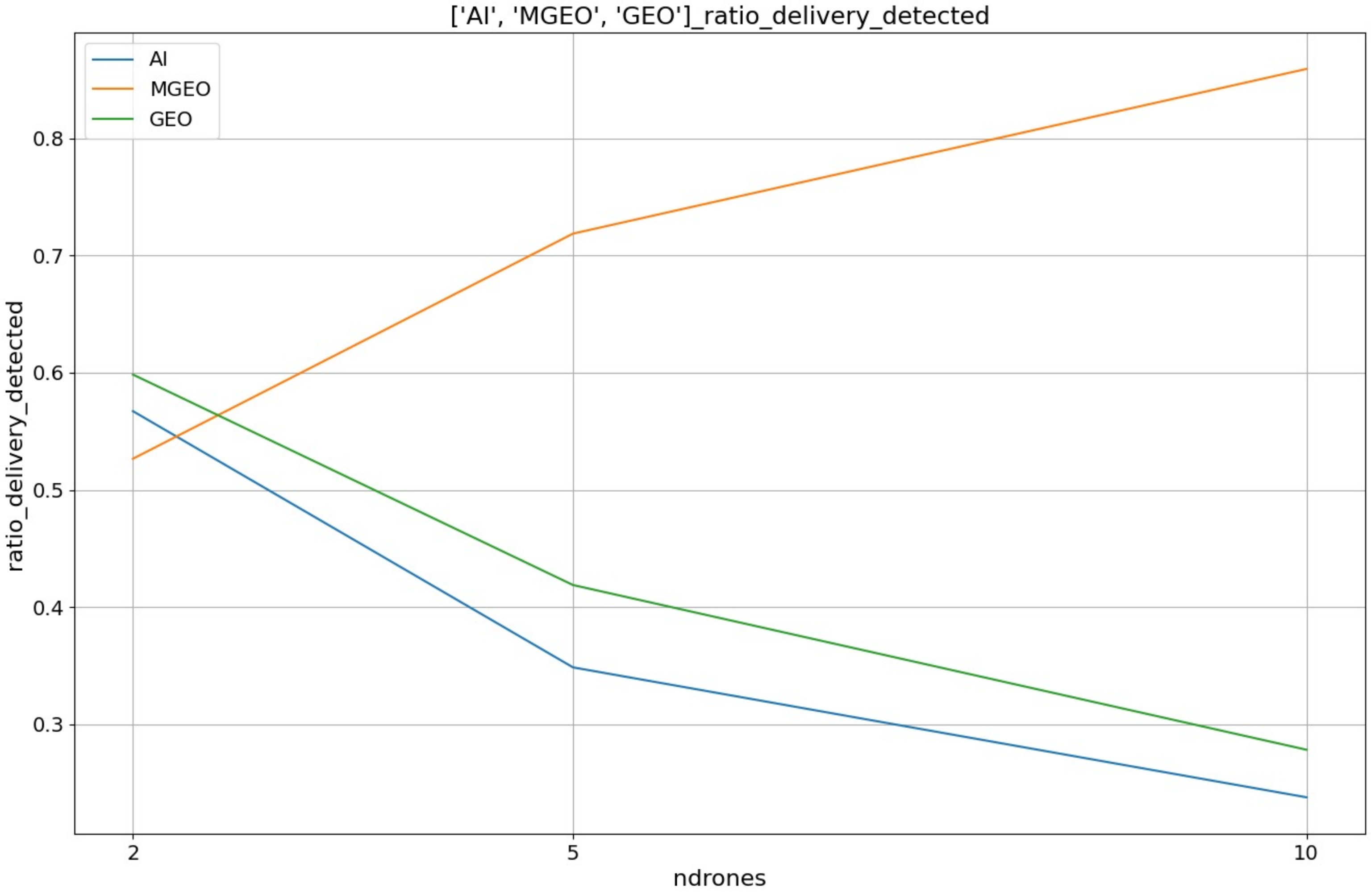
HOMEWORK 3 – RESULTS



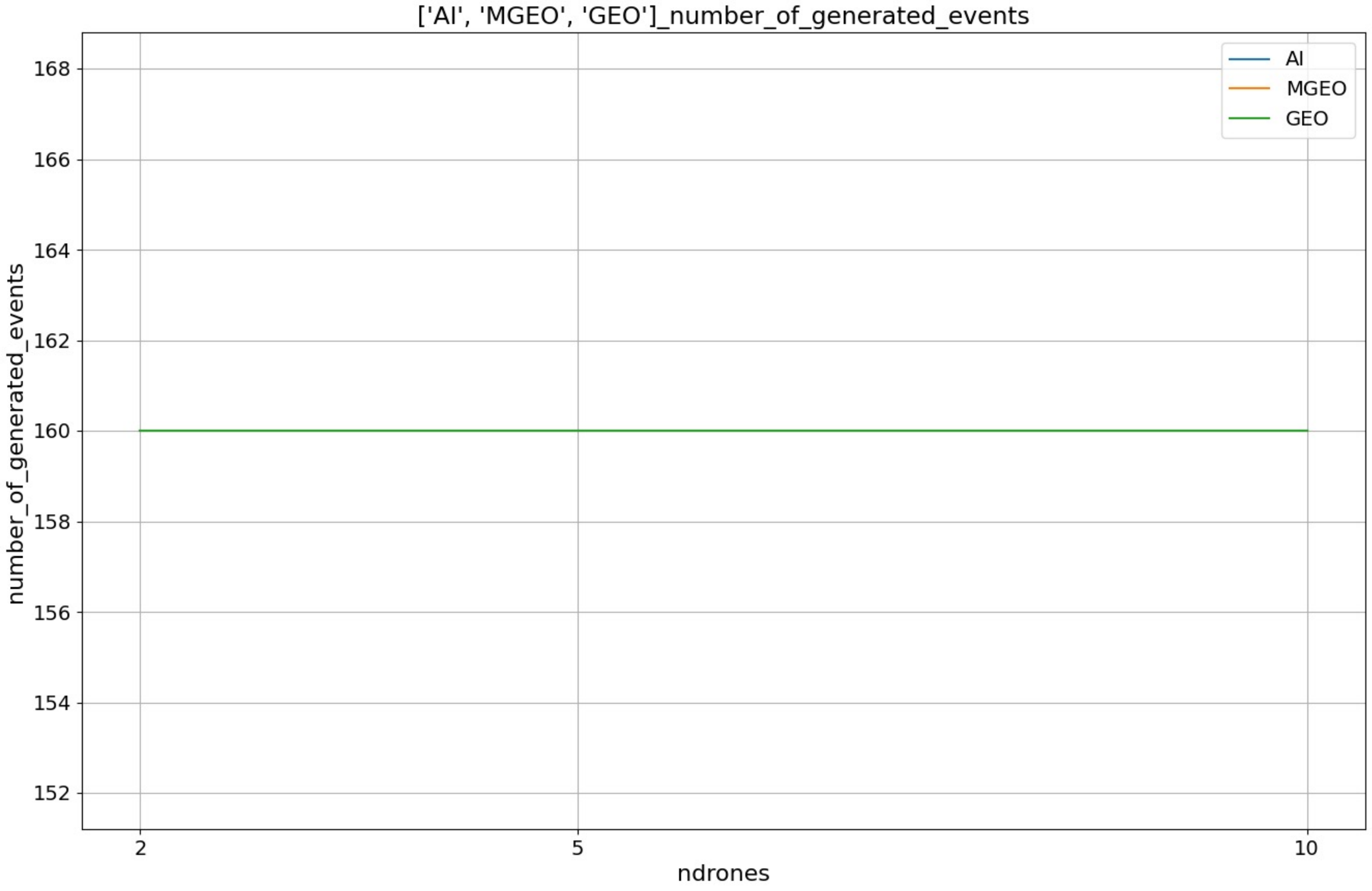
HOMEWORK 3 – RESULTS



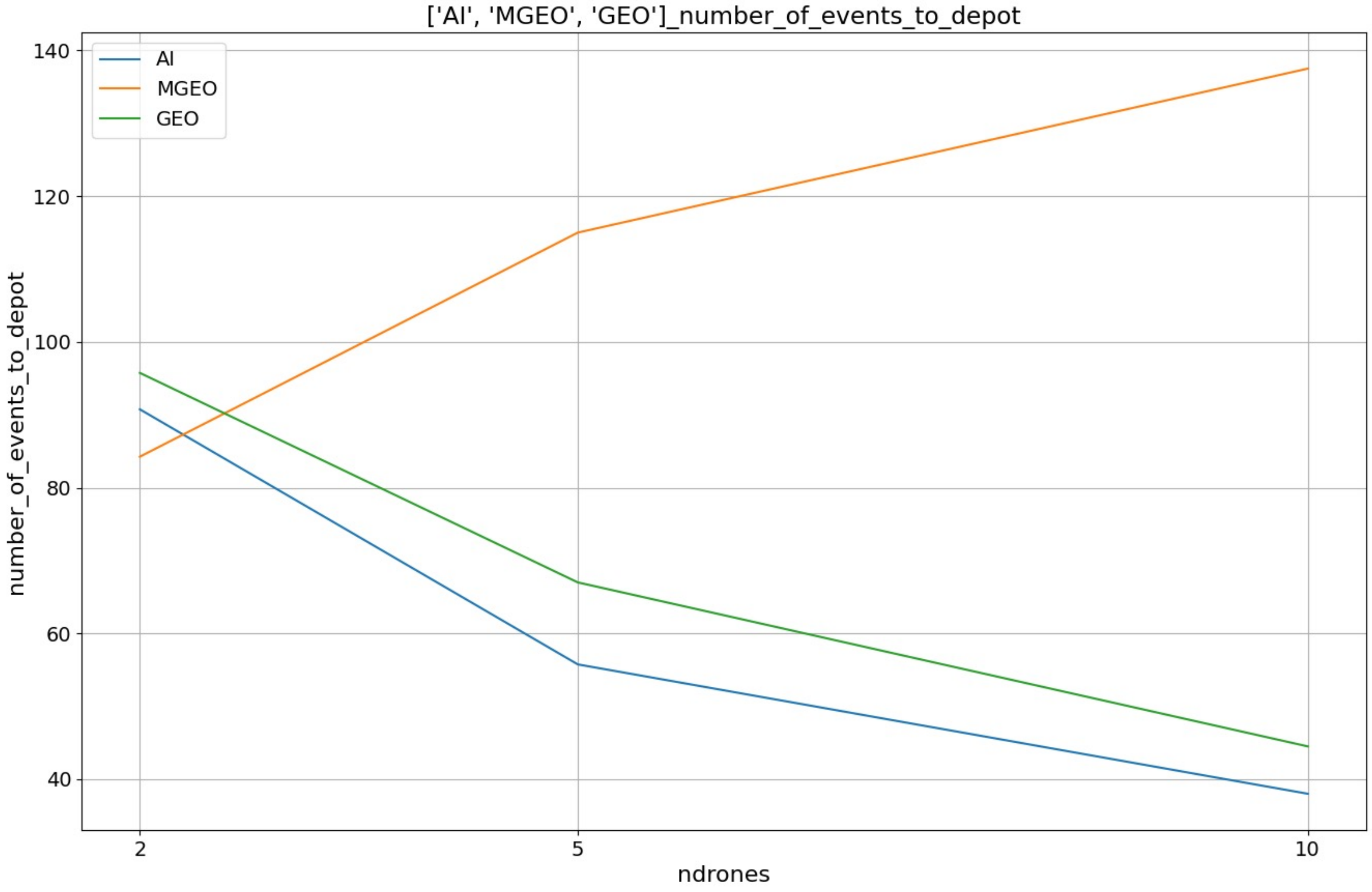
HOMEWORK 3 – RESULTS



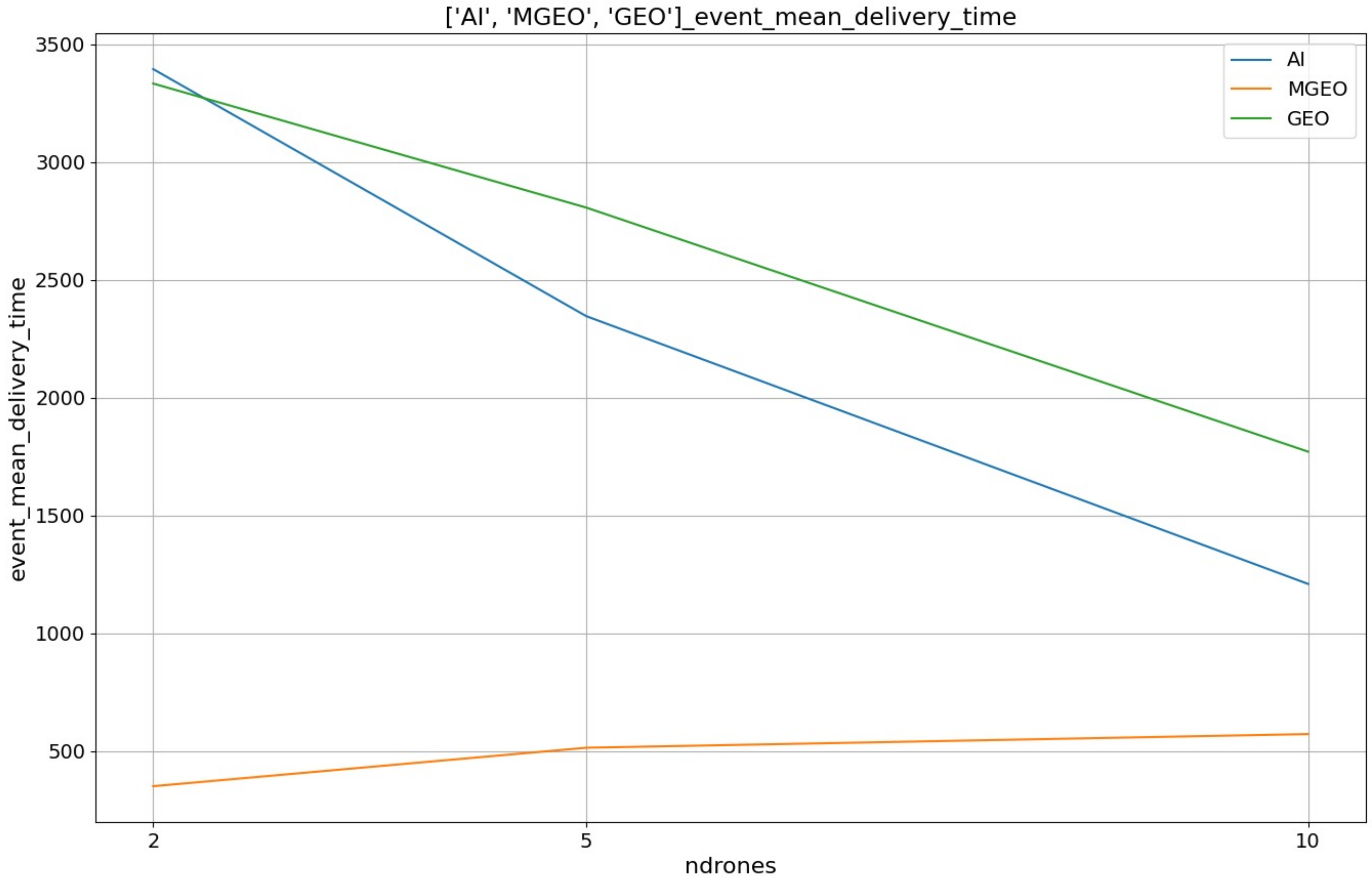
HOMEWORK 3 – RESULTS



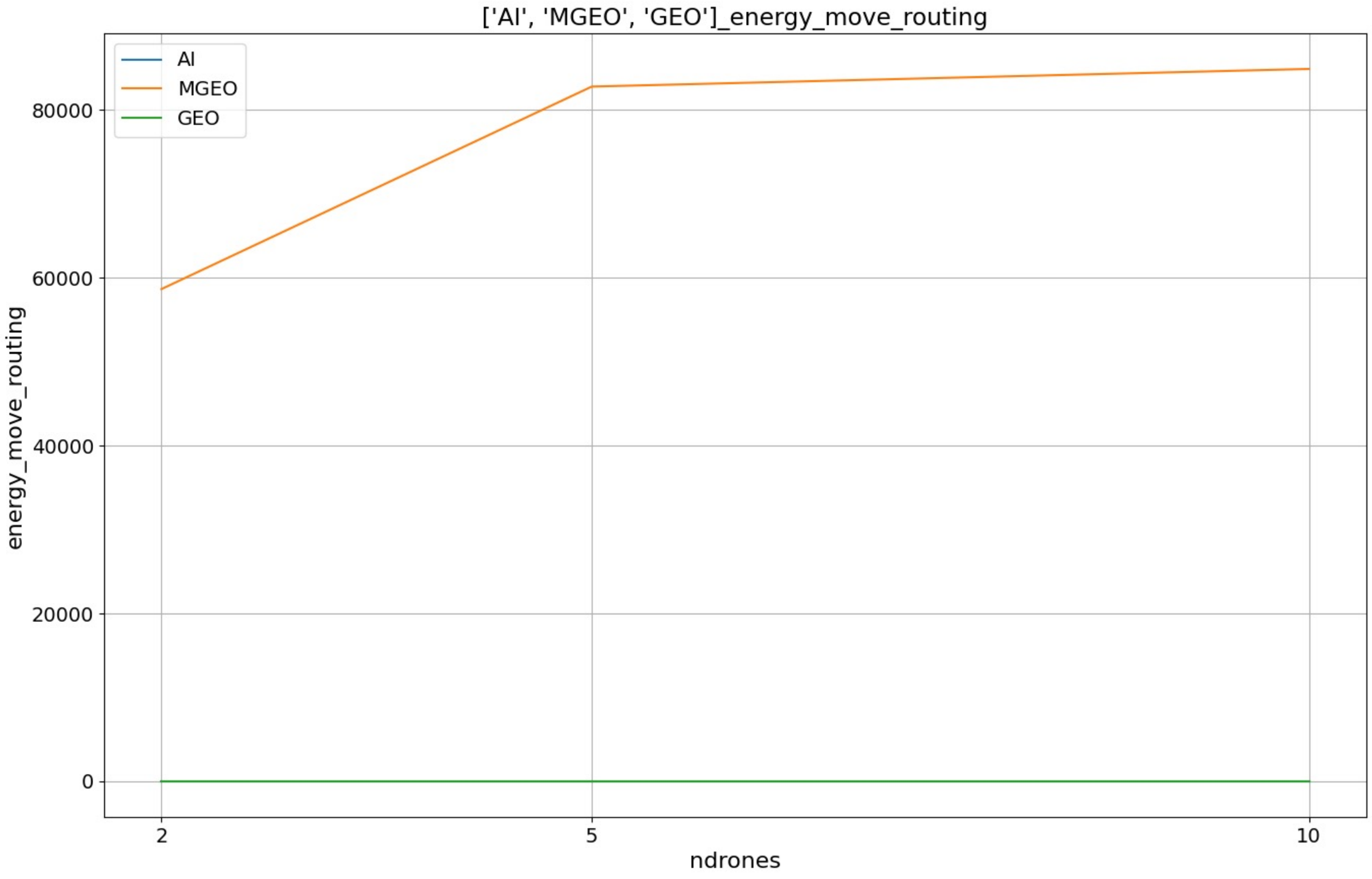
HOMEWORK 3 – RESULTS



HOMEWORK 3 – RESULTS



HOMEWORK 3 – RESULTS



HOMEWORK 3 - SUBMISSION

How:

- **email** (**subject=**[Autonomous Networking - A.A. 2021-2022] – HMW3)
- **Classroom**

Format:

A unique submission for each group is enough!

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 only this file but change the file name and the algorithm name (see example here).

HOMEWORK 3 – END

Is fine to slightly change the solution of hmw2, but try to add a more theoretical analysis of the problem/solution in the report.

Possible write questions here:

https://docs.google.com/spreadsheets/d/1PbrruWdEf2w3eAmlYkXpNnypV191EFsxh0FTPhOXIHQ/edit?usp=sharing

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

Deadline: 8 January at 05:00 am