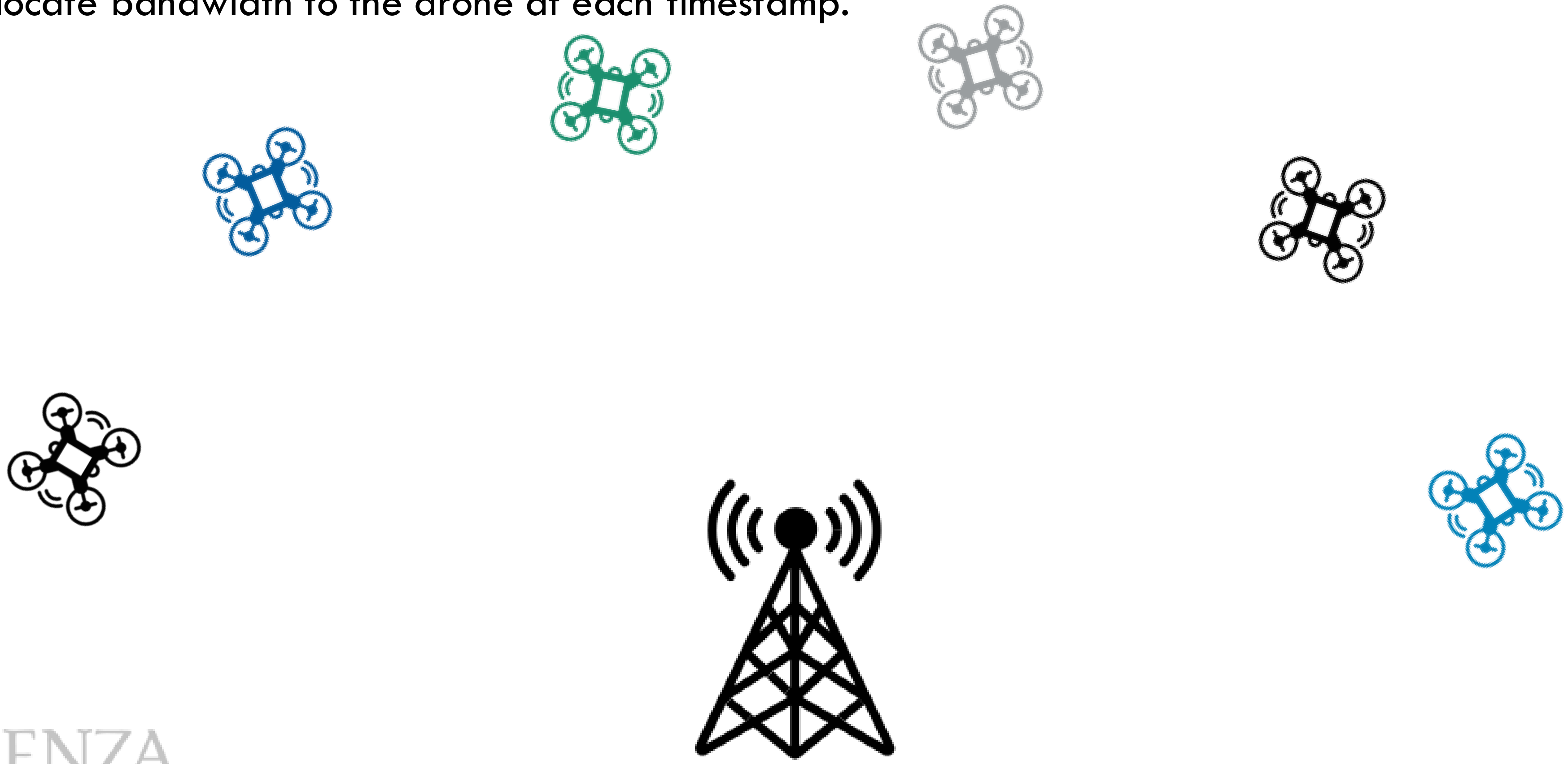
DroNET Simulator

Course: *Autonomous Networking - Prof. Gaia Maselli (A.A. 2020-2021)*
Speaker: *Dr. Andrea Coletta - 20-11-2020*

SAPIENZA
UNIVERSITÀ DI ROMA

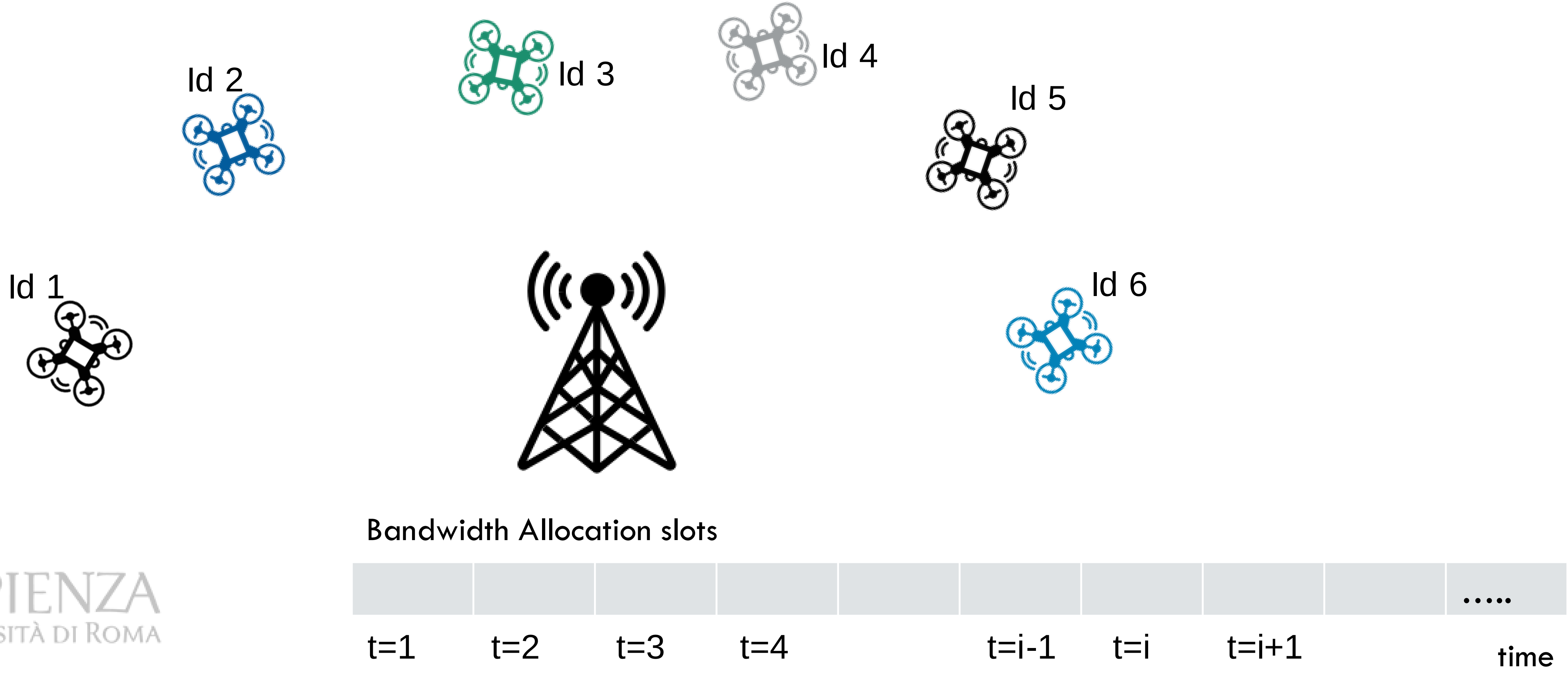
HOMEWORK 1 – MAC PROTOCOL

1) A set of drones that want communicate with the depot
2) The depot allocate bandwidth to the drone at each timestamp.



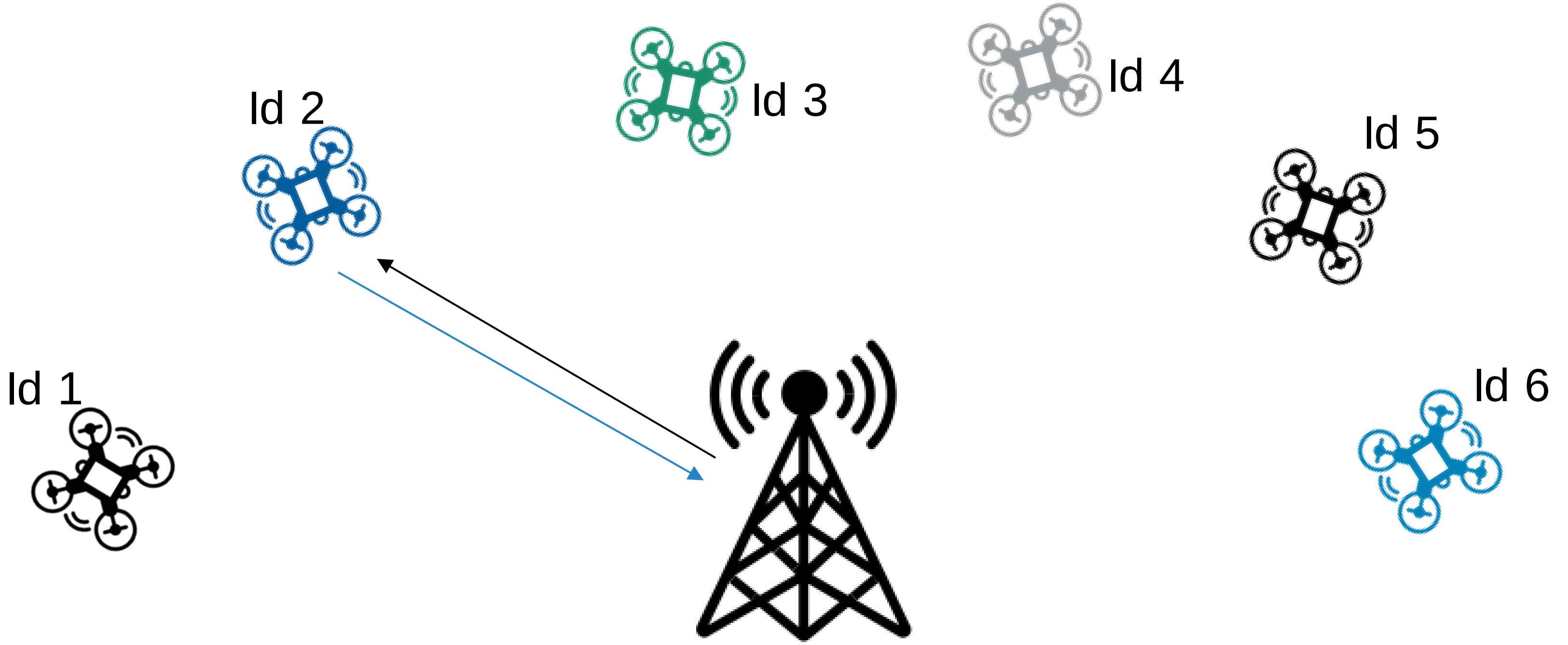
HOMEWORK 1 – MAC PROTOCOL

1) A set of drones that want communicate with the depot
2) The depot allocate bandwidth to the drone at each timestamp (**Time Slotted** Channel).



HOMEWORK 1 – MAC PROTOCOL T=1

1) A set of drones that want communicate with the depot
2) The depot allocate bandwidth to the drone at each timestamp (**Time Slotted** Channel).

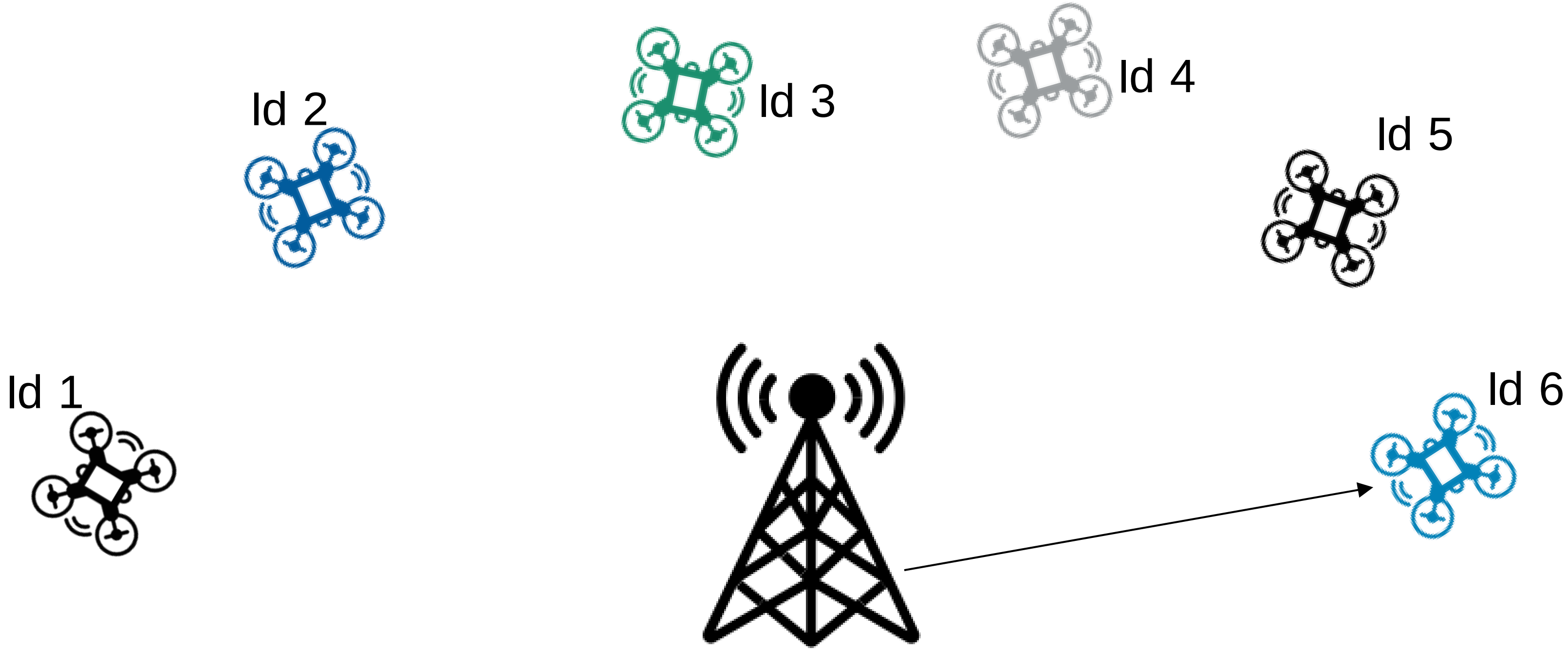


Bandwidth Allocation slots

Dr.2									…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL T=2

1) A set of drones that want communicate with the depot
2) The depot allocate bandwidth to the drone at each timestamp (**Time Slotted** Channel).

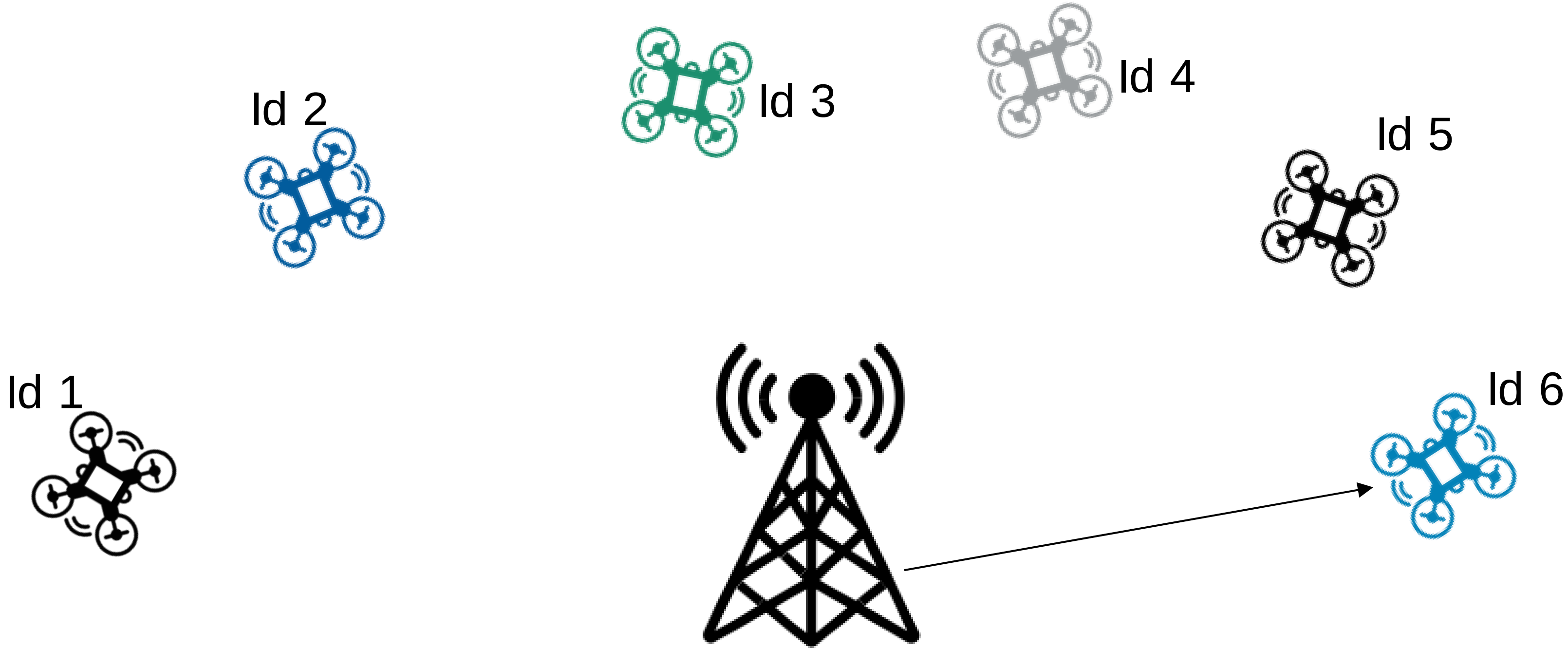


Bandwidth Allocation slots

Dr.2	**Dr.6**								…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL T=3

1) A set of drones that want communicate with the depot
2) The depot allocate bandwidth to the drone at each timestamp (**Time Slotted** Channel).

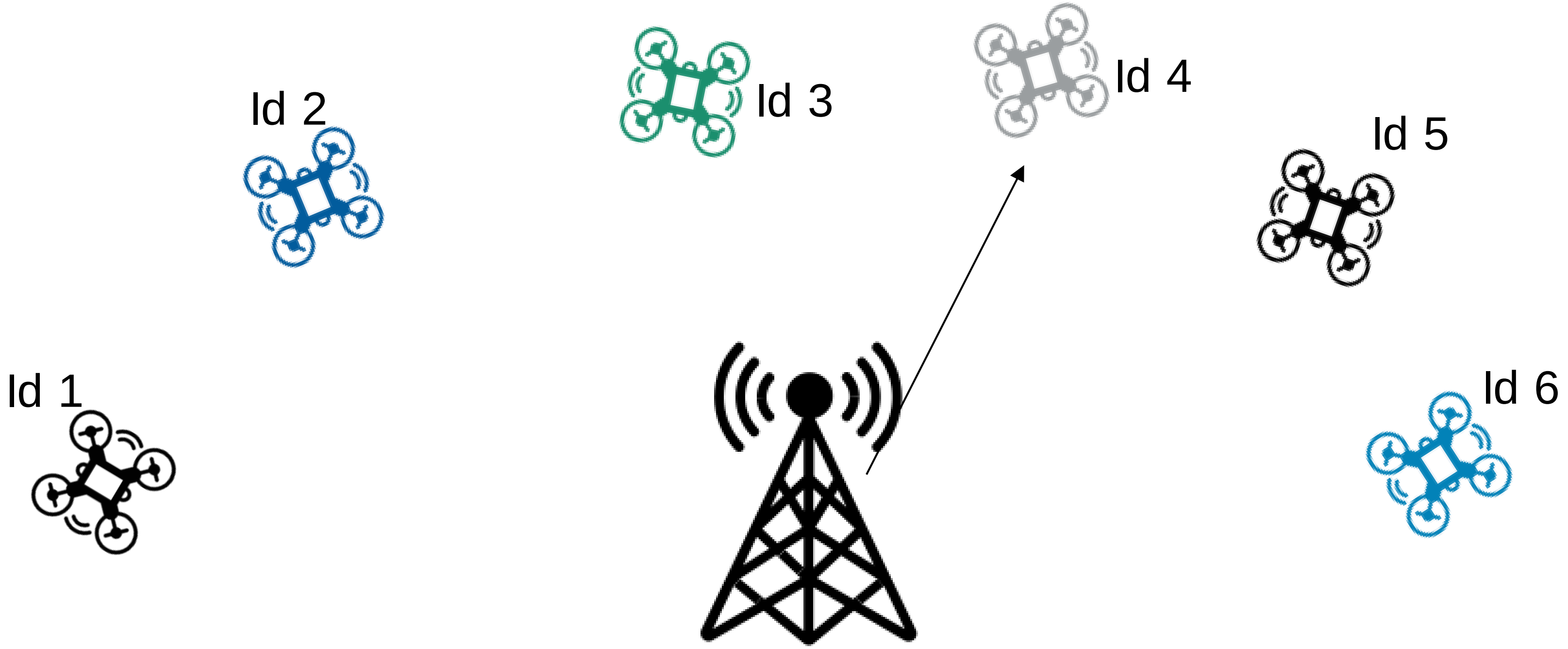


Bandwidth Allocation slots

Dr.2	**Dr.6**	**Dr.6**							…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL T=4

1) A set of drones that want communicate with the depot
2) The depot allocate bandwidth to the drone at each timestamp (**Time Slotted** Channel).

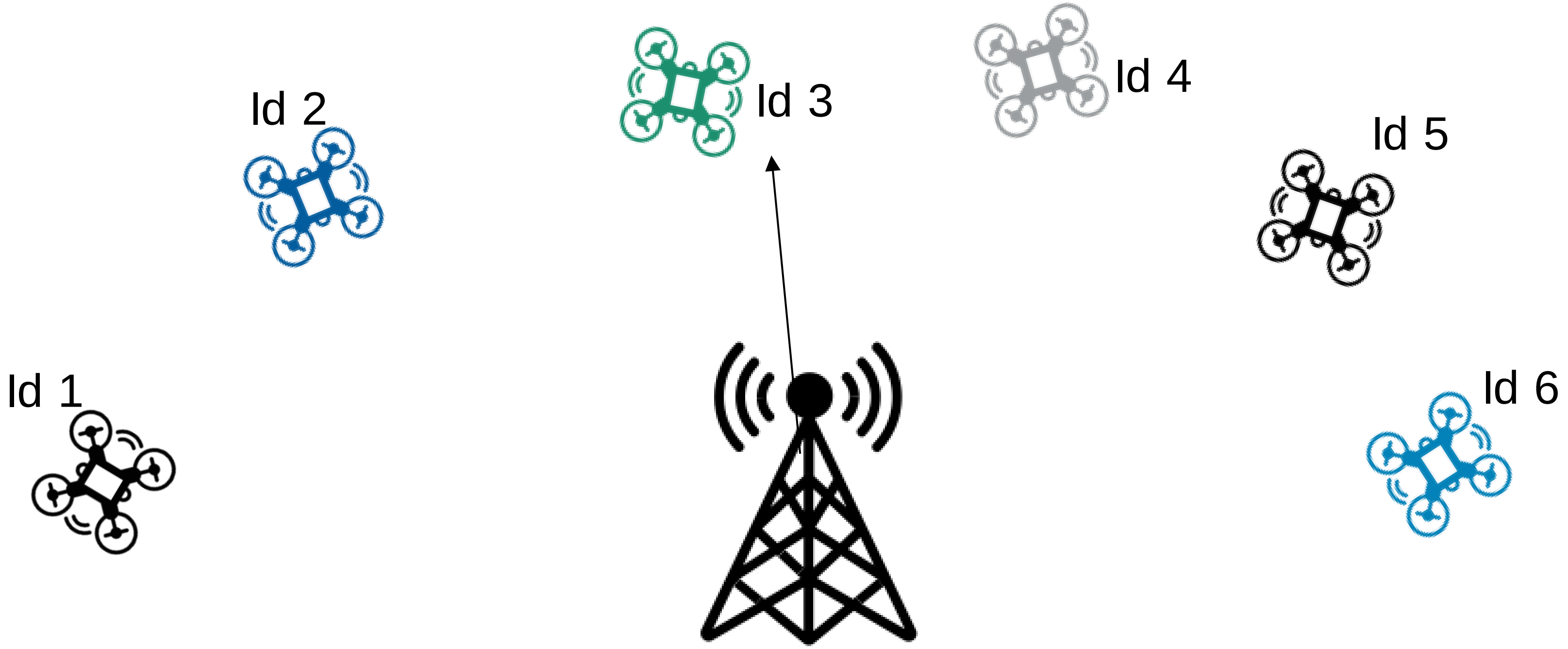


Bandwidth Allocation slots

Dr.2	**Dr.6**	**Dr.6**	**Dr.4**						…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL T=i-1

1) A set of drones that want communicate with the depot
2) The depot allocate bandwidth to the drone at each timestamp (**Time Slotted** Channel).

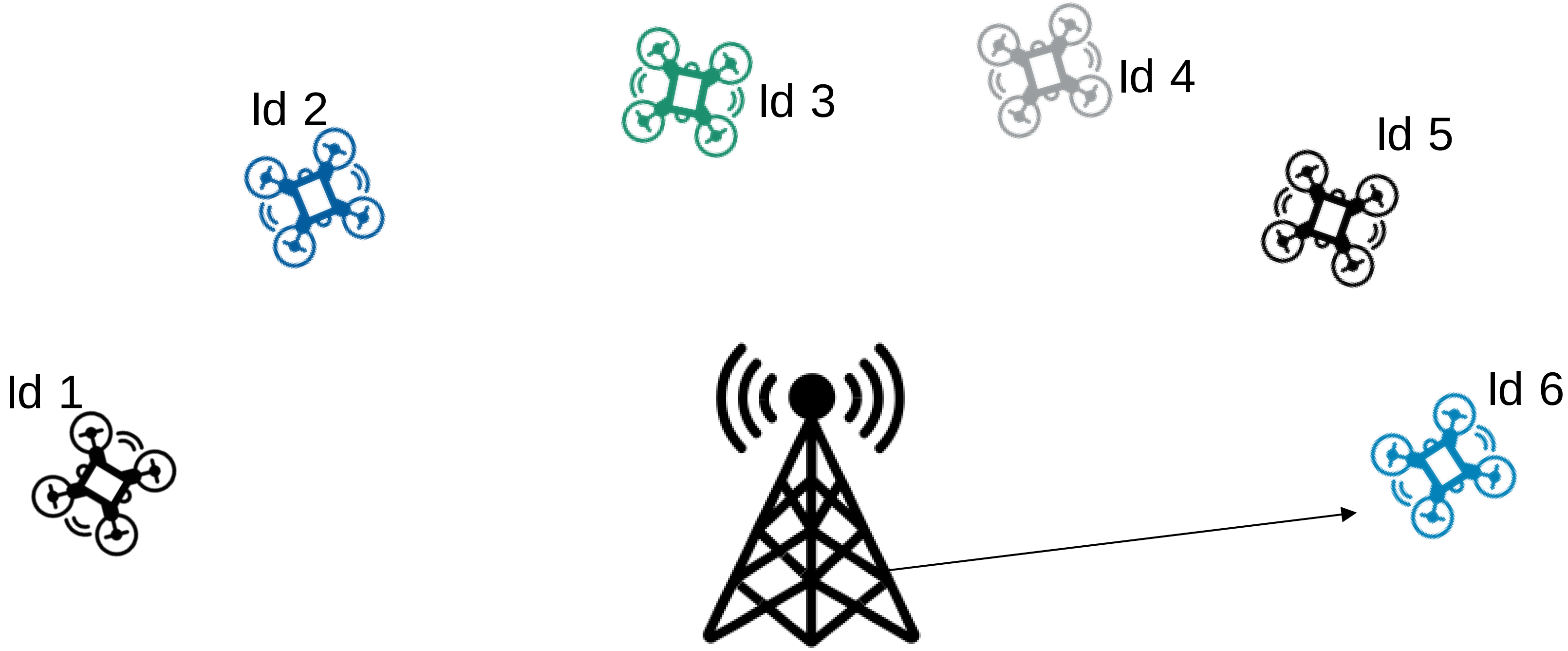


Bandwidth Allocation slots

Dr.2	**Dr.6**	**Dr.6**	**Dr.4**	**…**	**Dr.3**				**…..**
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL T=i

1) A set of drones that want communicate with the depot
2) The depot allocate bandwidth to the drone at each timestamp (**Time Slotted** Channel).

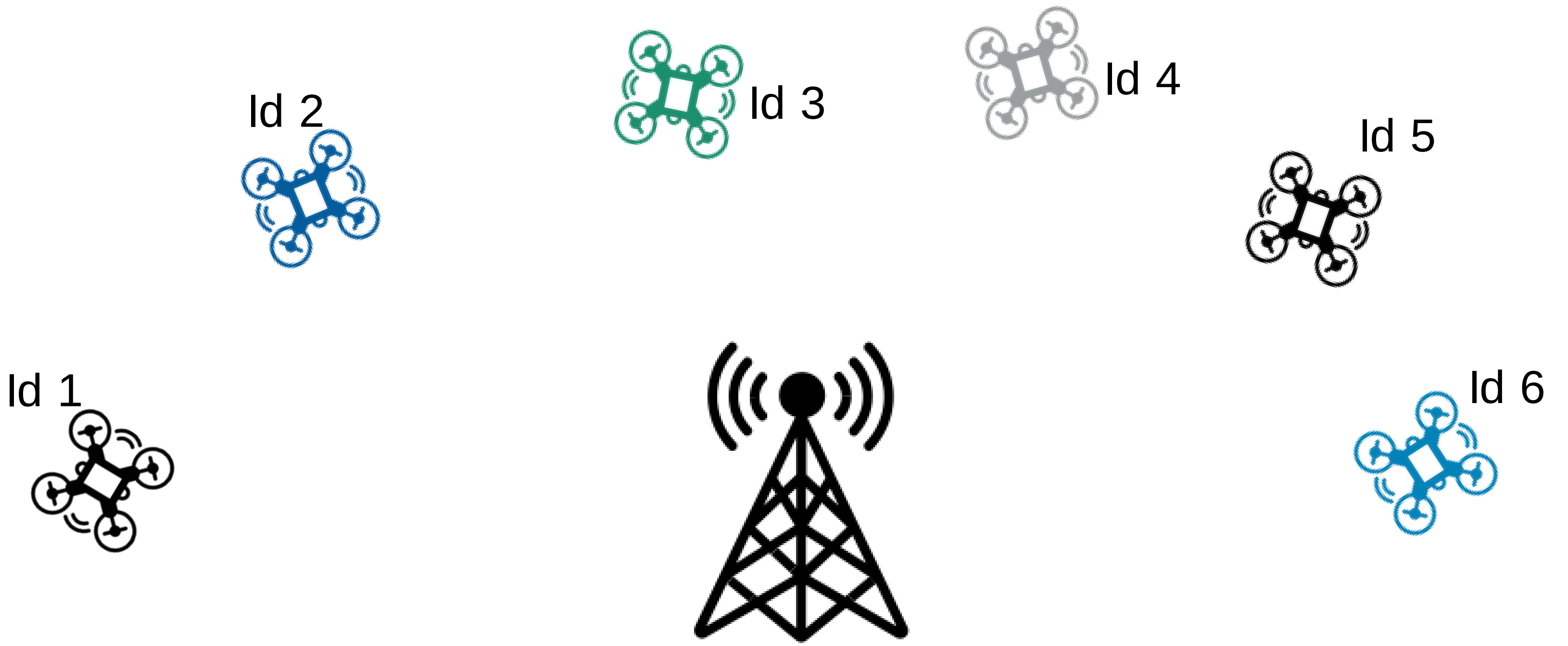


Bandwidth Allocation slots

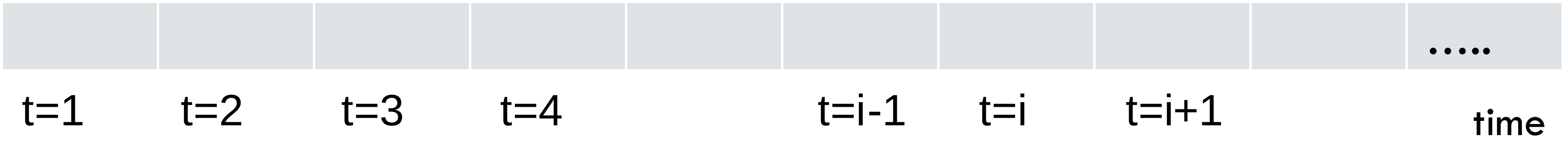
Dr.2	**Dr.6**	**Dr.6**	**Dr.4**	**…**	**Dr.3**	**Dr.6**	**…**	**…**	**…**
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) round robin

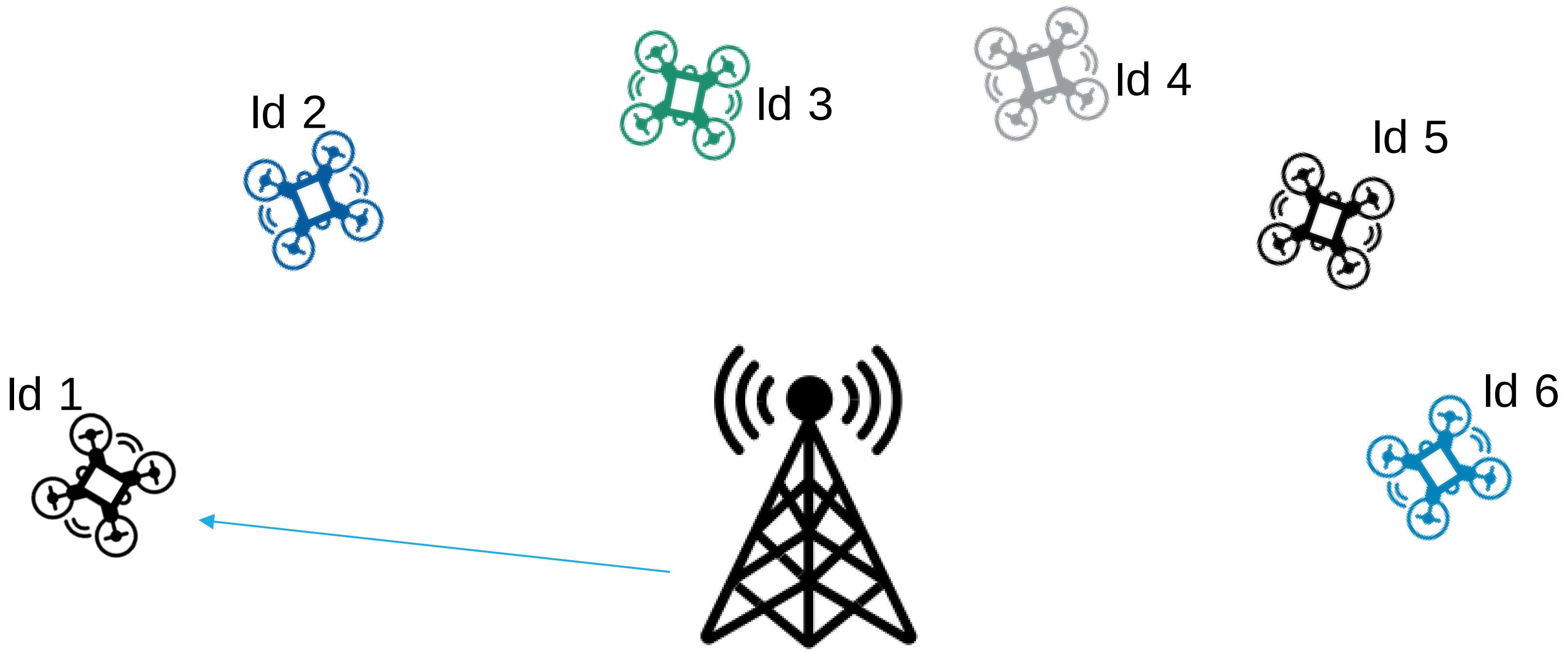


Bandwidth Allocation slots



HOMEWORK 1 – MAC PROTOCOL - HOW

1) round robin

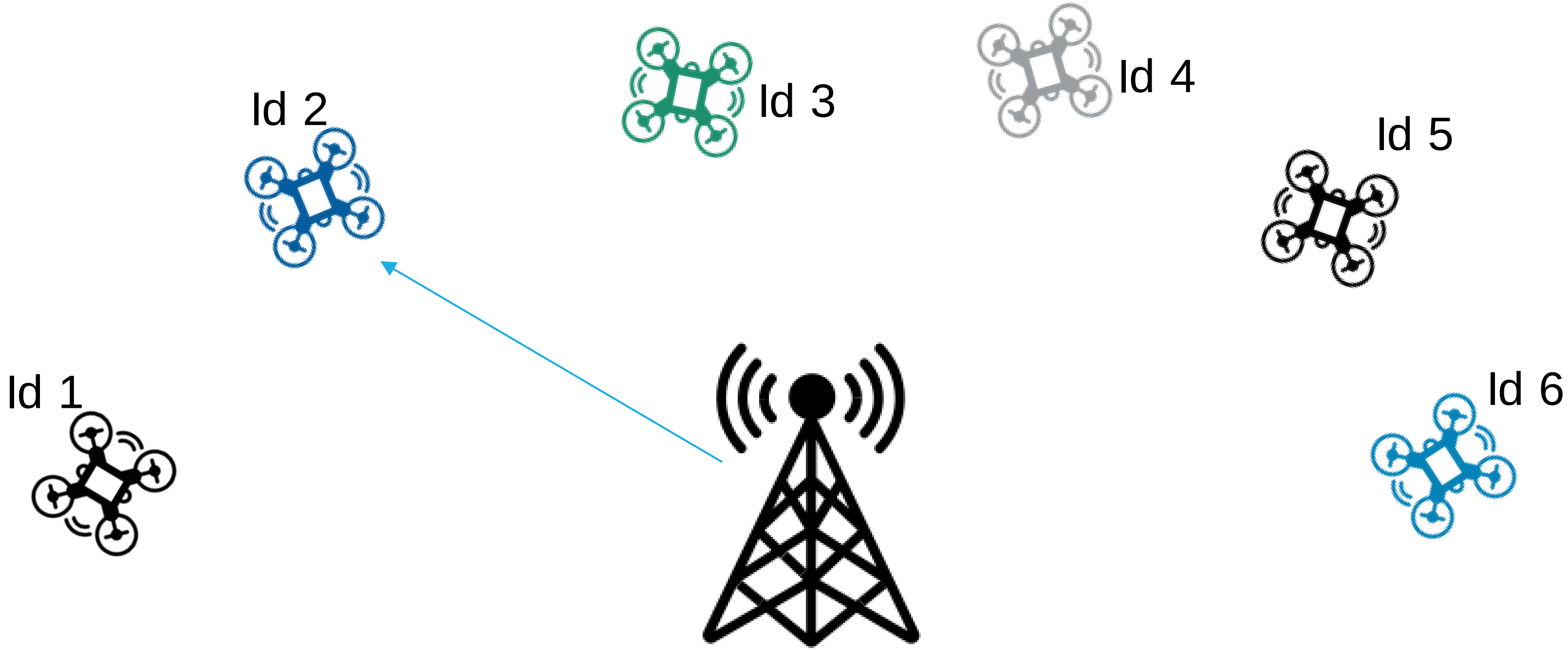


Bandwidth Allocation slots

Dr.1									…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) round robin

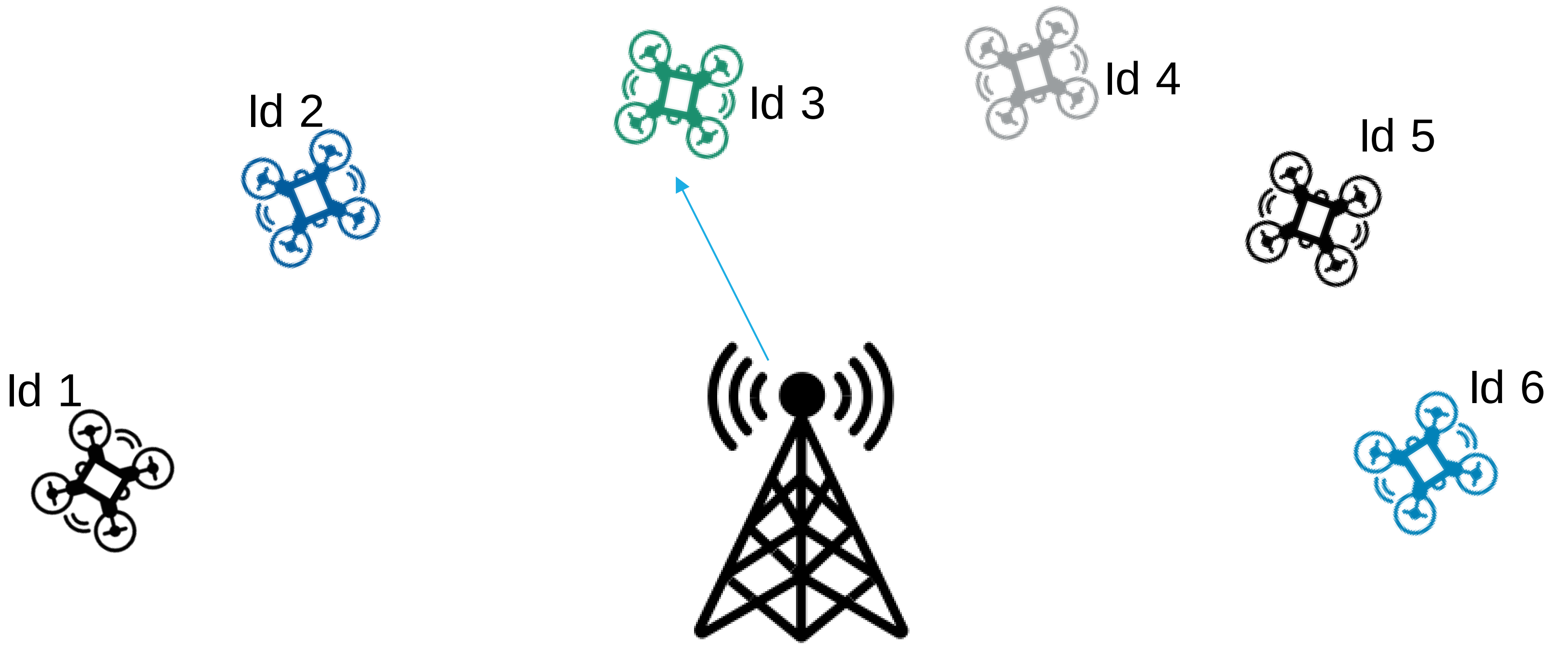


Bandwidth Allocation slots

Dr.1	**Dr.2**								…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) round robin

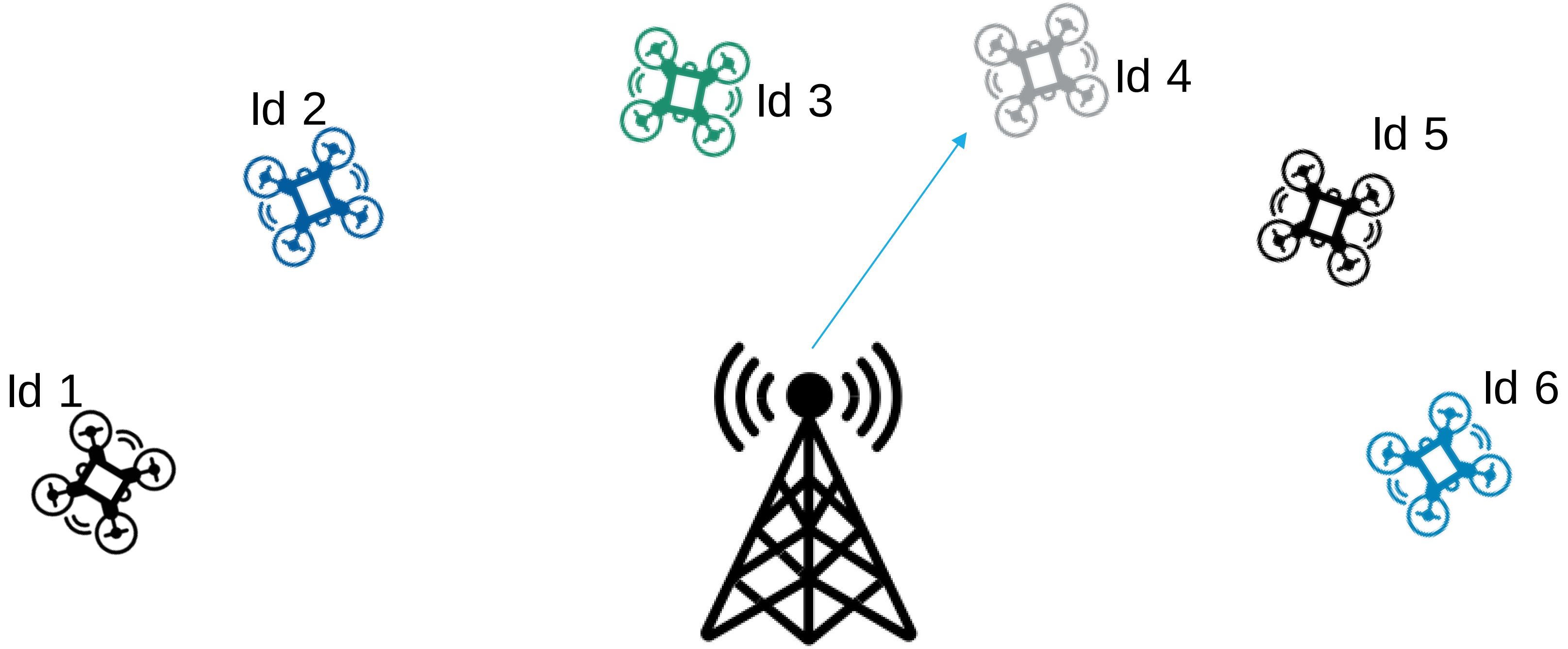


Bandwidth Allocation slots

Dr.1	**Dr.2**	**Dr.3**							…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) round robin

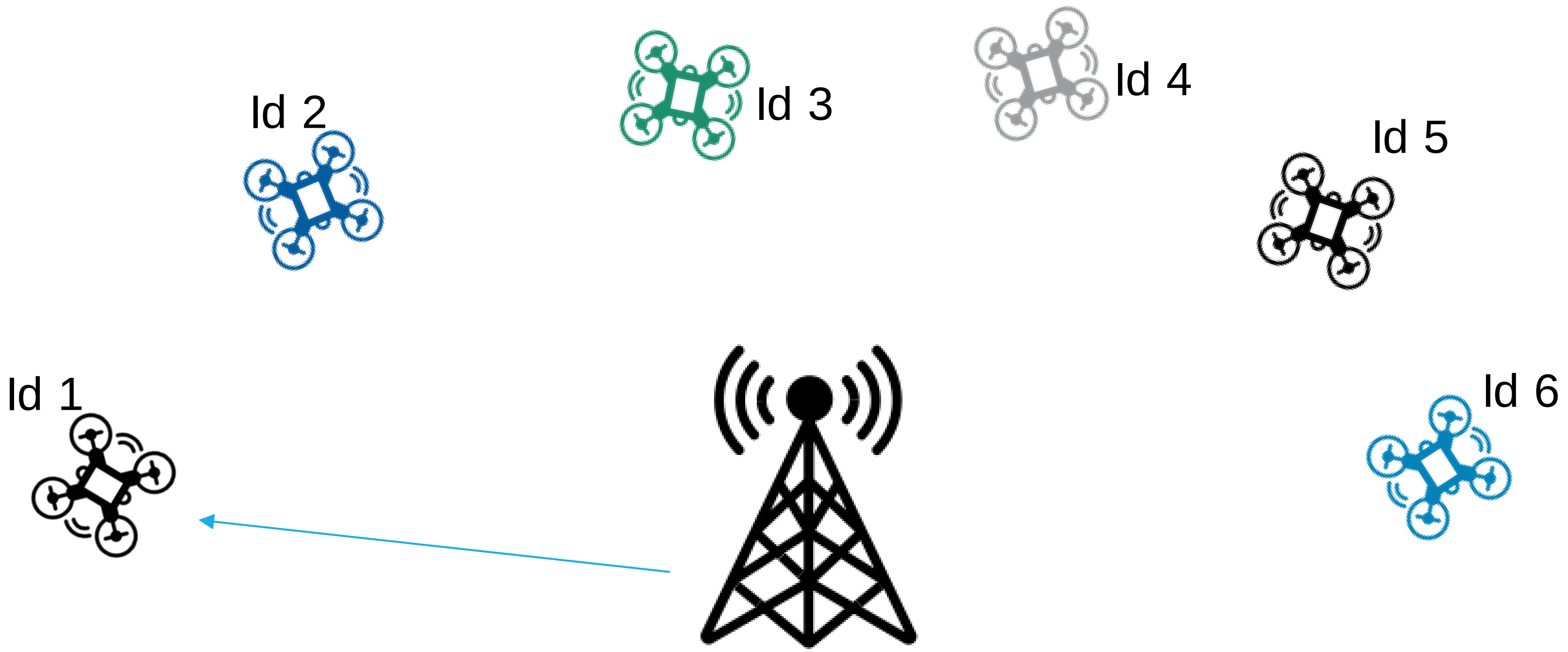


Bandwidth Allocation slots

Dr.1	**Dr.2**	**Dr.3**	**Dr.4**	**…**					**…..**
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) round robin

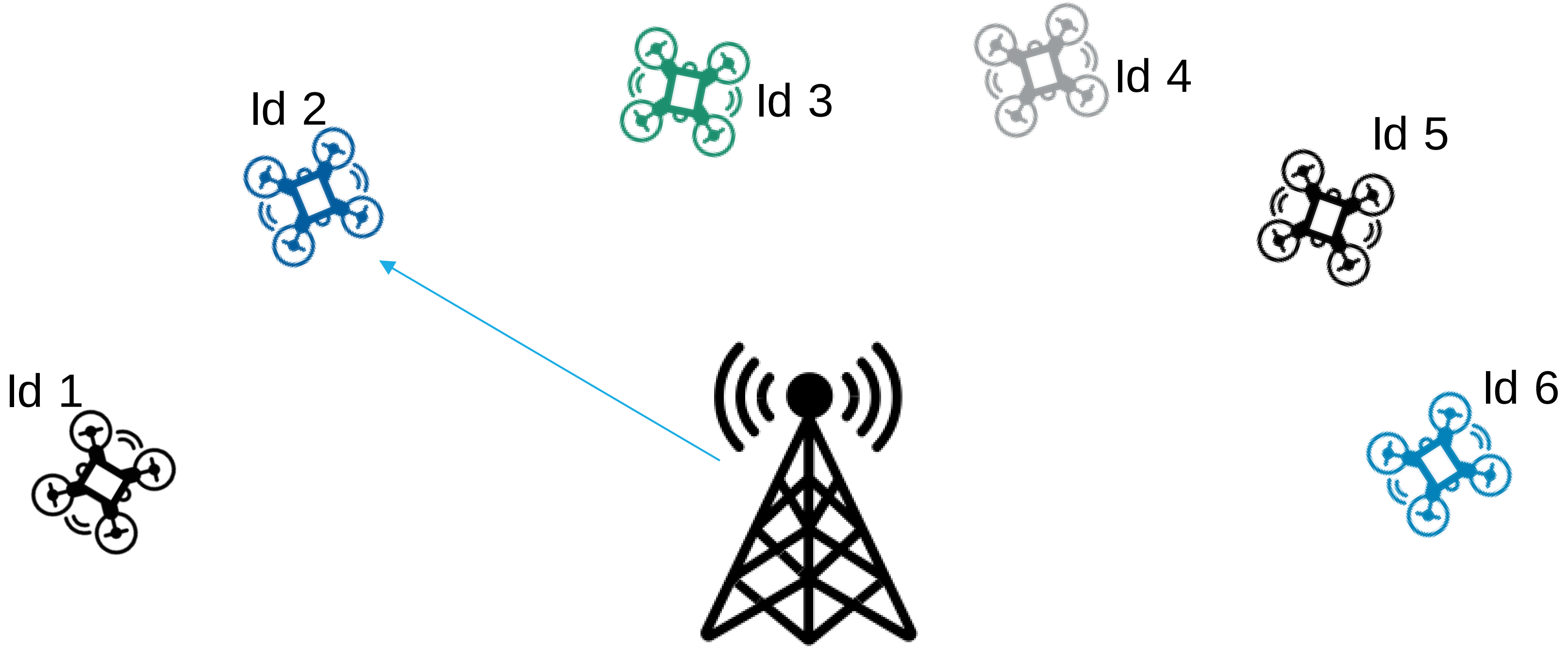


Bandwidth Allocation slots

Dr.1	**Dr.2**	**Dr.3**	**Dr.4**	**…**	**Dr.1**				**…..**
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) round robin

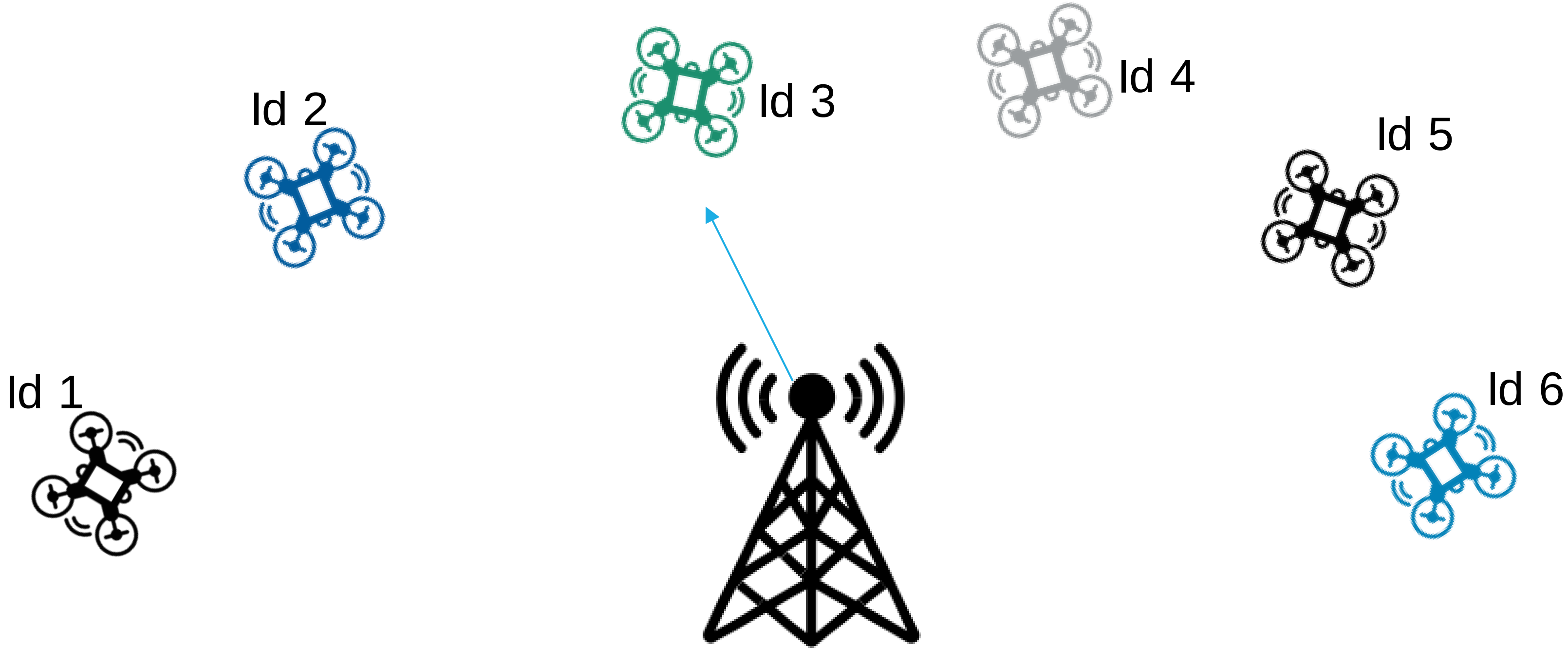


Bandwidth Allocation slots

Dr.1	**Dr.2**	**Dr.3**	**Dr.4**	**…**	**Dr.1**	**Dr.2**			**…..**
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) round robin

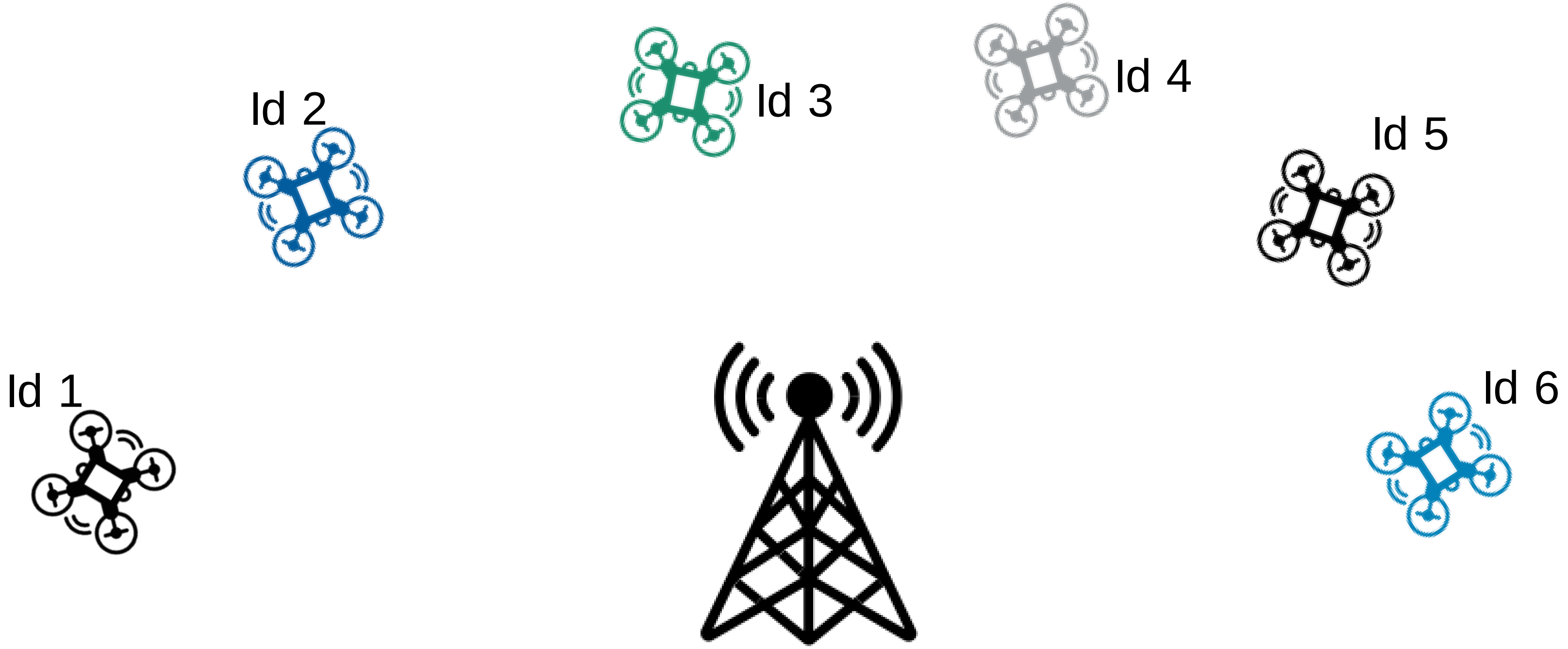


Bandwidth Allocation slots

Dr.1	Dr.2	Dr.3	Dr.4	…	Dr.1	Dr.2	Dr.3		…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) Why round robin may fail?

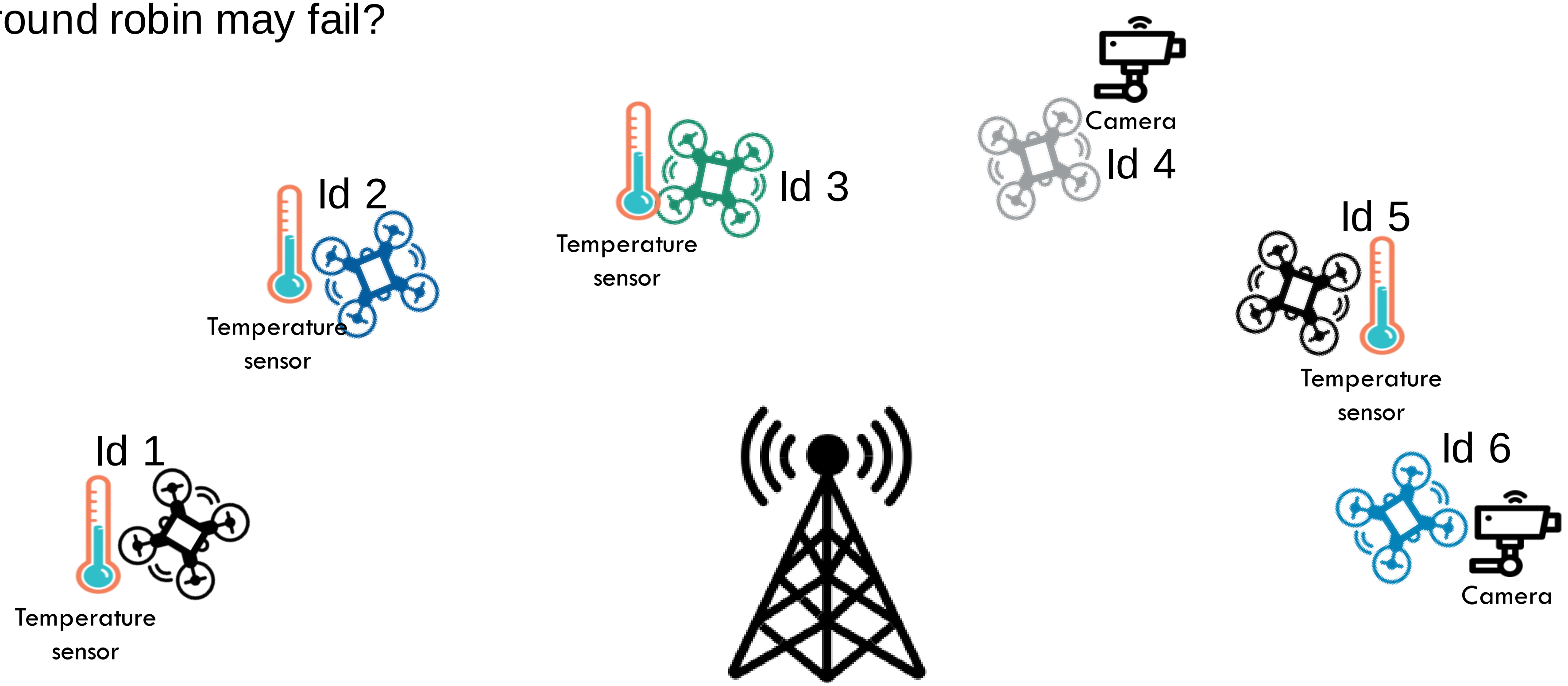


Bandwidth Allocation slots

Dr.1	Dr.2	Dr.3	Dr.4	…	Dr.1	Dr.2	Dr.3		…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) Why round robin may fail?



Bandwidth Allocation slots

Dr.1	Dr.2	Dr.3	Dr.4	…	Dr.1	Dr.2	Dr.3		…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

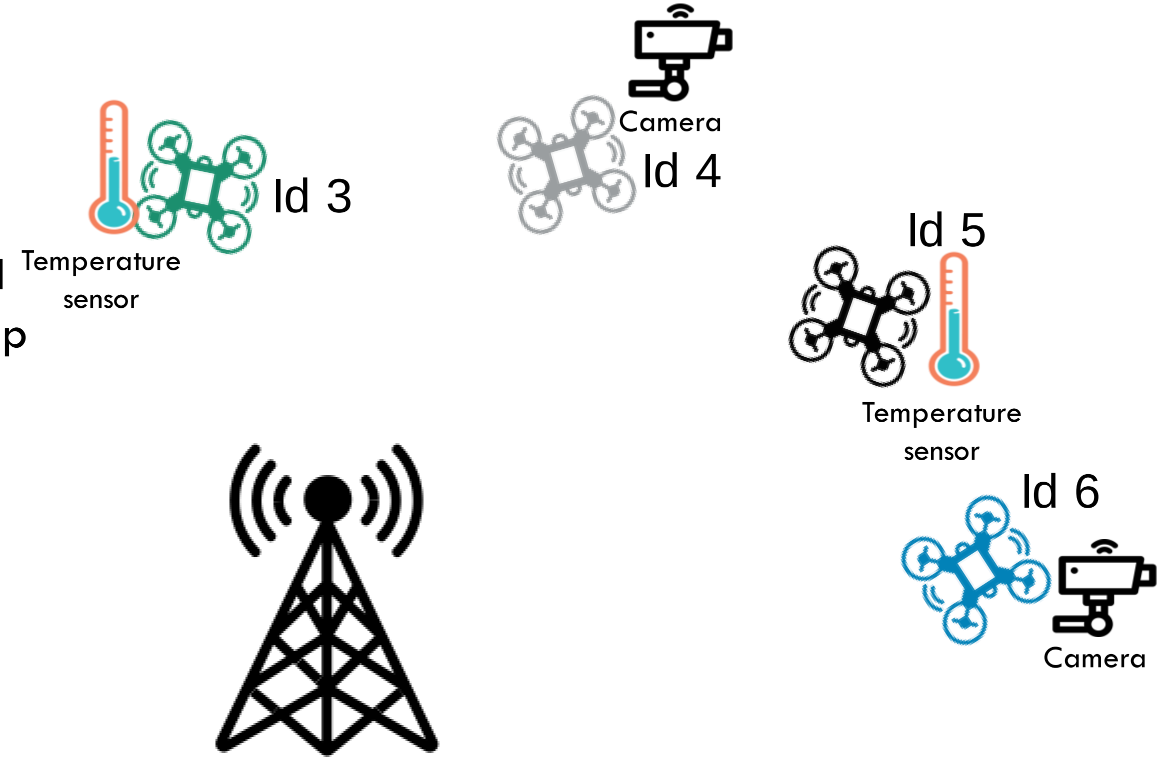
HOMEWORK 1 – MAC PROTOCOL - HOW

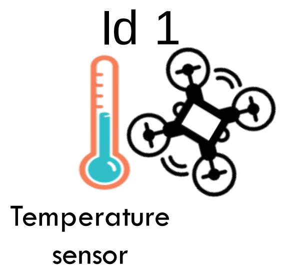
1) Why round robin may fail?

WE SPENT SAME TIME ON EACH DRONE!

We lost data!

The drones with cameras should be queried more frequently respect to drones with temp sensors!



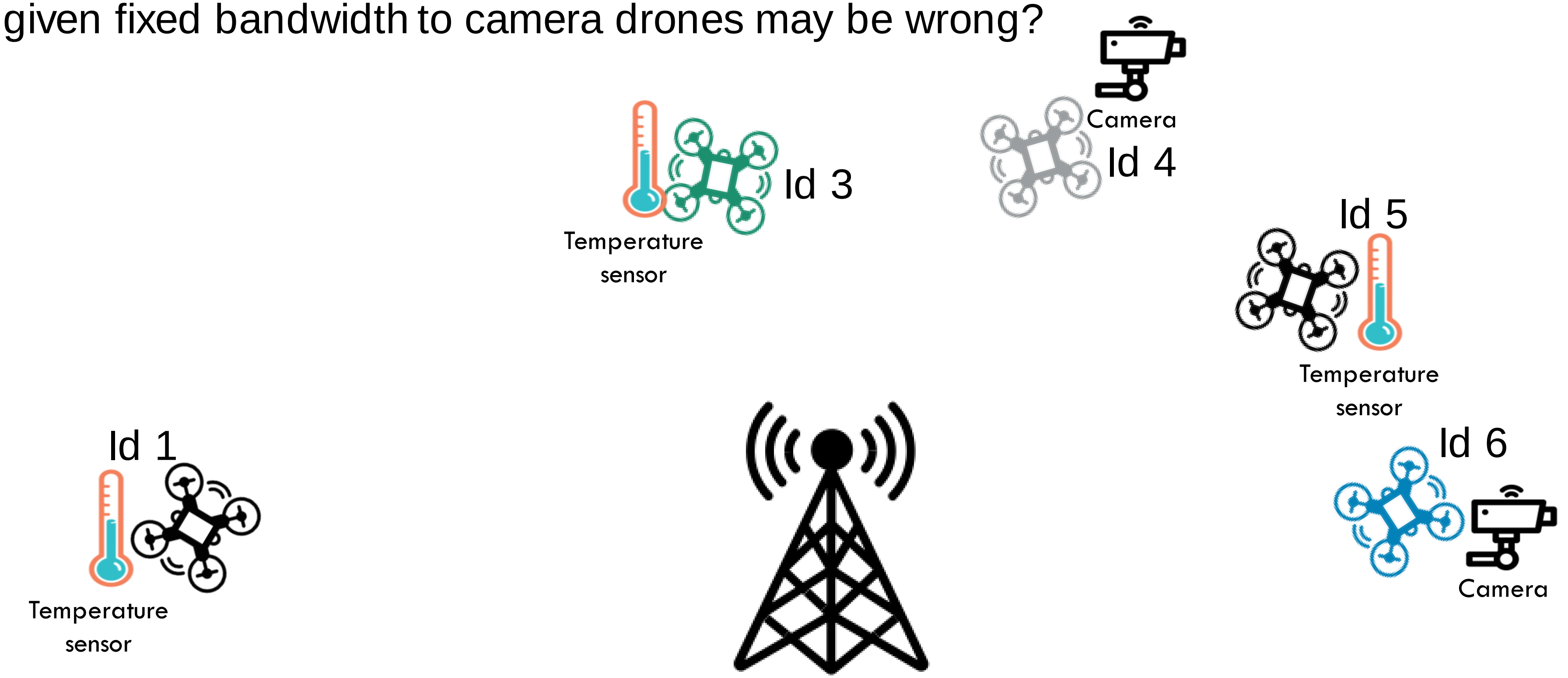


Bandwidth Allocation slots

Dr.1	Dr.2	Dr.3	Dr.4	…	Dr.1	Dr.2	Dr.3		…..
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL - HOW

1) Why given fixed bandwidth to camera drones may be wrong?



Bandwidth Allocation slots

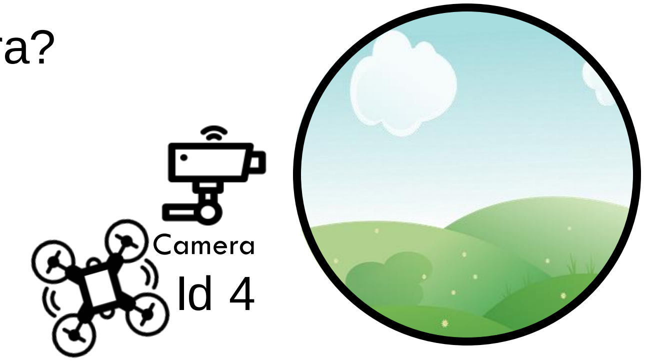
Dr.1	**Dr.2**	**Dr.3**	**Dr.4**	**…**	**Dr.1**	**Dr.2**	**Dr.3**		**…..**
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

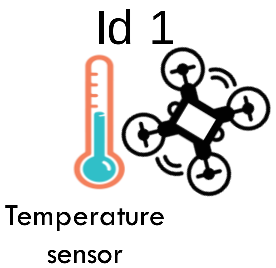
HOMEWORK 1 – MAC PROTOCOL - HOW

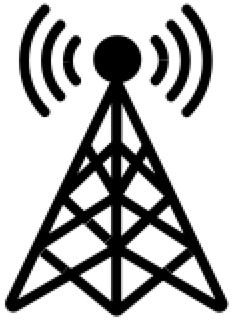
1) So we can just give more bandwidth to drones with a camera?

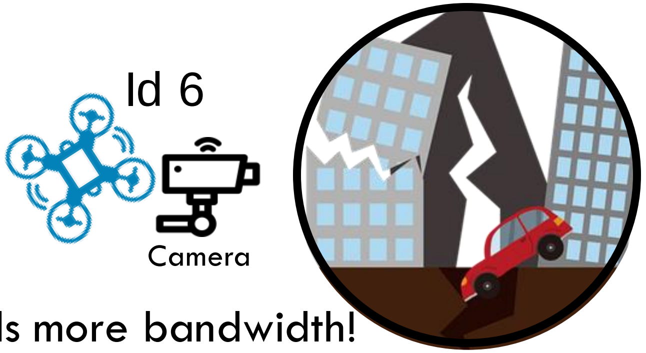
WE DON'T KNOW THE POSSIBLE SITUATION!

We need adaptive approach!









Drone 6 needs more bandwidth!

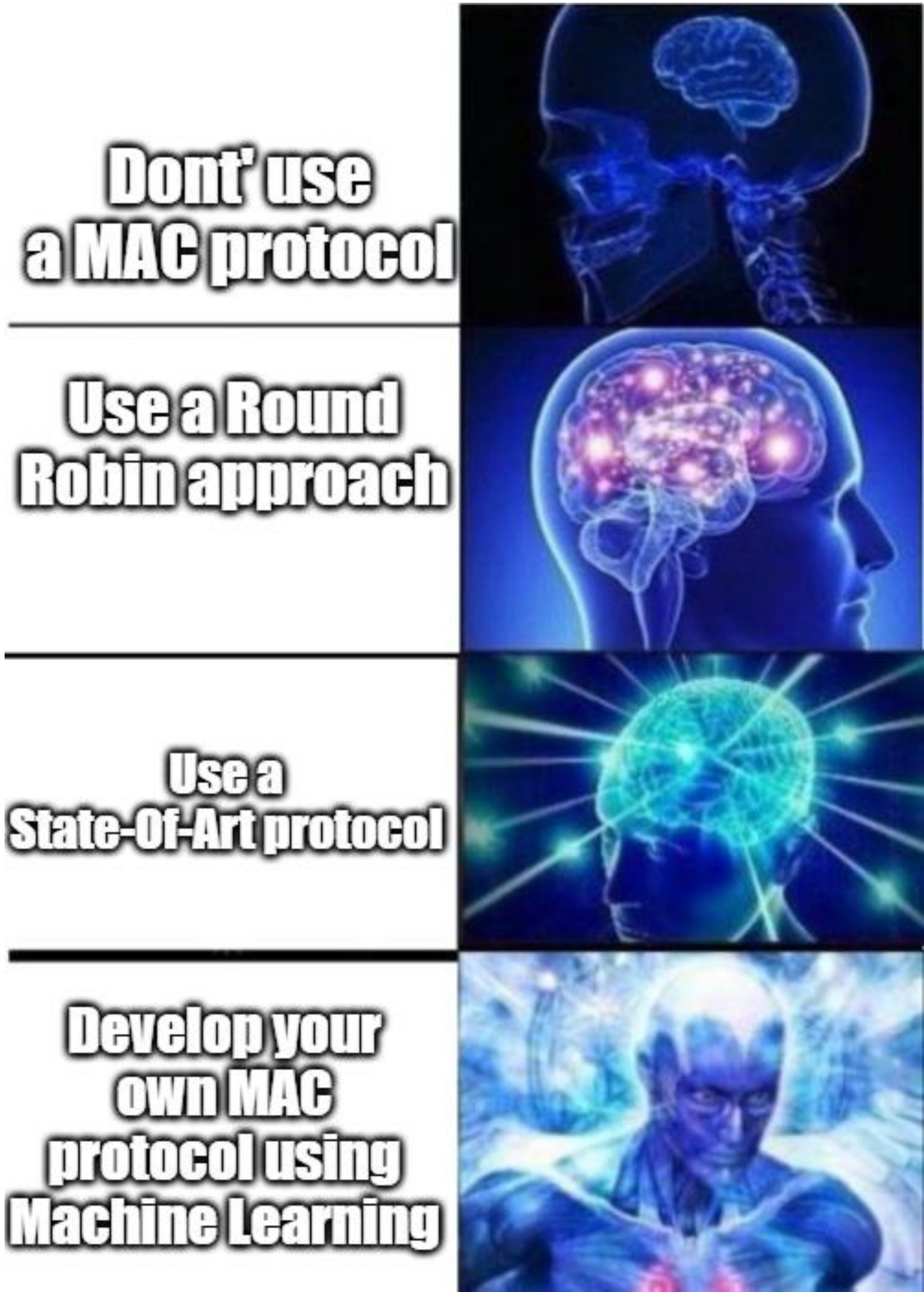
Bandwidth Allocation slots

Dr.4	**Dr.4**	**Dr.6**	**Dr.6**	**…**	**Dr.6**	**Dr.6**	**Dr.1**		**…..**
t=1	t=2	t=3	t=4		t=i-1	t=i	t=i+1		time

HOMEWORK 1 – MAC PROTOCOL

GOAL:

Create a **Reinforcement Learning** based MAC protocol!!



HOMEWORK 1 – MAC PROTOCOL

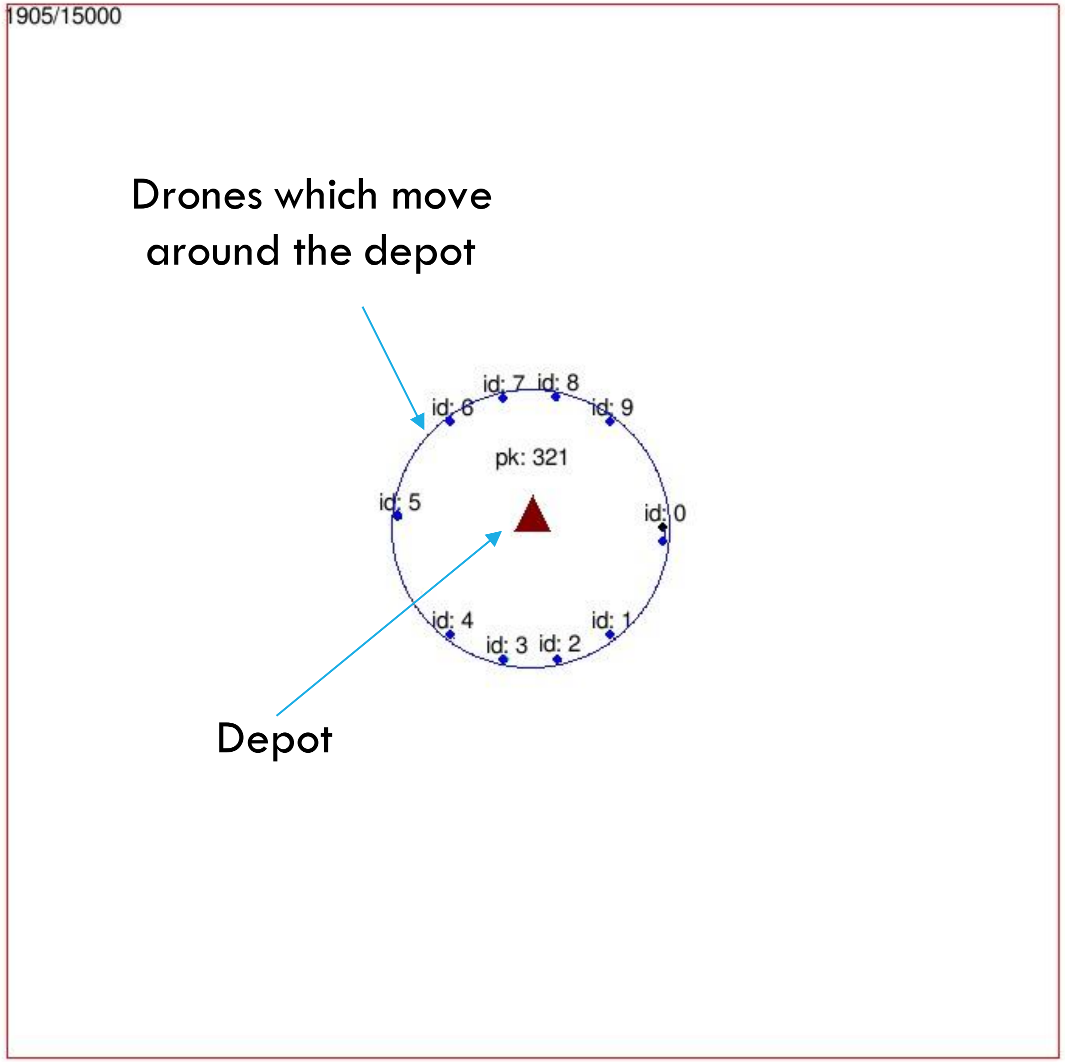
git clone https://github.com/Andrea94c/DroNETworkSimulator

git checkout hmw1

python3 –m src.main

To implement: src.mac_protocol.ai_depot_mac (used by the depot)

The method selects the next drone to query.



HOMEWORK 1 – MAC PROTOCOL

TASK: Implement the "allocate_resource_to_drone" method. (<u>DO NOT change other files</u>)

Use it by setting:

config.MAC_ALGORITHM = MACAlgorithm.AI

config.py × depot_mac_weighted.py × ai_depot_mac.py × depot_mac_rnd.py × depot_mac.py × uav_entities.py ×

```python

import ...

"""
The class is responsable to allocate communication resources to neighbors drones that want to offload data to the depot.
We work over an semplified TDMA approach, each time step only one drone can receive the resource and communicate a packet to the depot.
"""
class AIDepotMAC(DepotMAC):

    def allocate_resource_to_drone(self, drones : list, cur_step : int) -> Drone:
        """ Return the drone to who allocate bandwith for upload data in this step """
        # implement here your intelligent logic
        return None
```

If you need random generators, create them in the __init__

self.rnd_for_mac = np.random.RandomState(self.simulator.seed)

HOMEWORK 1 – MAC PROTOCOL - FEEDBACK

To learn and adapt according the scenario, the MAC stores a feedback from the last communication.

Self.last_feedback (when not None) contains a tuple (drone, transmission, feedback):

Transmission *(if the current transmission was usefull or not):*

False : if the drone does not send any packet (he has no alive packets)

True : if the drone sends a packet right now (he has at least one alive packet)

Feedback *(how many packets we missed):*

0 : *if the drone have no lost packets since last update (no packet has been generated since last communication)*

a positive integer (+pos) : *if the drone have some lost packets. The "pos" is the number of packets lost since last communication. A packet generated and expired counts as 1.*

FEEDBACK - EXAMPLE

1. Drone **A generates a new (1) packet** at time t_0 and **we query it immediately**.

   The method returns (A, True, 0) (transmission success and no packets lost since last query)

2. Then, drone A generates **new nine (9) packets**, at time t_1, **eight (8) expired** and **one (1) is still alive**, and we query it.

   At time t_1 it offload the last packet but the remaning (8) packets are lost as expired (not queried in time).

   The method returns (A, True, 8) (good query, a new packet, but we lost 8 packets, we should query it more frequently).

3. Then, drone A generates **4 packets** up to time t_2. **They all expire before we query it** at time t_3.

   thus, at time t_3 the drone A has not alive packets and we query it.

   The method return (A, False, 4) (bad query, no new packet, and we also lost 4 packets, we should query it more frequently)

4. Then, drone A does not generate packets anymore. At time t_4 **the drone A has not packets** and we query it.

   The method return (A, False, 0) (no new packets, useless query).

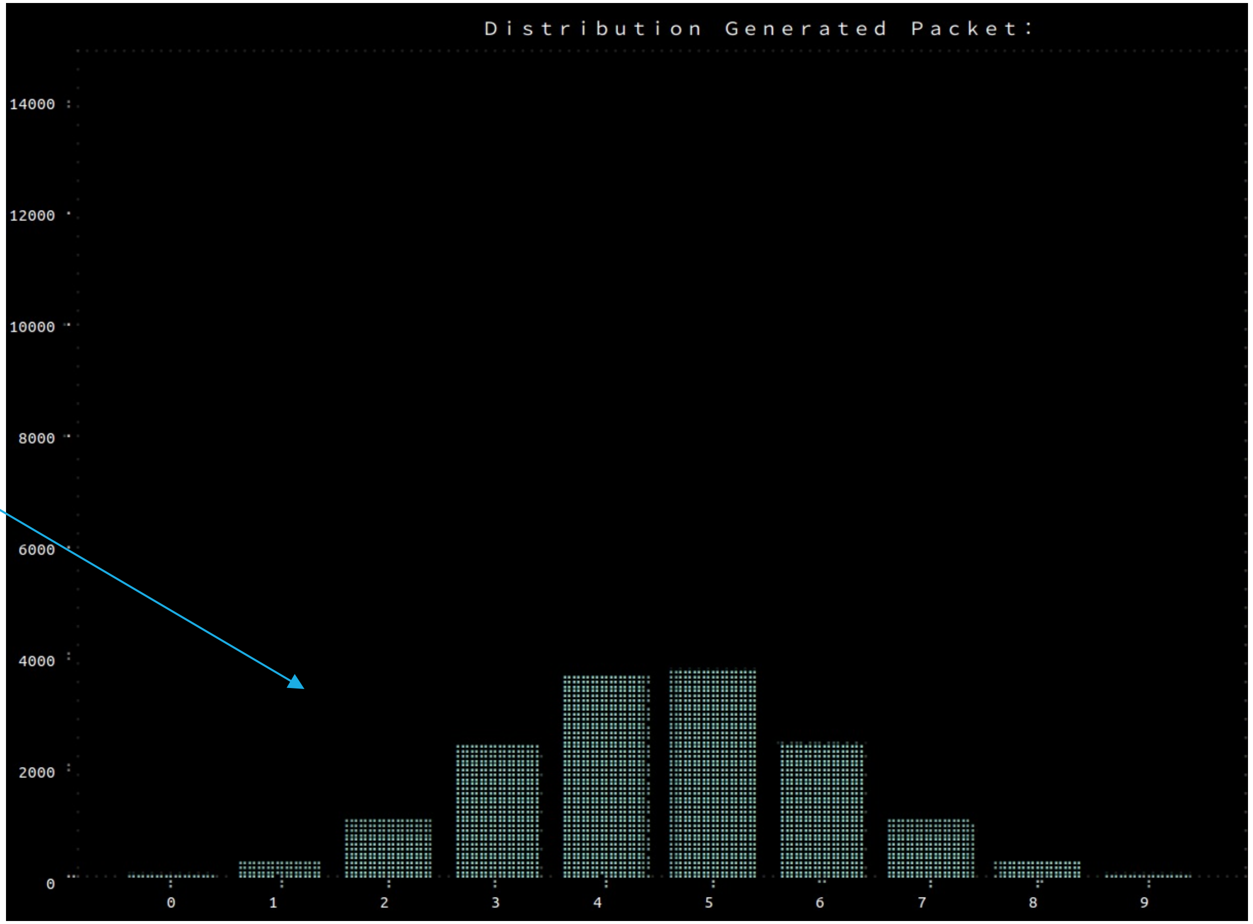
FEEDBACK - EXAMPLE

Enable debug print in: **config.MAC_PRINT_STATS = True**

```
Transmission for drone:  7  was  False - with feedback 4
Transmission for drone:  4  was  False - with feedback 5
Transmission for drone:  5  was  True - with feedback 1
Transmission for drone:  5  was  False - with feedback 0
Transmission for drone:  5  was  False - with feedback 0
Transmission for drone:  5  was  False - with feedback 0
Transmission for drone:  5  was  False - with feedback 0
Transmission for drone:  6  was  True - with feedback 0
Transmission for drone:  5  was  False - with feedback 0
Transmission for drone:  3  was  True - with feedback 1
Transmission for drone:  5  was  False - with feedback 0
Transmission for drone:  5  was  False - with feedback 0
Transmission for drone:  3  was  False - with feedback 0
Transmission for drone:  4  was  False - with feedback 4
Transmission for drone:  4  was  False - with feedback 0
```

HOMEWORK 1 – MAC PROTOCOL

Python3 –m src.main (at the end of the simulation)



The packets generated by each drone

Enable them in:
Config.PLOT_HISTOGRAMS =True

Two versions:

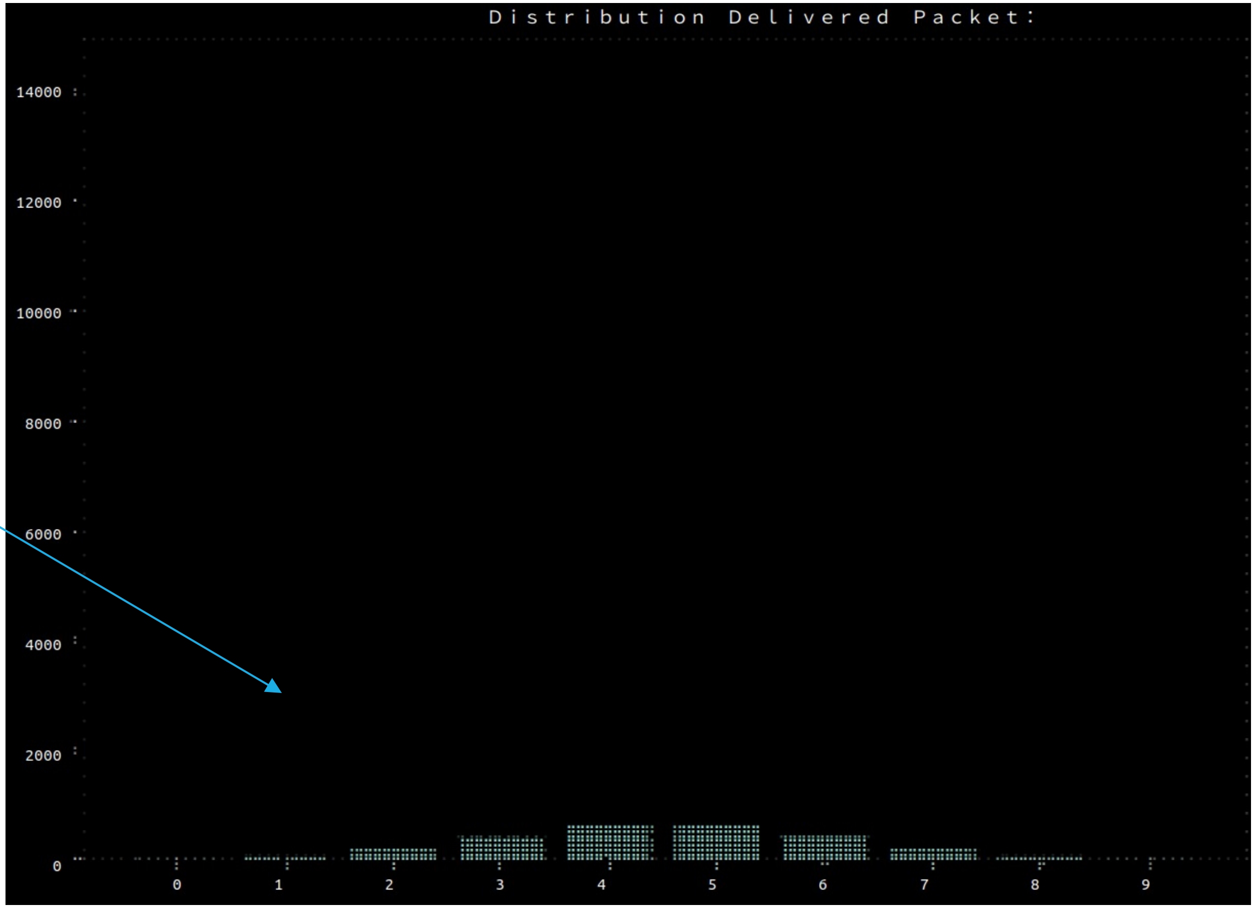
- config.MATPLOTLIB_TERMINAL=True
which needs "pip install matplotlib-terminal"

- **config.MATPLOTLIB_TERMINAL=False**
Simple text print.

HOMEWORK 1 – MAC PROTOCOL

Python3 –m src.main (at the end of the simulation)

The packets collected by the depot, from each drone



SCORE:

```
Collected Ratio: 0.18413333333333334
```

(packet collected) / (packet generated)

Higher is better! This is your main goal!

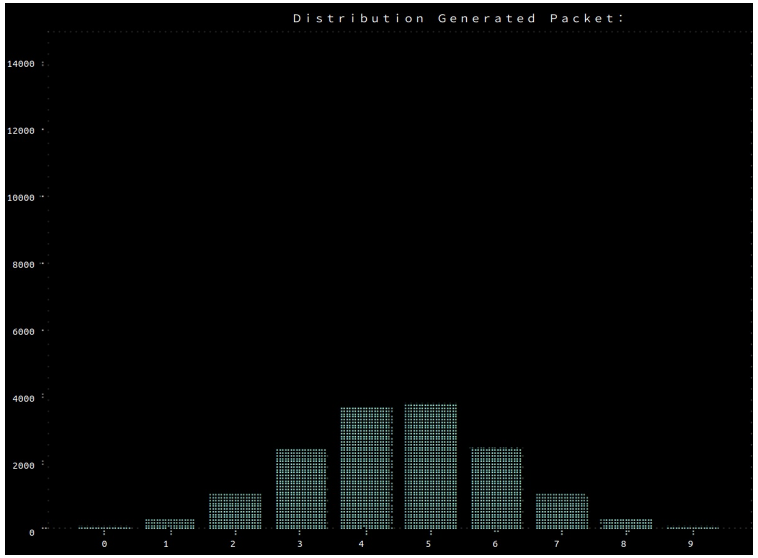
HOMEWORK 1 – MAC PROTOCOL

How the packets are generated? We use Stationary probability (mostly)

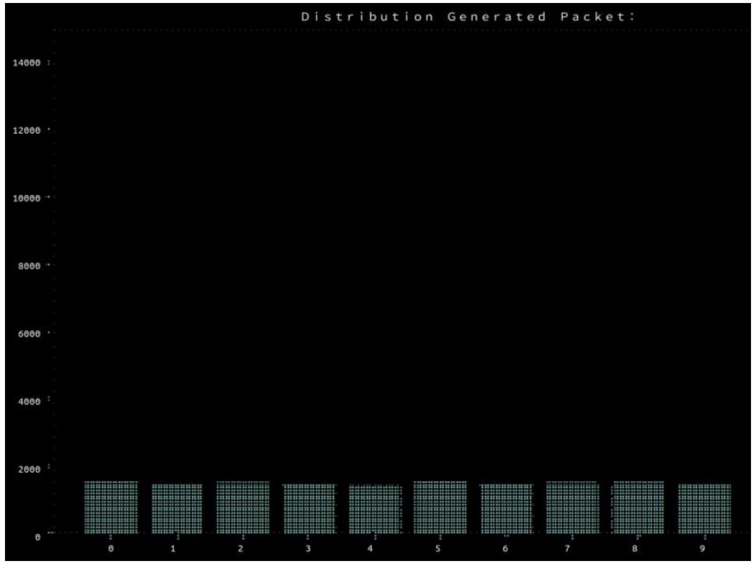
```
class GenerationPattern(Enum):
    GAUSSIAN = 0
    UNIFORM = 1
    FIXED_PROB = 2

RANDOM_GENERATION_PATTERN = GenerationPattern.UNIFORM
```

Select the generation pattern

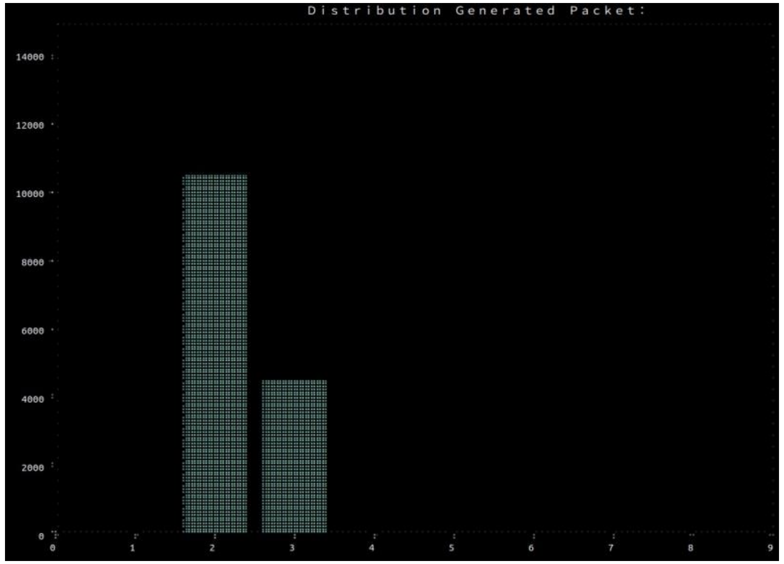
Gaussian



Uniform



With fixed probabilities



HOMEWORK 1 – MAC PROTOCOL

How to compare your approah/protocol? **./src/experiments/run_mac_experiments.sh**

Try different number of drones

The algorithms to compare

The effectively code to run.
Remove comments to run extensive simulations;
also set i_s = 0 and e_s 10.

```
#test baselines
for nd in "2" "5" "8" "10" "15";  # number of drones
do
    for alg in "RND" "RR" "WEIGHT" "AI";
    do
        echo "run: ${alg} - ndrones ${nd} "
        python3 -m src.experiments.experiment_mac -nd ${nd} -i_s 0 -e_s 2 -alg ${alg} &
        #python3 -m src.experiments.experiment_mac -nd ${nd} -i_s 10 -e_s 20 -alg ${alg} &
        #python3 -m src.experiments.experiment_mac -nd ${nd} -i_s 20 -e_s 30 -alg ${alg} &
        #wait
    done;
    #wait
done;
wait
```

Avoid to much parallelism!

How many seed to run (# experiments). range(i_s, e_s)

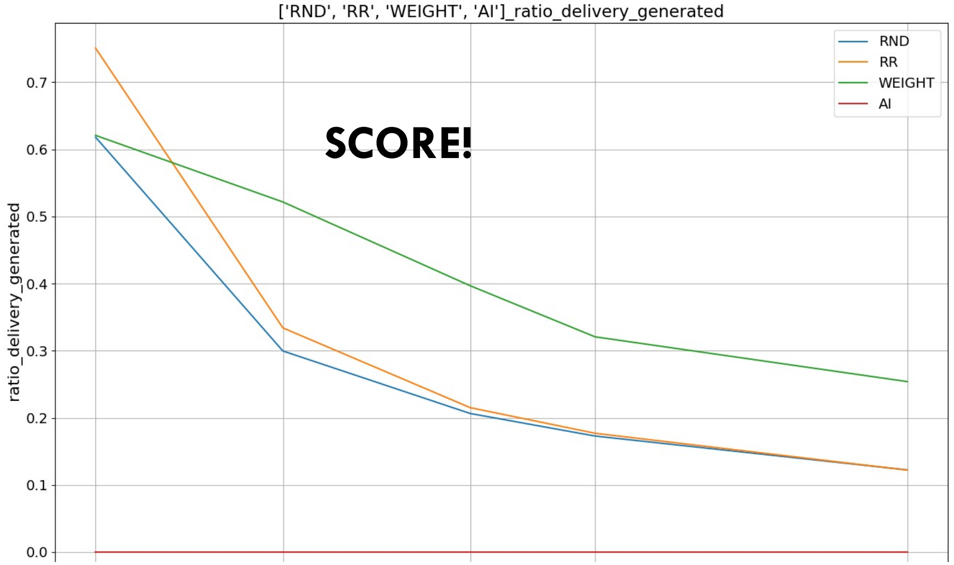
Plot code

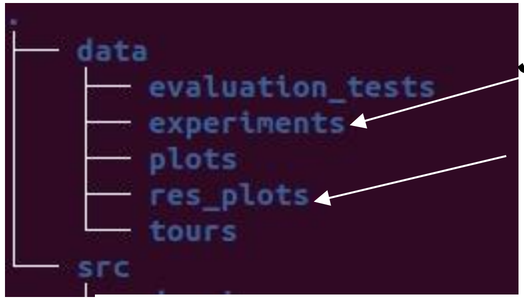
The number of seed used

Experiments to plot

```
python3 -m src.experiments.json_and_plot -nd 2 -nd 5 -nd 8 -nd 10 -nd 15  -i_s 0 -e_s 2 -exp_suffix RND -exp_suffix RR -exp_suffix WEIGHT -exp_suffix AI
```

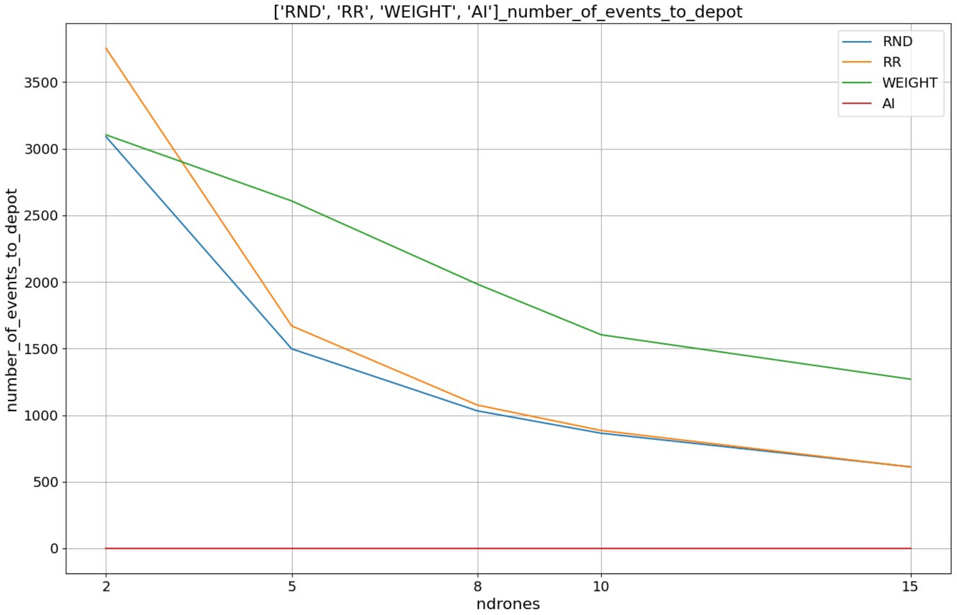
HOMEWORK 1
MAC PROTOCOL - RESULTS

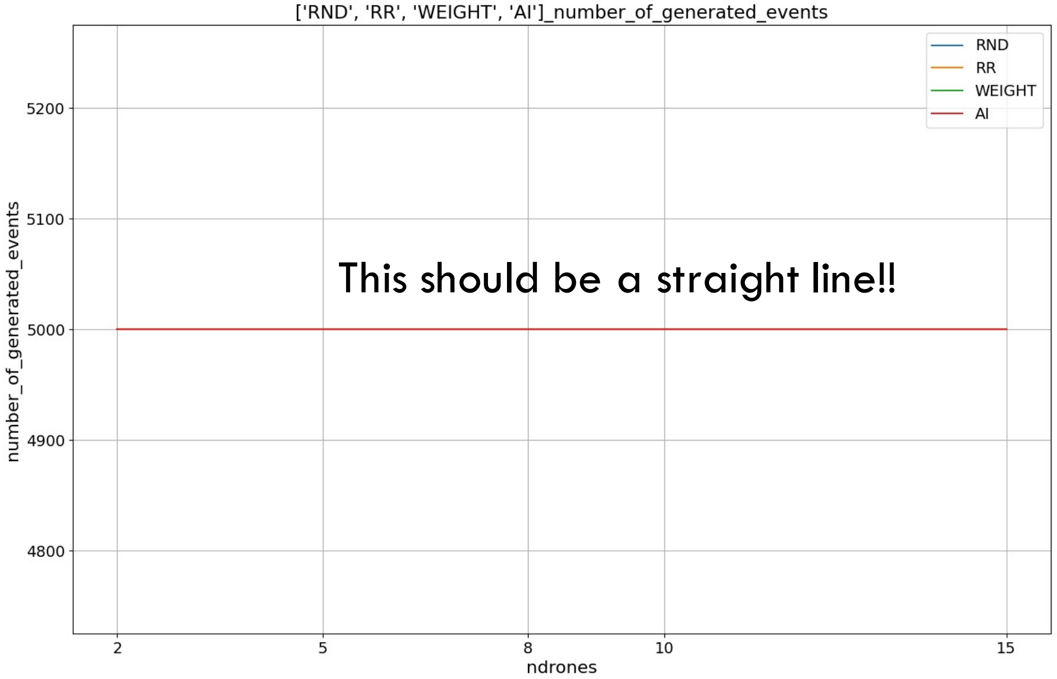




Json files with all the data

Plot with metrics





HOMEWORK 1!!

Deadline: 23:59:59 - 03/12/2020

How : in group of **1** to **3**

Submission: report + code

Score: up to 31 (30 cum laude)

Interaction lecture: 10:00 - 26/11/2020

FAQ: Common questions will be answered on classroom, above the homework post

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

Challenge: rank of top 10 algorithms.

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 20

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** (to be decided)

Note:

[1] – grade can eventually increase up to 26.

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

HOMEWORK 1 - SUBMISSION

How:

- **email** (**subject=**[Autonomous Networking - A.A. 2020-2021] – 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).
- 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_MAC", then, the delivery will be:
Zip: 999_black_01_donald.zip
Inside the zip:

- ***999_black_01_donald_report.txt***
- ***999_black_01_donald_mac_protocol.py*** (which contains the algorithm class called "**X_MAC_999**")

Currently the file algorithm is called "ai_depot_mac.py" and the algorithm is called AIDepotMAC.
Submit only this file but change 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 (if needed, not mandatory):

- 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 interaction to understand your score/position: https://bit.ly/hmw1_score

You can send to me your score with an email, subject ***'[Autonomous Networking - A.A. 2020-2021] – score'***

CONTACTS

Andrea Coletta:

coletta@di.uniroma1.it