AUTONOMOUS NETWORKING

PRESENTATION HW1-HW2

Master in Computer Science
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
La Sapienza Università di Roma
Academic year 2022-2023

Ballanti Chiara
ballanti.1844613@studenti.uniroma1.it

Bevilacqua Paolo Pio
bevilacqua.2002288@studenti.uniroma1.it

De Sio Ilaria
desio.2064970@studenti.uniroma1.it

PROBLEM DEFINITION

Problem

- **N drones** deployed in Area of interest (AoI)
- Send packet to **Depot D**
- Packet expiring after *t* time-steps
  - [HW1] *t* = 2000 ts
  - [HW2] *t* = 1500 ts

Aim

- [HW1] Implement an algorithm that allows multi-hop drone-to-depot (node-to-sink) communication.
- [HW2] Implement a state-of-the-art algorithm to intelligently guide the drones in a choice[1].

- Maximize the **number of packets delivered** to the depot on time while minimizing delivery time.

[1]Ardeep: Adaptive and reliable routing protocol.

HOW IT WORKS

Agent

Observations
The agent receives the observations from the environment

Relay selection
The agent chooses an action to select a relay for packets

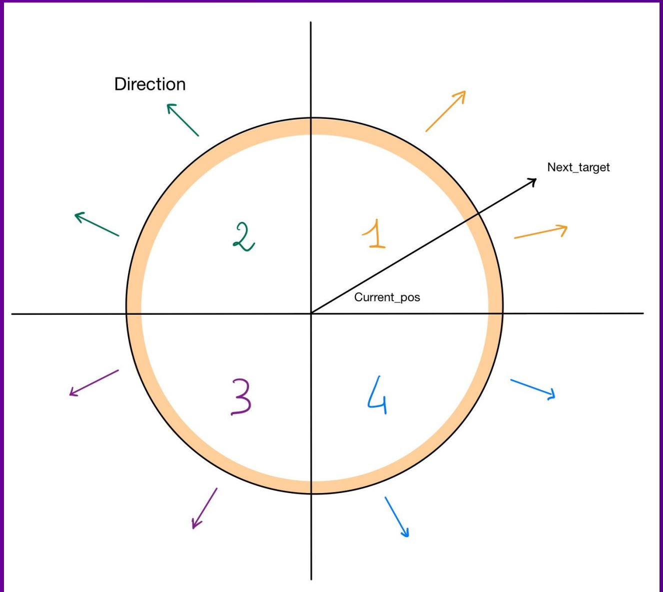
Action
The agent performs the chosen action

Feedback
The agent receives a reward on the performed action

HOMEWORK 1

STATES

State = (direction of the drone, index of the cell)



ACTIONS

- **Keep**: the drone decides to keep its packets.
- **Move:** the drone decides to send its packets to one of the neighbors.

RELAY SELECTION

We chose the **epsilon-greedy strategy** to decide if the agent should:

Exploration

$$A_t \sim \text{Uniform}(\{a_1, ..., a_k\})$$

with probability ε

Exploitation

$$A_t \sim \arg\max_a Q(s_t, a)$$

with probability $1 - \varepsilon$

If the chosen action is :

- **Keep,** the drone will ignore its neighbors to keep the packet.
- **Move,** a relay for packets will be selected according to geographic routing using *C2S* criteria.

FEEDBACK

The agent receives a **reward** when a packet expires or arrives to the depot.

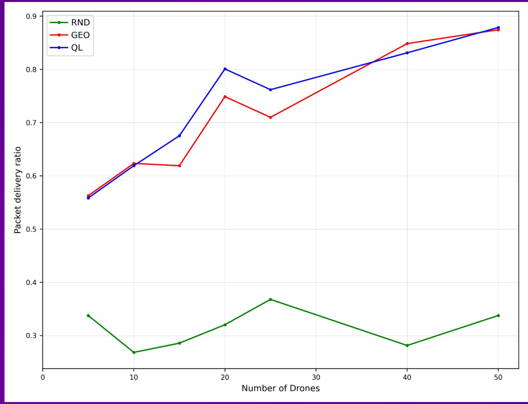
$$reward \leftarrow \begin{cases} \dfrac{(2000 - delay)}{50} & outcome = 1 \quad ☺ ✓ \\ -20 & outcome = -1 \quad ☹ ✗ \end{cases}$$

Once the reward is calculated, it is used to update the Q-table according to the Bellman equation:

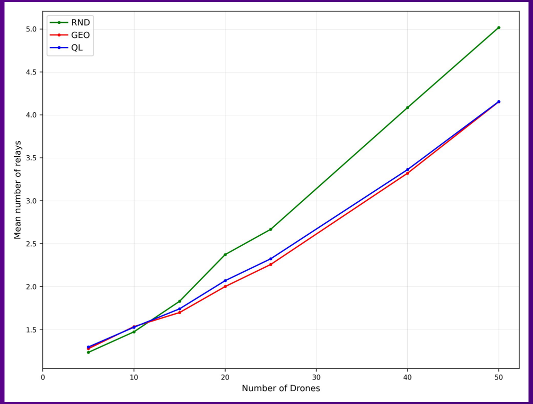
$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \cdot td$$

$$td = reward_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$$

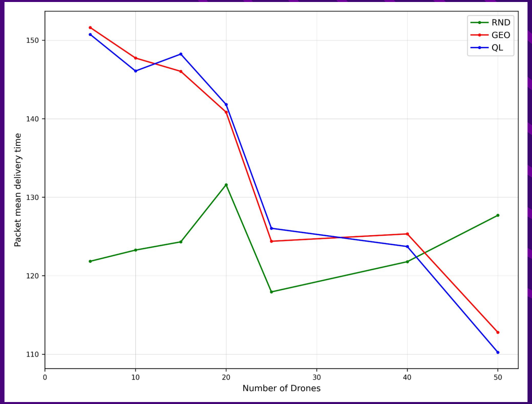
PLOTS AND CONCLUSIONS



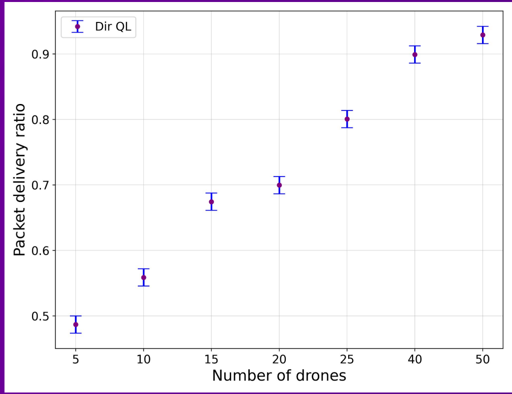
Packet delivery ratio



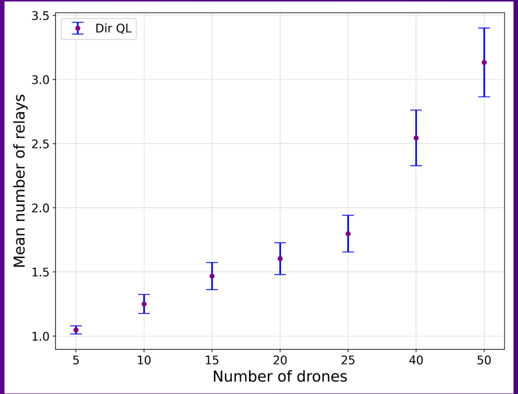
Mean number of relays



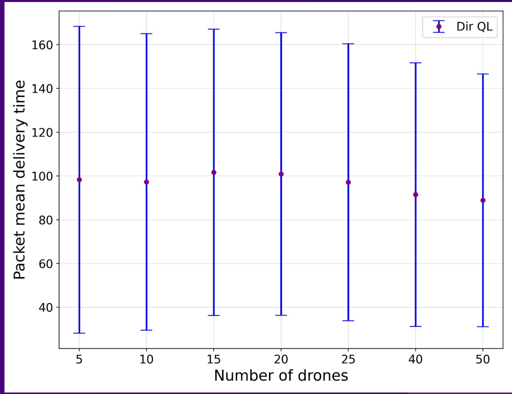
Packet mean delivery time



Packet delivery ratio with STDEV



Mean number of relays with STDEV



Packet mean delivery time with STDEV

HOMEWORK 2

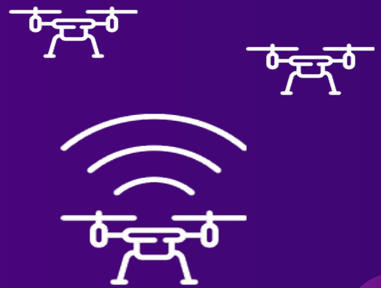
STATES

$$C_{u_i,b_j} = (ct_{u_i,b_j}, PER_{u_i,b_j}, e_{b_j}, d_{b_j,des}, d_{min})$$

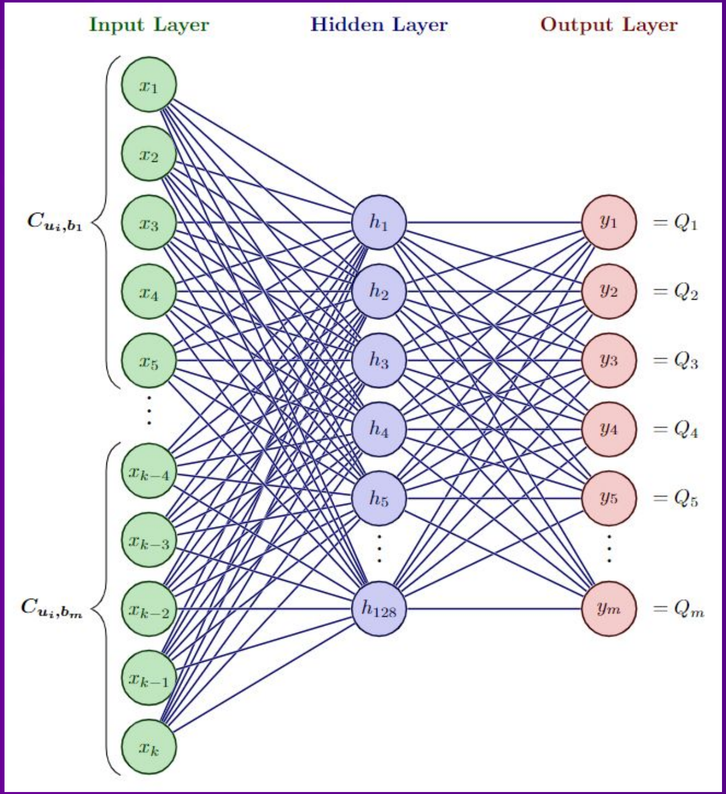
- ct_{u_i,b_j} expected connection time of the link.
- PER_{u_i,b_j} packet error rate of the link.
- e_{b_j} remaining energy of the neighbor.
- $d_{b_j,des}$ distance between the neighbor and destination.
- d_{min} minimum distance between a two hop neighbor and destination.

ACTIONS

Action denotes itself (keeping the packet) or the selected neighbor to forward the packet.



DQN ARCHITECTURE



- *n_observations = n_actions = n_drones*
- *k = n_observations × 5*
- *m = n_actions*
- Q_i is the **Q-value** related to the neighbor b_i
- Activation functions: ***Leaky ReLu***
- **Mask** to filter the real neighbors

RELAY SELECTION

We chose the **epsilon-greedy strategy** to decide if the agent should:

Exploration

$$A_t \sim \mathrm{Geo}(\{\mathrm{a}_1, \ldots, a_k\})$$

with probability ε

Exploitation

$$A_t \sim \arg\max_a DQN(s_t, a)$$

with probability $1 - \varepsilon$

In order to record the performed actions we use a data structure called **taken actions**, described as follows:

$$taken\_actions = \{id\_event : (current\_state,\ action,\ next\_state)\}$$

FEEDBACK

The agent receives a **reward** when a packet expires or arrives to the depot.
The reliable distance is calculated as:

$$D_{i,j} = \frac{d_{u_i,des}}{d_{b_j,des}} * (1 - PER_{u_i,b_j}) * \beta \qquad \text{with } \beta = \begin{cases} 1 & ct_{u_i,b_j} \geq ct_{min} \\ 0 & ct_{u_i,b_j} < ct_{min} \end{cases}$$

And the **reward function** is defined as:

$$r_t = \begin{cases} R_{max} & \text{when } \textit{outcome} = 1 \text{ and the neighbor } b_j \text{ is the destination} \\ x & \text{when } \textit{outcome} = 1 \text{ and the neighbor } b_j \text{ is not the destination} \\ -R_{max} & \text{when } \textit{outcome} = -1 \text{ and the neighbor } b_j \text{ is a local minimum} \\ -(1-x) & \text{when } \textit{outcome} = -1 \text{ and the neighbor } b_j \text{ is not a local minimum} \end{cases}$$

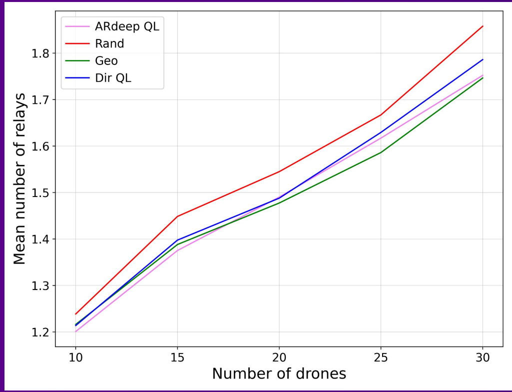
$$\text{where } \; x = \omega \; D_{u_i,b_j} + (1-\omega)\left(\frac{e_{b_j}}{E_{b_j}}\right)$$

SIMULATOR CHANGES

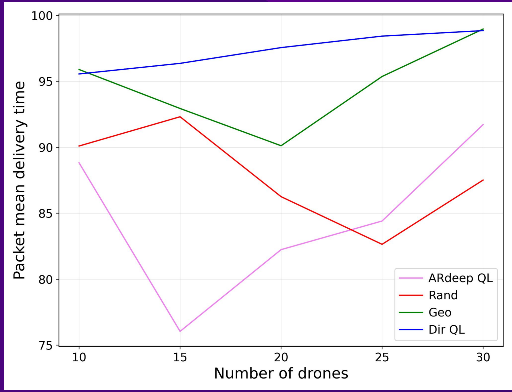
Expected connection time of the link

Remaining energy of the neighbor

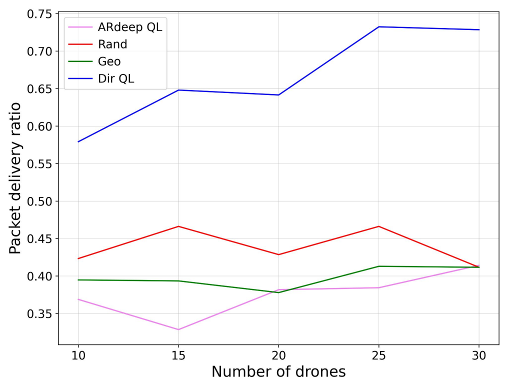
PLOTS AND CONCLUSIONS



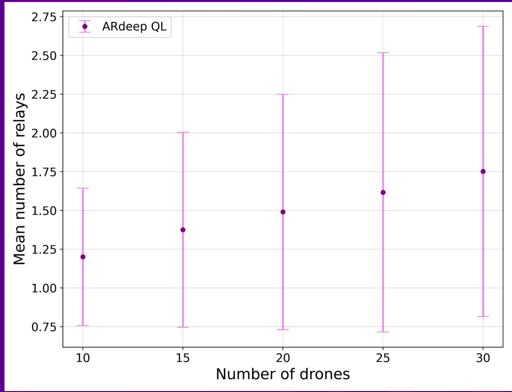
Mean number of relays



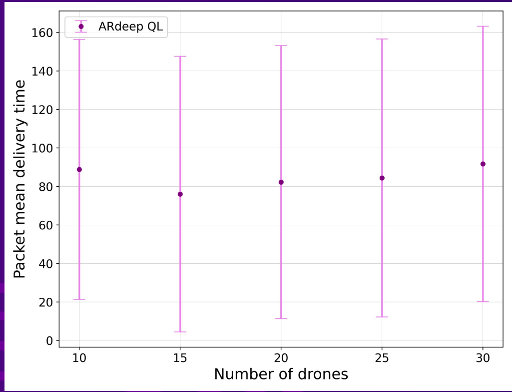
Packet mean delivery time



Packet delivery ratio



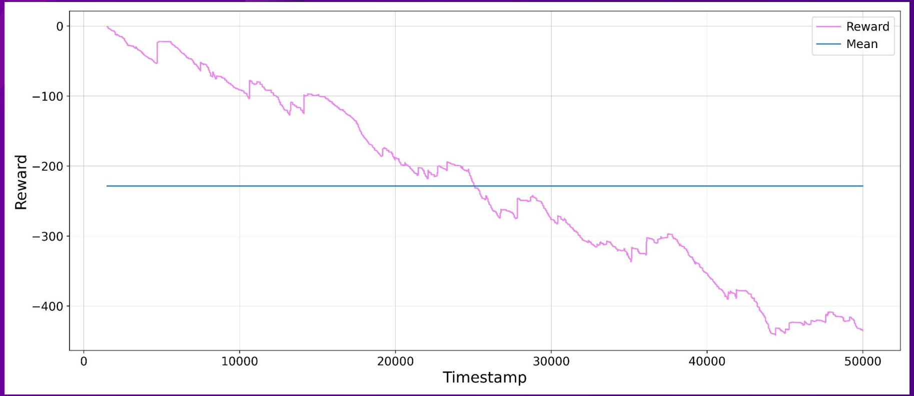
Mean number of relays with STDEV



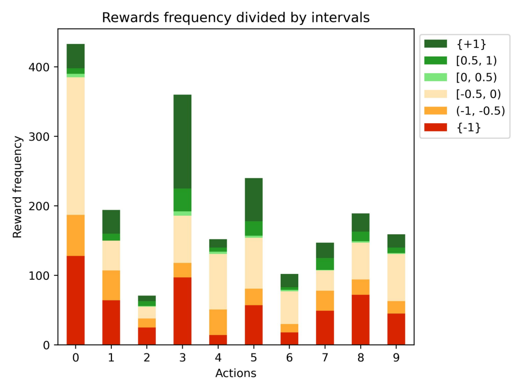
Packet delivery time with STDEV

HOMEWORK 2

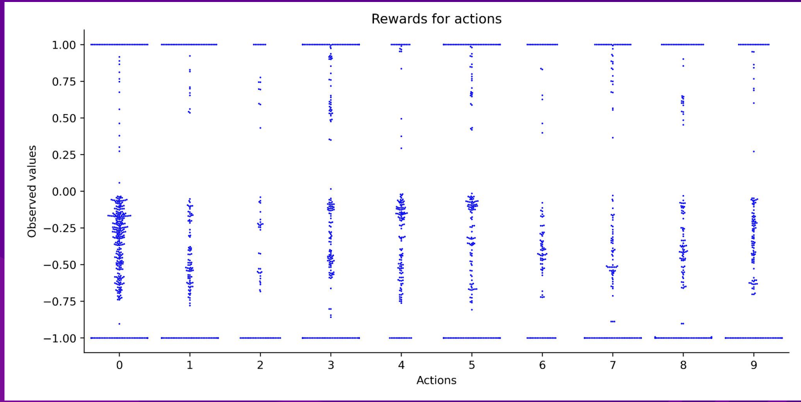
PLOTS AND CONCLUSIONS



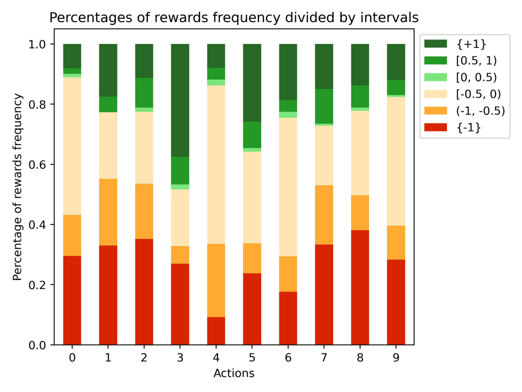
Sum of the rewards at each time step



Reward frequency per actions



Swarm plot of the rewards given to actions



Reward frequency per actions (%)

REFERENCES

[1] Jianmin LIU, Qi WANG, Chentao HE, and Yongjun XU. Ardeep: Adaptive and reliable routing protocol for mobile robotic networks with deep reinforcement learning. In *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, pages 465–468, 2020.

THANKS
FOR THE ATTENTION

Master in Computer Science
Facoltà di Ingegneria dell’Informazione, Informatica e Statistica
La Sapienza Università di Roma
Academic year 2022-2023

Ballanti Chiara
ballanti.1844613@studenti.uniroma1.it

Bevilacqua Paolo Pio
bevilacqua.2002288@studenti.uniroma1.it

De Sio Ilaria
desio.2064970@studenti.uniroma1.it

Sapienza
Università di Roma