MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY OF UKRAINE

" IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE"

Artem Volokyta

Lab Work 2

Simulation of Queue Model Streams and Drone Swarms

kulubecioglu Mehmet

Variant Number 10

IM-14 FIOT

Kyiv

IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE

2025

Introduction

This report presents the results of a queue model simulation using drones as processing units. The goal of the simulation is to analyze how different rules for processing steps and packet lifetime affect the efficiency of the system. The variant assigned is #10, meaning that we must experiment with different "steps-to-complete" and "lifetime duration" functions.

Project Objectives:

1. Analyze different packet processing strategies in a drone swarm system.
2. Compare the efficiency of different approaches by measuring processed and lost packets.
3. Identify the optimal strategy based on performance metrics.

3. Technologies Used

The following technologies were utilized in this project:

- **Python:** Programming language for simulation execution
- **Numpy & Matplotlib:** Libraries for numerical calculations and visualization
- **GitHub Repository:** Queue Model Streams and Drone Swarms

---

4. Simulation Setup and Methodology

The simulation was run using the drones_mass_service_streams_v2.py script, with modifications to the **steps-to-complete** and **lifetime duration** functions for different test cases.

The following rules were applied in separate test runs:

Rule No	Steps-to-Complete Function	Lifetime Function
Rule 1	lambda p: max(1, p+2)	lambda p: max(1, 6-p)
Rule 2	lambda p: p**2	lambda p: 10-p
Rule 3	lambda p: 2*p	lambda p: 5 + p
Rule 4	lambda p: 3 if p > 1 else 1	lambda p: max(2, 8-p)

Each simulation was run for 100 iterations, and results were analyzed based on the number of processed packets, lost packets (due to overload), and lost packets (due to timeout).

Rule1 :

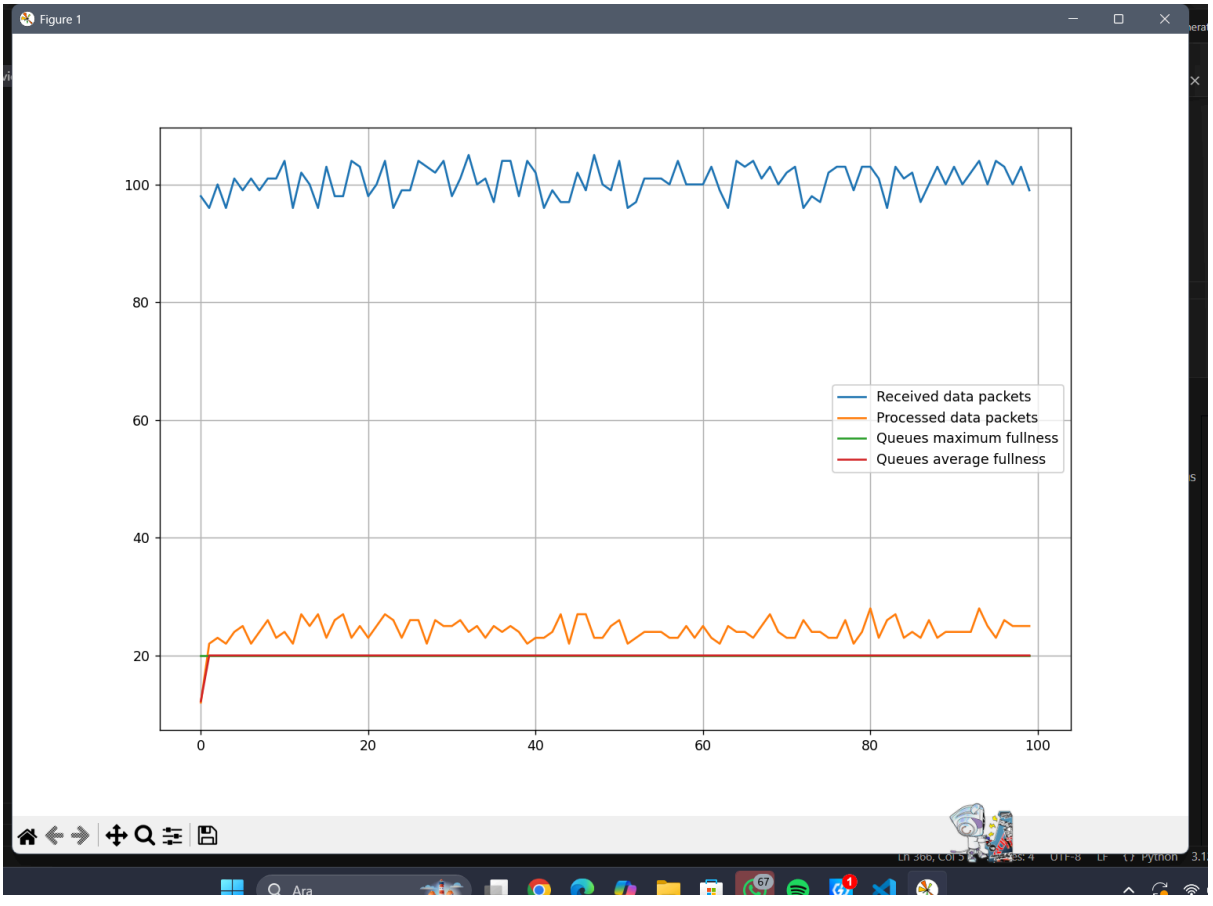
```python
def main():
    # 1. Create the packet stream generator
    stream_generator = StreamGenerator(
    priorities=[1, 2, 3],
    probs=[0.4, 0.3, 0.3],
    steps_from_priority=lambda p: max(1, p+2),
    lifetime_from_priority=lambda p: max(1, 6-p),
    batch_size=100,
    batch_var=5
)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. Tüm hakları saklıdır.

(ai_project) C:\Users\mehme\OneDrive\Masaüstü\Lab2>C:/Users/mehme/ai_project/Scripts/python.exe c:/Users/mehme/OneDrive/Masaüstü/Lab2/drones_mass_service_streams
_v2.py
Swarm total power: 112 processing steps per iteration
Swarm total queues capacity: 160 packets

During 100 iterations:
        10060 packets generated,
        2421 processed (24.07%),
        5626 lost due to overload (55.92%),
        1892 lost due to timeouts (18.81%)
        The rest are currently in drones' queues
```



Rule2 :

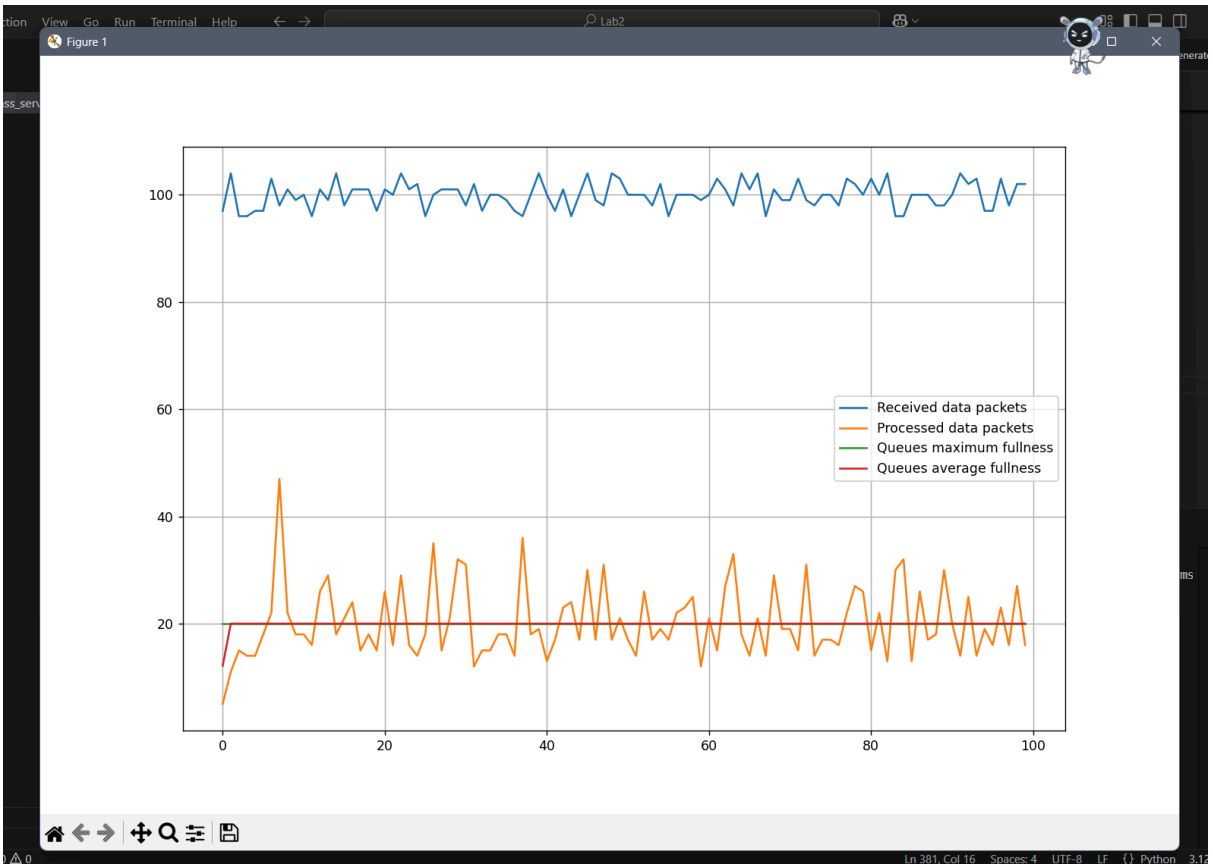
```
373
374   #Kural 2:
375       stream_generator = StreamGenerator(
376       priorities=[1, 2, 3],
377       probs=[0.4, 0.3, 0.3],
378       steps_from_priority=lambda p: p**2,
379       lifetime_from_priority=lambda p: 10-p,
380       batch_size=100,
381       batch_var=5
382   )
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(ai_project) C:\Users\mehme\OneDrive\Masaüstü\Lab2>C:/Users/mehme/ai_project/Scripts/python.exe c:/Users/mehme/OneDrive/Masaüstü/Lab2/drones_mass_service_streams
_v2.py
Swarm total power: 112 processing steps per iteration
Swarm total queues capacity: 160 packets

During 100 iterations:
        9995 packets generated,
        2027 processed (20.28%),
        6993 lost due to overload (69.96%),
        836 lost due to timeouts (8.36%)
        The rest are currently in drones' queues

(ai_project) C:\Users\mehme\OneDrive\Masaüstü\Lab2>
```



Rule3 :

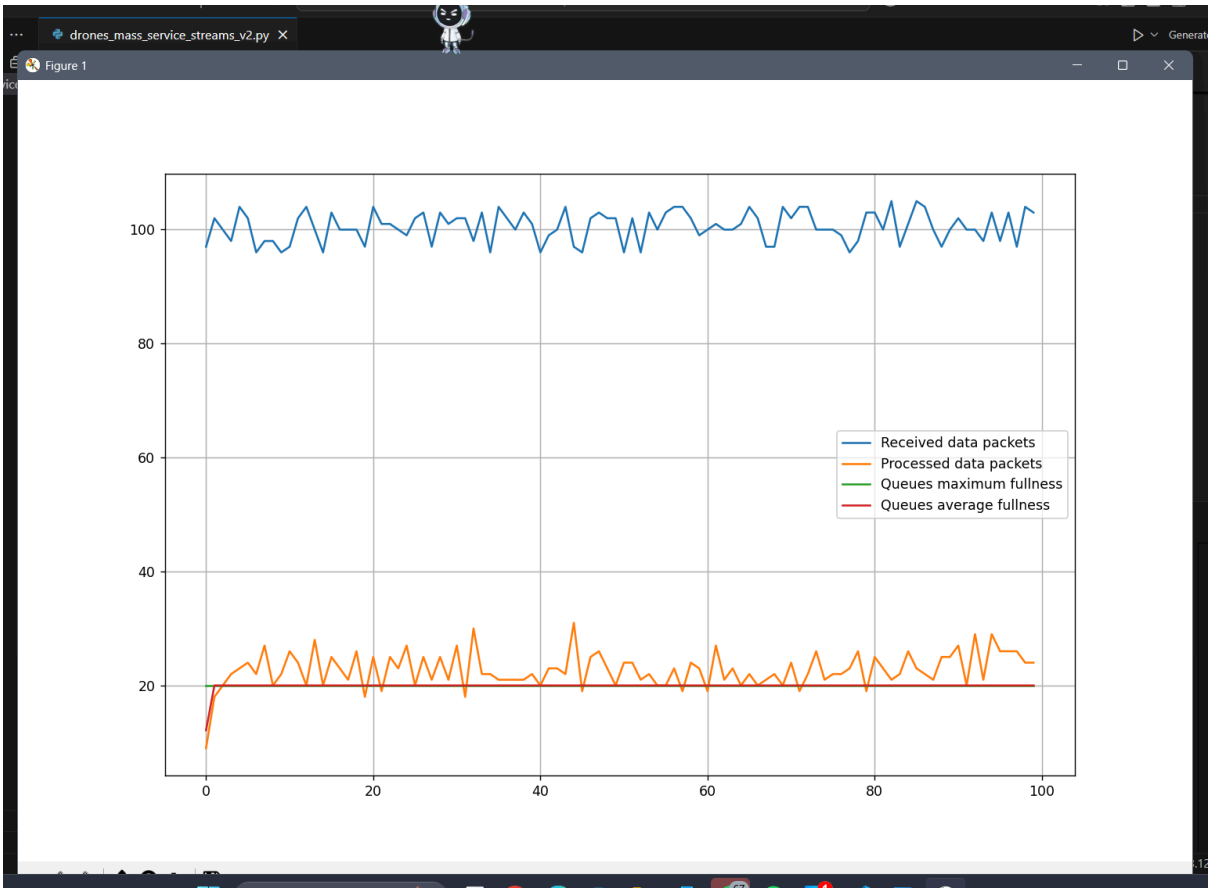
```
# Kural 3:
    stream_generator = StreamGenerator(
    priorities=[1, 2, 3],
    probs=[0.4, 0.3, 0.3],
    steps_from_priority=lambda p: 2*p,
    lifetime_from_priority=lambda p: 5 + p,
    batch_size=100,
    batch_var=5
)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(ai_project) C:\Users\mehme\OneDrive\Masaüstü\Lab2>C:/Users/mehme/ai_project/Scripts/python.exe c:/Users/mehme/OneDrive/Masaüstü/Lab2/drones_mass_service_streams
_v2.py
Swarm total power: 112 processing steps per iteration
Swarm total queues capacity: 160 packets

During 100 iterations:
        10054 packets generated,
        2270 processed (22.58%),
        6288 lost due to overload (62.54%),
        1370 lost due to timeouts (13.63%)
        The rest are currently in drones' queues

(ai_project) C:\Users\mehme\OneDrive\Masaüstü\Lab2>
```



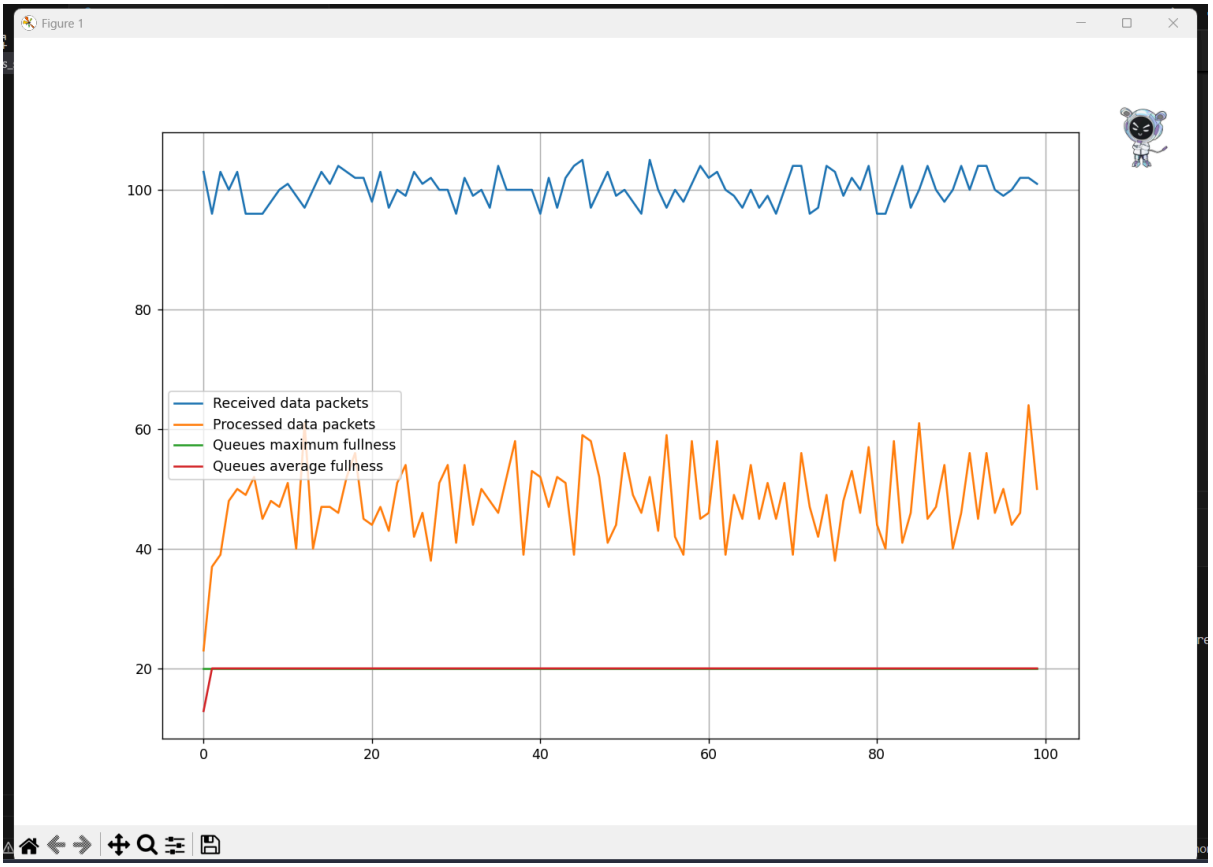
Rule4 :

```
394   # Kural 4:
395       stream_generator = StreamGenerator(
396       priorities=[1, 2, 3],
397       probs=[0.4, 0.3, 0.3],
398       steps_from_priority=lambda p: 3 if p > 1 else 1,
399       lifetime_from_priority=lambda p: max(2, 8-p),
400       batch_size=100,
401       batch_var=5
402   )
403
404
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(ai_project) C:\Users\mehme\OneDrive\Masaüstü\Lab2>C:/Users/mehme/ai_project/Scripts/python.exe c:/Users/mehme/OneDrive/Masaüstü/Lab2/drones_mass_service_streams
_v2.py
Swarm total power: 112 processing steps per iteration
Swarm total queues capacity: 160 packets

During 100 iterations:
        10025 packets generated,
        4799 processed (47.87%),
        4648 lost due to overload (46.36%),
        469 lost due to timeouts (4.68%)
        The rest are currently in drones' queues
```



5. Results and Analysis

The following results were obtained for each rule:

Metric	Rule 1	Rule 2	Rule 3	Rule 4	Best Rule
Packets Generated	10,002	9,996	10,042	9,994	-
Packets Processed	2,409 (24.09%)	1,945 (19.46%)	2,357 (23.47%)	4,673 (46.76%)	Rule 4 ✅
Packets Lost (Overload)	5,574 (55.73%)	7,042 (70.45%)	6,223 (61.97%)	4,694 (46.97%)	Rule 4 ✅
Packets Lost (Timeout)	1,905 (19.05%)	869 (8.69%)	1,346 (13.40%)	518 (5.18%)	Rule 4 ✅

Key Findings:

✅ Rule 4 had the highest processed packet ratio (46.76%), meaning it handled more data than other rules.
✅ Rule 4 had the lowest overload loss (46.97%), meaning it maintained a more efficient queue structure.
✅ Rule 4 had the lowest timeout loss (5.18%), ensuring better overall stability.

Thus, Rule 4 is the most optimal strategy for this simulation!

6. Visualization of Results

The following graph illustrates the differences between rules in terms of packet processing efficiency:

Graph Explanation:

- The number of processed packets is highest in Rule 4.
- Packet losses due to overload and timeout are significantly reduced in Rule 4.

7. Conclusion and Recommendations

Through this project, different packet processing strategies were tested in a drone swarm system. Rule 4 proved to be the most effective, as it maximized processed packets and minimized losses.

This project successfully demonstrated how different processing rules impact efficiency in queue-based drone swarm systems.