Task Scheduling for Heterogeneous Swarm of Drones

Ariana Bruno & Justin Hu

Advised by Prof. Christina Delimitrou

Introduction

The project has three main goals:

1. Setup and develop applications for a swarm of drones: image recognition, obstacle avoidance and coordinated movement.
2. Benchmark local versus remote execution performance between the drone and a backend cloud
3. Design centralized swarm coordination control system and compare with decentralized approaches for different scheduling policies.

Swarm Applications

The application for the swarm of drones is defined as a coordinated sequence of tasks that is split up and divided among the drones.

The ideal swarm application is for each drone in the swarm to execute a route, performing object recognition and obstacle avoidance along the way.

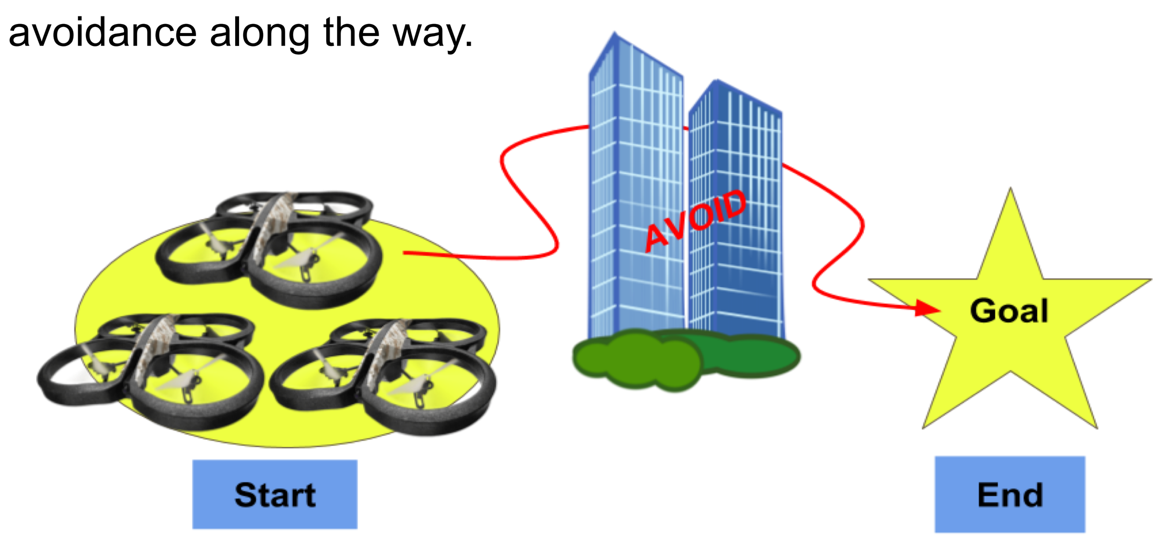

Fig. 1: Illustration of the Swarm of Drone Application

System Specifications

Table 1: Drone Specifications

Drone Type:	Parrot AR Drone 2.0
Operating System:	Linux 2.6.32
Available Memory:	11 GB
RAM:	1GB at 200 MHz
Connection:	WiFi

Table 2: Software Specifications

Framework:	Node.js
Open Source Libraries:	ardrone_autonomy
	Cylon
	OpenCV

Swarm Task Scheduler

The task scheduler is developed to assign tasks to the drones and performs in two ways:

- **Push**: the manager can push tasks to drones via the worker.
- **Pull**: the worker pulls tasks from a queue created by the manager.

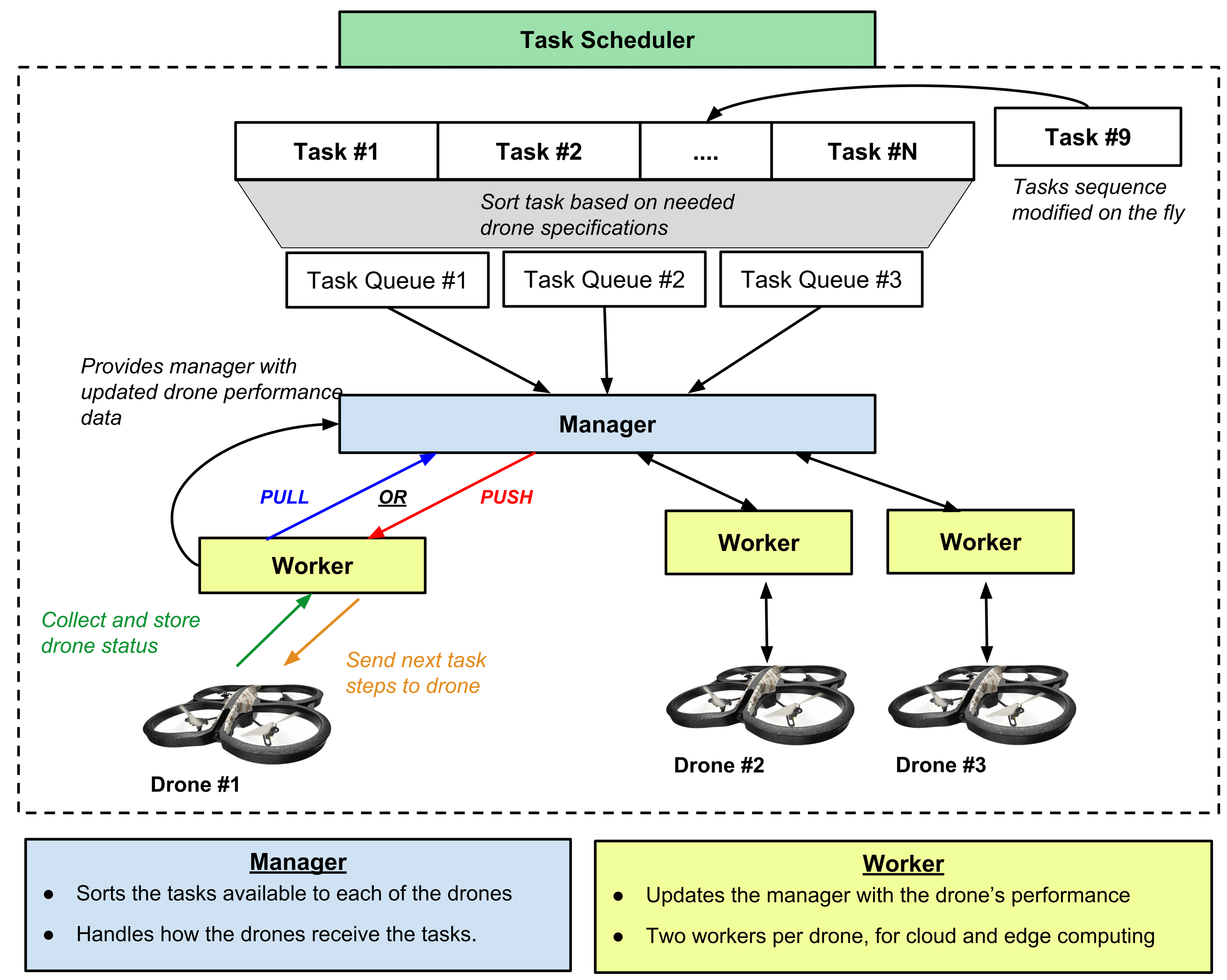

Fig. 2: Illustration of the tasks scheduler process for the swarm of drone

Execution Environment

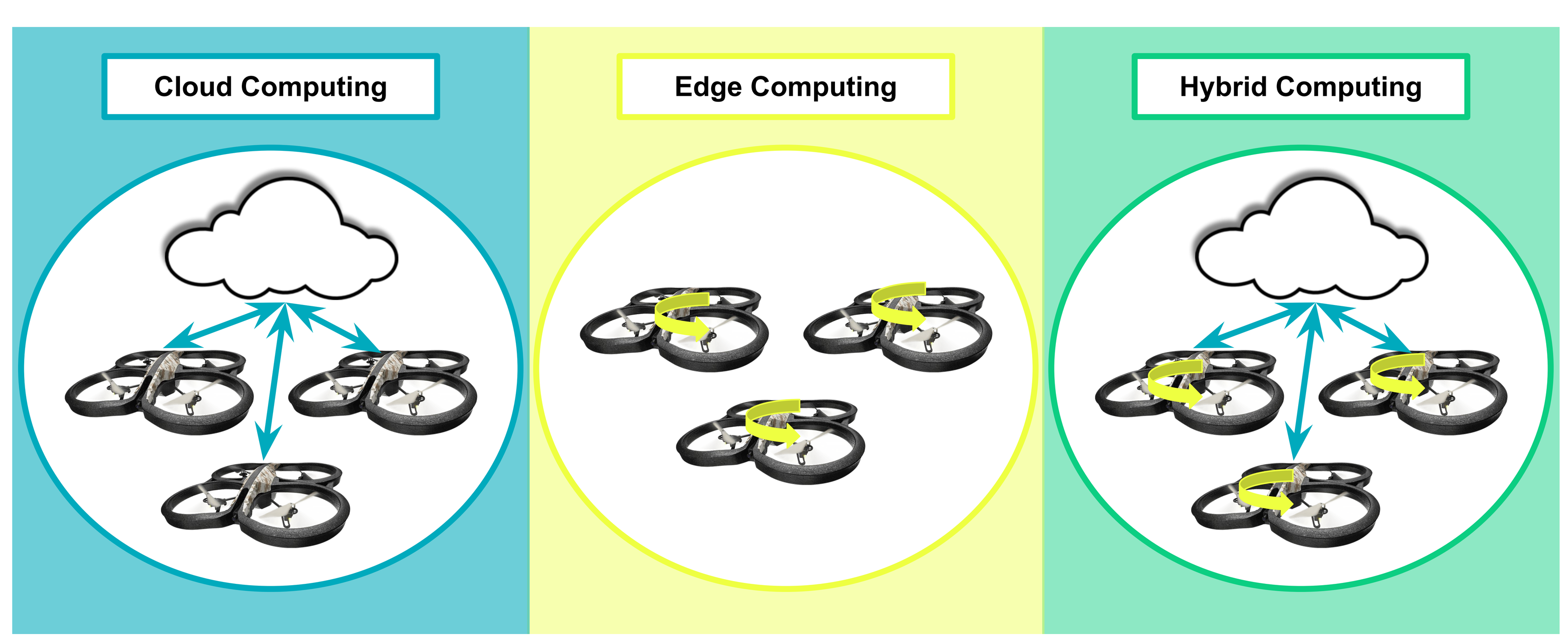

Fig. 3: Illustration of the cloud, edge, and hybrid computing for the Swarm

- Cloud computing: all computation and execution required for the drone's task occurs in the cloud.
- Edge computing: all the computation and execution occurs on the drone's operating system.
- Hybrid computing: the computation and execution can occur both in the cloud or on the drone.

Results

Benchmark results were collected for, network latency, TCP latency, drone movement latency, drone detection latency, and power consumption for drone movement and drone detection.

Table 3: Network Latency - Ping

Connection Type	Average (ms)	St. Dev. (ms)
Direct connection to Drone	1.720	2.086
Connection to Drone through Router	16.229	30.140

Fig. 4: Drone's Average TCP Latency

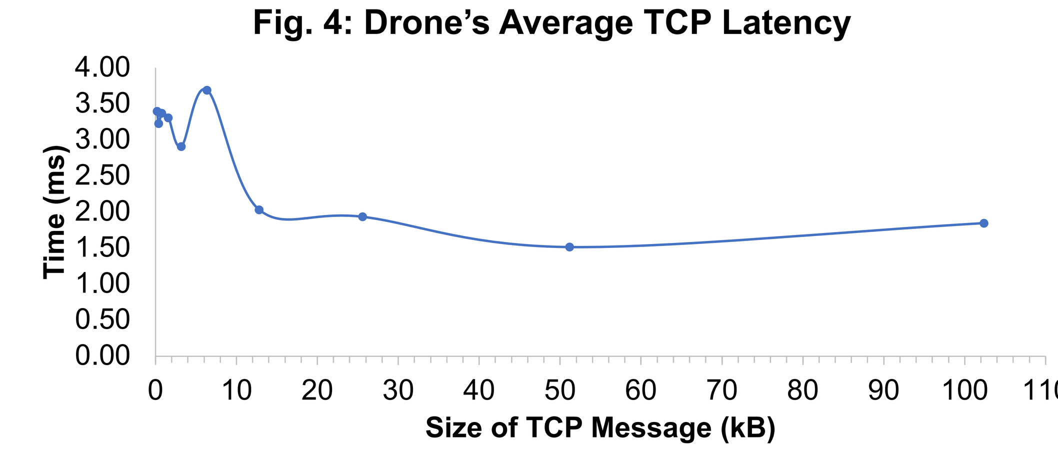

Table 4: Detection Latency

Detection Type	Average (ms)	St. Dev. (ms)
On Drone Tag Detection	32.093	231.861
OpenCV Cloud Tag Detection	75.096	39.705
OpenCV Cloud Face Detection	94.904	32.663

Table 5: Movement Latency & Battery Power Consumption

Movement	Avg. (s)	St. Dev. (s)
Take off	7.365	0.396
Turn	3.089	0.460
Horizontal Motion	2.432	0.681
Vertical Motion	5.231	1.271
Land	1.202	0.055

Table 6: Battery Power Consumption

Task Type	Average (%)	St. Dev. (%)
On Drone Tag Detection for 1 min.	<1%	<1%
OpenCV Cloud Tag Detection for 1 min.	<1%	<1%
Constant Drone Movement for 1 min.	10%	1.4%

Future Work

1. Further optimize how the task scheduler prioritizes tasks for drones based on each drone's performance status.
2. Investigate the centralized coordination control system for scalability challenges and quantify the results to improve the control system to be more practical for large-scale swarms
3. Improve object detection and obstacle avoidance procedures to handle more practical events such as building obstruction

CornellEngineering
Electrical and Computer Engineering

Acknowledgements: Thank you Professor Christina Delimitrou, we truly appreciate all of your help and guidance throughout this research project. Additionally, we would like to thank our fellow graduate students that collaborated on this project, Brian Ritchken (M.Eng ECE '18), and Brendan Jackson (M.Eng ECE '18)

Contact: For additional information or questions please contact:
Ariana Bruno (M.Eng ECE '18) - amb633@cornell.edu
Jusrin Hu (M.Eng CS '18) - amb633@cornell.edu