AFRL-RI-RS-TR-2022-046

SWARMCONTROL: TOWARD ELASTIC, PROGRAMMABLE, OPTIMIZED SWARM NETWORKING

NORTHEASTERN UNIVERSITY

MARCH 2022

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2022-046 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
ELIZABETH S. BENTLEY
Work Unit Manager

/ S /
GREGORY J. HADYNSKI
Assistant Technical Advisor
Computing & Communications Division
Information Directorate

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	END DATE
MARCH 2022	FINAL TECHNICAL REPORT	SEPTEMBER 2018	SEPTEMBER 2021

4. TITLE AND SUBTITLE
SWARMCONTROL: TOWARD ELASTIC, PROGRAMMABLE, OPTIMIZED SWARM NETWORKING

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-18-C-0122	N/A	62788S
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
		R2KQ

6. AUTHOR(S)
Tommaso Melodia

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Northeastern University, Institute for Wireless Internet of Things Dept. of Electrical and Computer Engineering Boston MA 02115	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Air Force Research Laboratory/RITGB 525 Brooks Road Rome NY 13441-4505	RI	AFRL-RI-RS-TR-2022-046

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The main goal of this project has been to design, develop and prototype an automated and self-organizing control framework for Unmanned Aerial Vehicles (UAVs). We have successfully achieved our goal by combining softwarization and abstraction principles, optimization and Artificial Intelligence (AI), which allowed us to develop a prototype that is capable of understanding network operator goals and adapting networking parameters and functionalities to respond to changing environment conditions and guarantee high performance.

15. SUBJECT TERMS
Swarm networking, autonomous swarms, artificial intellgence

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
U	U	U	SAR	20

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
ELIZABETH S. BENTLEY	N/A

Table of Contents

Table of Figures

1. SUMMARY

The main goal of this project has been to design, develop and prototype an automated and self-organizing control framework for Unmanned Aerial Vehicles (UAVs). We have successfully achieved our goal by combining softwarization and abstraction principles, optimization and Artificial Intelligence (AI), which allowed us to develop a prototype that is capable of understanding network operator goals and adapting networking parameters and functionalities to respond to changing environment conditions and guarantee high performance.

2. INTRODUCTION

Intelligent unmanned aerial vehicles (UAVs) have gained momentum as a tool to provide new capabilities, to extend the infrastructure of wireless networks, and to make networks more flexible [1-4]. Thanks to their unique characteristics such as fast deployment, high mobility, processing capabilities, and reduced size, UAVs are an enabling technology for numerous future wireless applications. Among these, increasing network coverage, and providing context-aware network services to users are notable examples.

Despite the above advantages, how to deploy swarms of UAVs that are capable of self-adapting and self-organizing to deliver reliable, high-performance and ubiquitous services is not an easy task. Indeed, the wireless and distributed nature of the network exposes UAVs to interference, obstacles and changing channel conditions that might negatively impact (or, in the worst case, prevent completely) networking operations. These challenges are even more important in applications where success depends on the reliability and efficiency of wireless communications between UAVs, mobile end users, base stations, operations centers, etc.. **In this project, we have advanced the state of the art on networked swarms of UAVs by designing, developing and prototyping SwarmControl, a new software-defined control framework for swarms of UAVs whose goal is to enable automated and intelligent reconfiguration of networking operations to guarantee high performance and adapt to continuously changing network conditions.** SwarmControl combines a variety of optimization and data-driven tools, software-defined radios (SDRs), and distributed network control principles to provide network operators with a unified abstraction of the networking and flight control functionalities. With SwarmControl, the operator can define and implement complex network control problems by specifying high-level control directives and requirements for the UAV networks on a centralized abstraction. SwarmControl (i) constructs a network control problem representation of the directives of the network operator; (ii) decomposes it into a set of distributed sub-problems; and (iii) automatically generates both data-driven and distributed optimization solution algorithms to be executed at individual UAVs.

In the following, we provide a detailed overview of the research activities we have performed in this project, with specific focus on describing the evolution of SwarmControl from an optimization-based framework [2-4] to a data-driven one [3], highlighting how our prototyping efforts and the corresponding experimental results have helped us in demonstrating its effectiveness

3. METHODS, ASSUMPTIONS, AND PROCEDURES

This project evolved across two different phases. The first phase involved model-based optimization where UAVs adjust their location and networking functionalities via traditional optimization tools [2,4,5]. The second phase, instead, concerned research activities where optimization tools have been replaced with model-free AI solutions where the underlying model of the network is learned by a set of cooperating agents (i.e., the UAVs) that coordinate their mobility and networking operations to meet the network operator's goal [3].

3.1. Phase I: A traditional optimization-based approach

The research activities carried out in the first phase of the project focused on the development of a network operating system for swarms of UAVs that leveraged decomposition and distributed optimization theories to achieve network operator goals. For this reason, we have developed the system illustrated in Figure 1.

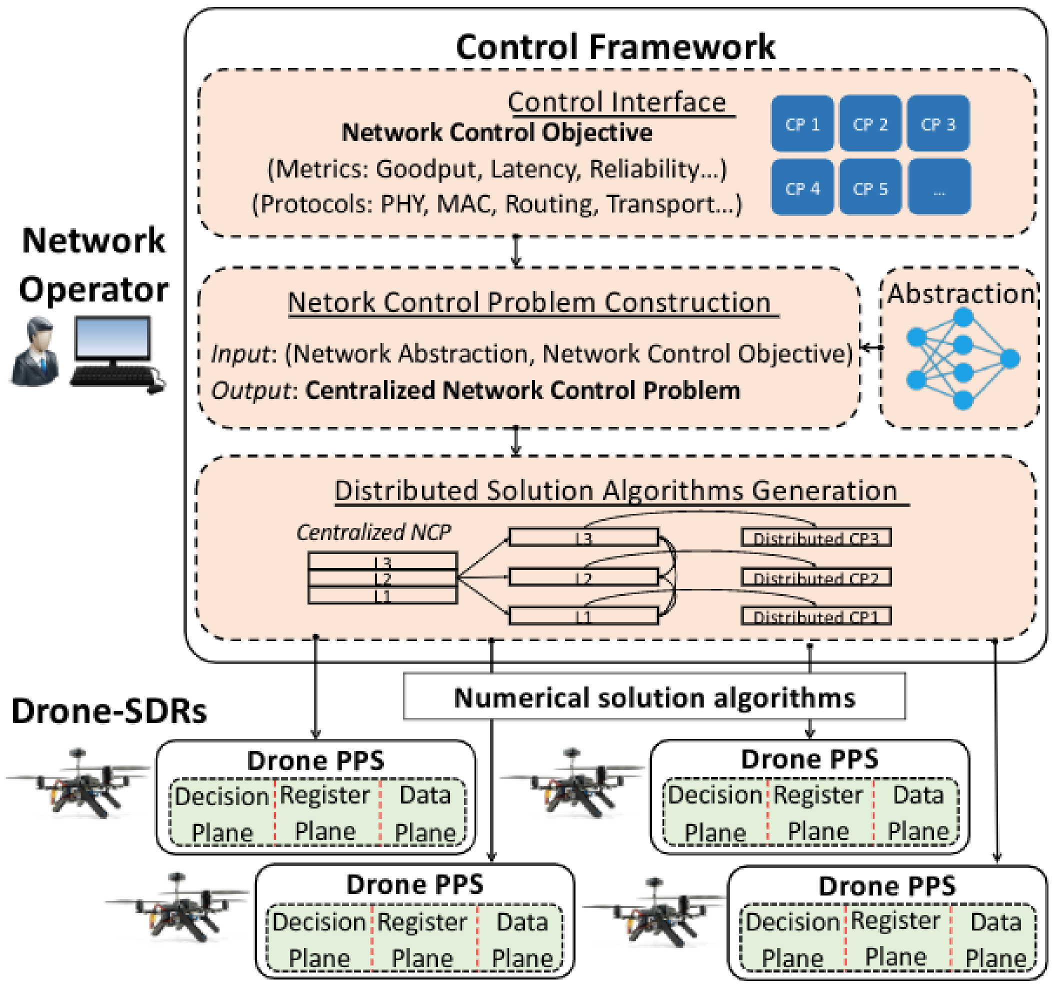

Figure 1 SwarmControl Architecture

The system includes two key components: Control Framework and Drone Programmable Protocol Stack (DPPS).

3.1.1. Control framework

As illustrated in Figure 1, this component is responsible for (i) providing the network operator with a control interface to specify the desired network behavior; (ii) constructing a mathematical Network Control Problem (NCP) representation of the directives of the network operator; and (iii) decomposing the NCP into a set of independent sub-problems and distributing them to individual UAVs.

- **Control Interface.** The interaction with the network operator is implemented through a Control Interface. Through a few input characters on the Control Interface, the network

- operator can specify the desired network behavior, the network protocols to implement, and node-specific constraints. Examples of high-level directives include boosting network performance by maximizing the network throughput, prolonging UAV network lifetime by minimizing energy consumption, ensuring QoS requirements by specifying minimal rate constraints, and covering a particular aerial space, among others. The Control Interface provides the network operator with an abstraction of the UAV network hiding the low-layer network functionalities and details of the underlying network architecture such as the number of UAVs as well as their computing capabilities and battery level, among others. With SwarmControl, controlling a UAV network becomes as simple as choosing among pre-defined control templates, selecting the preferred network protocols, and specifying individual-node constraints.
- **Network Control Problem Construction.** Once the network control problem has been defined, SwarmControl transforms the network operator's directives and requirements into a set of mathematical expressions, which are then merged and rearranged into a NCP. The resulting NCP is a centralized representation of the high-level network behaviors defined by the network operator through the Control Interface, spanning both the networking and the flight control domains, involving multiple nodes and all layers of the protocol stack.
- **Distributed Solution Algorithms Generation**. The cross-layer nature of the NCP obtained through the Network Control Problem Construction and the coupling among its variables make it hard to compute a desirable solution in a distributed manner. To address this challenge, SwarmControl employs horizontal and vertical decomposition theories to decouple the NCP with cross-layer and cross-node dependencies into a set of distributed sub-problems, each involving only a single network node and a single layer of the protocol stack. For each of the resulting sub-problems, SwarmControl generates in an automated manner a distributed numerical solution algorithm, which is then forwarded to individual UAVs and executed at network run time based on local network state information.
- **Drone programmable protocol stack (DPPS).** As shown in Figure 1, the DPPS is installed at each individual UAV to solve the numerical solution algorithms received from the Control Framework in a distributed and automated fashion. The DPPS spans all layers of the network protocol stack and tightly interacts with the flight controller firmware. The DPPS provides the building blocks and primitives necessary to prototype complex cross-layer and cross-domain network protocols, allowing complete control of the network, sensing and motion parameters at all layers of the protocol stack.

•

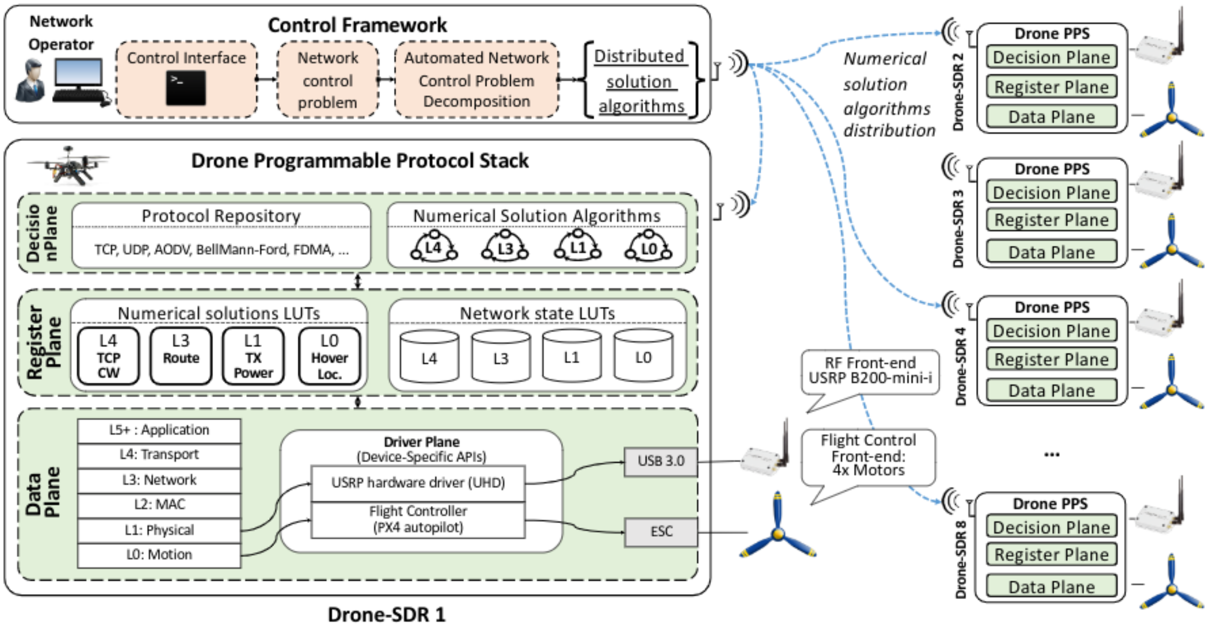

Figure 2 Overview of the SwarmControl DPPS

3.1.2. Details of the DPPS

The SwarmControl DPPS receives the distributed numerical solution algorithms (e.g., motion solution algorithm, transport rate solution algorithm) to be executed at individual UAVs and runs them at its Decision Plane as shown in Figure 2. This plane features a Protocol Repository containing the software implementations of different network protocols and motion strategies (e.g., TCP, Bellman-Ford routing algorithm), together with the mathematical solvers to run the dispatched scripts.

The Decision Plane oversees running the distributed optimization algorithms in real-time, using up-to-date network state and motion information as input parameters (e.g., noise power, queue status, distance from other UAVs). Such information is retrieved from the Register Plane, which is also employed to store the computed numerical solutions. The DPPS configures at run-time the networking and flight control operating parameters contained in the Data Plane based on the computed numerical solutions (e.g., change the current UAV location based on the optimized routing table, configure the TCP window size based on the optimized application-layer rate injected into the network).

This plane implements a fully programmable and re-configurable protocol stack spanning all the networking layers and motion layers, and interfaces with the radio and motion front-end through the SDR and the flight controller drivers as illustrated in the lower part of Figure 2.

The Data Plane has full control over both the radio front-end, implemented by a software defined radio and the motion front-ends, as well as on protocol stack functionalities; and it is in charge of feeding actual network state and UAV location information back to the Register Plane. As shown in Figure 2, both the network state information and the computed numerical solutions are stored in dedicated look up tables (LUTs) in the Register Plane.

Each DPPS layer has a dedicated Network State LUT in the Register Plane, where to store all the layer-related network state parameters (e.g., signal-to-interference-plus-noise ratio (SINR) and link capacity in physical layer L1 LUT; set of neighbors and their distance in network layer L3 LUT; physical location and obstacles in the vicinity in motion layer L0 LUT). Numerical solutions are stored in a similar way in dedicated Numerical Solution LUTs, one per DPPS layer (e.g., TCP window size in transport layer L4 LUT; routing tables for network later L3 LUT; location for physical layer L1 LUT).

3.1.3. SwarmControl Phase I prototype

The first challenge toward the evaluation of SwarmControl is the lack of commercial off-the-shelf UAV platforms featuring SDRs. To address this, we design and build a custom UAV network node platform, referred to as Drone-SDR, by mounting an Ettus Research Universal Software Radio Peripheral (USRP) B205mini-i SDR on an Intel Aero Ready-to-Fly Drone, as illustrated in Figure 3.

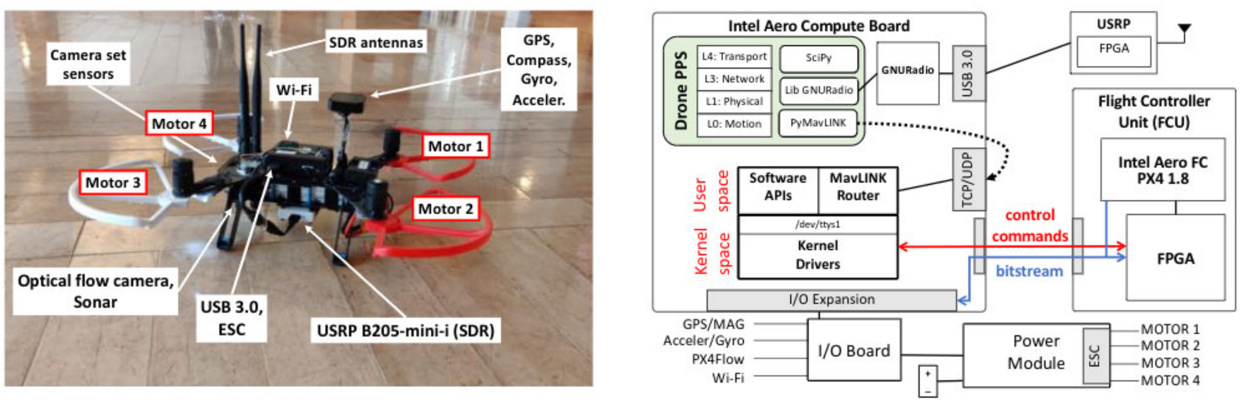

Figure 3 SwarmControl prototype with SDRs: (Left) picture of the prototype; (Right) architecture

With a flight autonomy of over 20 minutes, a hub-to-hub diagonal length of 360 mm, and a base-to-top height of 222 mm, Intel Aeros offer high portability and maneuverability. Similarly, B205mini-i SDRs are the most compact, lightweight and low-power SDR devices available on the market. Intel Aero houses a Compute Board providing sufficient computational power to run Ubuntu 16.04 and SDR development frameworks such as GNU Radio. Flight management, motors control, and sensors fusion are performed on a Pixhawk 4 Flight Controller Unit (FCU) directly connected to the Compute Board. All FCU parameters and commands (e.g., remote control and sensor readings) are accessed through UDP communications via the MAVLink Router. Different from legacy UAVs, SwarmControl UAV nodes are endowed with a DPPS Motion Layer (L0: Motion in Figure 3) that hosts a Pymavlink-based control implementation, allowing each node to execute flight control operations autonomously.

It is worth pointing out that SwarmControl fully relies on open-source software. Specifically, the DPPS is entirely implemented in a high-level scripting language (i.e.,Python) and runs on native Linux OS, which directly interfaces with both the FCU and GNU Radio. This makes SwarmControl compatible with every MAVLink-based programmable drone interface (e.g., Pymavlink, DroneKit). Figure 3 shows an overview of the Drone-SDR prototype, its architecture, and its hardware design.

3.2.Phase II: Artificial Intelligence for swarms of UASs

In the first phase of the project, we have demonstrated how SwarmControl can boost network performance and leverage automated and distributed control to adapt to current network conditions via optimization theory. However, the first generation of the SwarmControl framework relied on model-based optimization which might result in inaccurate approximations when the environment is too complex to be captured by a tractable model. Indeed, the performance of model-based optimization approaches is often limited by the accuracy of the approximations and relaxations necessary to solve the UAV network control problem through convex optimization or similar techniques, and by the accuracy of the channel network models used. To address these challenges, **the second phase of this project focused on developing a new architectural framework to control and optimize UAV networks based on model-free Deep Reinforcement Learning (DRL).** To overcome the limitations of model-based optimization, we have also developed a virtualized, \`ready-to-fly' emulation environment to generate the extensive wireless data traces necessary to train DRL algorithms, which are notoriously hard to generate and collect on battery-powered UAV networks. The training environment integrates the DPPS developed in the first phase of the project with the CORE/EMANE emulation tool for accurate emulation of the UAV network.

3.2.1. A Two-tier Architecture for UAV Network Control

We have designed a two-tier architecture consisting of a Control Framework and a DRL DPPS. The network operator uses the Control Framework to dictate the desired behavior of a distributed UAV network. Our solution automatically generates a set of DRL agents (i.e., a set of policies in the form of Neural Networks (NNs)) that are trained in a virtual environment within the Control Framework. Once trained, the NN configurations are tested and automatically distributed to the individual network nodes, where they will be used to control networking and motion parameters in the DRL DPPS. In this way, the individual UAVs distributively implement the network operator's objective by optimizing their network performance in real time.

By distributing the NN configuration once, and by enforcing the desired network control policy at the edge nodes of the network, this approach does not suffer from stale information retrieval and delayed command typical of centralized control systems. Moreover, the proposed NN-based policies envision full-stack and cross-layer optimization of flight and wireless networking parameters alike, thanks to the use of programmable motion and RF front ends.

3.2.2. A Data-driven Control Approach

The newly developed SwarmControl framework solves the UAV network control problem via DRL. We consider a multi-agent DRL scenario where each UAV is a different agent, and collectively train complex UAV fielding in a virtual environment for a specific flight mission. Upon training completion, we test and distribute mission-tailored NN configurations to individual UAVs. These use them to compute networking and motion policies to achieve the network operator's desired network behavior by adapting to the dynamic network conditions.

Compared to model-based optimization, our data-driven approach addresses inaccurate modeling formulation and optimization approximations. Unlike optimization approaches, the DRL agents do not suffer from optimization solver latency and can derive policies with O(1) complexity.

3.2.3. A `Ready-to-Fly' Virtual Environment

To collect extensive performance data for battery-powered UAV networks, we have developed a highly representative emulation virtual environment. We have revisited the DPPS developed in the first phase and integrated it with Deep Reinforcement Learning (DRL) features and refer to it as DRL DPPS. We have integrated the DRL DPPS with the CORE/EMANE emulation tools to obtain a high-fidelity virtual environment that captures the motion, wireless channel, and higher-layer protocol stack interactions alike. We systematically employ our `ready-to-fly' virtual environment to collect extensive high-fidelity network performance data. Ultimately, this integration effort produces a highly representative emulation environment that allows us to scale up our learning time and to train our DRL agents with a high degree of realism.

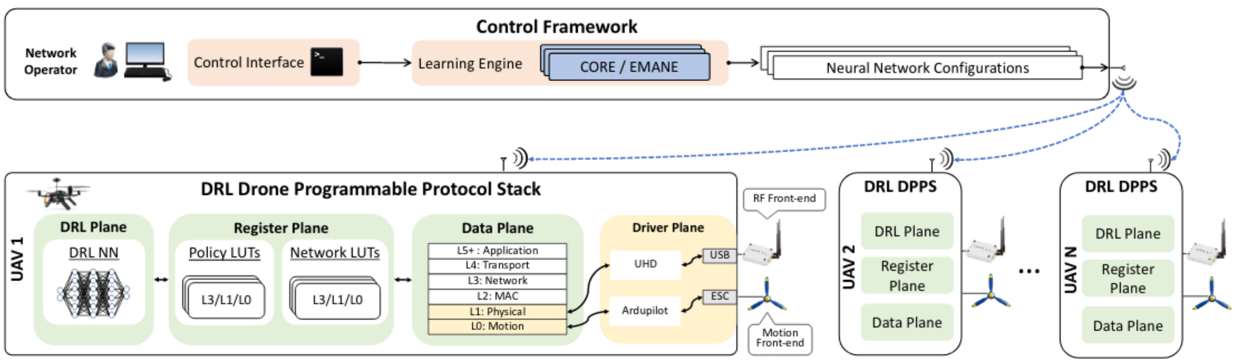

Figure 4 Overview of the DRL DPPS developed in the second phase

3.2.4. A new DRL-based DPPS

An overview of the DRL Drone Programmable Protocol Stack (DRL DPPS) architecture is reported in Figure 4. The DRL DPPS is used at individual UAVs to carry out motion and wireless operations at all layers of the protocol stack, as well as in the Control Framework's Learning Engine to train and test the NN policy making for specific mission objectives. In the latter, the Physical layer and Motion operations are performed by the virtualized CORE/EMANE environment, while in the former these operations are implemented through hardware motion and RF front-end.

By employing the whole DRL DPPS architecture in the Control Framework's Learning Engine (with the exclusion of the hardware front-ends), we obtain a realistic emulation environment which is key to our high-fidelity performance data collection and effective DRL training.

We have extended the DPPS developed in the first phase of the project by replacing the decision plane with the new DRL plane yet maintaining its architectural functionality; to optimize the networking and motion control parameters at once in a cross-layer fashion. Specifically, the control logic is performed by adopting a DRL variant called Q-learning, which aims at optimizing an estimate (called Q function) of the objective function we are trying to maximize (i.e., the network operator's objective). The NN employed by the DRL is a Deep Q-Network (DQN) which uses stochastic gradient descent (SGD) to approximate the Q-function.

3.2.5. The emulation environment for AI-driven swarms of UAVs

Among the main contributions of this project is the integration of the DRL DPPS with the CORE/EMANE emulation tools.

The goal of this integration effort is two-fold:

- to develop a high-fidelity emulation environment capturing both real-time wireless channel phenomena (e.g., path-loss, delay spread, interference) and networking operations at all layers of the protocol stack (e.g., packetization, segmentation, re-transmissions, traffic bursts, processing delay) which are hard or expensive to model and can only be approximated in simulations;
- to provide researchers with a reconfigurable emulation tool to design different UAV network configurations and topologies and collect high-fidelity UAV network performance data at scale. This effort serves as an effective alternative to the collection of experimental performance data for battery-powered UAV networks that are both time- and resource-expensive.

At the same time, this emulation tool can be employed to test UAV network configurations before experimental implementation.

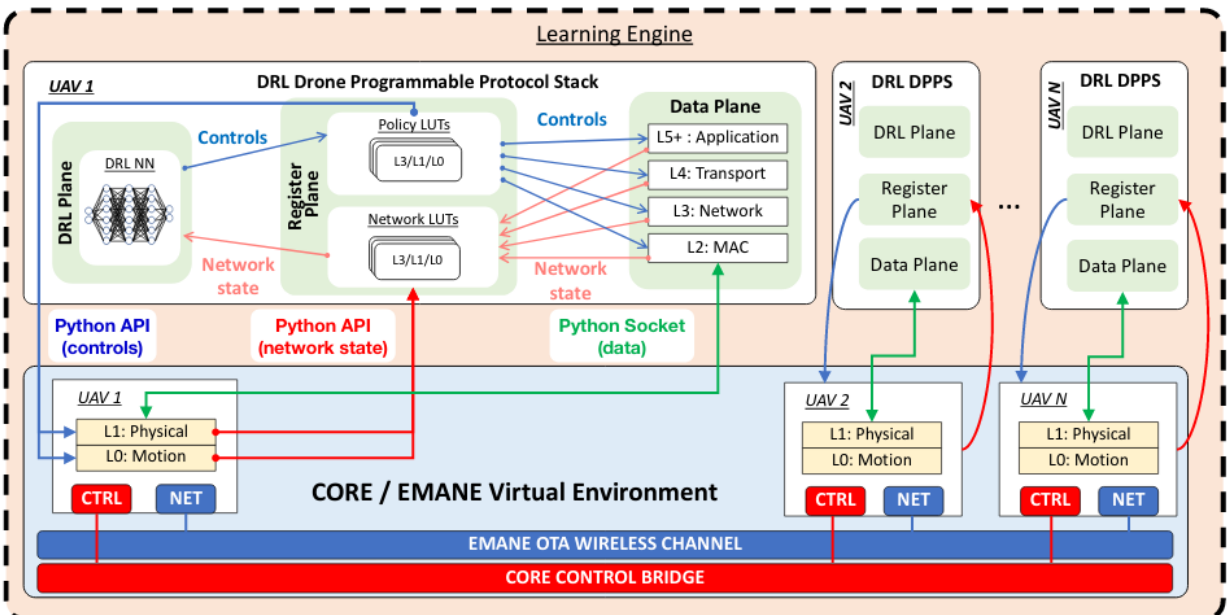

Figure 5 Overview of the Learning Engine and its integration with CORE/EMANE

The architectural integration between the DRL DPPS and CORE/EMANE is illustrated in Figure 5. The same architecture is used in the Control Framework's Learning Engine both by the Virtual Training Environment and by the Virtual Testing Environment. Moreover, the interactions between the DRL agents (i.e., the UAVs embedding the DRL DPPS of SwarmControl) are described in Figure 6, where we show how multiple DRL agents interact with the same environment emulated in CORE/EMANE. It is worth mentioning that although each DRL agent executes in a separate container, they all interact via the same CORE/EMANE container that is shared across all DRL agents.

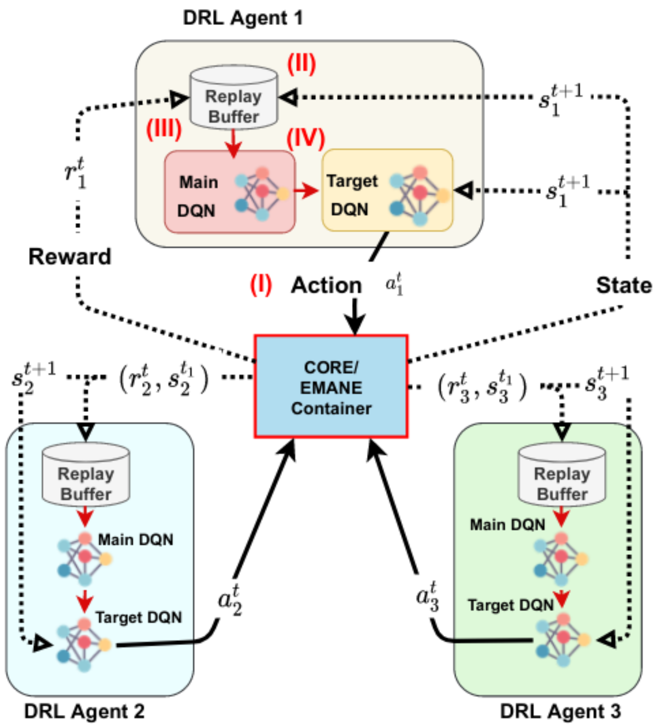

Figure 6 Interactions between multiple DRL agents and CORE/EMANE

4. RESULTS AND DISCUSSION

This section summarized the major results we obtained by developing and prototyping our solutions. Results are given for our solutions developed during both Phase I and Phase II of the project.

4.1. Phase I results

Here we report results obtained by conducting flight experiments in a state-of-the-art UAV lab built to allow UAV flight testing in an indoor RF controlled environment. The facility is a 15x15x7m anechoic chamber, entirely shielded outdoor and indoor (Figure 7).

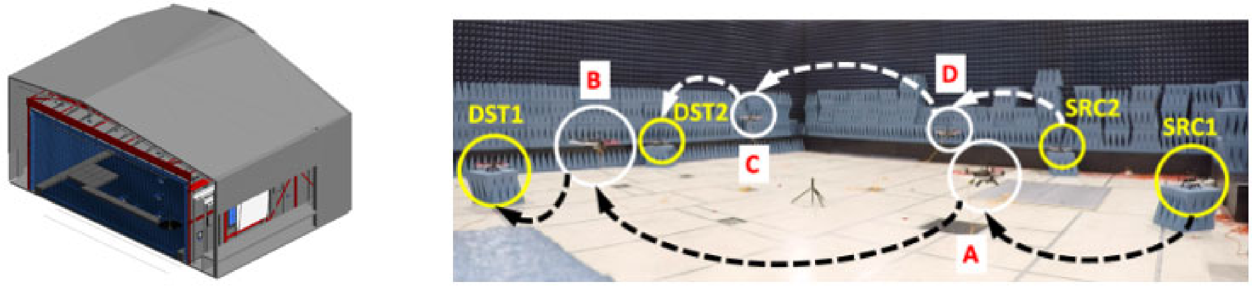

Figure 7 Anechoic chamber (Left) and experiment deployment scenario (Right)

Even though the absorbing walls and the anechoic chamber create ideal conditions for radio communications, the total absence of Global Positioning System (GPS) signal and Earth magnetic field pose severe challenges to distributed flight coordination. To mitigate the absence of universal reference signals, we equip our Drone-SDR prototypes with high frame-per-second optical flow cameras and a sonar to determine local positioning and ground distance. The flight controller automatically adapts the new hardware as primary positioning system without requiring any modification of the Drone PPS.

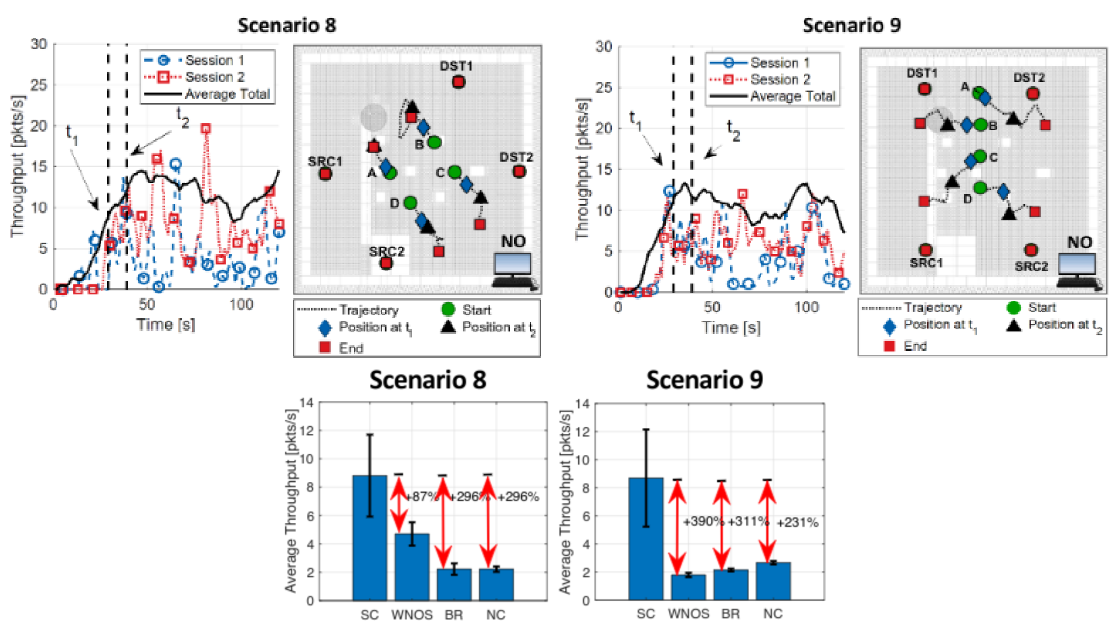

Figure 8 Throughput measurements and physical deployment for two different scenarios (Top) and comparison with other control strategies (Bottom)

In this set of experiments, we evaluate the effectiveness and flexibility of SwarmControl distributed optimization spanning all layers of the prototyped PPS: motion, physical, network, and transport layers.

To that end, we compare SwarmControl's performance with three other control schemes: Best Response (BR), No Control (NC) and Wireless Network Operating System (WNOS) (a recently developed open-source wireless operating system for infrastructure-less ad hoc wireless networks, which does not account for mobility and UAV-related features).

In Figure 8 (top), we present single-run experiments for two scenarios (Scenarios 8 and 9) with different starting topologies. SwarmControl implements network control directives by automatically optimizing networking and flight control strategies at each UAV-SDR in a distributed fashion. More specifically, Figure 8 (top) showcases how individual UAV-SDRs distributively optimize their trajectories to improve the SINR of the individual session links. For both initial deployment scenarios, we can observe the trajectories of the UAVs over time converging toward a reduced mutual interference topology, which eventually results in increased network capacity and overall network throughput improvement.

For each control scheme and deployment scenario, we conduct ten independent 2-minute-long flight experiments. As it can be observed in Figure 8 (bottom), SwarmControl obtained an average throughput gain of 87% and 231% over the second-best performer for the two scenarios, 8 and 9, respectively. This verifies the effectiveness of SwarmControl's unique joint networking and flight control optimization approach. A demo of the flight experiments showcasing the distributed network control implementation of SwarmControl is available at https://mega.nz/#!DIhGlYhQ!ZieAyMtTME9j1pTTFfiwMavx1l5PSLS6p-5co5Mgn9Q.

4.2. Phase II results

In this section, we present a selection of the most relevant results we obtained by assessing the performance of the DRL-based version of SwarmControl and comparing it with other control strategies, as well as the model-based version of SwarmControl developed in the first phase of the project.

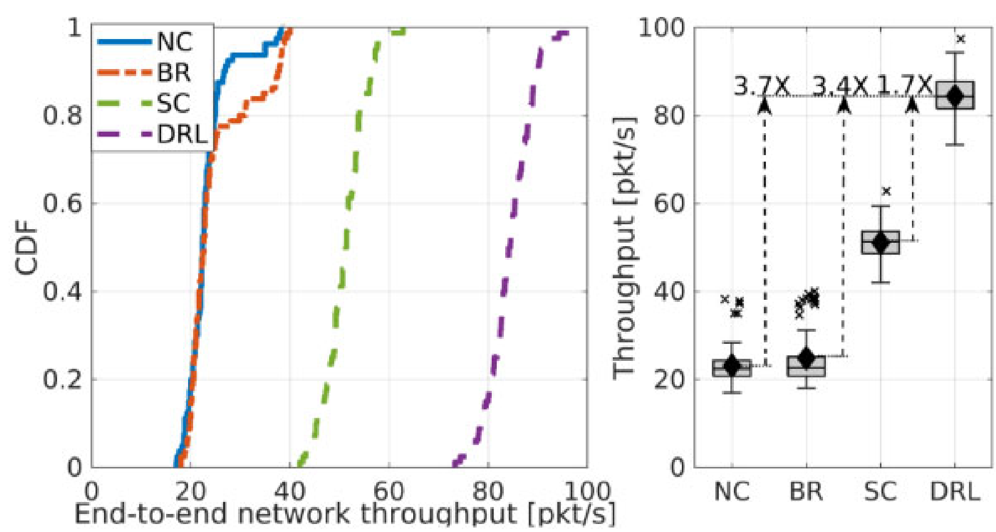

Figure 9 Throughput measurements for different control schemes

The performance comparison between the convex optimization-based control approach developed in the first phase and the data-driven control approach developed in the second one is illustrated in Figure 9. The network deployment involves 8 UAVs, two sensing areas, and two reporting areas.

The control instead involves both transmission power and location of the UAVs. As shown in Figure 9 (right), the proposed data-driven optimization control scheme (DRL) outperforms the convex optimization-based control schemes, achieving 1.7x better performance.

The reason behind this performance gap is that the performance of the model-based optimization is tightly coupled to the accuracy of the models employed in the problem formulation and to the quality of the approximation and relaxation necessary to solve the UAV network control problem via convex optimization. On complex UAV networks, which require conspicuous modeling and approximation effort, these effects combined can result in sub-optimal solutions that result in a performance gap between modeling and performance assessment.

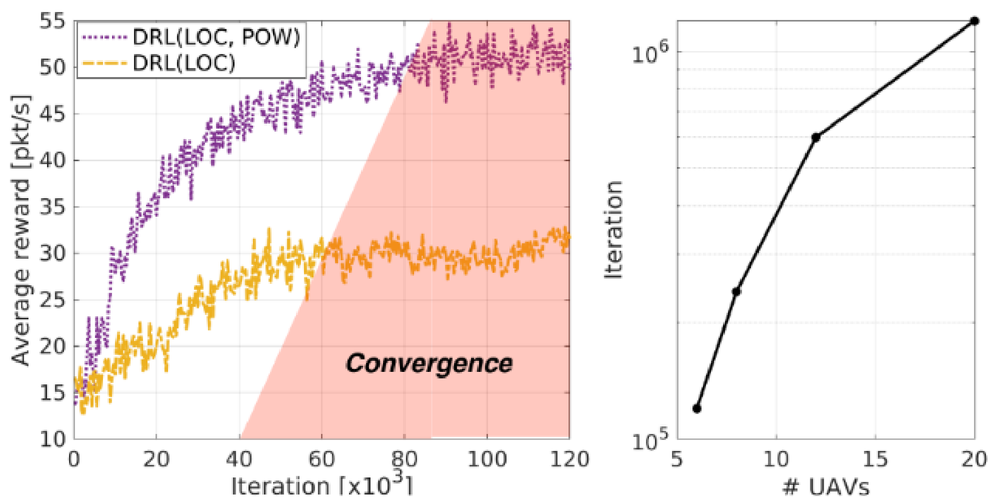

Figure 10 Convergence analysis of the DRL-based SwarmControl framework

Figure 10 (left) shows the reward obtained by the DRL agents during the training procedure in the CORE/EMANE environment. As expected, the DRL agent increases the reward by selecting the most appropriate transmission and mobility actions. Instead, Figure 10 (right) shows the total number of iterations that are required to achieve convergence as a function of the number of UAVs involved in the mission.

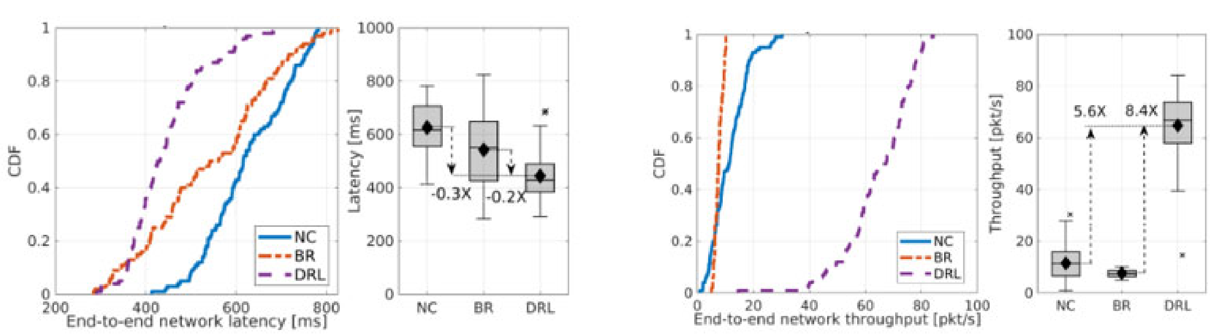

Figure 11 Performance comparison between the data-driven SwarmControl and other control schemes: (Left) Latency; (Right) Throughput

Figure 11, instead, shows the benefit of DRL-based control over other control solutions that do not adapt to time-varying network conditions. Our results show that DRL-based control reduces latency by 30% while guaranteeing throughput values that are 5.6x higher than the second-best performer.

5. CONCLUSIONS

In this project, we have designed, developed and prototyped SwarmControl, a software-defined and optimization-based control framework for UAV networks. SwarmControl leverages the reconfigurability and flexibility of UAVs endowed with software defined radios to provide network operators with an abstraction of the motion and networking functionalities that simplifies and automates the control of the network. This centralized abstraction can be used to define the desired network behavior through a few lines of code. SwarmControl automatically transforms centralized control directives into distributed optimization problems or DRL agents that are dispatched to and executed at individual UAVs. We implemented SwarmControl on SDR-based UAV network platform prototypes, and we assessed its performance through an extensive experimental campaign in both model-based and model-free configurations. Performance evaluation results demonstrate that SwarmControl provides flexibility, fast adaptability, and throughput gains up to 230% when compared to state-of-the-art solutions.

6. REFERENCES

[1] Z. Guan, N. Cen, T. Melodia, and S. Pudlewski, "Joint Power, Association and Flight Control for Massive-MIMO Self-Organizing Flying Drones," to appear in IEEE/ACM Transactions on Networking.

[2] Bertizzolo, L., D’Oro, S., Melodia, T., Pudlweski, S, Bentley, E. S. SwarmControl: A decentralized control solution to optimize infrastructure-less UAV networks. To be submitted to IEEE/ACM Transactions on Networking.

[3] Cheng, H., Bertizzolo, L., D’Oro, S., Buczek, J., Melodia, T., and Bentley, E. S. (2021). Learning to fly: A distributed deep reinforcement learning framework for software-defined uav network control. IEEE Open Journal of the Communications Society, 2, 1486-1504.

[4] Bertizzolo, L., D’Oro, S., Ferranti, L., Bonati, L., Demirors, E., Guan, Z., Melodia, T. and Pudlewski, S. (2020, July). SwarmControl: An automated distributed control framework for self-optimizing drone networks. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications (pp. 1768-1777). IEEE.

[5] Software defined drone network control system, Tommaso MELODIA, Lorenzo BERTIZZOLO, Zhangyu GUAN, Salvatore D'ORO, WO2020209935A3

7. LIST OF ACRONYMS

AI	Artificial Intelligence
AODV	Ad-hoc On-demand Distance Vector
API	Application Programming Interface
BR	Best response
CNN	Convolutional Neural Network
CORE	Common Open Research Emulator
DNN	Deep Neural Network
DPPS	Drone Programmable Protocol Stack
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
EMANE	Extendable Mobile Ad-hoc Network Emulator
FCU	Flight Controller Unit
FDMA	Frequency Domain Multiple Access
FPGA	Field Programmable Gate Array
LUT	Look-up Table
MAC	Medium Access Control
MAG	magnetometer
ML	Machine Learning
NC	No control
NN	Neural Network
OS	Operating System
PHY	Physical Layer
PPS	Programmable Protocol Stack
QoS	Quality of Service
RF	Radio Frequency
SC	SwarmControl
SDN	Software-Defined Networking
SDR	Software-Defined Radio
SGD	Stochastic Gradient Descent
SINR	Signal-to-noise-plus-interference Ratio
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
WNOS	Wireless Network Operating System