SwarmControl: An Automated Distributed Control Framework for Self-Optimizing Drone Networks

Lorenzo Bertizzolo†, Salvatore D'Oro†, Ludovico Ferranti†, Leonardo Bonati†, Emrecan Demirors†,
Zhangyu Guan‡, Tommaso Melodia†, Scott Pudlewski*
‡Institute for The Wireless Internet of Things, Northeastern University, Boston, MA 02115, USA
‡Dept. of Electrical Engineering, The State University of New York (SUNY) at Buffalo, Buffalo, NY 14260, USA
*Air Force Research Laboratory (AFRL), Rome, NY 13440, USA
Email: {bertizzolo.l, s.doro, ferranti.l, bonati.l, e.demirors, melodia}@northeastern.edu,
guan@buffalo.edu, scott.pudlewski.1@us.af.mil

***Abstract*—Networks of Unmanned Aerial Vehicles (UAVs), composed of hundreds, possibly thousands of highly mobile and wirelessly connected flying drones will play a vital role in future Internet of Things (IoT) and 5G networks. However, how to control UAV networks in an automated and scalable fashion in distributed, interference-prone, and potentially adversarial environments is still an open research problem. This article introduces SwarmControl, a new software-defined control framework for UAV wireless networks based on distributed optimization principles. In essence, SwarmControl provides the Network Operator (NO) with a unified centralized abstraction of the networking and flight control functionalities. High-level control directives are then automatically decomposed and converted into distributed network control actions that are executed through programmable software-radio protocol stacks. SwarmControl (i) constructs a network control problem representation of the directives of the NO; (ii) decomposes it into a set of distributed sub-problems; and (iii) automatically generates numerical solution algorithms to be executed at individual UAVs.**

We present a prototype of an SDR-based, fully reconfigurable UAV network platform that implements the proposed control framework, based on which we assess the effectiveness and flexibility of SwarmControl with extensive flight experiments. Results indicate that the SwarmControl framework enables swift reconfiguration of the network control functionalities, and it can achieve an average throughput gain of 159% compared to the state-of-the-art solutions.

***Index Terms*—Drone Networks, Software-Defined Networking, Distributed Network Control.**

I. Introduction

Intelligent unmanned aerial vehicles (UAVs, or "drones") are attracting the interest of the networking community as a "tool" to provide new capabilities, to extend the infrastructure of wireless networks and to make it more flexible [1]. 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 [2–4]. Among these, increasing network coverage [1], providing advanced network services such as location-aware content delivery [5], and massive MIMO transmissions [6] are notable. UAV-aided wireless networks will enable present and future Internet of Things (IoT) and 5G applications, and be a driver for new military and civilian applications spanning battlefield inspection [7], border control and aerial surveillance [8], precision agriculture [9], environmental monitoring [10], transportation and delivery of goods [11–13].

While networks of UAVs can certainly enable a broad range of new applications, UAV orchestration is often performed through centralized control at the core of the infrastructure or manual operations. How to design simple, elastic, and optimal control strategies for infrastructure-independent UAV networks is still a challenging and open issue. First, commercially available UAVs rely on inflexible wireless interfaces (e.g., RC or Wi-Fi), which are sensitive to spatially and temporally varying topologies, dynamic RF environments, and adversarial attacks. Consequently, even basic functionalities such as network formation and point-to-point communications are impaired by unstable channel conditions and fragile network connectivity typical of infrastructure-less aerial scenarios. Second, traditional network control schemes often rely on the assumption that the network operator is aware of real-time network state information and of low-level network infrastructure details and protocol implementations (e.g., UAVs location, network topology, spectrum availability, and modulation schemes); an assumption that often does not hold in distributed aerial networks. Finally, controlling the network behavior and flight operations in a dynamic environment requires a deep understanding of the interactions between the motion and networking functionalities at all layers of the protocol stack. As of today, a widely accepted framework distributively controlling the networking and motion functionalities of large-scale UAV networks is still missing.

To address these challenges, in this paper we propose *SwarmControl*, a new software-defined principled framework to control the behavior of *distributed UAV infrastructure-independent wireless networks*. SwarmControl *provides a centralized abstraction of the UAV network hiding the low-level details of the protocol stack and of the flight control functionalities, as well as the distributed nature of the network control problem*. It also embraces the flexibility of the software defined radio (SDR) paradigm to support UAV communications in dynamic, time-varying, and potentially adversarial infrastructure-less environments. With SwarmControl, the network operator (NO) can programmatically control the overall network behavior without a-priori knowledge of the network topology, UAVs

This article is based upon work supported in part by the Air ForceResearch Laboratory under Contract FA8750-18-C-0122.

mobility patterns, and the details of the distributed control implementation. SwarmControl implements a self-organizing and coordinated UAV network that dynamically adapts to location and network state changes, and ultimately guarantees reliable connectivity and optimized communication with minimal human intervention. In doing so, SwarmControl attempts to provide a software-defined principled approach to jointly and seamlessly control networking and motion functionalities for UAV networks. The main contributions of this article can be summarized as follows:

- **SwarmControl framework.** We propose SwarmControl, a novel software-defined networking control framework for wireless swarms of UAVs endowed with software radios. SwarmControl provides a unified abstraction of networking and motion functionalities that enables the definition of complex network control problems. SwarmControl employs control decomposition theories to generate distributed control problems that are then solved at each individual UAV;
- **Drone programmable protocol stack**. We develop a new Drone Programmable Protocol Stack (Drone PPS) spanning all layers of the network protocol stack as well as the flight control functionalities. The SwarmControl Drone PPS is based on SDN principles and follows a three-plane structure: (i) Decision Plane, (ii) Register Plane, and (iii) Data Plane. The Drone PPS executes the distributed solution algorithms generated by the SwarmControl framework, and enforces optimal networking and motion strategies on each UAV;
- **Prototyping and assessment**. We implemented SwarmControl on a SDR-based UAV network platform prototype, and we assessed its performance through an extensive experimental campaign in an indoor UAV Lab. Experiments demonstrate that SwarmControl effectively improves the network performance (up to 231% of throughput gain) and dynamically adapts the networking strategies to different control objectives, topologies, channels, and interference conditions.

The rest of this paper is organized as follows. In Section II we present a design overview of SwarmControl architecture and we discuss its network abstraction principles. We describe the SwarmControl Drone Programmable Protocol Stack design in Section III and present SwarmControl prototyping and experimental evaluation in Section IV. We discuss related work in Section V, and draw the main conclusions in Section VI.

II. Control Framework

The architecture of SwarmControl is illustrated in Fig. 1. It includes two key components: a *Control Framework* interfacing the network operator at a centralized location and *Drone Programmable Protocol Stack (Drone PPS)* executed at each UAV. In this section, we describe in detail the procedures executed within the control framework. As illustrated in Fig. 1, this component is responsible for (i) providing the network operator (NO) with a control interface to specify the desired network behavior (Section II-A); (ii) constructing a mathematical Network Control Problem (NCP) representation of the NO directives (Section II-B); and (iii) decomposing the NCP into a set of independent sub-problems and distributing them to individual UAVs (Section II-C). We conclude discussing a toy example to showcase the application of the SwarmControl decomposition approach in UAV networks (Section II-D).

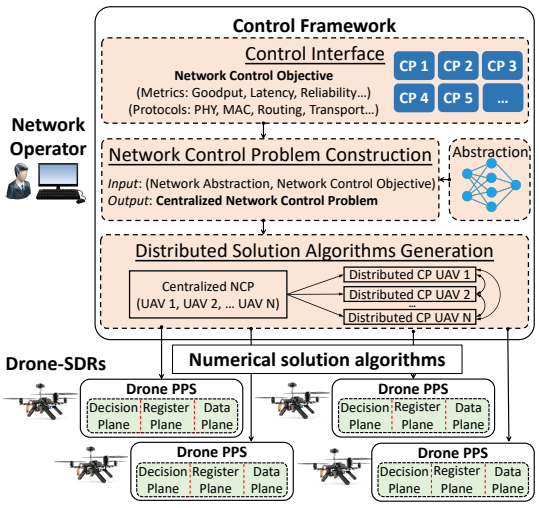

Fig. 1: SwarmControl Architecture.

A. Control Interface

The interaction with the network operator (NO) is implemented through a *Control Interface*, which consists of a set of high-level APIs and protocol libraries. 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, e.g., the number of UAVs as well as their computing capabilities and battery level.

Through the control interface, the NO can express directives defining: (i) the desired network behavior (ii) which layers of the protocol stack to involve in the optimization process and which protocols to implement at each layer; and (iii) node- and layer-specific constraints and QoS requirements. Examples of high-level objectives include maximizing the end-to-end network throughput, prolonging network lifetime by minimizing energy consumption and covering a particular aerial space, among others. By selecting the network protocols and the protocol layers to be optimized the NO can define the desired optimization problem, for example opting out MAC and transport protocols optimization while optimizing motion, routing, and transmission power strategies. Node- and layer-specific constraints can involve, for example, physical layer transmission power, flight speed, ground distance, transmission rate, or a combination of them. Finally, the NO can select among a list of network control templates, or custom design its own optimization problem through the provided APIs.

B. Network Control Problem Construction

The NCP construction is the first step toward distributed control of a UAV network. Once the optimization problem has been defined (e.g., maximizing the overall network throughput), SwarmControl converts the network operator's directives and requirements into a set of mathematical expressions, which are then rearranged in the form of a network control problem (NCP). The resulting NCP is a centralized representation of the high-level network behavior defined by the network operator

through the *Control Interface* spanning both the networking (e.g., transmission power, routing policies, session rates) and the flight control domains (e.g., mobility patterns, flight speed), involving multiple nodes and all layers of the protocol stack.

C. Distributed Algorithms Generation

The resulting NCP cannot be solved at a central controller that has no access to the time-varying network state information, (e.g., UAV locations, routing paths, interference levels). The overhead and the delay to retrieve such information in an infrastructure-less scenario might result in inefficient and sub-optimal network solutions. On the other hand, the cross-layer nature of the obtained NCP and the coupling among its variables (e.g., end-to-end session rate with link capacities with UAVs mobility) make it hard to compute a desirable solution in a distributed fashion. To address this challenge, SwarmControl employs decomposition theories to "loosen" the coupling among optimization variables and generate a separable-variables version of the NCP to be decomposed. The outcome of this procedure is a set of independent sub-problems that can be solved at individual network nodes by exchanging local information with the neighboring UAVs (e.g., intermediate-step solutions and penalization terms). This procedure consists of three fundamental steps, which are discussed in what follows.

1) Optimization variables and parameter detection: First, SwarmControl parses the objective function and the constraints of the constructed NCP and detects the optimization variables and parameters involved in the optimization problem. In doing so, it assigns variables to protocol layer functionalities to be optimized (e.g., transmission power, UAV location, routing tables), while network state parameters (e.g., channel gain coefficients, noise level) and variables excluded from the NCP optimization are treated as constants.

2) Problem decomposition: Given a set of protocol layer functionalities to be optimized, the objective of the decomposition is to loosen the coupling between optimization variables of the NCP. To do so, SwarmControl identifies which network nodes and network layers have control over which optimization variables and generates a *layered coupling graph* $G = (E, V)$. In this abstract representation, vertexes V are optimization variables of the NCP and are associated to a specific layer and a specific node (e.g., transmission power belongs to the physical layer of transmitters and relays); while edges E are coupling relationships between variables of the problem. The layered coupling graph is used to classify dependencies into horizontal (i.e., among variables controlled by different UAVs) and vertical (i.e., among variables controlled by the same UAV but belonging to different layers). We illustrate a portion of the layered coupling graph for a small UAV network in Fig. 2, which will be discussed in detail in Section II-D. Based on this abstract representation of the NCP, SwarmControl uses "tools" such as Decomposition by Partial Linearization [14] and Lagrangian Duality [15] to relax the coupling among variables into cross-layer penalization terms. The decomposition process produces a separable-variable version of the NCP that can be decomposed into independent sub-problems solved at individual UAVs through distributed control actions.

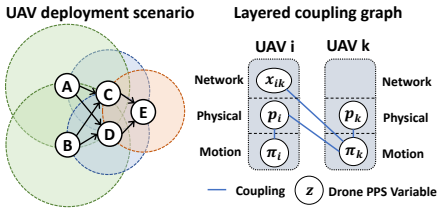

Fig. 2: The network scenario considered in Section II-D and a portion of the corresponding layered coupling graph.

3) Distributed algorithm generation: The final step is the generation and distribution of the numerical solution algorithms to each node. For each of the decomposed sub-problems, an algorithm (e.g., sequential quadratic programming) is automatically generated to calculate the numerical solution associated with a given network control variable. Each algorithm is then translated into an executable script where variables and parameters appear as keywords in text format. All keywords used in SwarmControl are stored in a dedicated library installed at all network nodes. Nodes employ the library to interpret keywords (e.g., whether a specific keyword is a variable to be optimized, a network parameter or a penalization term) and replace them with real-time numerical values (if parameters) or computed numerical solutions (if optimization variables). The resulting scripts are executed to compute optimized numerical solutions for the networking and flight control functionalities. When needed, nodes distributively exchange the intermediate-step computed solutions and penalization terms with other nodes in close proximity over the wireless interface. A toy example showcasing how SwarmControl first constructs a centralized NCP and then generates distributed executable scripts for a simple UAV network scenario is described below.

D. Example of UAV Network Control

We consider a simple network scenario as depicted in Fig. 2, where two source nodes, A and B (e.g., exploration UAVs), are in charge of collecting and delivering strategic data toward a destination E (e.g., the sink UAV) employing relay nodes C and D. We assume the transmission range of each node $i \in \mathcal{N} = \{A, B, C, D, E\}$ depends on its transmission power p_i and that the source nodes' transmission range is not large enough to reach the destination E. Based on routing polices, source nodes independently employ relay nodes C and D to forward their traffic toward the destination. We consider a scenario where the network operator (NO) aims at minimizing the overall power consumption (`min_power`) by optimizing the transmission power, routing tables and UAV locations. We assume that the NO specifies QoS constraints such that each data transmission enjoys a minimum SINR level γ, and imposes fixed flying locations to A and B, in charge of collecting information over specific locations, and to E, to keep it close to a target. Before going through the automated network control process, let us introduce some notation. Let $\mathcal{N}$ be the set of UAVs, $x_{kn} \in \{0, 1\}$ be the routing strategy at node k such that $x_{kn} = 1$ if node k routes its traffic through relay node n, and $x_{kn} = 0$ otherwise, while π_i and p_i represent the location and transmission power of node i, respectively.

As mentioned in Section II-B, SwarmControl processes the control directives of the network operator and, by relying on a UAV network abstraction, constructs the following abstract network control problem:

$$\underset{\mathbf{x},\mathbf{p},\boldsymbol{\pi}}{\text{minimize}} \quad \sum_{i\in\mathcal{N}} p_i, \tag{1}$$

$$\text{subject to} \quad \text{SINR}_{ik}(\mathbf{p},\boldsymbol{\pi}) \geq x_{ik}\gamma, \quad \forall i,k \in \mathcal{N} \tag{2}$$

where $\mathbf{x} = (x_{ik})_{i,k\in\mathcal{N}}$, $\mathbf{p} = (p_i)_{i\in\mathcal{N}}$, $\boldsymbol{\pi} = (\pi_i)_{i\in\mathcal{N}}$, and $\text{SINR}_{ik}(\mathbf{p},\boldsymbol{\pi})$ is the SINR experienced by node k when receiving useful signals from node i. Specifically, we have $\text{SINR}_{ik}(\mathbf{p},\boldsymbol{\pi}) = \frac{G(\pi_i,\pi_k)p_i}{N+\sum_{j\in\mathcal{N}\setminus\{i,k\}} G(\pi_j,\pi_k)p_j}$, where N is the ambient noise power and $G(\pi_i,\pi_k)$ is the channel coefficient between two nodes i and k as a function of their position. For the sake of illustration, we have deliberately omitted all upper and lower bound constraints from Problem (1)-(2).

First, SwarmControl identifies the optimization variables and classifies their coupling with the aid of the layered coupling graph introduced in Section II-C. A portion of the layered coupling graph generated by SwarmControl, with specific focus on coupling introduced by Constraint (2), is illustrated in Fig. 2 (right). For example, the transmission power p_i of node i is coupled with its own location (π_i) and with the location of node k (π_k) through the SINR formulation term in constraint (2).

Once the optimization variables and their dependencies have been identified, SwarmControl performs the decoupling process to generate a separable-variable version of the NCP. This is done by leveraging decomposition tools such as partial linearization [14], Taylor series linearization, and Lagrangian duality [15]. To understand the basics of this procedure, we now showcase the decoupling process for Constraint (2). Following the definition of SINR_{ik}, we can redefine Constraint (2) as

$$\theta_{i,k}(\mathbf{x},\mathbf{p},\boldsymbol{\pi}) = G(\pi_i,\pi_k)p_i - x_{ik}\gamma\left(N + \sum\nolimits_{j\in\mathcal{N}\setminus\{i,k\}} G(\pi_j,\pi_k)p_j\right) \geq 0 \tag{3}$$

Note that optimization variables at different UAVs and different layers of the protocol stack are coupled together by the nonlinear relationships in (3). By applying Taylor series linearization to (3), we first (i) generate a linearized constraint, and then (ii) use Lagrangian duality to include the linearized constraint into the objective function of Problem (1)-(2). Let $\tilde{\theta}_{ik}(\mathbf{x},\mathbf{p},\boldsymbol{\pi})$ be the linearized – and thus with separable variables – version of (3), we can generate the following Lagrangian dual function

$$\begin{aligned} L(\boldsymbol{\lambda},\mathbf{x},\mathbf{p},\boldsymbol{\pi}) &= \sum_{i\in\mathcal{N}} p_i - \sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{N}\setminus\{i\}} \lambda_{ik}\tilde{\theta}_{ik}(\mathbf{x},\mathbf{p},\boldsymbol{\pi}) \\ &= \sum_{i\in\mathcal{N}} p_i + \Gamma_i(\boldsymbol{\lambda},\mathbf{x}_i,p_i,\pi_i) \end{aligned} \tag{4}$$

where $\mathbf{x}_i = (x_{ik})_{k\in\mathcal{N}\setminus\{i\}}$, $\boldsymbol{\lambda} = (\lambda_{ik})_{i,k\in\mathcal{N}}$, $\lambda_{ik} \geq 0$ is the Lagrangian multiplier associated to the linearized constraint (2), and Γ_i is a node-specific function such that $\sum_{i\in\mathcal{N}}\Gamma_i(\boldsymbol{\lambda},\mathbf{x}_i,p_i,\pi_i) = \sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{N}\setminus\{i\}}\lambda_{ik}\tilde{\theta}_{ik}(\mathbf{x},\mathbf{p},\boldsymbol{\pi})$. From (4), it is easy to see that Γ_i contains variables controlled by node i only, while the Lagrangian multipliers $\boldsymbol{\lambda}$ keep track of the previous coupling with other variables. Thus, for each node i we can formulate the following node-specific iterative optimization sub-problem

$$(\mathbf{x}(t),\mathbf{p}(t),\boldsymbol{\pi}(t)) = \underset{\mathbf{x}_i,p_i,\pi_i}{\arg\min} \; p_i - \Gamma_i(\boldsymbol{\lambda}(t-1),\mathbf{x}_i,p_i,\pi_i) \tag{5}$$

where t represents the iteration index and the Lagrangian coefficients are updated as follows:

$$\lambda_{ik}(t) = [\lambda_{ik}(t-1) - \alpha(t)\tilde{\theta}_{ik}(\mathbf{x}(t-1),\mathbf{p}(t-1),\boldsymbol{\pi}(t-1))]^+ \tag{6}$$

Once the decomposition procedure has been completed, SwarmControl automatically generates numerical solution algorithms (based on interior-point methods) to solve the decomposed optimization sub-problem. The numerical solution algorithms are then turned into executable template scripts where optimization variables and network parameters are represented through textual keywords, which in turn are replaced by available information and exchanged penalization terms at each UAV. For the sake of completeness, an illustrative example of the numerical solution algorithm generated by the Control Framework is presented in Fig. 3. The template script reflects the optimization problem in (5). Due to space limitations, we can report only a portion of the code generated from (5). For example, `MSG.lambda` reflects penalty terms defined in (6), `HW.min_snr` is the minimum SINR level γ in (3), and `opt_var.net.nextHop.isRouteActive` identifies x_{ik}. It is worth noting that the template generated by the framework and dispatched to the UAVs is node-independent and does not contain any UAV-specific term. The template instructs each UAV to optimize transmission power (`opt_var.phy`), routing strategies (`opt_var.net`), and location (`opt_var.loc`); and indicates which parameters to exchange across UAVs (`MSG.lambda`). Upon receiving the template, each UAV updates the script variables and parameters according to its role in the network. This is illustrated in Fig. 3, where we show how the script is handled differently by nodes A and C. As discussed above, sources are instructed to hover over a specific location, accordingly, Fig. 3 shows how A removes `opt_var.location` from the optimization variable set and adopts the fixed location `HW.location`.

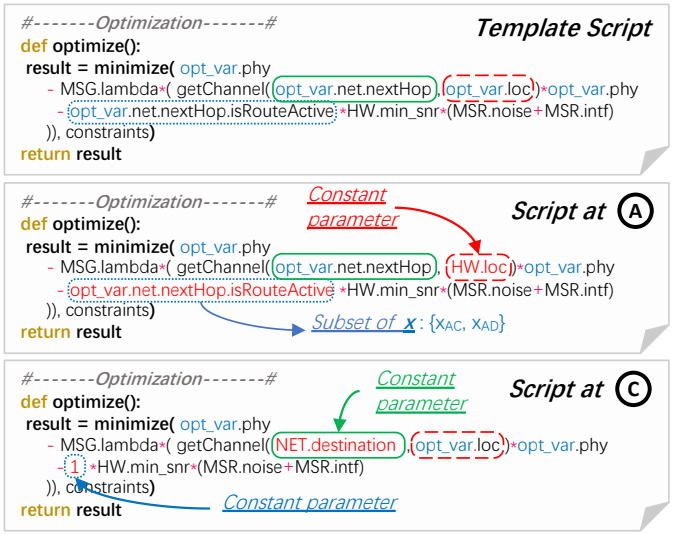

Fig. 3: Example of the script generated by the Control Framework.

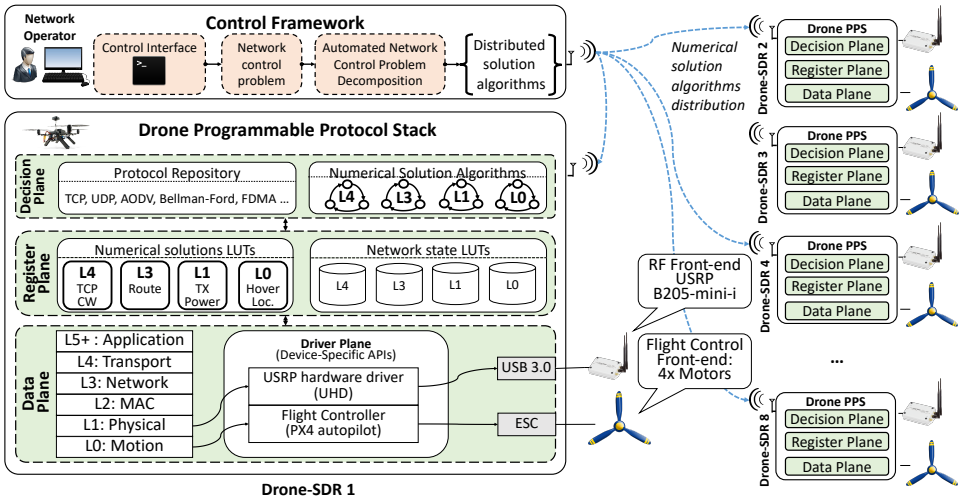

Fig. 4: Drone Programmable Protocol Stack Prototype.

Similarly, relays are instructed to deliver data generated by the two source nodes to the destination E. Hence, C fixes the next hop equal to the destination E (`NET.destination`) and removes the routing strategies from the optimization variable set (`opt_var.net.nextHop.isRouteActive` $= 1$). On the contrary, source node A must select the best relay node between C and D. Consequently its routing strategy space `opt_var.net.nextHop.isRouteActive` is represented by the subset $\{x_{AC}, x_{AD}\}$.

III. Drone Programmable Protocol Stack

The core contribution of SwarmControl is the implementation of automated and distributed control of UAVs' networking and flight functionalities through a new Drone Protocol Stack. As shown in Fig. 1, the *Drone PPS* is installed at each individual UAV to solve the numerical solution algorithms received from the *Control Framework* in an automated and distributed fashion. To compute a desirable network operating point, each individual UAV node executes the distributed optimization solution algorithms generated by the *Control Framework* with up-to-date and accurate network state information. The computed numerical solutions are then implemented at the network protocol stack and the flight controller. This is accomplished by the Drone PPS, organized in three tightly interacting planes, namely *Decision Plane*, *Register Plane*, and *Data Plane*. The design architecture of the three planes is illustrated in Fig. 4.

A. Decision plane

Upon receiving the distributed numerical solution algorithms generated by the *Control Framework* (e.g., motion solution algorithm, transport rate solution algorithm), the SwarmControl Drone PPS runs them in its *Decision Plane* of each individual UAV as shown in Fig. 4. This plane contains a Protocol Repository with 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* is in charge of running the distributed optimization algorithms in real-time based on up-to-date network state and motion information as input parameters (e.g., noise power, queue status, UAV locations). Such information is retrieved from the *Register Plane*, which is also employed to store the computed numerical solutions.

B. Data plane

The *Data Plane* is responsible for implementing the computed optimal solutions by re-configuring the networking and flight control operating parameters. To do so, this plane implements a fully-programmable re-configurable protocol stack spanning all the networking and motion layers. The protocol stack provides the building blocks and primitives necessary to prototype complex cross-layer and cross-domain network protocols and motion strategies, allowing complete control of the network, sensing, and motion parameters at all layers of the protocol stack. The control interface between the protocol stack and the distributed solution algorithms is defined so that (i) the solution algorithms can retrieve network state information from the *Data Plane* through the *Register Plane* (e.g., noise and interference power level, queue status, node location, among others), and use it as input parameters of the distributed optimization problems; and (ii) based on the optimized solutions, the Drone PPS configures the networking and motion parameters of the adopted protocol stack in the *Data Plane* (e.g., change the current UAV location based on the optimized motion pattern, configure the TCP window size based on the optimized transport-layer rate). The lower layers of the implemented protocol stack interface with the radio and motion front-ends through the software defined radio (i.e., USRP hardware driver (UHD)) and the flight controller (i.e., PX4 flight control) drivers. Finally, the *Data Plane* controls the external radio and motion hardware through its drivers on the universal serial bus (USB 3.0) and electronic speed control (ESC) interfaces, as illustrated in Fig. 4.

C. Register plane

As shown in Fig. 4, the *Register Plane* acts as a middleware allowing the *Decision Plane* to retrieve fresh network state information from the *Data Plane* and making the computed optimal solutions available to the *Data Plane* through a set of

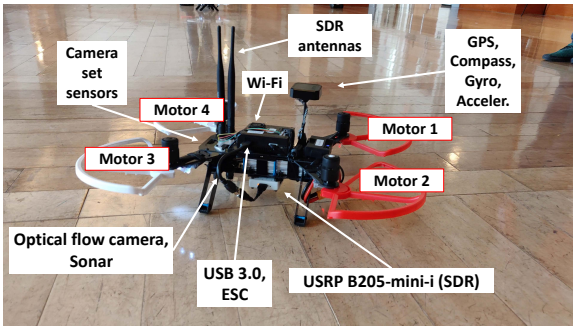

Fig. 5: Drone-SDR prototype.

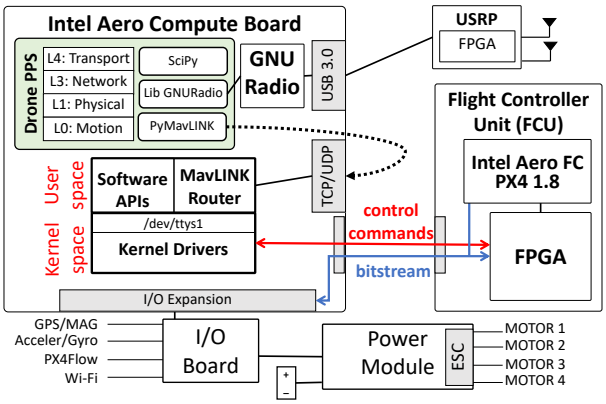

Fig. 6: Drone-SDR prototype hardware design.

dedicated look up tables (LUTs). Each protocol stack layer has a dedicated Network State LUT in the *Register Plane*, where to store all the layer-related network state parameters, e.g., the physical location and the obstacles vicinity in the motion layer `LUT_L0`, the SINR and the link capacity in the physical layer `LUT_L1`; the set of neighbors and their distances in the network layer `LUT_L3`. Numerical solutions are stored in a similar way in dedicated Numerical Solution LUTs, one per protocol stack layer, e.g., the location for the physical layer `LUT_S1`; the routing tables for the network layer `LUT_S3`; the TCP window size in the transport layer `LUT_S4`.

IV. Prototype and Performance Evaluation

In this section, we assess the performance of SwarmControl as presented in Sections II and III by comparing it to other state-of-the-art solutions on a variety of network configurations. We first describe the Drone-SDR platform prototype we have developed for our experiments in Section IV-A and we summarize the experimental setup in Section IV-B. Finally, experimental results are discussed in Sections IV-C, IV-D and IV-E.

A. Drone-SDR 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 designed and built 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 Fig. 5. 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 *Drone PPS Motion Layer* (*L0: Motion* in Fig. 6) 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 Drone PPS 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). Figures 5 and 6 show an overview of the Drone-SDR prototype, its architecture, and its hardware design.

B. Experimental Setup

We test SwarmControl on a plethora of network control problems and network deployments, such as fully aerial networks, where the goal is to transmit data from source UAVs toward destination UAVs in a multi-hop fashion; and hybrid ground/aerial networks, where a UAV network is employed to relay data between ground nodes; varying the number of nodes, number of sessions, topologies, and testing environments. Given the complexity of the network control problems, we tested the effectiveness and flexibility of SwarmControl in an incremental fashion, obtaining intermediate results to highlight the impact of different features on the overall system performance. We demonstrate how SwarmControl efficiently handles cross-layer optimization by considering joint

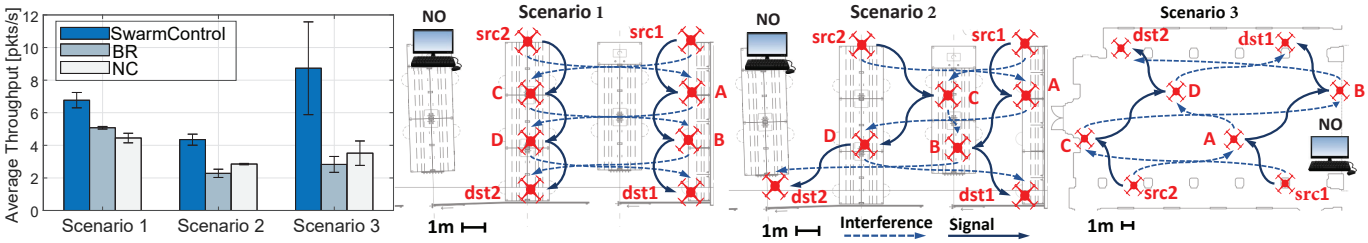

Fig. 7: Network average throughput and network scenarios for fully aerial experiments.

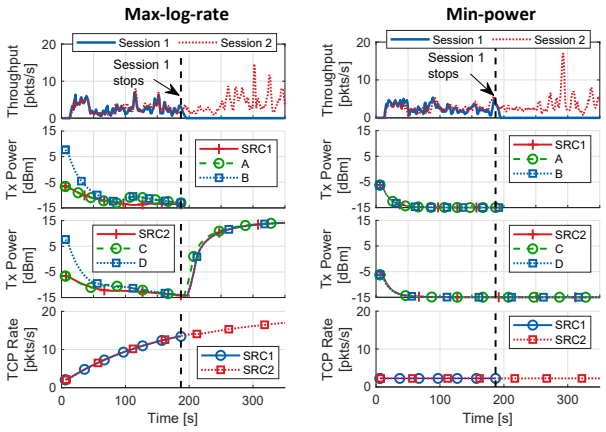

Fig. 8: Network operations comparison under two different control problems for Scenario 1.

optimization operations that span across four layers of the prototyped drone programmable protocol stack (Drone PPS). Specifically, we optimize the transport layer transmission rate, routing decisions, transmission power, and flight control of the UAVs. Furthermore, we compare the performance of the UAV network under four different distributed control schemes: (i) *SwarmControl*, which jointly optimizes networking and motion parameters at all layers of the Drone PPS; (ii) *Best Response* (BR), independently optimizing the parameters at different layers of the Drone PPS; (iii) *No Control* (NC), which does not use any network optimization mechanism and operates only with average networking parameters; and (iv) *WNOS* [16]. The latter is a recently developed open-source wireless operating system for infrastructure-less ad hoc wireless networks, with code openly available on the project website. In all our experiments, we assume that the NO selects TCP as transport layer protocol, single-path routing algorithm, frequency division multiplexing MAC, and GMSK modulation. Furthermore, all transmissions are performed in the 2.4GHz ISM frequency band and the bandwidth of transmitted signals is 500kHz. In all figures, links connecting UAVs represent a single snapshot of the network state, which may evolve throughout the experiment.

C. Fully Aerial Network

We first evaluate the performance of SwarmControl by deploying 8 Drone-SDRs in three different scenarios, namely Scenario 1, 2, and 3 (Fig. 7); comparing it to Best Response (BR) and No Control (NC) schemes under the *max-log-rate* control problem (i.e., maximize $\sum_{i\in\mathcal{N}} \log(x_i)$, with x_i the end-to-end session rate for source i). In these three scenarios, we constrained the mobility of Drone-SDRs to highlight the performance of SwarmControl in different static network topologies. Scenarios 1 and 2 feature dense indoor environments with obstacles, non-line of sight conditions, strong multipath effect, and high background interference. On the contrary, Scenario 3 presents a large obstacle-free space with low signal refraction and negligible multipath effect. Operating on the same spectrum bands, the deployed Drone-SDRs mutually interfere with each other. Therefore, the overall end-to-end network throughput depends on transmission power, routing policies and TCP session rate. The experiment includes two Drone-SDR

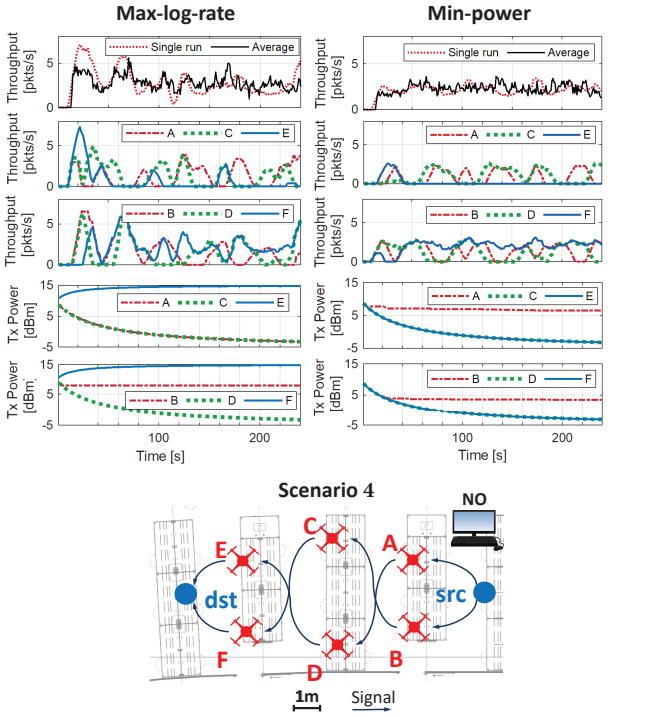

Fig. 9: Performance and network scenario for two different control problems in UAV relay network for disaster scenario.

source nodes opening two sessions toward two destinations and injecting traffic over the multi-hop Drone-SDR network.

In all three considered scenarios, SwarmControl - by jointly and distributively optimizing TCP rate, routing strategies, and transmission power - significantly outperforms the BR and NC schemes in terms of overall network throughput. As shown in Fig. 7 (left), the average performance gain of SwarmControl with respect to the second-best performing scheme is 52%, 90%, and 208% for Scenarios 1, 2, and 3, respectively.

Modifying the network behavior. As discussed in Section II, SwarmControl makes the operation of modifying the behavior of a whole aerial network as simple as inputting a few characters into the Control Interface. In the current implementation, automated decomposition and re-distribution of the new optimization problems to all UAVs take less than 3 seconds, which allows modifying the network behavior in real-time. To illustrate how different control objectives lead to different distributed control actions in SwarmControl, in Fig. 8 we compare optimized PPS parameters for *max-log-rate* and *min-power* (i.e., minimize $\sum_{i\in\mathcal{N}} p_i$, with p_i being the transmission power for node i) control problems for Scenario 1. *max-log-rate* simultaneously maximizes fairness and overall network application throughput, while *min-power* aims at minimizing the overall consumed network power while maintaining the minimum-session-rate QoS requirements. Figure 8 shows the measured network throughput together with the physical layer transmission power levels and TCP transport rates at individual nodes for a single experiment realization under *max-log-rate* and *min-power* network control problems. We terminated Session 1 after 180 seconds in both experiments. Despite the fact that stopping Session 1 causes a general throughput gain at Session 2 because of the reduced interference, under the *max-log-*

rate control problem the nodes increase the transmission power and TCP transport rate to pursue overall network throughput maximization. On the contrary, under the *min-power* control problem, nodes belonging to Session 2 increase neither the transmission power nor the session rate, which results in lower power consumption, i.e., the objective of the *min-power* network control problem.

D. Mixed Ground/Aerial network

Here, we consider a swarm of Drone-SDR nodes integrated with a ground wireless infrastructure. In this experiment, we show the use of SwarmControl aiming at improving the performance of a mixed aerial-ground network. We demonstrate the effectiveness of SwarmControl on a 6-UAV swarm used to restore and optimize the network connectivity following a ground infrastructure collapse. Scenario 4 in Fig. 9 presents two ground nodes, source and destination, unable to communicate, and a multi-hop aerial relay network of 6 Drone-SDRs deployed to restore the connectivity.

Figure 9 reports the performance of the relay UAV network for two different control problems, namely *max-log-rate* and *min-power*. The bottom of the figure presents the single-run individual optimal transmission power values of the 6 Drone-SDRs for each control problem. The top of the figure instead reports the individual forwarding rates contribution of the 6 Drone-SDRs achieving optimal traffic distribution across the network, together with the overall recovered network throughput for the considered single run and an average over ten 4-minute long experiments. It can be seen that Drone-SDRs use higher transmission power implementing *max-log-rate* control problem compared to *min-power*. Overall, the results prove that the relay UAV network can successfully implement different control objectives. For example, node E uses over 20 dBm higher transmission power and accordingly relays over 20% more packets per second in *max-log-rate* control problem realization. These higher individual forwarding rates translate into an overall network throughput improvement up to 100% if compared to the *min-power* problem.

E. Open Sky Experiments

We conclude our evaluation section by reporting on extensive flight experiments conducted in a state-of-the-art UAV lab built to allow UAV flight testing in an indoor RF controlled environment. The facility is a $15\ \mathrm{m} \times 15\ \mathrm{m} \times 7\ \mathrm{m}$ anechoic chamber,

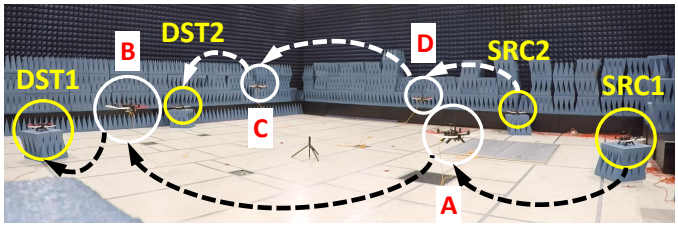

Fig. 10: Snapshot of Scenario 5 flight experiment in the UAV lab.

entirely shielded outdoor and indoor (Fig. 10). The chamber also contains $0.5\ \mathrm{m}$ hi-performance RF absorbing pyramidal foam that covers all surfaces. The absorbers were removed from the floor to simplify take off and landing operations (see Fig. 10). The shielded enclosure provides $>\ 100\ \mathrm{dB}$ of isolation between $300\ \mathrm{MHz}$ and $18\ \mathrm{GHz}$, while the foam creates an environment simulating free-space transmission that can be used as open sky environment for UAV network testing. 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 UAV 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. SwarmControl automatically detects the new hardware and set the camera as the primary positioning system without requiring any modification of the Drone PPS.

In this set of experiments, we evaluate the effectiveness and flexibility of SwarmControl distributed optimization spanning all layers of the prototyped Drone 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 WNOS [16]. We conduct our evaluation on an 8-UAV swarm flying wireless network under the *max-log-rate* control problem, in two different deployments scenarios, Scenarios 5 and 6. The two scenarios represent two different UAV network deployments where two source UAVs aim at retrieving data at two specific locations and delivering it to two destination UAVs located somewhere else by employing a flying UAV network. Scenarios 5 and 6 differ for the location of the source and destination Drone-SDRs, the initial positioning of the four relay Drone-SDRs, and their relative distances, which implies different initial SINR conditions and different intermediate operational points. The mobility of

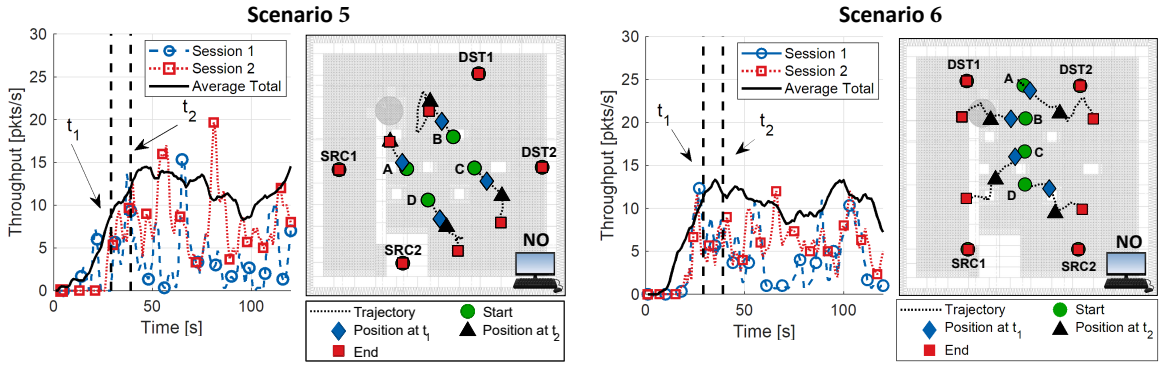

Fig. 11: Network throughput and instantaneous UAVs trajectories for two network scenarios in UAV lab flight experiments.

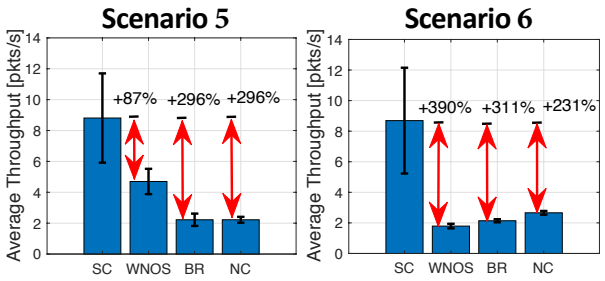

Fig. 12: Average throughput for different control schemes in indoor flight experiments at UAV lab.

sources and destinations is constrained to the regions of interest by setting them in "hold mode", while other Drone-SDRs are let free to move according to the network optimization results. Similar to previous experiments, we consider two source nodes opening two sessions toward two destinations, while the rest of the network participates in multi-hop traffic forwarding.

In Fig. 11, we present single-run experiments for the two scenarios. It can be seen that SwarmControl implements network control directives by automatically optimizing networking and flight control strategies at each Drone-SDR in a distributed fashion. More specifically, Fig. 11 shows how individual Drone-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 drones over time converging toward a reduced mutual interference topology, which eventually results in increased network capacity and overall network throughput improvement. Figure 10 shows a snapshot of the multi-hop Drone-SDR network flight experiments. For each control scheme and deployment scenario, we conduct ten independent 2 minute-long flight experiments. As it can be observed in Fig. 12, SwarmControl obtained an average throughput gain of 87% and 231% over the second-best performer for the two scenarios, 5 and 6, respectively. This verifies the effectiveness of SwarmControl's unique joint networking and flight control optimization approach. A demo video of the flight experiments showcasing the distributed network control of SwarmControl is available at [17].

V. Related Work

Works such as OpenFlow [18], OpenRadio [19], Soft-RAN [20], CellSDN [21], OpenRoads [22] and SDN-WISE [23] have pioneered software defined networking as an enabling technology for both wired and wireless networks and for the next-generation Internet. They are based on a few key principles: (i) removing control decisions from hardware; (ii) enabling the hardware decisions to be programmable through open and standardized interfaces, and (iii) allowing a network operator to define (in software) the behavior of the network infrastructure on a centralized abstraction. These approaches have simplified introducing and deploying new applications and services, as well as configuring network policy and enhance the performance of the system, e.g., improving network resource utilization efficiency, simplifying network management, reducing operating cost, and promoting innovation and evolution.

Compared to infrastructure-based SDN approaches, enabling SDN in infrastructure-less wireless networks is much more challenging and far from being well explored. There are only a few prior research efforts in this field. In [24], Zhu et al. proposed an SDN-based routing scheme for Vehicular Ad Hoc Network (VANET), where a central controller collects network information from switches and computes optimal routing strategies. In [25], the authors discussed a hybrid SDN architecture for wireless distributed networks (WDNs) to alleviate the multi-hop flooding operation of routing information. In this way, the computational complexity of route discovery is split between the SDN controller and the distributed forwarding nodes, eliminating the need for collecting all the link-state information to select routes. In [26], Wu et. al. propose a multi-UAV wireless communication system to optimize the multi-user communication scheduling and association, together with the UAV trajectory and power control for cellular networks. In this way, they maximize the downlink throughput to ground users maintaining good fairness performance. [27] introduces SkyCore, a new EPC design for UAV cellular networks pushing the EPC functionality to the edge of the core network. The proposed lightweight solution is co-located with the BS at the UAV nodes, overcoming the limitations of traditional orchestration typical of wireless UAV environments. In WNOS [16], the authors present an optimization-based SDN framework for ad hoc networks. However, only static and ground-based ad hoc networks are considered, which results in problems that are significantly easier to solve given the pre-determined traffic paths and the lack of mobility. Moreover, [16] only optimizes transport and physical layer, while it does not consider the dynamics of network formation and location-aware routing operations; or the interdependencies between control of the networking functionalities and flight control in a swarm of drones. These and other papers are either designed for a single-drone architecture [28–31], employ centralized network control [26, 32–35], focus only on one single protocol layer [36–38], or limit their evaluation to simulation-based experiments [39–42]. Differently, in this work, we focus on designing and controlling infrastructure-less UAV networks, in a software-defined, distributed, and cross-layer fashion, and evaluate our performance on a swarm testbed with Drone-SDR prototypes.

VI. Conclusions

We presented 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 the network operator with an abstraction of the motion and networking functionalities. 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 that are decoupled, dispatched to and solved distributively at individual UAVs. We implemented SwarmControl on SDR-based UAV network platform prototypes, and we assessed its performance through an extensive experimental campaign. Performance evaluation results demonstrate that SwarmControl provides flexibility, fast adaptability, and throughput gains up to 230% when compared to state-of-the-art solutions.

REFERENCES

[1] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, N. Lagraa, and M. B. Yagoubi, "Intelligent UAV-assisted routing protocol for urban VANETs," *Elsevier Computer Communications*, vol. 107, pp. 93–111, July 2017.

[2] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, Nov. 2015.

[3] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: Multi-tier drone-cells," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–55, Nov. 2016.

[4] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, "Drone-aided communication as a key enabler for 5G and resilient public safety networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 36–42, Jan. 2018.

[5] T. Fang, H. Tian, X. Zhang, X. Chen, X. Shao, and Y. Zhang, "Context-aware caching distribution and UAV deployment: A game-theoretic approach," *Applied Sciences*, vol. 8, no. 10, p. 1959, Oct. 2018.

[6] P. Chandhar, D. Danev, and E. G. Larsson, "Massive MIMO for communications with drone swarms," *IEEE Trans. on Wireless Communications*, vol. 17, no. 3, pp. 1604–1629, Mar. 2018.

[7] Military.com. (2019) Army to buy $39.6 million worth of pocket-sized drones. https://www.military.com/defensetech/2019/02/14/army-buy-396-million-worth-pocket-sized-drones.html.

[8] Business Insider. (2018) China is testing creepy drones that look and fly like real birds to monitor citizens. https://www.businessinsider.com/china-is-testing-creepy-dove-drones-to-monitor-citizens-2018-6.

[9] E. Honkavaara, H. Saari, J. Kaivosoja, I. Pölönen, T. Hakala, P. Litkey, J. Mäkynen, and L. Pesonen, "Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture," *Remote Sensing*, vol. 5, no. 10, pp. 5006–5039, Oct. 2013.

[10] Shell. (2019) Eye in the sky. https://www.shell.com/inside-energy/eye-in-the-sky.html.

[11] Amazon. (2016) Amazon Prime Air. https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011.

[12] DHL. (2018) PARCELCOPTER: DHL's drone. https://discover.dhl.com/business/business-ethics/parcelcopter-drone-technology.

[13] The Economist. (2018) How e-commerce with drone delivery is taking flight in china. https://www.economist.com/business/2018/06/09/how-e-commerce-with-drone-delivery-is-taking-flight-in-china.

[14] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J. S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. on Signal Processing*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[15] D. Y. Gao, *Duality principles in nonconvex systems: theory, methods and applications*. Springer Science & Business Media, 2013.

[16] Z. Guan, L. Bertizzolo, E. Demirors, and T. Melodia, "WNOS: An optimization-based wireless network operating system," in *Proc. of ACM MobiHoc*, Los Angeles, CA, USA, June 2018.

[17] Authors, "Swarmcontrol demo," 2019. [Online]. Available: https://www.youtube.com/watch?v=0fOJhc8gr6A

[18] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[19] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "OpenRadio: A programmable wireless dataplane," in *Proc. of ACM HotSDN*, Helsinki, Finland, Aug. 2012.

[20] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proc. of ACM HotSDN*, Hong Kong, China, Aug. 2013.

[21] L. E. Li, M. Z. Mao, and J. Rexford, "CellSDN : Software-defined cellular networks," Tech. Rep., 2012, online. ftp://ftp.cs.princeton.edu/techreports/2012/922.pdf.

[22] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "OpenRoads: Empowering research in mobile networks," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 125–126, Jan. 2010.

[23] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks," in *Proc. of IEEE INFOCOM*, Kowloon, Hong Kong, 2015.

[25] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein, "Software-defined wireless networking: centralized, distributed, or hybrid?" *IEEE Network*, vol. 29, no. 4, pp. 32–38, July 2015.

[26] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, Jan. 2018.

[27] M. Moradi, K. Sundaresan, E. Chai, S. Rangarajan, and Z. M. Mao, "SkyCore: Moving core to the edge for untethered and reliable UAV-based LTE networks," in *Proc. of ACM MobiCom*, New Delhi, India, Oct. 2018.

[28] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in uav enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, June 2018.

[29] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, "Backhaul-aware robust 3D drone placement in 5G+ wireless networks," in *Proc. of IEEE ICC Workshops*, Paris, France, May 2017.

[30] U. Ali, H. Cai, Y. Mostofi, and Y. Wardi, "Motion and communication co-optimization with path planning and online channel prediction," in *Proc. of IEEE ACC*, Boston, MA, USA, July 2016.

[31] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. of IEEE GLOBECOM*, San Diego, CA, USA, Dec. 2015.

[32] B. Barritt, T. Kichkaylo, K. Mandke, A. Zalcman, and V. Lin, "Operating a UAV mesh & internet backhaul network using temporospatial SDN," in *Proc. of IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 2017.

[33] S. ur Rahman, G.-H. Kim, Y.-Z. Cho, and A. Khan, "Deployment of an SDN-based UAV network: Controller placement and tradeoff between control overhead and delay," in *Proc. of IEEE ICTC*, Jeju Island, Korea, Oct. 2017.

[34] R. Kirichek, A. Vladyko, A. Paramonov, and A. Koucheryavy, "Software-defined architecture for flying ubiquitous sensor networking," in *Proc. of IEEE ICACT*, Bongpyeong, South Korea, Feb. 2017.

[35] S. ur Rahman, G.-H. Kim, Y.-Z. Cho, and A. Khan, "Positioning of UAVs for throughput maximization in software-defined disaster area UAV communication networks," *IEEE Journal of Communications and Networks*, vol. 20, no. 5, pp. 452–463, Nov. 2018.

[36] L. Ferranti, S. D'Oro, L. Bonati, E. Demirors, F. Cuomo, and T. Melodia, "HIRO-NET: Self-organized robotic mesh networking for internet sharing in disaster scenarios," in *Proc. of IEEE WoWMoM*, Washington, D.C., USA, June 2019.

[37] V. Sharma, M. Bennis, and R. Kumar, "UAV-assisted heterogeneous networks for capacity enhancement," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1207–1210, Apr. 2016.

[38] Z. Yuan, X. Huang, L. Sun, and J. Jin, "Software defined mobile sensor network for micro UAV swarm," in *Proc. of IEEE ICCRE*, Singapore, 2016.

[39] A. French, M. Mozaffari, A. Eldosouky, and W. Saad, "Environment-aware deployment of wireless drones base stations with google earth simulator," *arXiv preprint arXiv:1805.10424*, May 2018.

[40] O. Shrit, S. Martin, K. Alagha, and G. Pujolle, "A new approach to realize drone swarm using ad-hoc network," in *Proc. of IEEE Med-Hoc-Net*, Budva, Montenegro, Apr. 2017.

[41] G. Secinti, P. B. Darian, B. Canberk, and K. R. Chowdhury, "Resilient end-to-end connectivity for software defined unmanned aerial vehicular networks," in *Proc. of IEEE PIMRC*, Montreal, QC, Canada, Oct. 2017.

[42] H. Iqbal, J. Ma, K. Stranc, K. Palmer, and P. Benbenek, "A software-defined networking architecture for aerial network optimization," in *Proc. of IEEE NetSoft*, Seoul, South Korea, June 2016.