On the Design of the Drone Control Layer

Giovanni Grieco, *Graduate Student Member, IEEE*, Giovanni Iacovelli, *Graduate Student Member, IEEE*, Pietro Boccadoro, *Member, IEEE*, Luigi Alfredo Grieco, *Senior Member, IEEE*

Department of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy
CNIT, Consorzio Nazionale Interunivesitario per le Telecomunicazioni, Bari
Email: {giovanni.grieco, giovanni.iacovelli, pietro.boccadoro, alfredo.grieco}@poliba.it

***Abstract*—The Internet of Drones (IoD) is a networking architecture born to leverage the huge potential of Internet-connected drones. A key challenge for the IoD is represented by the technological fragmentation of its components. To this end, a middleware solution, namely Drone Control Layer (DCL), is proposed hereby, which enables complex mission design when heterogeneous swarms are considered. The DCL is made of four interfaces which abstract underlying drivers and hardware, provide a set of common primitives to applications, enable communications between drones, and other logical entities. To illustrate its applicability, relevant scenarios of interest are analyzed in details.**

***Index Terms*—Internet of Drones, 5G, Modeling Techniques.**

I. Introduction

The IoD is a major leap in telecommunications. It is a cutting-edge networking architecture aiming at interconnecting drones to the rest of the Internet [1]. Indeed, the IoD eases trajectory planning, mission design, flight control, resource optimization, and swarm management at scale [2]. Indeed, drones' intrinsic versatility allows them to be employed in several scenarios and applications, such as monitoring, surveillance, payload transportation, and Flying Base Stations (FBSs) [3], to name a few. Moreover, Unmanned Aerial Vehicle (UAV) technological landscape is extremely heterogeneous, therefore interoperability issues could arise in complex missions. As a matter of fact, current available applications strongly depend on specific hardware and software environments which threatens/slows down drones' employment at scale. To boost the deployment of the IoD, such dependencies should be avoided thanks to a de-verticalizing platform that grants portability. Furthermore, drone infrastructures require highly qualified personnel with expertise spanning over multiple aspects of hardware, mechatronics, network, and software engineering. A middleware with a flexible software interface eases the management of multiple subsystems and their requirements to be satisfied [4].

Thanks to the recent advancements in cloud and edge computing, the use of a middleware in the IoD enables complex mission design, off-the-shelf software-defined components, integrated service provisioning, and management at a glance. Mission plan can be envisioned as a composition of different containers in a micro-service development environment [5]. On these bases, this work designs a middleware solution, namely DCL, that is located between the transportation and application layer. It helps to define the underlying platform to abstract mission planning from drone peculiarities, while providing a safe and unified control structure. Meanwhile, the DCL grants the management of drones by means of a common set of interfaces with predictable responses. This facilitates applications' portability. The DCL identifies each elementary component of a given mission plan in order to further assign them to dedicated, yet specific, core modules, which cooperate towards mission accomplishment. The middleware has four interfaces which enable (i) the abstraction of underlying drivers and hardware, (ii) the use of common primitives for application development, (iii) the communication between drone and other logical entities, and (iv) core functionality extensibility.

Differently from the current state of the art [6]–[11], which envisions drone as standalone entities, the DCL allows seamless management and coordination among swarms of heterogeneous drones. Moreover, the DCL enables the deployment across multiple unmanned systems.

Inspired by cloud and edge practices, DCL-enabled drones working together with other logical entities in the IoD (e.g., Unmanned Traffic Management, Air Traffic Control (ATC) services, weather stations, and recharging stations) can be envisioned as resources, thus naturally becoming Platform as a Service (PaaS). The proposed solution allows a wide applicability in several scenarios of interest. For the sake of concreteness, public safety and FBS scenarios are deeply investigated.

II. Reference Scenario

The reference scenario (Figure 1) envisions swarms of drones assigned to different missions. Each mission is composed by a specific set of operations. Drones are grouped to form a swarm based on their on-board equipment and capabilities. Mission assignment and drones' enrollment are handled by dedicated logical entities:

- Mission Control Center (MCC): ground control infrastructure that designs the mission and monitors the swarm.
- Drone Orchestration Center (DOC): a central hub used to enroll drones in a mission by matching their characteristics with the operations to be done.
- Cloud-based Application Services: high-level functionalities for data mining and analysis related to the mission.

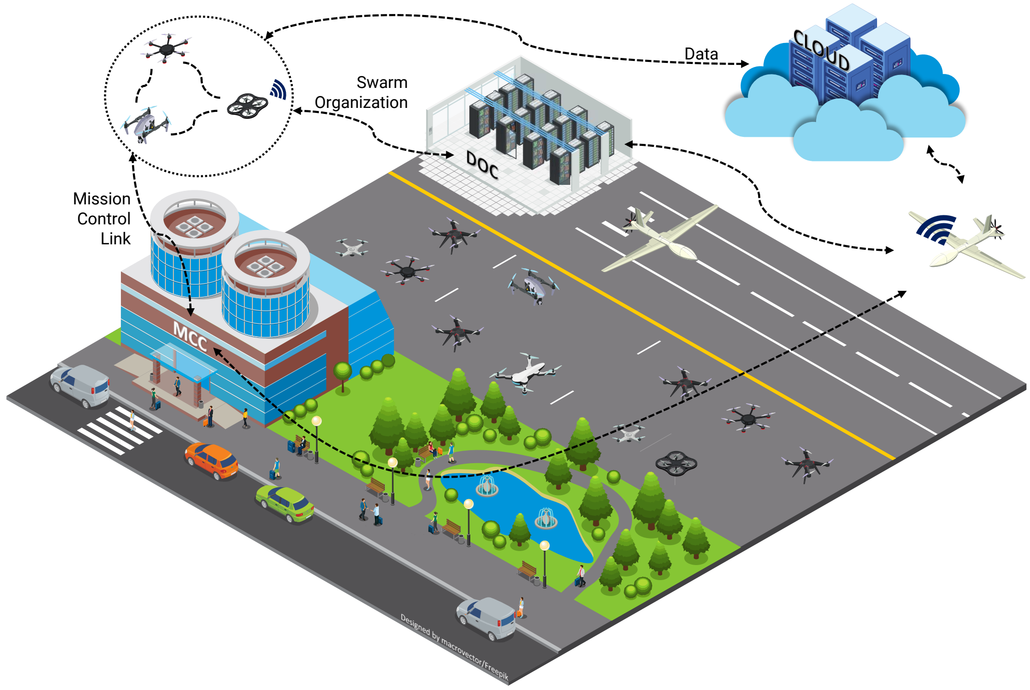

Fig. 1: Schematic representation of the envisioned scenario.

Drones are assigned to a specific mission before take off. The assignment procedure is illustrated in Figure 2 and composed by the following steps:

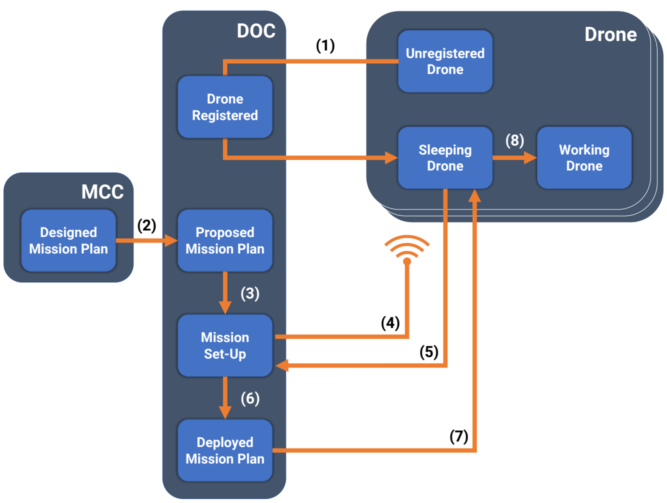

Fig. 2: Addressing solution for the identification of the swarms and roles related to drones.

1) Drones join the IoD network thanks to a registration procedure at the DOC. On top of it, the drone is able to announce its availability to the network. As a consequence, the drone is enlisted among those that the DOC can assign to a mission. To this end, the drone receives a logical address, namely Global Drone Identifier (GDI).
2) The design of the configuration includes the mission plan and the list of roles to be accomplished within the mission. This is carried out by the MCC. Once the DOC is notified with the published configuration, drones can be assigned to the mission.
3) The DOC is now able to process the received information, to specify the mission goal in terms of required drones' capabilities, to achieve the desired objective.
4) The mission plan is announced to all drones in the IoD network.
5) The idle (i.e., not involved in a mission) and suitable (i.e., with the proper characteristics) drones are required to probe the pre-flight checklist and announce its outcome to the DOC. In particular, their compatibility can either be *partial* or *full*.
6) The received information are processed by the DOC to select drones that are most suitable for the mission. As a consequence, a structure that couples each drone to its activity is set up.
7) The DOC leverages the GDI to communicate to each drone its assignment. It might happen that the number of idle drones does not cover the mission needs. In this case, the DOC notifies it to the MCC to abort the mission.
8) If the assignment is concluded successfully, a bootstrap procedure takes place to define the logical swarm domain. Members of the group become logically identifiable by a context-based addressing schema focused on swarm participation and drone identification.

Once the swarm formation setup is completed, drones can start exchanging data with the MCC and execute the mission tasks.

III. Interfaces Characterization

The architecture of the DCL is graphically introduced in Figure 3. The DCL includes four main interfaces: Northbound Interface (NbI), Southbound Interface (SbI), Westbound Interface (WbI), and Eastbound Interface (EbI).

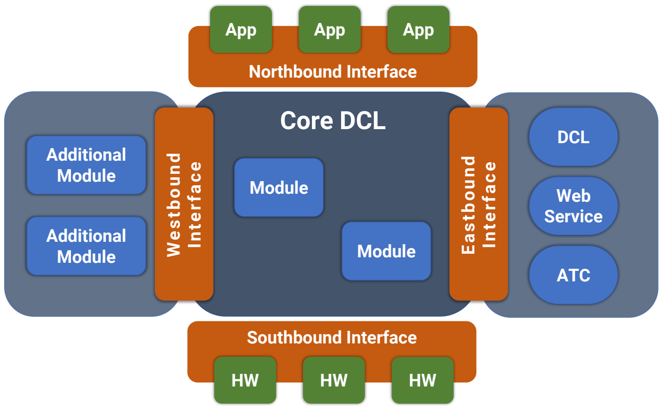

Fig. 3: High level overview of DCL Interfaces and their types of applicability.

A. Northbound Interface

The NbI supports the development of high-level applications on top of the DCL to ease drone's control, communications and mission planning. As illustrated in Figure 5, the NbI provides an event-driven notification mechanism that eases information exchanges towards upper-level applications. For instance, drones in a swarm are able to react to updates and dynamically adjust their configuration to newer roles, if needed.

At same time, applications may use Input/Output functionalities to exchange data and information with other DCL modules according to their exposed features on the interface. Messages travel in both directions using push-based, pull-based, pub/sub mechanisms according to application design and its requirements. Through the NbI, the application can inspect the network and automatically discover its topology and recognize other network entities. It can also exchange mission information to cooperate for task accomplishment. In the same manner, it would be possible to optimize flight control operations and make use of hardware capabilities.

In this way the application can influence drone operations and profits from its capabilities, while the DCL can ensure operations are correctly applied to the drone and notifies the application in case of state changes.

B. Southbound Interface

The SbI enables the interactions among the on-board resources and all the upper layers. Such a component solves the problems arising from vendor-related dependencies. In fact, this interface recognizes and supports multiple protocol stacks, radio interfaces, flight control primitives, and hardware resources in order to provide a set of software abstractions. The SbI is organized in three logical contextual blocks, as depicted in Figure 5. The Protocol Stack Context abstracts on-board radio interfaces providing low-level networking primitives. The Flight Stack Context is a collection of drivers to control mechanical components of the drone and, hence, its motion. Finally, Hardware I/O Context aims at managing interactions with underlying peripherals, i.e., sensors and actuators.

C. Westbound Interface

The WbI enables the development of DCL integration with additional modules and customized algorithms in order to optimize operations and support emerging applications. Extensions can cover a wide range of further developments, such as energy optimization.

D. Eastbound Interface

The EbI allows horizontal logical communications with other DCL-enabled entities to synchronize information, e.g., tasks, missions, and establish swarm networks. It also allows to establish direct connections between the drone and the outer world, i.e., Internet.

DCL entities in the IoD can be reached by means of contextual logical addresses. In particular, as shown in Figure 4, addressing is two-folded: drones have an address of their own (i.e., the GDI), while, in case they belong to a swarm, they can be addressed as a group entity. Such addressing scheme is independent from the particular communication stack used by drones and will be explained in detail in the following Section.

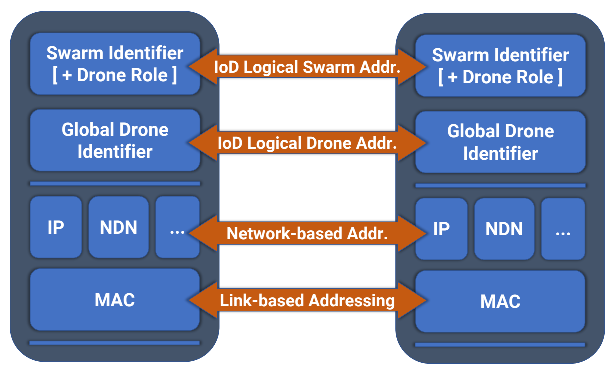

Fig. 4: Logical addressing types supported by the DCL EbI.

IV. Core Characterization

The architectural components of the DCL, and its organization (Figure 5), can be considered as the base fabric of all primitives that assist the planning of a mission. These elements characterize the kernel of a general purpose drone. What will be herein described may either be used as off-the-shelf functionalities or further extended to enable new services on top of them.

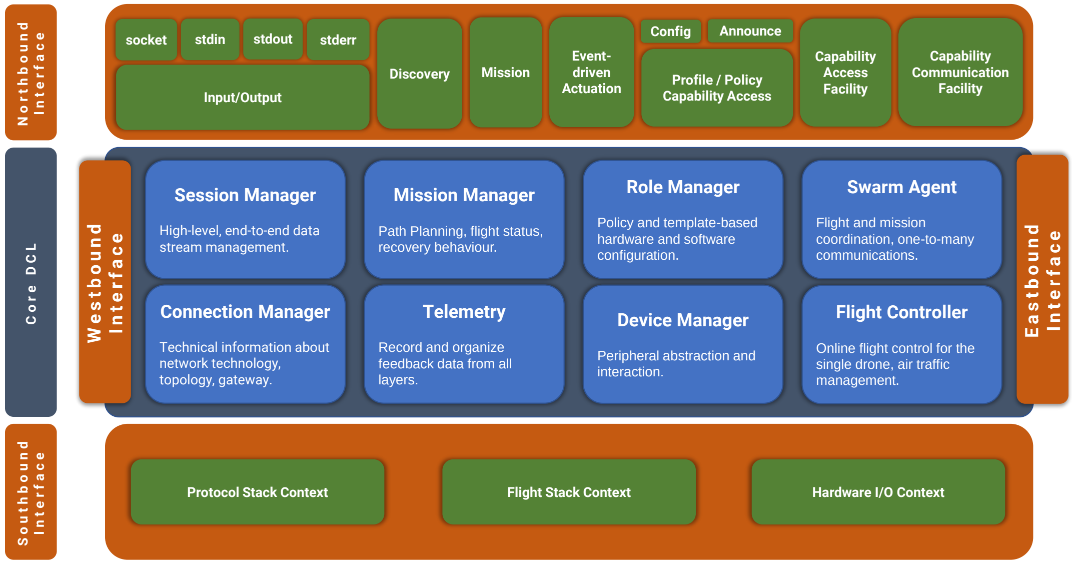

Fig. 5: High level architecture of the DCL.

A. Device Manager

The Device Manager provides handlers to orchestrate drone's hardware in terms of its capabilities. As a consequence, the Device Manager can be used to validate the suitability of the drone for the specific mission plan. Each hardware component is managed by a specific *Device Driver*. The latter exposes multiple high-level capability objects to communicate with such hardware component. Moreover, it publishes a manifest which includes a description of hardware functionalities together with its state, as represented in Figure 6. For instance, from an application point of view, the Device Manager does not provide detailed information about the specific on-board camera. Instead, it will indicate the drone with an on-board camera as enlistable for a mission during which images/video signals have to be gathered.

A single capability offers a data-driven, bidirectional communication channel to exchange data with the *Capability Communication Facility*. The joint adoption of the Device Manager and its drivers decouples high-level communication at the NbI from the specific hardware communication methodology (sequential or random access, exclusive or multiple access) and the detection of the component itself (driver, bus, battery, peripheral, and hot-plug support.)

B. Role Manager

Leveraging the high-level description provided by the Device Manager, the Role Manager is responsible for matching drones to a mission plan. To this aim, the specific mission requirements are encoded in a structured format and described as follows:

- *Rule*: it enables drone's configuration with reference to the working context and it is composed by:
  - *Subject*: the drone capability affected by the *Rule*.
  - *Predicate*: the action or condition that the *Subject* has to observe.

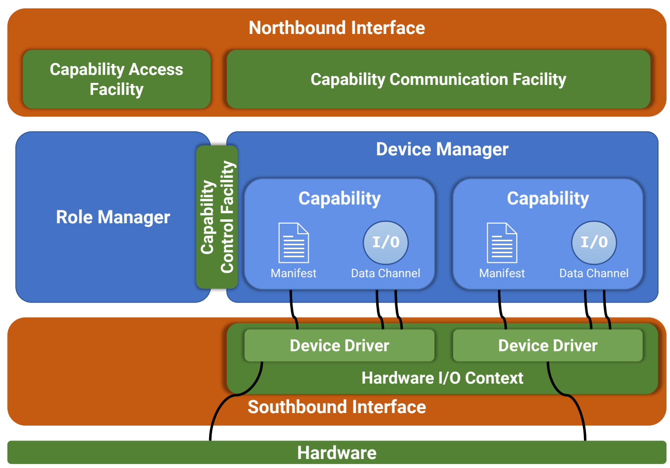

Fig. 6: Device Manager schematic characterization.

  Each *Rule* can either be *Observed* or *Not Observed*. To provide a concrete example, a certain amount of energy could be specified by the MCC as a requirement for the given mission plan. In case the condition for that mission is satisfied, the *Rule* is set to *Observed*.
- *Policy*: it is described by a decision tree, where each node uniquely identifies a *Rule*. The tree exploration is performed to evaluate drone suitability, given the mission requirements, encoded in these Rules. This structure has been chosen to ease rules evaluation through recursion. In case a rule cannot be observed, its subtree will not be satisfied, as well. According to this structure, the *Policy* can either be *Mandatory* or *Optional*: if the *Policy* is *Mandatory* the drone must observe each *Rule*, otherwise it may observe them, based on its capabilities. Thus, the *Policy* can assume three distinct states: *Observed*,

Partially Observed, and *Not Observed*.

- *Role*: one or more *Policies* characterizing the working context of a drone and defines its abilities with respect to the mission requirements. Each *Role* is uniquely identified by a Global Role Identifier (GRI) and can either be in the state of *Unassigned*, if the *Role* has not been assigned to a drone, or *Assigned*, if so. More drones can be employed to cover the same *Role* in the mission, hence they will be identifiable under the same GRI.
- *RoleSet*: group of *Roles* that satisfies mission requirements by employing multiple drones. Each *RoleSet* can assume the state of *Unassigned*, if no drone have been selected yet, or *Assigned* if each swarm component is compliant with the specified *Roles*.

Following the aforementioned Policy-based model, the Role Manager has to detect the optimal *Role* for the drone. The declarative nature of drone capabilities allows an easier detection of drone compatibility with the candidate *Role*.

For each *Rule* of a *Policy*, the Role Manager has to link *Subject* with capability and *Predicate* with the functionalities declared on the capability manifest. This operation is achievable through the employment of a marshalling mechanism. In this way, it is possible to completely abstract drone hardware at mission design and control. Thus, a *Role* is the characterization of the ideal drone for the mission. The importance of the Role Manager not only includes the necessity to find an optimal drone for the mission, but also to configure and contextualize the candidate drone in order to perform mission tasks ahead. This also comprehends mission requirements, e.g., geofencing and restrictions in the working airspace.

The contextualization of DCL components, depending on the mission, is important for the Device Manager. A high-level application can use a capability by requesting its access at the *Capability Access Facility* (reported as building block of the upper layer in Figure 6), which is managed by the Role Manager. If such capability is compatible with *Role* confinements and requirements, the access is granted. The application can then interface with the *Capability Communication Facility*, managed by the Device Manager, to use the requested and initialized resource.

In case a drone might not be able to complete the mission as expected, the MCC notifies the DOC that a swarm is no longer suitable for that mission. Nonetheless, there could be new drones available, ready to deal with that mission in a more efficient way, otherwise the mission is aborted. In the former case, the DOC suspends the mission and re-configures its *RoleSet*. The process is similar to the previous one: the DOC announces the *RoleSet* availability, awaits drones' response, and selects again a suitable group that replaces the current operating swarm for the mission. When the new swarm is ready, the DOC notifies back the MCC and the mission is resumed.

C. Telemetry

The Telemetry module aggregates and elaborates drone data in an uniform and declarative way. Information derived from the SbI, e.g., IMU, state, and diagnostic data, converge into this module and are provided to applications through the NbI or external entities via the EbI. This continuous data stream enables the assessment of drone operations being correctly executed. Moreover, this module can also act as an emergency black box in order to analyze data related to an unexpected event, like a system failure.

D. Connection Manager

The Connection Manager is focused on connection abstraction, which decouples data exchange from the protocol stack in use, thus yielding to context independent communications. The module maintains links with remote hosts using multiple radio and networking technologies according to drone *Role* and application requirements. Concretely, the module is in charge of finding the optimal setup for the requested communication channel. A fundamental additional service is neighborhood discovery, such as nearby drones and other relevant entities, e.g., MCC and DOC.

E. Session Manager

Leveraging the functionalities exposed by the Connection Manager, the Session Manager handles simultaneous logical links to ease context-based information exchange to/from external entities. The module provides a simple interface to open/close connections and send/receive end-to-end, broadcast and multicast information. As depicted in Figure 4, drone addressing is done through the use of a GDI to establish a session with it. Based on mission context and drone role, two more multicast addresses can be used, namely Logical Swarm Address (LSA) and GRI. The former is useful to broadcast information to an entire swarm, while the latter is appropriate in situations where a specific group has to be reached. Moreover, the LSA can be combined with the GRI to interact with drones that are assuming a specific role within the swarm.

F. Flight Controller

The Flight Controller provides a common set of commands that allows flight maneuvers, addressing hardware and software complexity. The Flight Controller has the responsibility of accepting such commands and controlling drone movements using its specific functionalities and drivers. The control logic that characterizes the Flight Controller must be flexible enough in order to find and use particular drone flight capabilities, e.g., drone's trajectory expressed as a sequence of steps, to change course in mid-flight in a short time, to stop in mid-air, to do acrobatics, and to move with high accuracy in constrained environments.

G. Swarm Agent

The Swarm Agent is specifically designed to handle information regarding mutual coordination among drones forming a swarm. Such messages allow swarm cooperation in order to achieve mission tasks. Specific functionalities include flight coordination and collision avoidance that rely on a continuous exchange of the relative position and speed, on top of low-latency communication channels. The Swarm Agent entrusts the EbI and the Session Manager to synchronize with other Swarm Agents. Clearly, such feature is not available if the drone is not part of a swarm.

H. *Mission Manager*

The Mission Manager is focused on monitoring the drone status and on planning the sequence of tasks to be completed. This sequence of tasks can be received directly by the MCC in a structured format named *Mission Plan*, or be streamed from another drone locally coordinating the swarm. Upon schedule, the Mission Manager orchestrates DCL modules to achieve each task, disseminating the derived information to the appropriate modules. Furthermore, during the mission it is of importance to analyze the telemetry and detect anomalies or any other critical element that could suggest the activation of emergency procedures. It is also important that the Mission Manager is highly configurable and flexible, to confirm it as a general purpose endpoint. Task scheduling and operational strategies can differ among missions or be limited by environmental restrictions.

V. Application Domains and Relevant Use Cases

To prove the applicability of the DCL, two real-world use cases are described hereby, starting from the bootstrap procedure they share. Without loss of generality, it is hereby assumed that all the operations described in the reference scenario have been completed. At the beginning of the mission, the drone receives the mission plan by the DOC.

The mission plan is a set of information including: flight trajectories, specific role assigned to the drone, as well as its possible membership in a swarm. All the components of the mission plan are contextualized and dispatched to the reference logical modules to be initialized.

When the mission includes a single drone, the addressing scheme requires the GDI and the GRI, so that the drone will be assigned both a logical address and an information connected with its role in the mission. In case the mission requires a swarm, the addressing scheme will still require the GDI and the GRI, but the two will come along with the LSA. This is motivated by the fact that the LSA indicates the group of drones belonging to a specific logical network.

The bootstrap procedure continues with the notification of the new mission plan to the applications by the DCL through the NbI. When this bootstrap phase is over, the mission can begin.

A. *Public Safety*

Drone technology has been applied in monitoring and containing the COVID-19 pandemic in several countries worldwide. Crowd surveillance has been successfully applied in China to detect gatherings of people. In Spain, drones equipped with speakers were used to inform people about current regulations. In particular situations, drones were equipped with multi-spectral cameras to measure body temperature en masse, or with medical supplies to safely ship to the destination [12].

Fig. 7: Example of drones employed in a public safety use case.

In a similar use case, as depicted in Figure 7, two main roles for drones are envisioned. The former, namely announcer, is in charge of broadcasting safety measures, e.g., observing social distancing, frequently sanitizing hands, and using masks. The latter, namely detector, employs multi-spectral cameras to detect body temperatures and crowds.

Announcers have to roam the whole area of interest avoiding collisions. To this aim, core functionalities may be extended with optimization algorithms implemented through the WbI, and used by applications via NbI. Moreover, a collision avoidance mechanism, relying on intra-swarm communications, is provided by the Swarm Agent. Indeed, it is assumed that each drone periodically broadcasts its Global Position System (GPS) position through the EbI that relies on the Session Manager, leveraging the LSA. The application that characterizes this drone class picks pre-recorded audio messages available on-board and dispatches them through the audio capability exposed by the Device Manager. Then, the SbI takes care of encoding and playing audio messages.

Detectors have to patrol the same area of interest, searching for crowds and measuring body temperatures. Even in this case, optimization algorithms are necessary to maximize, without colliding, the covered area. Video signals, obtained from the camera capability, feed an application-level machine learning model. It is able to recognize gatherings, generating a notification that is sent to a cloud-based service monitored by authorities. At the same time, an alert is delivered to announcers to reach and warn the target audience. Both intra/inter swarm communications take place thanks to EbI, which leverages Swarm Agent and Session Manager primitives.

Throughout the mission, both classes continuously stream telemetry to the MCC for ATC activities. Telemetry messages are handled by the dedicated DCL module. Also, for each drone, Flight Controller relies on the Mission Manager, which constantly updates trajectory according to surroundings.

Fig. 8: Example of drones employed in a FBS scenario.

B. Flying Base Station

Among 5G&Beyond applications in which drones are involved, one of the most promising is the FBS. Scientific literature has been deeply analyzed and this use case has been discussed from an optimization point of view [13], [14]. In fact, drones are warmly recommended in all those situations in which connectivity is poor or lacking, e.g., rural zones, or disaster areas. This is due to their ability to extend, restore, or, in the worst case provide, radio coverage to Ground Users (GUs).

In this use case, illustrated in Figure 8, it is assumed that each drone is equipped with two radio interfaces. Each of them is dedicated to a communication link, one for UAV-GU and the other for UAV-Ground Base Station (GBS).

Upon reception of the GU traffic, packets reach the SbI, specifically the Protocol Stack Context and then are forwarded to the relay application, through the NbI. The application is in charge of processing and routing data over the service gateway via the EbI. To this aim, specific optimization frameworks are needed to manage radio resources, leveraging extension modules of the WbI. The same operations are applied to the inbound traffic received from the service gateway. Moreover, among the outputs of the optimization process there is the trajectory, which is utilized by the Mission Manager. This information, that is forwarded to the Flight Controller, is crucial to guarantee optimal coverage and service quality.

Based on the available intra-swarm communication technologies, drones can also be organized into multi-hop topologies to further extend the distance between UAV serving GUs and the GBS. Furthermore, the Swarm Agent is involved to coordinate the fabric of interconnected UAVs.

VI. Conclusion

In this work, the DCL middleware is proposed to provide seamless interoperability within the IoD. With DCL, mission management is addressed through a set of core functionalities exposed by modules, while interoperability with external entities is enabled by interfaces. On these bases, the DCL extends cloud concepts and applies them on this new fabric, proposing a PaaS that further enhances mission design. To demonstrate its effectiveness, two possible use cases are deeply examined, focusing on the interaction among the involved modules and interfaces. In the future, a concrete implementation of the proposal will be deployed on drones operating in real-world scenarios.

Acknowledgments

This work was partially supported by the Italian MIUR PON projects Pico&Pro (ARS01_01061), AGREED (ARS01_00254), FURTHER (ARS01_01283), RAFAEL (ARS01_00305) and by Warsaw University of Technology within IDUB programme (Contract No. 1820/29/Z01/POB2/2021).

References

[1] M. Gharibi, R. Boutaba, and S. L. Waslander, “Internet of drones,” *IEEE Access*, vol. 4, pp. 1148–1162, 2016.

[2] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, “Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges,” *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019.

[3] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, “A tutorial on uavs for wireless networks: Applications, challenges, and open problems,” *IEEE Commun. Surveys Tuts*, vol. 21, no. 3, pp. 2334–2360, 2019.

[4] N. Yumi, S. Yuki, and O. Tetsuya, “An effective visual programming tool for learning and using robotics middleware,” in *2016 IEEE/SICE International Symposium on System Integration (SII)*, 2016, pp. 156–161.

[5] J. Boubin and C. Stewart, “Softwarepilot: Fully autonomous aerial systems made easier,” in *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, 2020, pp. 250–251.

[6] A. Koubâa, B. Qureshi, M.-F. Sriti, A. Allouch, Y. Javed, M. Alajlan, O. Cheikhrouhou, M. Khalgui, and E. Tovar, “Dronemap planner: A service-oriented cloud-based management system for the internet-of-drones,” *Ad Hoc Netw.*, vol. 86, pp. 46–62, 2019.

[7] G. Kakamoukas, P. Sarigiannidis, and I. Moscholios, “High level drone application enabler: An open source architecture,” in *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2020, pp. 1–4.

[8] E. Frew, K. Glasheen, C. Alexander Hirst, J. Bird, and B. Argrow, “A dispersed autonomy architecture for information-gathering drone swarms,” in *2020 IEEE Aerospace Conference*, 2020, pp. 1–11.

[9] A. Essameldin and K. A. Harras, “The hive: An edge-based middleware solution for resource sharing in the internet of things,” in *3rd Workshop on Experiences with the Design and Implementation of Smart Objects*, 2017, p. 13–18.

[10] D. Marcheras, M. Ayaida, N. Messai, and F. Valentin, “A new middleware for managing heterogeneous robot in ubiquitous environments,” in *2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2020, pp. 1–5.

[11] J. Besada, L. Bergesio, I. Campaña, D. Vaquero-Melchor, J. López-Araquistain, A. Bernardos, and J. Casar, “Drone mission definition and implementation for automated infrastructure inspection using airborne sensors,” *Sensors*, vol. 18, no. 4, p. 1170, 04 2018.

[12] V. Chamola, V. Hassija, V. Gupta, and M. Guizani, “A comprehensive review of the covid-19 pandemic and the role of iot, drones, ai, blockchain, and 5g in managing its impact,” *IEEE Access*, vol. 8, pp. 90 225–90 265, 2020.

[13] J. Chakareski, S. Naqvi, N. Mastronarde, J. Xu, F. Afghah, and A. Razi, “An energy efficient framework for uav-assisted millimeter wave 5g heterogeneous cellular networks,” *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 37–44, 2019.

[14] N. Zhao, X. Pang, Z. Li, Y. Chen, F. Li, Z. Ding, and M. Alouini, “Joint trajectory and precoding optimization for uav-assisted noma networks,” *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3723–3735, 2019.