End-to-End Design for Self-Reconfigurable Heterogeneous Robotic Swarms

Jorge Peña Queralta[1], Li Qingqing[1], Tuan Nguyen Gia[1], Hong-Linh Truong[2], Tomi Westerlund[1]

[1] Turku Intelligent Embedded and Robotic Systems (TIERS) Lab, University of Turku, Turku, Finland

[2] Department of Computer Science, Aalto University, Finland

Emails: [1]{jopequ, qingqli, tunggi, tovewe}@utu.fi, [2] linh.truong@aalto.fi

Abstract—More widespread adoption requires swarms of robots to be more flexible for real-world applications. Multiple challenges remain in complex scenarios where a large amount of data needs to be processed in real-time and high degrees of situational awareness are required. The options in this direction are limited in existing robotic swarms, mostly homogeneous robots with limited operational and reconfiguration flexibility. We address this by bringing elastic computing techniques and dynamic resource management from the edge-cloud computing domain to the swarm robotics domain. This enables the dynamic provisioning of collective capabilities in the swarm for different applications. Therefore, we transform a swarm into a distributed sensing and computing platform capable of complex data processing tasks, which can then be offered as a service. In particular, we discuss how this can be applied to adaptive resource management in a heterogeneous swarm of drones, and how we are implementing the dynamic deployment of distributed data processing algorithms. With an elastic drone swarm built on reconfigurable hardware and containerized services, it will be possible to raise the self-awareness, degree of intelligence, and level of autonomy of heterogeneous swarms of robots. We describe novel directions for collaborative perception, and new ways of interacting with a robotic swarm.

Index Terms—Swarm Robotics; Edge Computing; Elasticity; Dynamic Resource Management; Autonomous Robots
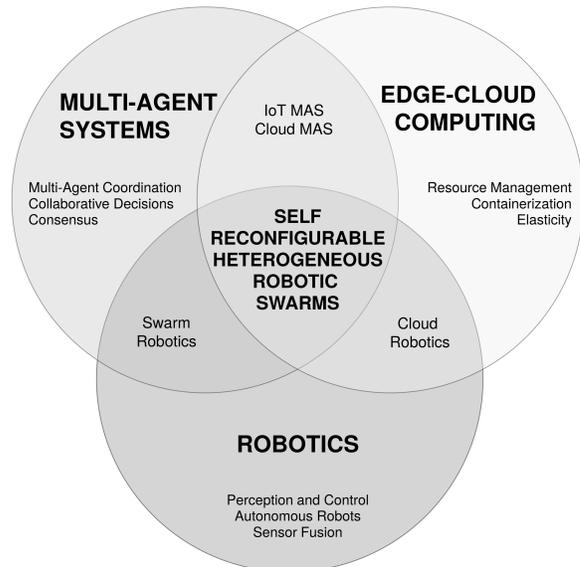
Fig. 1: Intelligent and self-reconfigurable robot swarms can be designed at the intersection of the multi-agent systems (MAS), Robotics and Edge-Cloud Computing domains.

I. INTRODUCTION

A recent trend in cyber-physical systems and the Internet of Things (IoT) domain is to shift towards more distributed computation, a trend that has crystallized through the edge computing paradigm [1]. Similarly, recent advances in containerization, elastic computing, and dynamic resource management are materializing a decentralized cloud [2], [3]. Multiple researchers have explored the possibilities of integrating Multi-Agent Systems (MAS) theory within the IoT and cloud computing towards IoT MAS and Cloud MAS [4]. On the other side, the combination of MAS and robotics has brought swarms of robots with the potential for enhancing human responses in safety-critical applications such as firefighting [5], or post-disaster scenarios [6], among others.

We extend these two approaches (swarm robotics and IoT MAS) towards the intersection of edge computing and robotics. and argue that with appropriate management and distribution of computing, sensing and communication resources within a robotic swarm, higher degrees of intelligence and operational flexibility and robustness can be achieved. This concept is illustrated in Fig. 1. The capabilities of robotic swarms are currently limited by different factors, from the lack of methods and distributed collaborative sensing algorithms [7], to static and inflexible resource management with non-optimal utilization of hardware resources due to separate hardware and software design. This includes embedded hardware with relatively constrained computational resources due to payload constraints and uniform resources in swarms [8], [9]. Some previous works have addressed these limitations through computational offloading and the definition of edge-cloud robotics architectures [10]. More recently, heterogeneous robotic swarms have been proposed to extend the flexibility and intelligence, opening the door to a wider array of more complex application possibilities [7]. Nonetheless, multiple challenges remain in terms of managing the collaboration within heterogeneous swarms [11].

A. Related Works

Multiple research efforts within the fields of multi-robot systems and cloud computing have been directed towards distributed task allocation and distributed load management. For instance, autonomous mobile programs (AMPs) were an early introduction of a dynamic computational load manage-

ment framework [12]. AMPs provided a distributed approach where autonomous agents were able to make decisions on a shared computational load, being aware of their own computational capabilities. AMPs share similarities with early load balancing techniques based and colony optimization for cloud computing [13]. More recently, multi-agent load balancing for resource allocation in a distributed computing environment has been proposed [14]. From the point of view of task allocation in multi-robot systems, K-means clustering and auction based mechanisms were introduced in [15]. In terms of spatial allocation, a workspace partitioning method was presented in [16] for indoor environments. In our work, we aim at combining these two approaches considering full reconfigurability through tight integration of methods form the edge computing domain and algorithms for cooperation in multi-robot systems. This is, to the best of our knowledge, the first paper presenting such approach.

In our recent works, we have presented initial results towards the definition of reconfigurable robotic swarms. In [17], we define the concept of resource ensembles from the perspective of the edge computing domain. This concept serves as the basis for abstracting edge resources and building dynamic management models on top of them. From the point of view of collaborative swarms of robots, we have presented a blockchain-based approach in [11]. In this work, we utilize a blockchain as a medium for achieving consensus for bandwidth allocation and data quality ranking in a distributed multi-robot system. This, again, serves as the starting point towards distributed sensing and data processing in swarms of robots, where sensing, network and computational resources are abstracted and managed through collaborative decision making.

B. Contribution and Structure

The techniques we propose have a clear impact on swarms of drones. Current solutions for drone swarms require operators to either manually control drones or perform analysis of streamed data at a ground control center. This is a limitation for the deployment of drones in large areas where there have been natural disasters such as fires, or where people have gone missing, as the human resources necessary are too large. Even if the data is processed by a computer at the ground station, the need for a high-bandwidth channel between drones and the base station still limits significantly their operational capabilities. Therefore, there is an evident need for more intelligent drones that are able to perform data analysis independently and autonomously navigate large areas. Reconfigurable drone swarms empower complex edge data analysis at the swarm level through distributed edge computing. At the same time, this allows for long-term autonomous operation when energy constraints allow, as well as reconnaissance in remote areas with poor network connectivity.

The main objective of this paper is to introduce a *new design approach* that enables efficient and dynamic resource management and autonomous reconfiguration of heterogeneous robotic swarms (Sections II & III). We discuss the optimization

of the various computing resources and sensing capabilities of robots in the swarm through a hardware (HW)/software (SW) co-design approach for the development of specialized robots. In particular, we incorporate elastic principles for coordinating and engineering collective capabilities of multiple heterogeneous robots. This elasticity takes into account the coordination at the level of multi-agent systems, but also the specific resources and algorithms utilized for robotic perception and navigation, among others.

Furthermore, we discuss how our proposed approach enables the reconfiguration of drone swarms and realize a distributed collective intelligence (Section IV). This requires embedding intelligence and information processing in the drones themselves. With current technology, deploying multiple drones requires coordination among their operators, binding valuable resources from the actual mission. Even more, when multiple drones are being deployed in parallel [18], [19]. However, neither isolated intelligent drones nor simple task list orchestration is sufficient [20], [21]. Thus, it is essential to establish a collective intelligence that enables autonomous coordination and collaboration among the drones.

The remainder of this paper is organized as follows. Section II describes the main concepts involved in swarm reconfigurability, interfacing and control. In Section III, we introduce a three-layered end-to-end design architecture for reconfigurable robotic swarms. Section IV then delves into enabled technologies, focusing on our initial work towards a reconfigurable drone swarm. In Section V, we discuss potential application scenarios. Section VI concludes the work.

II. MODELS

We address the aforementioned challenges through an end-to-end design: from the robot hardware and local decision making on the computational and sensing resources to providing the swarm and its capabilities as a service for end-users. Two key aspects in our work are:

- Reconfigurable hardware resources for flexible and resource-rich computation platform at the swarm level
- Distributed data processing at the edge for collective swarm intelligence.

Reconfigurability and distributed intelligence provides more computational resources for multi-modal sensor fusion and distributed computation. Our model leverages methodologies and engineering techniques for distributed and dynamic management of elastic resources for swarms improving quality of results. The models are:

1) Swarm-as-a-Service Model: A swarm provides services to end-users with a control interface or API through what we call a Swarm as a Service (SwaaS). SwaaS offers an edge service model for swarm applications. In this view, each robot with its specialization is an edge services provider (or just an edge provider). Edge provider's sensing, computational and external communication resources are considered edge resources. Edge resources (hardware and software) are co-designed to make robots richer in terms of the resources that they can provide. With this approach, we bring various

(a) A cloud robotics system.

(b) A swarm robotics system.

(c) A reconfigurable robotic swarm.

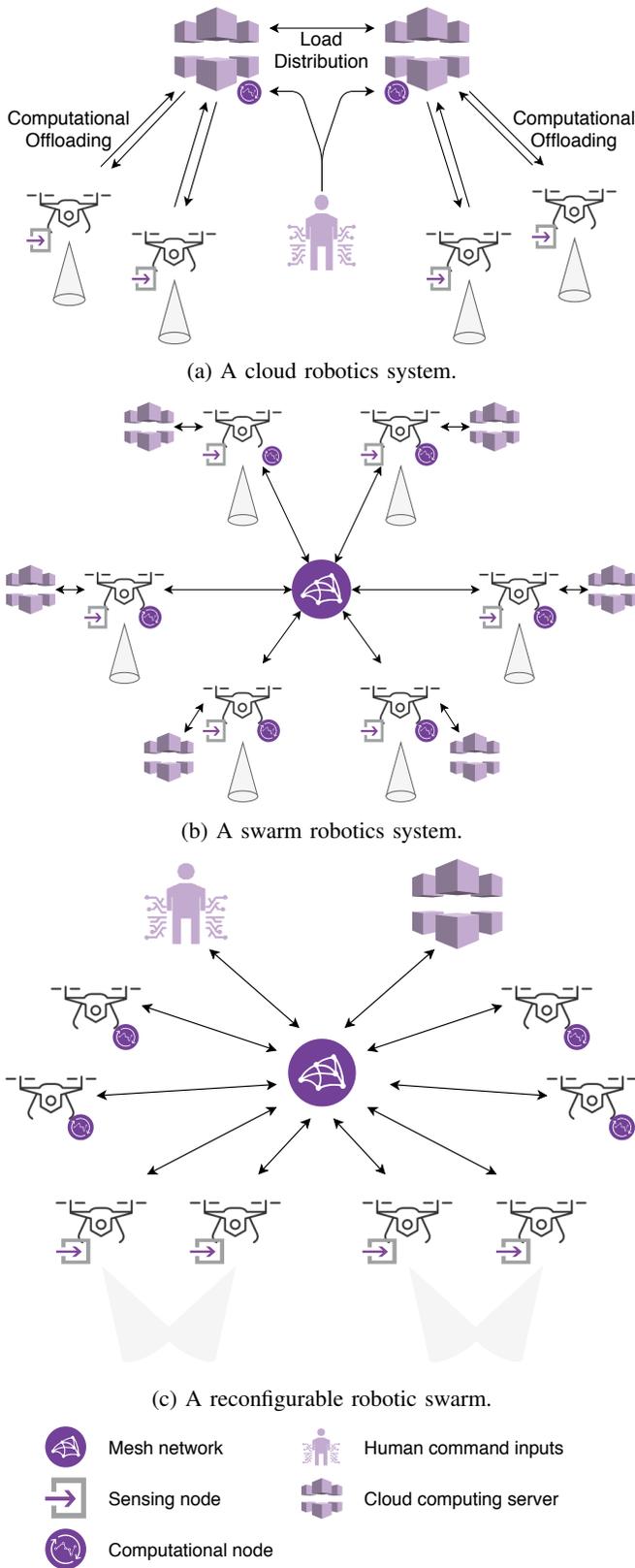|			
Mesh network		Human command inputs	
Sensing node		Cloud computing server	
Computational node			

Fig. 2: Illustration of different connectivity modalities for a swarm of drones, illustrating sensing and computing roles together with interaction modalities.

concepts of edge computing and services models to the field of multi-robot systems.

2) Application-Specific Resource Ensembles Model: Application Specific Resource Ensembles (ASREs) [17] define a specific organized set of edge resources (ASRE template or pattern) in swarms forming the edge infrastructure. An essential part of ASREs is the coordination and monitoring of resources together with swarm control and coordination. Resource management techniques enable service discovery, service end-to-end communication segmentation, and distributed task computation under unreliable and uncertain environments by incorporating uncertainty and elasticity [22]. For our knowledge, these have not beed applied before to drone swarms. For example, [23], [24] are dedicated for containers and virtual machines.

In our end-to-end vision, the central point is to provide dynamic and flexible swarms as an *elastic heterogeneous multi-robot system*. In the system individual robots have different sensing and computational capabilities, a mesh network takes care of intra-swarm communication, and distributed algorithms enable the swarm to perform collective decision making as if it were a single unit. By *elastic*, we mean that the different resources of robots are abstracted and can be reconfigured depending on the application needs [25]. An illustration of this concept, compared to current cloud robotic and swarm robotic systems, appears in Fig. 2. In Fig. 2a, each drone is independently connected to a cloud server, where it offloads part of its data processing. Any external control in this case goes through the cloud application, but direct control of drones could be enabled as well (for example, if the movement of the drone is controlled via a radio controller, and then mapping or other algorithms are run in the cloud). In Fig. 2b, the drones form together a swarm. Each individual drone in the swarm has the same role initially, and both sensing and data processing occur individually at each drone. Algorithms describing the collaboration between drones could then run at the swarm level, but each drone would be still a separate entity. Finally, our approach is illustrated in Fig. 2c, where all drones form a swarm as well. The key difference is that the swarm and its applications and resources are abstracted from individual drones and defined in a distributed manner at the swarm level. The communication with a controller or cloud services occurs from the swarm as a whole, and not from individual drones as separate entities. Moreover, sensing and computational resources, and the corresponding roles, are assigned dynamically among the swarm members. In the example illustrated in the figure, half of the drones take a sensing role as their main role while data processing is offloaded to other drones assigned as computing nodes.

III. ARCHITECTURE LAYERS

We design an architecture with three layers and building blocks as illustrated in Fig. 3. The layers are:

Physical Swarm Layer. The actual control of robots is carried out at the physical layer, where the different hardware and mesh communication solutions are defined. The definition
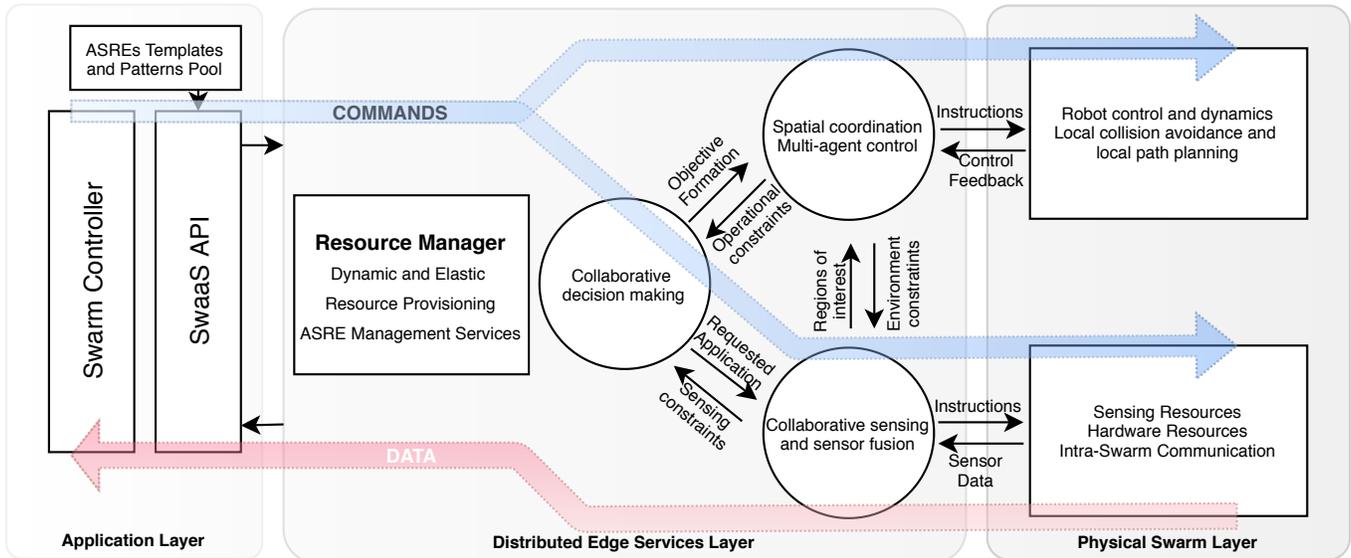
Fig. 3: Proposed Architecture and Building Blocks for SwaaS and ASREs

of a set of computing resources, sensors and actuators needs to be carried out as part of the swarm design in order to enable higher degrees of optimization when the sensing and computational resources are shared at the swarm level and applications run in a distributed manner.

Edge Services Layer. This is the main focus of the swarm design in terms of self-reconfigurability. This layer includes a Resource Manager (RM) for configuring and managing resources provided by the Physical Swarm Layer. Based on application requirements, the RM will assemble resources from edge providers into ASREs for SwaaS. The RM supports the automatic creation of ASREs by requesting, provisioning and orchestrating suitable resources.

This layer represents all the distributed services and processes running within the swarm and executed at each individual robot. These processes are classified in three main types: (1) spatial coordination (e.g., distributed formation control [26], [27]), (2) collaborative sensing (e.g. cooperative mapping [28]), and (3) collaborative decision making (e.g.). These three apparently different parts of swarm control and decision making have a high synergy and their optimal operation depends on feedback from each other. These three topics have mostly been studied separately in the previous works [29], [30], [31]. Therefore, we have focused on the design and development of techniques for efficient communication between these processes. In summary, the key novelty is that ASREs and the RM implicitly manage collaboration within the swarm.

At runtime, ASREs will form an elastic and resilient edge mesh of services across robots in a swarm. The RM will dynamically provision new resources from different providers elastically. This kind of elasticity is carried out in an end-to-end and bi-directional manner: the resources are provisioned dynamically when new services are required or when the

available resources change. The RM learns and optimizes the provisioning based on the reliability of resources, performance variations, bottlenecks, and failures. The RM provisioning is hidden from the application.

Application Layer. The application layer provides an interface for controlling and interacting with the swarm. We refer to this interface as the Swaas API. An external party or swarm controller can select from a pool of ASRE templates, which define the different patterns in which the swarm can be configured for different applications. By choosing an ASRE template through the SwaaS API, the swarm controller is implicitly selecting a set of resources and services. These resources are then provisioned and managed within the swarm itself through the RM. The services are provided based on the available distributed algorithms for sensing and coordination.

IV. RECONFIGURATION PROCESSES IN A DRONE SWARM

In this section, we discuss the specific technologies that enable the realization of the self-reconfigurable robotic swarm architecture defined in the previous section. We are in the process of applying these technologies to a swarm of drones.

A. Reconfigurability-enabling Technologies

Currently, most small mobile robots and aerial drones rely mainly on CPUs in order to perform all the navigation and mission related computation, and microcontrollers for low-level control such as flight controllers [32]. We are leveraging existing technologies from other domains to enable reconfiguration within a drone swarm.

As a computing platform, we utilize FPGAs with embedded processors to extend the existing algorithms with custom hardware accelerators. FPGAs are reconfigurable hardware accelerators that can be exploited in computationally intensive and highly parallelizable tasks for autonomous robots, with higher performance/size and performance/power ratio as we

have shown in previous works [33], [34]. Both the size and power consumption of hardware are essential aspects to take into account in drones. The use of FPGAs enables the RM to not only provision the existing resources but also dynamically provision new hardware accelerators on-demand. Some drones are equipped with FPGAs while others have embedded processors with GPU such as the NVIDIA Jetson TX2.

At the software level, we utilize containers to enable dynamic resource management and task execution. Container technologies are known but they have not been exploited for drones. Our goal is to use containers to enable dynamic resources management and task execution. All algorithms, from spatial coordination to collaborative sensing, are containerized and run in a distributed way. With efficient container orchestration, we are able to add flexibility and reconfigurability to the swarm. We bring specific techniques for computational load distribution, elasticity and resource management from the edge-cloud domain to the robotics domain.

In order to interface sensors, actuators, communication and the containerized algorithms, we utilize Robot Operating System (ROS2) which runs as a container application as well. ROS is the de-facto standard for robotic development in both academia and industry. ROS2 focuses on distributed multi-robot systems and real-time computing, and will allow us to exploit container technologies for drones.

For network interfacing, there are no solutions that integrate ROS2 and mesh networking so far. Our experiments will utilize more traditional solutions at first, with all drones connected to a single Wi-Fi access point. Nonetheless, we will work towards the integration of ROS2 and a Bluetooth 5 mesh network. Another recent technology that can provide significant advantages is ultra-wideband (UWB) [35]. UWB enables accurate localization in multi-robot systems and, in particular, in drones [36], and has the potential for simultaneous communication and localization. We will work on extending our current works on UWB-based mobile localization systems [37], studying the integration of UWB as a network interface between ROS2 nodes.

Finally, we leverage blockchain-powered consensus algorithms suitable for multi-robot systems. Recent works [38], [11], [39] have proposed design concepts for integrating next-generation low-latency and scalable blockchains within heterogeneous multi-robot systems. Blockchains can be utilized as a distributed framework to achieve consensus in a multi-robot system, and also validate identities. This can be then utilized by ASRE management services, which could, in turn, be implemented as distributed Smart Contracts for resource coordination. The idea of utilizing a blockchain-based framework for managing edge resources has already been explored in our previous works [40].

B. Edge computing algorithms for sensing

We classify the algorithms in the edge layer in three main types: spatial coordination, collaborative sensing, and decision making. In previous works, these are typically defined with strong dependencies. For instance, depending on the sensing variable robots might be required to be in a specific spatial formation [41]. However, in our architecture these algorithms are designed independently and abstracted as edge services. This modular architecture brings multiple advantages. For instance, spatial coordination algorithms take feedback from the sensing algorithms regarding the location of regions of interest that should be analyzed more closely. This feedback is used to rearrange the drones in the proper shape and location. At the same time, the role of each drone within the spatial pattern is given by the collaborative decision making process. Analogously, the spatial coordination algorithms give feedback to other processes about the movement constrains of the swarm surroundings.

C. Resource Management

The RM provisions and manages all resources, from hardware to edge services. Sensing resources are abstracted through ROS nodes (drivers) that produce data in standard formats. Each edge service is broken down into sub-services (for example, independent parts of an algorithm) and each of these is abstracted as a ROS node that consumes and produces different types of data (always in standard formats). All these ROS nodes are containerized and provisioned by the RM across the available the computing resources. The computing resources are modeled based on the amount and type of containers that they can run, and the performance for each containers. The provisioning is an optimization process that takes into account communication latency between data producers and consumers, and execution latency. The RM itself runs as a distributed and containerized application across the swarm, and manages the resources with elastic techniques.

Each application that utilizes the SwaaS API must define a Quality of Results (QoRs) requirement. We bring this concept from elastic computing models in which a QoR is defined in terms of performance, quality of data, type of output, and other measurable information [42], which the swarm provides to the controller through the SwaaS API. The QoR is essential for elastic resource management to dynamically provision the different resources taking the QoR requirement as an optimization constraint. We will extend the work by Mariani *et al.* on coordination-aware elasticity for developing primitives and algorithms to control the elasticity of swarms [43].

V. APPLICATION SCENARIOS

Drones are already being utilized by fire departments [44], police forces [45], and for creating ad-hoc networks in post-disaster scenarios [46]. Fire and police departments are using drones for monitoring and surveying emergency scenes and disaster sites as well as finding missing people. Going beyond perception-centric usage would be beneficial in all of these use cases. This would require embedding intelligence and information processing in the drones themselves, especially when authorities are starting to deploy multiple drones in parallel for the same applications. Deploying multiple drones requires coordination and collaboration among their operators

binding valuable resources from the actual mission. Therefore, it is essential to establish a collective intelligence that enables autonomous coordination and collaboration among the drones instead of passively receiving the tasks individually from a centralized control system.

Another key application area for UAVs is real-time surveying and monitoring. The same methods and technologies presented in this paper can be applied in different tasks such as environmental monitoring, land surveying or scientific study of different parameters in remote locations. The main benefit of utilizing reconfigurable drone swarms is the ability to perform complex data analysis at the Swarm level through distributed Edge computing. This allows for long-term autonomous operation when energy constraints allow, as well as reconnaissance in remote areas with poor network connectivity.

VI. CONCLUSION

We have proposed an architectural definition for reconfigurability in heterogeneous robotic swarms. This architecture is based on a combination of concepts and techniques from the robotics domain, multi-agent systems domain and edge-cloud computing domain. This is, to the best of our knowledge, the first work that proposes the abstraction and management of both hardware (sensors, actuators, computation and communication) and software (distributed sensing, coordination and decision making) as edge resources with elastic techniques. In particular, we explain how we are designing a reconfigurable drone swarm and what are the different hardware and software that make reconfigurability and elasticity possible.

REFERENCES

[1] W. Shi, G. Pallis, and Z. Xu, "Edge computing [scanning the issue]," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1474–1481, Aug 2019.

[2] A. J. Ferrer, J. M. Marques, and J. Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 111:1–111:36, Jan. 2019. [Online]. Available: http://doi.acm.org/10.1145/3243929

[3] F. Jalali, O. J. Smith, T. Lynar, and F. Suits, "Cognitive iot gateways: Automatic task sharing and switching between cloud and edge/fog computing," in *Proceedings of the SIGCOMM Posters and Demos*, ser. SIGCOMM Posters and Demos '17. New York, NY, USA: ACM, 2017, pp. 121–123. [Online]. Available: http://doi.acm.org/10.1145/3123878.3132008

[4] D. Calvaresi, M. Marinoni, A. Sturm, M. Schumacher, and G. Buttazzo, "The challenge of real-time multi-agent systems for enabling iot and cps," in *Proceedings of the international conference on web intelligence*. ACM, 2017, pp. 356–364.

[5] J. Penders, L. Alboul, U. Witkowski, A. Naghsh, J. Saez-Pons, S. Herbrechtsmeier, and M. El-Habbal, "A robot swarm assisting a human fire-fighter," *Advanced Robotics*, vol. 25, no. 1-2, pp. 93–117, 2011.

[6] J. De Greeff, N. Smets, K. Hindriks, M. A. Neerincx, and I. Kruijff-Korbayová, "Incremental development of large-scale human-robot teamwork in disaster response environments," in *12th Annual ACM/IEEE International Conference on Human-Robot Interaction, HRI 2017*. ACM, IEEE Computer Society, 2017, pp. 101–102.

[7] E. Masehian, M. Jannati, and T. Hekmatfar, "Cooperative mapping of unknown environments by multiple heterogeneous mobile robots with limited sensing," *Robotics and Autonomous Systems*, vol. 87, pp. 188–218, 2017.

[8] A. Kattepur, H. Dohare, V. Mushunuri, H. K. Rath, and A. Simha, "Resource constrained offloading in fog computing," in *Proceedings of the 1st Workshop on Middleware for Edge Clouds & Cloudlets*. ACM, 2016, p. 1.

[9] Z. Hong, H. Huang, S. Guo, W. Chen, and Z. Zheng, "Qos-aware cooperative computation offloading for robot swarms in cloud robotics," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4027–4041, April 2019.

[10] L. Qingqing, J. Peña Queralta, T. N. Gia, Z. Zou, H. Tenhunen, and T. Westerlund, "Visual Odometry Offloading in Internet of Vehicles with Compression at the Edge of the Network," in *The 12th International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, 2019.

[11] J. P. na Queralta and T. Westerlund, "Blockchain-powered collaboration in heterogeneous swarms of robots," *arXiv preprint arXiv:1912.01711*, 2020, symposium on Blockchain for Robotic Systems, MIT Media Lab.

[12] X. Y. Deng, P. Trinder, and G. Michaelson, "Autonomous mobile programs," in *2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. IEEE, 2006, pp. 177–186.

[13] Z. Zhang and X. Zhang, "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation," in *2010 The 2nd International Conference on Industrial Mechatronics and Automation*, vol. 2. IEEE, 2010, pp. 240–243.

[14] S. Banerjee and J. P. Hecker, "A multi-agent system approach to load-balancing and resource allocation for distributed computing," in *First Complex Systems Digital Campus World E-Conference 2015*. Springer, 2017, pp. 41–54.

[15] M. Elango, S. Nachiappan, and M. K. Tiwari, "Balancing task allocation in multi-robot systems using k-means clustering and auction based mechanisms," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6486–6491, 2011.

[16] A. Gautam, S. A. Ram, V. S. Shekhawat, and S. Mohan, "Balanced partitioning of workspace for efficient multi-robot coordination," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 104–109.

[17] H.-L. Truong, "Asre -towards application-specific resource ensembles across edges and clouds," 2020, preprint, working paper. [Online]. Available: https://www.researchgate.net/publication/340272410

[18] J. Gregory, J. Fink, E. Stump, J. Twigg, J. Rogers, D. Baran, N. Fung, and S. Young, "Application of multi-robot systems to disaster-relief scenarios with limited communication," in *Field and Service Robotics*. Springer, 2016, pp. 639–653.

[19] M. Hu, W. Liu, K. Peng, X. Ma, W. Cheng, J. Liu, and B. Li, "Joint routing and scheduling for vehicle-assisted multidrone surveillance," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1781–1790, 2018.

[20] N. Mendes, M. Safeea, and P. Neto, "Flexible programming and orchestration of collaborative robotic manufacturing systems," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. IEEE, 2018, pp. 913–918.

[21] V. Lomonaco, A. Trotta, M. Ziosi, J. d. D. Y. Ávila, and N. Díaz-Rodríguez, "Intelligent drone swarm for search and rescue operations at sea," *arXiv preprint arXiv:1811.05291*, 2018.

[22] H.-L. Truong, L. Berardinelli, I. Pavkovic, and G. Copil, "Modeling and provisioning iot cloud systems for testing uncertainties," in *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, 2017, pp. 96–105.

[23] V. Medel, O. Rana, J. A. Bañares, and U. Arronategui, "Modelling performance amp; resource management in kubernetes," in *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*, Dec 2016, pp. 257–262.

[24] R.-A. Cherrueau, A. Lebre, D. Pertin, F. Wuhib, and J. M. Soares, "Edge computing resource management system: a critical building block! initiating the debate via openstack," in *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*. Boston, MA: USENIX Association, 2018. [Online]. Available: https://www.usenix.org/conference/hotedge18/presentation/cherrueau

[25] S. Dustdar, Y. Guo, B. Satzger, and H. L. Truong, "Principles of elastic processes," *IEEE Internet Computing*, vol. 15, no. 5, pp. 66–71, 2011. [Online]. Available: https://doi.org/10.1109/MIC.2011.121

[26] J. Peña Queralta, C. McCord, T. N. Gia, H. Tenhunen, and T. Westerlund, "Communication-free and index-free distributed formation control algorithm for multi-robot systems," *Procedia Computer Science*, 2019, the 10th ANT Conference.

[27] C. McCord, J. Peña Queralta, T. N. Gia, and T. Westerlund, "Distributed Progressive Formation Control for Multi-Agent Systems: 2D and 3D deployment of UAVs in ROS/Gazebo with RotorS," in *2019 European Conference on Mobile Robots (ECMR)*.   IEEE, 2019.

[28] J. Peña Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "Collaborative mapping with ioe-based heterogeneous vehicles for enhanced situational awareness," in *IEEE Sensors Applications Symposium (SAS)*. IEEE, 2019.

[29] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[30] M. T. Nguyen, H. M. La, and K. A. Teague, "Collaborative and compressed mobile sensing for data collection in distributed robotic networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 1729–1740, 2017.

[31] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, 2017.

[32] E. Kaufmann, M. Gehrig, P. Foehn, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Beauty and the beast: Optimal methods meet learning for drone racing," in *2019 International Conference on Robotics and Automation (ICRA)*.   IEEE, 2019, pp. 690–696.

[33] L. Qingqing, F. Yuhong, J. Peña Queralta, T. N. Gia, Z. Zou, H. Tenhunen, and T. Westerlund, "Edge Computing for Mobile Robots: Multi-Robot Feature-Based Lidar Odometry with FPGAs," in *The 12th International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, 2019.

[34] J. Peña Queralta, F. Yuhong, L. Salomaa, L. Qingqing, T. Gia, Z. Zou, H. Tenhunen, and T. Westerlund, "Fpga-based architecture for a low-cost 3d lidar design and implementation from multiple rotating 2d lidars with ros," in *2019 IEEE SENSORS*.   IEEE, 2019, pp. 1–4.

[35] W. Shule, C. Martínez Almansa, J. Peña Queralta, Z. Zou, and T. Westerlund, "Uwb-based localization for multi-uav systems and collaborative heterogeneous multi-robot systems: a survey," *arXiv preprint arXiv:2004.08174*, 2020.

[36] J. Peña Queralta, C. M. Almansa, F. Schiano, D. Floreano, and T. Westerlund, "Uwb-based system for uav localization in gnss-denied environ-

[45] M. W. Miller *et al.*, "The use of small unmanned aircraft systems (suas) in law enforcement," 2016.

ments: Characterization and dataset," *arXiv preprint arXiv:2003.04380*, 2020.

[37] C. Martínez Almansa, W. Shule, J. Peña Queralta, and T. Westerlund, "Autocalibration of a mobile uwb localization system for ad-hoc multi-robot deployments in gnss-denied environments," *arXiv preprint arXiv:2004.06762*, 2020.

[38] V. Strobel, E. Castelló Ferrer, and M. Dorigo, "Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*.   International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 541–549.

[39] E. C. Ferrer, "The blockchain: a new framework for robotic swarm systems," in *Proceedings of the Future Technologies Conference*.   Springer, 2018, pp. 1037–1058.

[40] J. Peña Queralta and T. W. Li Qingqing, Zhuo Zou, "Enhancing autonomy with blockchain and multi-acess edge computing in distributed robotic systems," in *The Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*.   IEEE, 2020.

[41] J. Peña Queralta, L. Qingqing, T. N. Gia, H. Tenhunen, and T. Westerlund, "Distributed progressive formation control with one-way communication for multi-agent systems," in *2019 IEEE Symposium Series on Computational Intelligence*, 2019.

[42] H. L. Truong and S. Dustdar, "Principles of software-defined elastic systems for big data analytics," in *Proceedings of the 2014 IEEE International Conference on Cloud Engineering*, ser. IC2E '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 562–567. [Online]. Available: https://doi.org/10.1109/IC2E.2014.67

[43] S. Mariani, H.-L. Truong, G. Copil, A. Omicini, and S. Dustdar, "Coordination-aware elasticity," in *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, ser. UCC '14.   Washington, DC, USA: IEEE Computer Society, 2014, pp. 465–472. [Online]. Available: http://dx.doi.org/10.1109/UCC.2014.59

[44] W. F. Price *et al.*, "Public safety drones," 2019.

[46] K. G. Panda, S. Das, D. Sen, and W. Arif, "Design and deployment of uav-aided post-disaster emergency network," *IEEE Access*, vol. 7, pp. 102 985–102 999, 2019.