This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3293034

Dynamic Topology Organization and Maintenance Algorithms for Autonomous UAV Swarms

Anna Gaydamaka, Andrey Samuylov, Dmitri Moltchanov, Mateen Ashraf, Bo Tan, Yevgeni Koucheryavy

Abstract—The swarms of unmanned aerial vehicles (UAV) are nowadays finding numerous applications in different fields. While performing their missions, UAVs have to rely on external positioning information to maintain connectivity and communications between units in a swarm. However, some of the critical applications such as rescue missions are performed in locations, where this information is partially or fully not available, e.g., deep woods, mountains, indoors. In this paper, we propose a method for dynamic topology organization and maintenance in UAV swarms. In addition to the baseline functionality, we also design advanced features required for dynamic swarms merging and disjoining, making it suitable for practical applications. Specifically, the proposal is based on the virtual coordinates system allowing for the utilization of conventional geographical routing algorithms. We test the proposed algorithm in different swarm conditions to illustrate that: (i) it is insensitive to distance estimates up to at least 30% allowing for simple estimation techniques, (ii) the accuracy of the topology inference is at least 90% even under impairments caused by mobility and temporal loss of connectivity, and (iii) the impact of the developed merging algorithm for swarms lasts for multiple tens of time steps that correspond to just few seconds in practice. The set of developed algorithms can be utilized to ensure always connected topology in conditions where positioning information is partially or fully unavailable.

Index Terms—5G, UAV swarms, autonomous operation, GNSS-denied, topology construction and maintenance, geographical routing.

1 Introduction

Unmanned aerial vehicles (UAV) are nowadays considered as an effective platform to implement various services without direct human involvement. UAVs facilitate and sometimes even replace human labor in many areas, from agriculture and forestry [1], [2] to search and rescue and healthcare [3], [4]. Nowadays, these platforms have already taken over some of the tasks in telecommunications, both technical, e.g., monitoring equipment on top of towers, and more intellectual such as using UAV as a flying base station (BS), relay or anchor [5], [6]. Therefore, both ITU and 3GPP consider UAVs as future network users and aim at developing standards for their direct support in fifth generation (5G) and beyond-5G networks [7], [8].

Once the potential of individual UAVs has been discovered, the idea of using joint systems, UAV swarms, started to appear. UAV swarms have been utilized in a diverse range of applications, including precision agriculture, search, and rescue in disaster response, inspection and surveillance of infrastructure, and logistics and delivery in the urban area. Thanks to an ability to perform tasks collectively, such systems have striking advantages, e.g., to search and monitor a wider area, possibilities to carry heavier goods, perform tasks in a parallel mode and thus complete them faster [9], and allow to solve more complex intellectual tasks that are beyond the power of a single UAV. Still, a successful and safe launch of UAV swarms poses several challenges and technical issues, i.e., in-flight swarm localization and control, coordination of UAVs within the swarm, level of UAV autonomy for navigation, choreography of swarm actions, etc.

To enable the deployment of UAV swarms for versatile applications, we have identified several technical gaps that require addressing, such as developing a low-complexity and resilient network within the swarm, implementing a precise sensing and perception system for each individual node, designing a robust and agile formation control algorithm, and ensuring the scalability of the swarm. This paper emphasizes the merging and disjoining functions that are crucial for the formation control and scalability of UAV swarms, especially in autonomous missions where external infrastructures may be unavailable. The term ”merging” implies the process of combining two or more sub-swarm groups (clusters) of UAVs into a single coordinated swarm. ”Disjoining” alludes to the process of breaking apart the swarm or separating individual drones from the group.

While performing autonomous missions, UAV swarms typically not only gather information but collectively process it to make decisions in real-time [10], [11], [12]. These tasks require efficient routing of data flows between members of the swarm. Development of efficient routing of data flows is a challenging problem, especially for large swarms, where the efficiency of conventional routing protocols is affected by multiple factors such as: (i) complex three-dimensional (3D) topology of UAV swarms, (ii) constantly changing swarm topology due to relative mobility of UAVs within the swarm when avoiding collisions or performing reformations, (iii) temporal loss of connectivity due to changing propagation conditions.

The authors are with the Department of Electrical Engineering and Communications, Tampere University, Finland. Email: firstname.lastname@tuni.fi.
The work in this paper has been funded by the Academy of Finland within ACCESS (Autonomous Communication Converged with Efficient Sensing for UAV Swarms) project.

One of the potential ways to overcome the above-mentioned issues is to utilize topology organization and maintenance algorithms that would dynamically keep the swarm topology up-to-date. In presence of external positioning information, such as GPS-based, one can utilize conventional geographic routing algorithms that are known to be very effective in terms of overheads and routing performance, e.g. Greedy Perimeter Stateless Routing (GPSR) [13] or Compass [14]. However, when this information is not available, for example, when performing search and rescue operations in places inaccessible to Global Positioning System (GPS) or cellular infrastructure (deep woods, mountains, indoors), one needs to resort to conventional routing solutions based on either periodic exchange of routing tables or flooding-based on-demand routing. The former protocols are long known to produce excessive communications overheads [15], [16] while the latter introduces significant delay prior to the actual exchange of data [17]. As a result, both approaches are not suitable for mission-critical UAV swarm applications.

In this paper, we specifically target environmental and functional conditions for UAV swarm applications, where no external positioning information is available. For such types of use cases, we develop a completely distributed topology organization and maintenance algorithm that utilizes only two pieces of information: (i) possibly erroneous distance estimates to its one-hop neighbors that can be deduced by utilizing the propagation model and (ii) availability information about one-hop and two-hop neighbors that can be exchanged regularly in beacon packets. Owing to the dynamic nature of tasks and environments, UAV swarms necessitate adaptive merging or disjoining (splitting) capabilities. Typically, the merging function is employed to enhance capabilities, such as expanding coverage areas, amassing greater amounts of data, or transporting heavier loads over longer distances. Conversely, the disjoining function is utilized for searching and navigating specific targeted areas, routes, and hard-to-reach zones that are inaccessible to large-scale swarms. This paper devise both basic functionality related to maintaining a consistent view of the network at all the UAVs and advanced algorithms related to swarms merging and disjoining. To compare the performance of the proposed algorithms, we then develop topological and routing metrics and proceed assessing performance of the proposed technique. The proposed algorithm explicitly take into account environmental specifics such as mobility of UAVs and inaccuracy of distance estimates as well as use case specific impairments such as temporal loss of connectivity between UAVs in a swarm due to loss of spatial and temporal synchronization.

The main contributions of the study are:

- The virtual coordination system (VCS) is introduced for 3D topology construction, maintenance and message routing for autonomous UAV swarms. The VCS integrates geographical and links quality dynamics to enhance the resilience of swarming operations, particularly during the merging.
- A set of fully distributed, real-time and VCS-based topology construction and maintenance algorithms are proposed to support the essential swarming functions, e.g. merging and disjoining. In addition, the proposed algorithms improve the swarm autonomy level by using only the lightweight local (within the swarm) information and casting off the dependence on external location and connectivity infrastructure.
- The topological and routing metrics are introduced to gauge the performance of the proposed algorithms when the UAV swarm is used for collective tasks.
- Numerical results show that the proposed algorithm is robust to the positioning error, mobility variance and fluctuating UAV behaviour while achieving more than 90% match within the original physical topology. The simulation results also illustrate that the performance of VCS-based algorithms is resilient during the merging of swarms.

The rest of the paper is organized as follows. First, in Section 2, the review of related work is presented. The system model is introduced in Section 3. The proposed approach is formalized in Section 4. Section 5 introduces the metrics utilized for further performance assessment of the proposed approach. Numerical results are provided in Section 6. Finally, conclusions are drawn in Section 7.

2 Related Work

This section presents the related work. We start with modern approaches to topology control in UAV swarms, followed by the outlook of VCS-based approaches proposed so far. Concluding the section, a brief classification of routing protocols that can be utilized in UAV swarms is provided.

The approach considered in our paper is designed for efficient routing in swarms of UAVs, where the external information is unavailable. It allows to infer the topology of the swarm (i.e., graph) based on limited information that can be exchanged between UAVs in "Hello" packets. Since the developed approach is only needed when the external positioning information is not available, Section 2.1 describes existing approaches for inferring the coordinates of nodes in a swarm. A special approach that received considerable attention in the past in the context of mesh networks and can be applied to the case of UAVs swarms is VCS that is discussed in Section 2.2. Finally, since the proposed approach is in fact developed for routing in UAV swarms, a brief account of existing routing protocols is provided in Section 2.3.

2.1 Topology Control in UAV Swarms

One of the cornerstones in successful UAV swarming is maintaining the connectivity of the swarm. The straightforward approach for supporting connectivity and coordination in drone swarms is to rely upon some global navigation satellite system (GNSS) for UAV positioning, such as Global Positioning System (GPS). By following this approach, the authors in [18] consider the application of UAV swarm in disaster management, where the positions of individual UAVs in a swarm are detected via GPS. The authors in [19] introduce a system for monitoring and controlling the safety of structures based on computer vision. Using video images photographed by UAV, the system determines the cracking status of internal and external structures. The connectivity within the swarm is also controlled by GPS.

The main advantages of this approach are the reduction of maintenance costs and inspection time of the building. The study in [20] targets a problem of packet forwarding in aerial networks, more precisely a way to keep connectivity even in a moving swarm. The suggested algorithm routes a packet using the shortest path to the destination, if it exists. The information about the location of UAVs is provided by GPS. The authors tested the algorithm in a real UAV swarm and concluded that it decreases the delay and number of hops to the destination while keeping the delivery ratio high.

When performing their missions, UAVs may suddenly experience the loss of GPS signals, which can cause the absence of position information. To continue a safe flight without collisions in the so-called GPS-denied environments [21], the authors in [22] proposed to use a method of dead reckoning. The idea of dead reckoning is to calculate new coordinates using known initial coordinates and motion parameters estimated by utilizing the interval inertial positioning system based on a gyroscope and accelerometer. After conducting a number of experiments on simulated data of swarm trajectories and comparing the predicted position with the real, zero collisions were observed.

One more option when GNSS systems are not available is to rely upon a cellular infrastructure. In [23], the authors sum up the current state-of-the-art on the use of cellular networks as the communication infrastructure for UAV swarms and propose their own high-level architecture for swarm autonomy. They provide an UAV-to-UAV (U2U) network communication testbed, where a swarm of UAVs was able to follow the master UAV on a predefined path. A survey [24] focuses on synergies of 5G/B5G innovations for cellularly connected UAVs. The paper covers a wide range of topics from UAV swarm applications to network architectures and channel modeling.

One of the challenging use cases for autonomous swarm missions is operations in locations when GNSS and cellular infrastructure is not available or damaged, e.g., disaster situations, indoors, in deep woods and mountains, etc. In this case, UAVs need to rely upon swarm-internal methods for topology organization and maintenance. One of the approaches is to utilize internal positioning systems and/or radars [25]. In [26], the authors introduced a relative localization sensor system based on gyroscopes and accelerometers. In spite of positioning errors accumulating over time, it is demonstrated that the proposed approach allows to maintain swarm topology over long periods of time when GPS and/or cellular infrastructure is not available. The study [27] aims at reducing the number of isolated drones in a highly dynamic topology network. The approach is inspired by biological natural swarms, like swarms of dragonflies, and reinforces it with machine learning. The authors considered a case when UAVs are equipped with GPS, radar mechanisms, and altitude sensors to support connectivity. In general, the use of radars allows for precise topology maintenance but requires additional equipment to be available at each drone in the swarm which limits the application of this approach [28], [29].

The paper [30] proposes an integrated vehicular system using UAVs and UGVs for autonomous exploration, mapping, and navigation. It implements a two-layered exploration strategy, with a coarse exploration layer using UGVs and a fine mapping layer using UAVs. The authors in [31] focus on path planning algorithms in the absence of GPS signals, leveraging range information from stationary objects called Landmarks (LMs). The optimization problem for LM placement and vehicle routing is posed as an integer program, and two fast heuristics are presented to find feasible solutions. The paper [32] tackles the issue of persistent excitation-based relative localization for multi-UAVs in GPS-denied environments. It introduces synchronized sensor sample prediction and redesigns RL estimation to avoid error accumulation and achieve greater precision. Together, these works demonstrate the potential of unmanned systems to operate autonomously and collaboratively, using innovative approaches to overcome challenges in navigation and perception.

The authors in [33] develop a self-organizing flying network using a hybrid multitask algorithm for UAV swarms in communication relay networks and surveillance missions. To address three principal challenges of UAV topology maintenance and communications, the authors propose a hybrid solution based on market auction paradigms and a biologically inspired pheromone map. Even though the proposed solution depends on the correct parametrization, the numerical results demonstrate that it effectively serves the outlined performance goals.

2.2 Virtual Coordinate Systems

An alternative to the radars and inertial positioning systems in absence of GNSS and cellular infrastructure is to utilize virtual topologies that closely resemble physical ones. The research on such systems dates back to the seminal paper by Rao et. al [34], where the authors proposed to utilize virtual coordinate systems (VCS) for topology construction and further routing in wireless sensor networks (WSN). Motivated by the use of geographical routing on top of VCSs, the authors in [35] considered a static 2D network with several sink nodes acting as static anchors and devised a VCS utilizing the number of hops to the sinks as the main input parameter. Their numerical results show that the mean path length of greedy geographical routing algorithms running on top of the developed VCS exceeds the shortest path by just a few percent.

The study [36] presents Greedy Embedding Spring Coordinates (GSpring), a geographical routing algorithm that proposes the solution for the issue of packets stuck at dead ends. In their proposed approach, nodes detect situations that lead to dead ends during greedy forwarding and adjust their coordinates so as to increase the degree of connectivity of existing voids in the routing topology. Though GSpring improves the routing efficiency when compared to existing geographic routing algorithms, e.g., GPSR [13] or Compass [14], it operates only in networks with static nodes. Further, the study in [37] considers a network, where external positioning information is not available at the nodes. Using local connectivity information, they build a logical topology, on top of which the lightweight geographic routing protocol is applied. The authors test the proposed framework by using the packet delivery ratio and node power consumption.

Along these lines, a number of approaches for VCS construction have been proposed in the past. In general,

they can be classified to anchor-driven [36], [35], [38], [39], [37] and anchorless ones [40], [40], [41]. The former group assumes the presence of static nodes in the system which serve as universal virtual references. Anchorless systems rely upon either modern algorithms such as distributed hash tables (DHT) [40] or produce simple topologies such as chain formations [42], or circular ones [41]. Most of these systems are tailored to static deployment use cases such as those typical for WSNs. However, recently, the use of VCSs for topology construction in mobile mesh networks has been demonstrated in [43], where gradual algorithms have been proposed. This paper extends the work of [43] proposing an algorithm suitable for fully dynamic systems such as UAV swarms and develop enhancements such as merging and disjoining that are critical for the considered use case.

2.3 Routing in UAV Swarms

Over time, a considerable body of literature has addressed research on routing in communication networks [44]. These protocols can be classified into three categories: (i) proactive or table-driven, (ii) reactive or on-demand, and (iii) hybrid protocols that combine features of proactive and reactive mechanisms. The main advantage of proactive protocols is low route acquisition latency as the routes are always kept up-to-date. However, due to the need for regular updates, table-driven protocols are not well suited for mobile networks such as UAV swarms since the routing table updates happen often and create significant overheads. Typical examples of proactive protocols are Destination-Sequenced Distance Vector (DSDV) [45] and Optimized Link State Routing Protocol (OLSR) [46].

As opposed to the proactive design, reactive protocols find the routes on-demand. The process consists of two steps: broadcasting the route request packets to the network and receiving return messages with a route response. To avoid duplication, each packet has a sequence number. In the second step, when the destination node receives the packet, it responds with a route reply message that takes the reverse path in the network. The downside of this approach is the latency of route acquisition that might be critical for UAV applications [47]. The AODV protocol, proposed in [48] and standardized in RFC 3651, is an example of an on-demand routing protocol.

Hybrid routing protocols combine the functions of proactive and reactive routing protocols. Typically, they are used to determine optimal network destination routes and to report changes in network topology data. One possible implementation of this combination is to keep the routing tables from table-driven routing protocol for nodes in close proximity and use on-demand routing protocol for nodes on the periphery [49].

A separate class of routing protocols relies upon the use of external information to make routing decisions. For example, geographic routing protocols [50] use GPS locations instead of network addresses. In this case, the source sends packets to the destination based on its geographic location. This solves the problem of storage and communications overheads associated with table-driven protocols and alleviates long route acquisition delay of on-demand protocols. Nevertheless, the availability of external information is a basic requirement in these protocols.

Nowadays machine learning techniques are being employed in various fields, and routing protocols are no exception. A new routing protocol called Q-FANET for Flying Ad-Hoc Networks (FANETs) utilizes an improved Q-learning algorithm to reduce network delay in high-mobility scenarios [51]. To suit the dynamic behavior of FANETs, the proposed Q-FANET combines two routing protocols (QMR and Q-Noise+) and uses reinforcement learning on top of it. The results of the performance evaluation show that Q-FANET outperforms the other reinforcement learning-based routing protocols in terms of lower delay, lower jitter, and a minor increase in packet delivery ratio. The proposed protocol is relevant for UAV swarm organization as it can enhance the reliability and performance of communication networks between drones and ground control stations, which is critical for the safe and effective operation of drone swarms.

In this study, it is presumed that the developed VCS will provide location-related information to all the network nodes in a fully distributed manner. More specifically, the developed approach is in fact a VCS that does not rely upon anchor nodes as compared to most of the algorithms reviewed in Section 2.2. Thus, the developed VCS provides the routing nodes with geo-location of the destination node, and then any geo-routing protocols can be utilized. Since the choice of the geo-routing protocol is not important for our paper, we have compared the performance of routing over the developed VCS by specifying a general routing protocol independent metric.

3 System Model

In this section, we formalize the system model. We start with the network graph inference problem with a constrained set of readability information. Then, we proceed with specifying models of impairments affecting UAV connectivity in the swarm including loss of temporal/spatial synchronization, mobility model, etc. The section is concluded by the definition of metrics of interest.

3.1 Deployment Model

Consider the deployment model illustrated in Fig. 1. The swarm is assumed to move in a certain direction with an average speed v_S m/s. Specifically, we consider an area of size $L_1 \times L_2 \times L_3$ cubic meters with N nodes uniformly distributed within. It is assumed that no external GNSS information is provided for the swarm and that no UAVs within the swarm have cellular connectivity. The typical use cases considered are disaster management, indoors, deep woods, and mountain operations, where this information is completely unavailable or temporarily blocked.

In UAV swarms, the inter-node links are inherently dynamic due to wireless propagation specifics that may lead to the temporal loss of connectivity due to, e.g., loss of synchronization between nodes. This situation is modeled by randomly chosen nodes appearing and disappearing in the swarm, i.e. switching between connected and disconnected states. These processes are assumed to be mutually independent at all the UAVs comprising the swarm.

Another impairment related to internal UAV mobility inside the swarm occurs while performing a mission task

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3293034

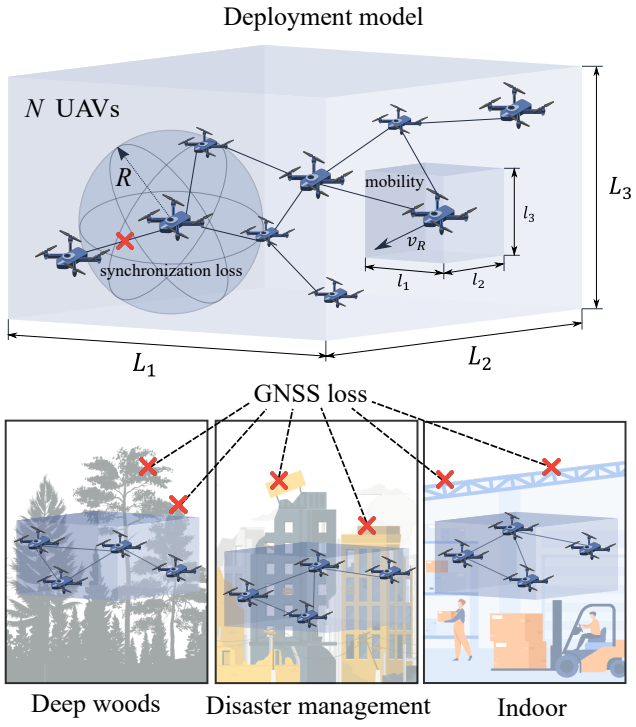


Fig. 1. UAV deployment model (red crosses in the bottom represent the loss of GNSS signal).

or going around obstacles. Often the movements of UAVs within the swarm are limited to small changes in the UAV locations for a subset of units. To capture this effect, it is assumed that at any given instant of time, no more than g percentage of UAVs are involved in relative movements with respect to the swarm movement. The relative speed of these units with respect to $(1-g)\%$ of UAVs is denoted by v_R. These UAVs move within the cubic compartments with sizes $l_1 \times l_2 \times l_3$ cubic meters centered around UAV's randomly chosen location according to the random direction mobility model (RDM, [52]). According to this model, UAV selects a direction of movement randomly and uniformly within $(0, 2\pi)$ and then moves toward the selected direction at constant speed v_R for exponentially distributed time with mean β^{-1}. The process is then repeated. We assume a peculiar reflection of the compartment boundaries.

The proposed algorithms in this paper are independent of the transmission technology utilized by the UAVs and hence can be used for any particular transmission technology. The transmission ranges r_i of radios equipped at UAVs $i = 1, 2, \ldots, N$, are known apriori or can be estimated by utilizing radio part parameters, e.g., carrier frequency, transmit and receive gains, receiver sensitivity, and noise floor. All the neighbors within the coverage radius r_i are assumed to hear and decode transmissions. If millimeter wave communications with highly directional antennas are utilized it is assumed that UAV may maintain connectivity with as many neighbors as needed.

3.2 Network Graph Inference Problem

It is assumed that each UAV possesses information about (i) distance estimates to the one-hop neighbors within its coverage area r_i and (ii) information about the two-hop neighbors. The former can be obtained by utilizing the information about at least the received signal strength and subsequently applying the propagation model for the considered carrier frequency, such as free-space path loss (FSPL). More comprehensive distances estimation algorithms can be utilized such as those based on the angle/zenith of arrival (AoA/ZoA) and the associated times of arrival if available. The use of radars for distance estimation is not precluded as well. To capture a wide range of potential distance estimate methods, we model the positioning error as αD, where D is the actual distance between neighbors and $\alpha \in (0, 1)$ is the error factor. Specifically, the information about the two-hop neighbors actually include IDs of UAVs that are directly connected to one-hop neighbors of a certain UAV and distances to them. This information is directly available at one-hop neighbors and is assumed to be delivered to the considered UAV in "Hello" packets. No other local or global information is assumed to be available at each UAV locally.

The information about the neighbors can be exchanged in "Hello" packets or in specifically designated "Beacon" packets exchanged on regular intervals, $i = 0, 1, \ldots,$ of duration T, i.e. $T = t_i - t_{i-1}, \forall i$. The latter variable defines the time step of algorithm execution at different nodes. These time steps are not necessarily synchronized between all the UAVs in a swarm.

To keep the application as wide as possible, we assume that directions towards neighbors is not explicitly utilized in the algorithm. However, observe that the information about two-hop neighbors allows to implicitly capture not only principle reachability but the directionality as well in a very simple form. In this way, a certain UAV running the algorithm sees branches of UAVs connected to different single-hop neighbors.

Having the above-mentioned information, the algorithm is expected to be executed independently at each node at each time step. Since the main usage is to maintain UAV swarm topology there are two types of tasks to be solved: (i) assignment of virtual coordinates within the swarm and (ii) enabling seamless real-time merging and disjoining of different swarms.

3.2.1 Assignment of Virtual Coordinates

Essentially the task is to construct distributed graph mapping physical space to a virtual one. To find virtual coordinates it is convenient to formulate the problem in terms of graph theory. We denote an oriented graph $G = (V, A)$ with a set of vertices $V = \{1, 2, ..., N\}$, a set of edges $A \subset \{(i, j) | i, j \in V, i \neq j\}$, and edge lengths $d_{i,j} > 0, (i, j) \in A$. That is, the goal is to find nodes' virtual coordinates $\mathbf{s}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, $i = 1, 2, \ldots, N$ at any time step t_i, such that for all edges $(i, j) \in A$ the distance between $\mathbf{s}_i$ and $\mathbf{s}_j$ should be close to $d_{i,j}$.

3.2.2 Merging and Disjoining

When performing autonomous missions, there might be the need to change UAV swarm formations by adding individual UAVs or swarms of UAVs to the original swarm. Alternatively, some of the UAVs might be lost during the mission. For this reason, for the VCS to be of practical interest, it has to allow for smooth swarm merging and disjoining. Note that there might be no need for special algorithms for

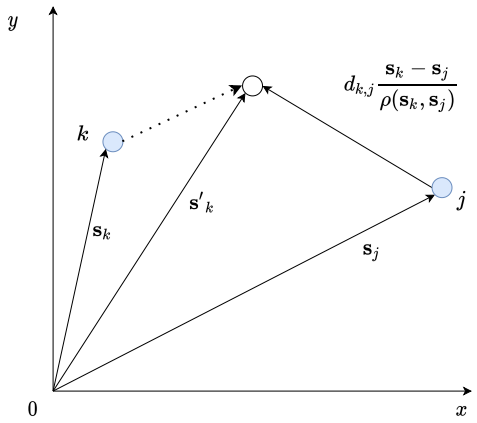


Fig. 2. Illustration of a solution to the VCS addressing the problem.

disjoining as nodes leaving the swarm functionality can be incorporated into the VCS maintenance procedure.

3.3 Metrics of Interest

As the main goal of this paper is to introduce VCS topology organization and maintenance algorithms for efficient geographical routing without the use of GNSS systems, topological and routing-related metrics of interest are considered. The topological metric is represented by the Pearson correlation coefficient between pairwise distances in virtual and physical topologies. Observe that since the aim is to enable efficient routing in dynamic UAV swarms, the developed algorithm may not produce the virtual topology exactly matching a physical one although it is preferable. Further, to benchmark the routing performance without resorting to details of specific routing protocols proposed to date, in Section 5, we propose a generic routing metric.

4 The Proposed Algorithms

In this section, we first develop our algorithm for VCS address allocations. Then, we proceed with specifying the algorithms for swarm merging.

4.1 VCS Address Allocation Problem

4.1.1 Problem Formulation

Consider a UAV swarm represented as a finite graph $G = (V, A)$ whose vertices may in general be dynamic, e.g., may disappear and move. The following notation is proposed. The vertices of the graph, $V = \{1, 2, ..., N\}$, are the UAVs. UAV i is able to receive a signal from other devices within its transmission range r_i, $i \in V$, which is defined by the UAV radio characteristics. The UAVs are connected by an edge if they are within the transmission range of each other. The set of edges is denoted by

$$A \subset \{(i,j)|i,j \in V, i \neq j\}, \tag{1}$$

and also define edge lengths as $d_{i,j} > 0, (i,j) \in A$.

Fig. 2 illustrates the intuitive solution to the VCS address allocation problem. Specifically, it highlights the process of adjusting the virtual coordinates of node k. Here, node j, as a one-hop neighbor of k, affects the choice of virtual coordinates of node k, denoted by $\mathbf{s}_k$. Ideally, node k should move to the point $\mathbf{s}'_k$. However, in a real scenario, node k has multiple one-hop neighbors (not shown in Fig. 2) which also affect the choice of virtual coordinates. Thus, making this process non-trivial.

Let $S_i = \{j \in V|(i,j) \in A\}$ be the set of vertices that are endpoints of the edges associated with vertex i, $i \in V$. We denote by $S_j^{-1} = \{i \in V|j \in A\}$ the set of vertices which are incident to the edges of vertex j. Further, let

$$r(\mathbf{s}_i, \mathbf{s}_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, \tag{2}$$

be the distance between $\mathbf{s}_i$ and $\mathbf{s}_j$, $\boldsymbol{\sigma} = (\mathbf{s}_1, \ldots, \mathbf{s}_N)$.

As an example, consider an arbitrary vertex $\mathbf{s}_k$

$$\mathbf{s}'_k = \mathbf{s}_j + d_{k,j}\frac{\mathbf{s}_k - \mathbf{s}_j}{r(\mathbf{s}_k, \mathbf{s}_j)} = \mathbf{s}_k + \\ + \left(1 - \frac{d_{k,j}}{r(\mathbf{s}_k, \mathbf{s}_j)}\right)(\mathbf{s}_j - \mathbf{s}_k), \tag{3}$$

at a distance $d_{k,j}$ from $\mathbf{s}_j$, see Fig. 2.

To make the distance between j and k close or equal to $d_{k,j}$ the node $\mathbf{s}_k$ should be moved to the new position $\mathbf{s}'_k$. If one averages the desired positions of the node $\mathbf{s}_k$ with respect to all surrounding nodes $j \in S_k$, then

$$\mathbf{s}'_k = \mathbf{s}_k + \frac{1}{|S_k|}\sum_{j \in S_k}\left(\frac{1 - d_{k,j}}{r(\mathbf{s}_k, \mathbf{s}_j)}\right)(\mathbf{s}_{n,j} - \mathbf{s}_{n,k}). \tag{4}$$

This leads to the following iterative procedure

$$\mathbf{s}_{n+1,k} = \mathbf{s}_{n,k} + \frac{\varepsilon_n}{|S_k|}\sum_{j \in S_k}\left(1 - \frac{d_{k,j}}{r(\mathbf{s}_{n,j}, \mathbf{s}_{n,k})}\right) \times \\ \times (\mathbf{s}_{n,j} - \mathbf{s}_{n,k}) = \mathbf{s}_{n,k} - \frac{\varepsilon_n}{|S_k|} \times \\ \times \sum_{j \in S_k}\left(1 - \frac{d_{k,j}}{r(\mathbf{s}_{n,j}, \mathbf{s}_{n,k})}\right)(\mathbf{s}_{n,j} - \mathbf{s}_{n,k}), \tag{5}$$

for any $k = 1, 2, \ldots, N$, $n = 1, 2, \ldots$, where $0 < \varepsilon_n < 1$ is the parameter determining how drastically the new value $\mathbf{s}_{n+1,k}$ of the vector $\mathbf{s}_k$ differs from its previous value $\mathbf{s}_{n,k}$.

4.1.2 Convergence Properties

The iterations in (5) are very similar to those utilized to minimize some function $F(\boldsymbol{\sigma})$ using iterations of the form

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n - \varepsilon_n \boldsymbol{\pi}_n, \; n = 1, 2, \ldots. \tag{6}$$

There are various methods for choosing the parameters ε_n and vectors $\boldsymbol{\pi}_n$. The classic method is the fast descent method [53], [54], [55], where the gradient $\boldsymbol{\pi}_n$ is taken as a vector $\boldsymbol{\pi}_n = \nabla F(\boldsymbol{\sigma}_n)$, ∇ denotes the gradient.

Consider the problem of minimizing a function

$$F(\boldsymbol{\sigma}) = \sum_{i=1}^{N}\sum_{j \in S_i} F_{i,j}(r(\mathbf{s}_i, \mathbf{s}_j)), \tag{7}$$

with differentiable terms $F_{i,j}(x)$.

The gradient $\nabla F(\boldsymbol{\sigma}_n)$ of this function is

$$\nabla F(\boldsymbol{\sigma}) = (\nabla_1 F(\boldsymbol{\sigma}), \nabla_2 F(\boldsymbol{\sigma}), \ldots, \nabla_N F(\boldsymbol{\sigma})), \tag{8}$$

where the individual gradient terms are given by

$$\nabla_k F(\boldsymbol{\sigma}) = \sum_{j \in S_k} \frac{f_{k,j}(r(\mathbf{s}_k, \mathbf{s}_j))}{r(\mathbf{s}_k, \mathbf{s}_j)} + \\ + \sum_{j \in S_k^{-1}} \frac{f_{k,j}(r(\mathbf{s}_k, \mathbf{s}_j))}{r(\mathbf{s}_k, \mathbf{s}_j)}, \quad (9)$$

and $f_{i,j}(x) = \frac{d}{dx} F_{i,j}(x)$.

Note an important special case of graphs $G = (V, A)$ with symmetric adjacency matrix A, for which the following conditions are satisfied

$$S_i^{-1} = S_i, i \in V, F_{i,j}(x) = F_{j,i}(x), j \in S_i, i \in V. \quad (10)$$

For such graphs (9) takes the form

$$\nabla_k F(\boldsymbol{\sigma}) = 2 \sum_{j \in E_k} \frac{f_{k,j}(r(\mathbf{s}_k, \mathbf{s}_j))}{r(\mathbf{s}_k, \mathbf{s}_j)} (\mathbf{s}_k - \mathbf{s}_j). \quad (11)$$

Let the graph $G = (V, A)$ have symmetric adjacency matrix A and $d_{i,j} = d_{j,i}$ for all $(i, j) \in A$. Consider two versions of the functions $F_{k,j}(x)$ and the corresponding gradients of the function $F(\boldsymbol{\sigma})$

$$F_{k,j}(x) = \frac{A_k}{4}(x - d_{k,j})^2, f_{k,j}(x) = \frac{A_k}{2}(x - d_{k,j}), \\ \nabla_k F(\boldsymbol{\sigma}) = A_k \sum_{j \in S_k} \left(\frac{1 - d_{k,j}}{r(\mathbf{s}_k, \mathbf{s}_j)} \right) (\mathbf{s}_k - \mathbf{s}_j), \quad (12)$$

and also

$$F_{k,j}(x) = \frac{A_k}{8}(x^2 - {d_{k,j}}^2)^2, f_{k,j}(x) = \frac{A_k}{2} x (x^2 - {d_{k,j}}^2), \\ \nabla_k F(\boldsymbol{\sigma}) = A_k \sum_{j \in S_k} (r^2(\mathbf{s}_k, \mathbf{s}_j) - {d_{k,j}}^2)(\mathbf{s}_k - \mathbf{s}_j). \quad (13)$$

It is easy to see that the considered intuitive solution (5) of the VCS address allocation problem is equivalent to the solution of the minimization problem of function in (7) with components in the form (12), where $A_k = |S_k|^{-1}$.

4.1.3 Single-Step Address Allocation

The iterative procedure in (5) can be implemented in different ways. One of the algorithms, called *single step allocation*, constructs a target optimization function that incorporates distance assessments to one-hop neighbors. The optimization problem is solved to obtain the most fitting virtual address for each UAV at each step independently of other UAVs.

Consider the following utility function

$$F_o(\mathbf{s}_{k,j}) = \sum_{j \in S_i} \Bigg[F_1(r(\mathbf{s}_i, \mathbf{s}_j), d_{i,j}) + \\ + \sum_{j \in M_i} F_2(r(\mathbf{s}_i, \mathbf{s}_j), d_{i,j}) \Bigg], \quad (14)$$

where S_i is the set of one-hop neighbors, and M_i is the set of two-hop neighbors.

The main idea of (14) is to find the virtual coordinates of UAV i by minimizing the penalty assigned to it when a set of rules is not obeyed. The first rule is that the distance between the UAV i and one-hop neighbors should be within a certain predefined range d_n. Specifically, UAV i should not be closer than a physical distance estimation and, at the same time, should not move away too far, as it is a directly connected neighbor. The second rule is that the distance between the UAV i and two-hop neighbors should be greater than or equal to a certain predefined limit d_m. Otherwise, a two-hop neighbor falls inside the area where UAV i is capable to receive a signal from UAV j, and UAV j becomes a one-hop neighbor by definition.

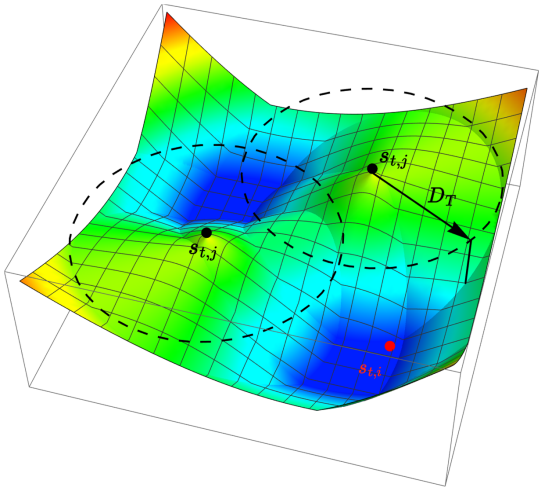


Fig. 3. Visual illustration of single step address allocation algorithm.

In order to implement the first rule we defince a function $F_1(r(\mathbf{s}_i, \mathbf{s}_j), d_{i,j})$, $j \in N_i$ which operates with the current UAV i and its one-hop neighbors which are denoted by N_i. It compares the virtual distance $r(\mathbf{s}_i, \mathbf{s}_j)$ between UAV i and its one-hop neighbor j and the estimation of physical distance between i and j and assigns a certain penalty based on its difference. In general, the function is defined as

$$F_1(d, d_n) = \begin{cases} ||d - D_L|| & d \le D_L(d_n), \\ 0 & D_L(d_n) < d \le D_H(d_n), \\ ||d - D_H|| & D_H(d_n) < d. \end{cases} \quad (15)$$

where $||\cdot||$ denotes Euclidean distance. The limit points of the penalty-free interval are denoted as D_L and D_H, where the former stimulates the virtual coordinates of the node i not to overlap with its one-hop neighbor's, and the latter indicates the transmission range of the current one-hop neighbor.

Similarly to (15), $F_2(r(\mathbf{s}_i, \mathbf{s}_j), d_{i,j})$ penalizes the current UAV i for being too close to its two-hop neighbors $j \in M_i$, as they are not in the coverage. It is given by

$$F_2(d, d_m) = \begin{cases} ||d - D_T|| & d \le D_T(d_m), \\ 0 & d > D_T(d_m). \end{cases} \quad (16)$$

An illustration of the penalty function (16) is shown in Fig. 3. The nodes with coordinates $\mathbf{s}_{t,j}$ are two-hop neighbors of the node i. Knowing that D_T is the transmission range of two-hop neighbors, the optimal coordinates of the node i should be outside of it. The three-dimensional nature of the illustration reflects the fact that three coordinates (x, y, z) are searched, while the color represents the value of the penalty function.

Algorithm 1: VCS Update Algorithm

Input:
1) signaling data packets $p_j \in P_i$ with data:
   - j - ID of the one-hop neighbor of UAV i
   - $\mathbf{s}_{n,j}$ - virtual coordinates of UAV j at iteration n
   - N_j - a list of one-hop neighbors' IDs of UAV j
   - a set of virtual coordinates of N_j
2) $d_{i,j}$ - an estimate of the physical distance between UAVs i and j

Result: virtual coordinates $\mathbf{s}_{n+1,i} = (x_{n+1,i}, y_{n+1,i}, z_{n+1,i})$ of UAV i at iteration n of the algorithm

1 During a specified time interval Δt UAV i may receive signaling data packets p_i;
2 Calculate $\mathbf{s}_{n+1,i}$;
3 **if** $P_i = \emptyset$ **then**
4 **if** $\mathbf{s}_{n,i} \neq NULL$ **then**
5 $\mathbf{s}_{n+1,i} = \mathbf{s}_{k,i}$
6 **else**
7 $\mathbf{s}_{k+1,i} = rand()$
8 **else**
9 **if** *low mobility* **then**
10 Single step Address Allocation
11 **else**
12 Gradual Address Convergence
13 Ḃroadcast p_i, $p_i = \{i, \mathbf{s}_{n,i}, N_i, \mathbf{s}_{n,j}, j \in N_i\}$;
14 Ṙepeat from point 1.

4.2 Proposed Algorithms

Recall that UAV calculates its virtual coordinates in three cases: (i) when UAV just joined the network and obtains its first virtual coordinates, (ii) when UAV needs to update its virtual coordinates at iteration n, (iii) on-demand when merging procedure is initiated. In this section, we introduce a VCS update **Algorithm 1** and describe how it can be utilized to determine and dynamically update the virtual coordinates of UAV i at step n of algorithm iteration. The merging procedure is described in Section 4.3.

To initiate the algorithm, UAV i within a certain period of time receives a signaling data packet from UAV that is in the set $V \cap S_i$ (line 1). Assume that UAV j is within the transmission range of UAV i. Then, UAV i will receive a signaling packet p_j with information about (i) UAV j's ID, (ii) virtual coordinates of UAV j, (iii) the list of one-hop neighbors of UAV j and their coordinates. In addition to the signaling data packets, UAV i estimates the physical distance to its one-hop neighbors, $d_{i,j}$, $j = 1, 2, \ldots, N_j$. The signaling data packets from the one-hop neighbors and the physical distance estimate are the two inputs to **Algorithm 1**. When the input data is collected, UAV i computes its virtual coordinates based on (15) and (16) (line 2). Once virtual coordinates are obtained, UAV i starts broadcasting signal packets with updated virtual coordinates (line 3).

UAV may not receive any signaling data packets. This means that UAV does not have any one-hop neighbors. This can either be because UAV is the first UAV to join the network or UAV is far from the connected component of the network. In the first case, UAV is assigned random virtual coordinates, while in the second case, the UAV keeps its previous virtual coordinates until the next update.

4.3 UAV Swarms Merging

We now proceed specifying the merging procedure by proposing an algorithm to recalculate virtual UAV coordinates in case a UAV belonging to more than one swarm appears. In practice, this procedure shall be initiated when UAV having a different swarm ID joins the current swarm formed by one or more UAVs. The proposed algorithm specifies two separate parts: (i) the operations of UAV belonging to several swarms ("merging" UAV), and (ii) the operations of all other UAVs.

Consider a network consisting of multiple, $c > 2$, UAV swarms represented by a disjoint graph, where UAV swarms are connectivity components of the graph. The set $C = 0, 1, \ldots, c$ denotes the set of all swarms. It is assumed that there are $N(i)$ UAVs in the swarm $i \in C$. We denote the virtual coordinates of the n-th UAV within i-th swarm by $(x_{i,n}, y_{i,n})$. Further, assume that a randomly chosen UAV in swarm i comes in the transmission range of a UAV that belongs to a different swarm. We denote the virtual coordinates of this UAV in the swarm i by $(x_{i,0}, y_{i,0})$. The polar coordinates of the vector $(x_{i,n} - x_{i,0}, y_{i,n} - y_{i,0})$ are denoted by $(r_{i,n}, \theta_{i,n})$. The task is to determine new virtual coordinates of the UAVs in the new joint swarm.

The idea of coordinate reassignment is to first convert the coordinates of all UAVs into a single coordinate system. Since for Cartesian coordinates such an operation is not straightforward, we propose to use a polar coordinate system, $(r_{i,n}, \theta_{i,n})$. It simplifies the algorithm by the fact that only the second coordinate, $\theta_{i,n}$, needs to be recalculated, while the first coordinate is determined by calculating the distance to the reference "merging" UAV. Finally, at the last step, the merged polar coordinates are recalculated into Cartesian coordinates and broadcasted in both swarms. To avoid ambiguity, the merging UAV is marked by 0.

Algorithm 2: Swarms merging: merging UAV 0

Input:
1) 0 - ID of the merging UAV
2) $(x_{i,0}, y_{i,0})$ - virtual coordinated of UAV 0 in the coordinate system of swarm i
3) N_0 - a list of one-hop neighbors' IDs of UAV 0 in swarm i

1 For all i, $i \in C$:
2 Send a message $\texttt{polar}(0, x_{i,0}, y_{i,0})$ to all j, $j \in N_0$.
3 After receiving a response message $\texttt{angles}(j, \varphi, \psi)$ from all j, $j \in N_0$, update parameters $\varphi_{i,0}$ and $\psi_{i,k}$:
4 $\varphi_{i,0} = \min_{n \in C_i} \theta_{i,n}$,
5 $\psi_{i,0} = \max_{n \in C_i} \theta_{i,n}$, $i = 1, 2, ...c$.
6 Calculate
7 $\theta_i = \theta_{i,min}$,
8 $\delta_i = \sum_{j=1}^{i-1} (\theta_{j,min} - \theta_{j,min})$.
9 Send a message $\texttt{cartesian}(0, \theta_i, \delta_i)$ to all j, $j \in N_0$;
10 After receiving a response message $\texttt{ready}(j)$ from all j, $j \in N_0$, take the coordinates (0,0).

The merging procedure is considered from the point of view of the merging UAV combining the swarms. The operations that need to be performed on the merging UAV are presented in Algorithm 2, while those that need to be performed at all the other UAVs – in Algorithms 3 and 4.

In Algorithm 2, the merging UAV initiates a Cartesian coordinate recalculation operation in each swarm i to which it belongs. The merging UAV sends a $\texttt{polar}(0, x_{i,0}, y_{i,0})$ message to all its neighbors in swarm i (line 2). The `polar` message contains the ID of the UAV from which the message is sent and the virtual coordinates of the merging UAV in swarm i. Once other UAVs receive `polar`, they calculate their polar coordinates, taking the merging UAV as the center of coordinates, see Algorithm 3. Then, the merging UAV waits for a response message $\texttt{angles}(j, \varphi, \psi)$ from all its neighbors (line 3). When the merging UAV has received `angles` from all its neighbors, it updates the φ (line 4) and ψ (line 5) parameters and calculates the polar angles of the UAVs from all the swarms (line 7,8). This operation is required further to recalculate the polar coordinates of all the UAVs in the same coordinate system. Next, the merging UAV converts the polar coordinates into Cartesian coordinates by initiating this process with a $\texttt{cartesian}(0, \theta_i, \delta_i)$ message (line 9), starting Algorithm 4. When the last UAV has performed the Cartesian translation, the merging UAV receives the message $\texttt{ready}(j)$ and assigns itself the coordinates $(0, 0)$ (line 10).

Algorithm 3 describes the actions for any UAV in the merging swarms that receives a `polar` message. UAV n, after receiving this message for the first time (line 1), determines its polar coordinates $(r_{i,n}, \theta_{i,n})$ and saves parameters $\varphi_{i,n}, \psi_{i,n}$ (line 3). These parameters will be further utilized to create a unified polar coordinate system. Finally, UAV n sends a `polar` message to all its neighbors except for the UAV from which it received a `polar` message (line 4), and then waits for a response message `angles`. If UAV n receives a `polar` message again, it immediately sends an `angles` message to the sending UAV. When UAV n receives $\texttt{angles}(m, \varphi, \psi)$ from its neighbor m, it updates the values

Algorithm 3: Polar coordinates: UAV n

Input:
1) $(x_{i,n}, y_{i,n})$ - virtual coordinated of UAV n in the coordinate system of swarm i
2) N_n - a list of one-hop neighbors' IDs of UAV n

1 UAV n received a message $\texttt{polar}(k, x_{i,0}, y_{i,0})$ from UAV k, $k \in N_n$.
2 If the message $\texttt{polar}(l, x_{i,0}, y_{i,0})$ was received earlier from some UAV l, send a message $\texttt{angles}(n, \varphi_{i,n}, \psi_{i,n})$ to UAV l.
3 Calculate the polar coordinates $(r_{i,n}, \theta_{i,n})$, save the parameters $\varphi_{i,n} = \theta_{i,n}$, $\psi_{i,n} = \theta_{i,n}$.
4 Send a message $\texttt{polar}(n, x_{i,0}, y_{i,0})$ to N_n, except k.
5 After receiving a response message $\texttt{angles}(n, \varphi, \psi)$, update the parameters $\varphi_{i,k} = \min(\varphi, \varphi_{i,k})$, $\psi_{i,k} = \max(\psi, \psi_{i,k})$.
6 After receiving a response message $\texttt{angles}(j, \varphi, \psi)$ from all j, $j \in N_n, j \neq k$, send $\texttt{angles}(n, \varphi, \psi)$ to UAV k.

Algorithm 4: Cartesian coordinates: UAV n

Input:
1) N_n - a list of one-hop neighbors' IDs of UAV n

1 UAV n received a message $\texttt{cartesian}(k, \theta_i, \Delta_i)$ from UAV k, $k \in N_n$
2 If the message $\texttt{cartesian}(l, \theta_i, \Delta_i)$ was received earlier from some UAV l, send a message $\texttt{ready}(n)$ to UAV l.
3 Send a message $\texttt{cartesian}(n, \theta_i, \Delta_i)$ to N_n, except k.
4 Calculate the Cartesian coordinates (x_n, y_n) using the polar coordinates $(r_{i,k}, \theta_{i,k} - \theta_i + \Delta_i)$.
5 Broadcast the Cartesian coordinates (x_n, y_n) to all j, $j \in N_n$.
6 After receiving a response message $\texttt{ready}(j)$ from all j, $j \in N_n, j \neq k$, send $\texttt{ready}(n)$ to UAV k.

φ, ψ (line 5). The update is designed to find the minimum and maximum polar angles. When the `angles` message is received from all the neighbors of UAV n except for the UAV from which it received `polar` message, UAV n sends the `angles` message to the UAV from which it received `polar` (line 6). Repeating recursively, the message `angles` reaches the merging UAV.

Once the described procedures in Algorithms 2 and 3 are performed, all UAVs in the merging swarms have a pair of coordinates – Cartesian coordinates in the original swarms and polar coordinates computed with the help of the merging UAV. The goal of Algorithm 4 is to assign Cartesian coordinates to all UAVs in a new joint swarm. The steps are similar to Algorithm 3, but now the process is initiated by the `cartesian` message (line 1). After converting the coordinates to the Cartesian system (line 4), UAV n informs all its neighbors and sends them the message `ready` (line 5). This message is needed to notify the merging UAV that the transition to Cartesian coordinates is now complete.

5 Topological and Routing Metrics

We now introduce metrics of interest reflecting the topological and routing performance of the proposed algorithm. First, the correlation metric is specified, and then we proceed with the routing one.

5.1 Topological Metric

As an indicator of the similarity between physical and virtual network topologies, the so-called topology similarity index is used. This index is defined as the Pearson correlation coefficient between pairwise distances in the virtual and physical topologies, i.e.,

$$K = \frac{cov(D_{phys}, D_{virt})}{\sigma_{D_{phys}} \sigma_{D_{virt}}}, \tag{17}$$

where D_{phys} and D_{virt} are the pairwise distance matrix for physical and virtual graphs respectively, cov is the covariance, and σ is the standard deviation.

5.2 Routing Metric

The main goal of the designed VCS system is to serve as a routing underlay for geographical routing. Observe that perfect matching of the physical and virtual topologies ensures that routing performance over them will be the same. However, when non-perfect matching is observed it is difficult to make definitive conclusions. To this aim, below we define a general routing metric that can be utilized for benchmarking the routing performance without resorting to specific details of various routing algorithms proposed so far.

Recall, that the network is represented by graph $G = (V, A)$. The virtual coordinates of a UAV i, $i \in V$, are denoted as $\mathbf{s}_i = (x_i, y_i, z_i)$. All the vertices that are associated with UAV i are contained in the set S_i. We introduce a set $\overline{S_i}$, such that $\overline{S_i} = S_i \cup \{i\}$. The goal of the routing algorithm is to determine the path from the UAV i to the UAV k in a swarm over virtual and physical topologies. Further, let $n_k(i)$ be UAV to which data is sent from UAV i over the path from i to k, $n_k(i) \in \overline{S_i}$. To determine the route to UAV $n_k(i)$, the following optimization problem should be solved

$$n_k(i) = \underset{j \in S_i}{\operatorname{argmin}} r(\mathbf{s}_j, \mathbf{s}_k),\ i \neq j. \tag{18}$$

Basically, the solution for (18) looks for a UAV from S_i that is closest in a geographical sense to the destination, i.e., UAV k. Each UAV sends a message with probability $1/N$ to one of the arbitrary N UAVs. We define a stochastic matrix $\mathbf{P} = [P(i,j)]$ and a unit vector $\mathbf{u} = (1, 1, \ldots, 1)$. Matrix $\mathbf{P}$ characterizes routing and its elements are

$$P(i,j) = \begin{cases} \frac{1}{N} \sum_{k=1}^{N} \delta_{n_k(i),j}, & i \neq j, \\ 1/N, & i = j, \end{cases} \tag{19}$$

where $j \in S_i$, $\delta_{n_k(i),j}$ takes a value of 1 if the routing from UAV i to UAV k includes a certain UAV and 0 otherwise.

By solving the matrix equation $\mathbf{pP} = \mathbf{p}$, $\mathbf{p}^T\mathbf{u} = 1$, the so-called routing vector $\mathbf{p}$ is derived. The i-th element of $\mathbf{p}$ represents the proportion of the time UAV i is involved in the routing. The final metric is defined as

$$\rho = \|\mathbf{p}_1 - \mathbf{p}_2\|, \tag{20}$$

where $\mathbf{p}_1$ and $\mathbf{p}_2$ are vectors for physical and virtual graphs.

If the virtual graph exactly matches the physical one, then the routing will be performed over the same UAVs. Accordingly, the frequency of visiting UAVs in both graphs will be the same. In this ideal case $\mathbf{p}_1 = \mathbf{p}_2$ and the metric $\|\mathbf{p}_1 - \mathbf{p}_2\| = 0$. When matching is imperfect, the constructed routes will be different and some components $\mathbf{p}_1$ and $\mathbf{p}_2$ will not coincide. It means that the routs in physical and virtual graphs are different, which influences the frequencies of visiting nodes. In the worst case, the routing in the graphs follows completely different routes and the metric takes value 2. It is caused by the low accuracy of virtual coordinates.

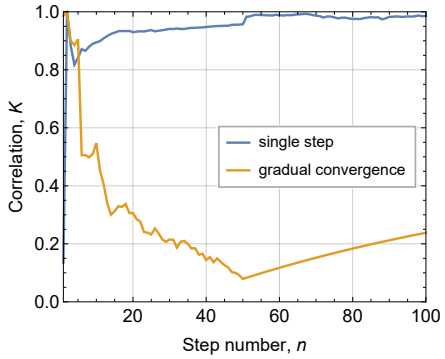
6 Numerical Results

In this section, we elaborate on our numerical results by assessing the performance of the proposed approach. We start with single UAV swarm performance by assessing the impact of different densities of UAVs, different types of topologies, relative mobility, and inaccuracy of distance estimates. As performance metrics of interest, we will utilize topological and routing metrics defined in Section 5. The results of the single-step algorithm are compared with those reported in [43], where the gradual convergence approach has been proposed.

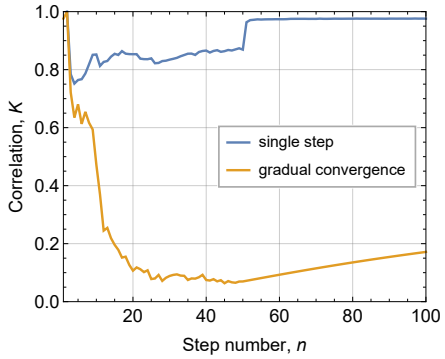
TABLE 1
Simulation parameters

Notation	Description	Value
N	Number of UAVs	50 units
R	Transmission range	30, 50 m
L_1,L_2	Sides of the area	100 m
L_3	Topology (flat or 3D)	5 m, 30 m
g	Percentage of moving UAVs	100 %
v_R	Relative UAV speed	0.1, 3 m/s
γ	Positioning error	1, 30 %
f	Synchronization impairment	On/Off

The system parameters are provided in Table 1. We note that such parameters as the number of UAVs and sides of the considered area can vary depending on the application scenario. Transmission range values are based on the IEEE 802.11n/ac standards. The standardization efforts are still ongoing with respect to the UAV traffic regulations. As for the percentage of moving UAVs, we consider the worst case, when all the UAVs move with respect to each other. We underline that by introducing the relative speed of UAVs, we consider scenarios where UAVs have to dynamically change their trajectories due to presence of obstacles and



(a) Positioning error 1%



(b) Positioning error 30%

Fig. 4. Topological metric as a function of positioning error.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3293034

thus move with respect to other UAVs in the swarm. For positioning error we consider the best case (1%) corresponding to the use of advanced localization techniques (e.g., based on time and angles of arrival) and worst case (30%) when the FSPL model is utilized for estimating the distance to the neighbors. Finally, we also consider that UAVs might lose connectivity with their neighbors due to imperfections of the communication technology. We take such scenarios into account by introducing a synchronization impairment parameter.

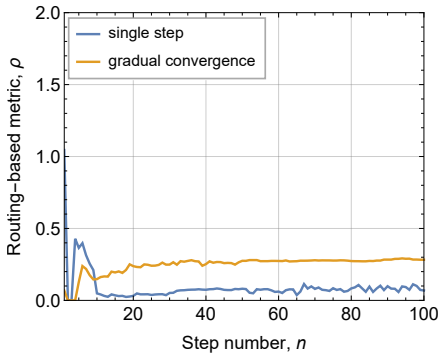
The considered scenario for performance assessment is as follows. In the initial phase, the UAVs are added one by one to the swarm every second corresponding to typical swarm initiation. Once all the considered number of UAVs are added, we introduce various impairments, such as mobility and loss of synchronization. First, the behavior of the single swarm system is observed for various realizations (trajectories) of UAV swarm in terms of topological and routing metrics. Here, the metrics of interest are topological and routing performance before and after merging.

The Cartesian coordinate system is used to simulate mobility. Once all UAVs are connected to the network, a certain percentage of UAVs, g, is selected to be mobile. Here, we consider the worst-case scenario, where all UAVs are mobile, i.e. $g = 100\%$. The movement of UAVs is characterized by two parameters: the range of mobility and the relative UAV speed. The range of mobility of one UAV is limited by a compartment with dimensions $l_1 \times l_2 \times l_3$. In addition to mobility, UAVs in the network may suddenly lose synchronization (connectivity) with their neighbors. This process is simulated as follows. At each algorithm iteration dictated by the frequency of message exchanges between UAVs, a randomly selected UAV is removed from the swarm. The remaining UAVs in the network redefine their virtual coordinates without the removed UAV. In the next time step, the removed UAV appears in the swarm again and the network adapts its virtual coordinates according to the new information.

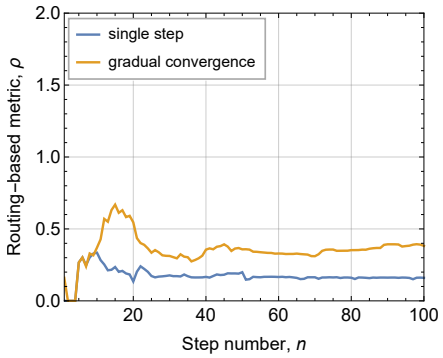
6.1 Single Swarm Performance

We start with characterizing the sensitivity of the proposed approach to the accuracy of the positioning information. To this aim, Fig. 4 illustrates the topological characteristic as a function of time, for $N = 50$ UAVs each having coverage of 50 m, where the height of the swarm is 5 m, there is no mobility and under perfect synchronization. Here, Fig. 4(a) shows the case of 1% positioning error that can be attained using advanced localization algorithms, while Fig. 4(b) corresponds to the 30%, i.e. the accuracy attained by the simple approaches such as those based on FSPL model. As one may observe, the proposed approach drastically outperforms the gradual convergence originally proposed in [43] by, e.g., 3-5 times. In fact, for the considered set of parameters, the proposed approach achieves a similarity of more than 90% between virtual and physical topologies. Note that this holds not only for static swarm behavior but also holds true during the UAV swarm formation phase, where UAVs are added one by one to the swarm.

Analyzing the data presented in Fig. 4 further, one may observe that the accuracy of the distance estimates between



(a) Positioning error 1%



(b) Positioning error 30%

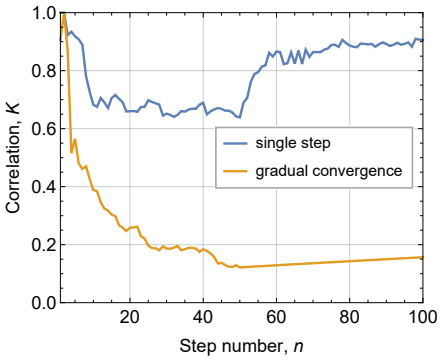
Fig. 5. Routing metric as a function of positioning error.

UAVs utilized for virtual topology construction does not affect the performance of the proposed algorithm. That is, even introducing a positioning error of 30% only slightly deteriorates the considered metrics during both swarm formation and operational phases. Importantly, once the swarm is formed, the metric jumps to approximately 1.0.

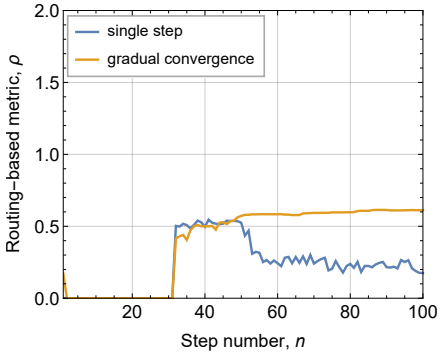
Recall that topology matching does not ensure identical routing. Fig. 5 illustrates the corresponding routing performance of the proposed approach for the same set of chosen parameters. Recall that the range of the metric is $(0, 2)$ where 0 implies perfect matching of the shortest path routing. As one may observe, for both 1% and 30% of positioning error, the considered metric is close to zero with a slight increase observed for 30% case. Importantly, it holds for both formation and operational phases implying that data can be routed even when the swarm formation is not finalized. Finally, we highlight that the proposed approach outperforms the gradual convergence significantly.

The swarm topology considered in Fig. 4 and 5 assumes a nearly flat swarm formation with a height of just 5 m. When performing real missions, the formation may dynamically change, e.g. when avoiding obstacles. To this aim, Fig. 6 shows topological and routing metrics for swarm height of 30 m as compared to the Fig. 4 and 5, where the height is 5 m. The rest of the parameters and swarm behavior are the same. By cross comparing the illustrations, one may observe that both metrics slightly degrade. The reason is that higher swarm height leads to fewer one- and two-hop neighbors utilized by the drones to compute their locations in the virtual topology. One may observe similar behavior when the communications range of UAV decreases (not shown

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3293034

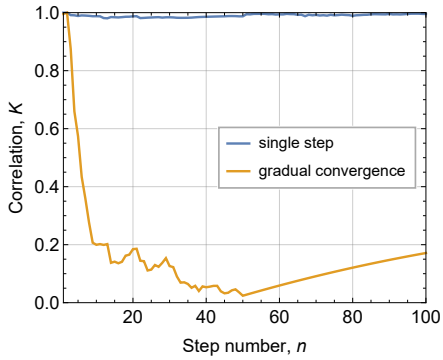


(a) Topological metric

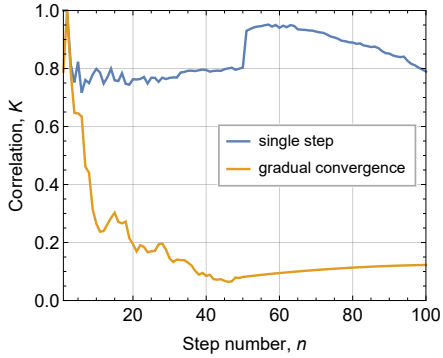


(b) Routing metric

Fig. 6. Topological and routing metrics for swarm height of 30 m.



(a) Mobility speed is 0.1 m/s



(b) Mobility speed is 3.0 m/s

Fig. 7. Topological metric as a function of UAVs' mobility speed.

here). Thus, we may conclude that the proposed approach is best suited for dense UAV swarms.

The first typical impairment we consider is the relative mobility of UAVs in the swarm shown in Fig. 7 for topological metric and two relative mobilities 0.1 m/s and 3 m/s. Here, $N = 50$ UAVs in the swarm, positioning error of 1%, swarm height of 5 m, and no temporal loss of synchronization between UAVs. There are two important observations. First by cross comparing the results in Fig. 7(a) and Fig. 4(a), one may deduce that slight relative mobility actually improves algorithms performance as an absolutely perfect topology match is observed in Fig. 7(a). This is due to the additional location relaxation factor for the optimization problem that is solved at each UAV. However, when the relative speed increases to 3 m/s the mobility becomes a deteriorating factor. Albeit, the topological metric still remains higher than 0.8.

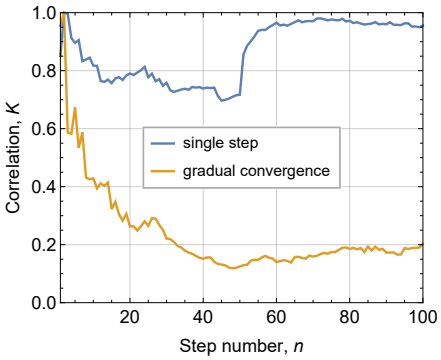
Note that one may observe two types of "jumps" in the plots. The first "larger" type of jumps are caused by swarm topology changes. In the initial phase, the UAVs are added one by one to the swarm. Since the proposed algorithm depends on the number of neighboring UAVs and their location, continuously changing the number and location of neighbors affects the accuracy of the algorithm. However, once the swarm is formed, the algorithm is able to calculate the coordinates more precisely, which causes a significant improvement (jump) in the plot. The second type of jump is less significant and appears due to certain orientations of the UAVs. It may appear for several reasons: specific geometric positions of UAVs in the swarm (for instance, when a certain subset of drones make a "line"), lack of a sufficient number of neighbors at any given instant of time, etc.

Yet another impairment for UAV swarms is temporal loss of synchronization between UAVs. Recall that we model it by temporarily putting UAVs "off" and "on" with a duration of 1 s. Fig. 8 illustrates the topological and routing metrics for $N = 50$ UAVs in the swarm, positioning error of 1%, swarm height of 5 m, relative mobility of 1 m/s, and temporal flashing on/off UAV behavior. Similarly to the case of low mobility, flashing helps to improve the correlation between the physical and the virtual graph, making them more similar. A rationale for this behavior is again additional location relaxation in the virtual domain induced by the partial absence of connected UAVs. One may observe that once the swarm is formed, the correlation value remains close to 1 for the rest of the experiments.

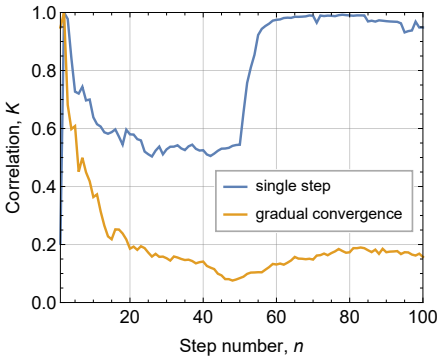
Another important impairment is the loss of packets utilized for exchange of the sets of neighbors between adjacent UAVs in a swarm. These losses may happen as a result of e.g., fast fading phenomenon. Fig. 9 demonstrates the algorithm behavior in the case when the packets are lost with the probability 0.05 and 0.1. Numerical results reveal that the proposed algorithm isn't affected significantly by packets' loss. With a loss probability of 0.05, the proposed algorithm achieves a similarity of more than 90% between virtual and physical topologies (Fig. 9(a)). The probability of 0.1 slows down the process, nevertheless, the final value of correlation is more than 90%. Fig. 9(b) shows behavior of the routing-based metric. With the range of (0, 2), where 0 implies perfect matching of the shortest path routing in physical and virtual graphs, the metric is close to zero.

Finally, Fig. 10 illustrates the impact of swarm density on

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3293034

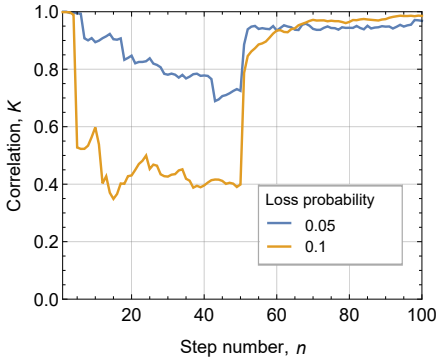


(a) Positioning error 1%

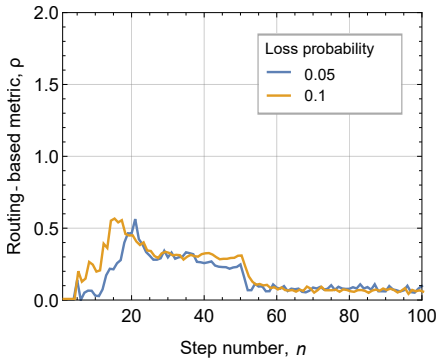


(b) Positioning error 10%

Fig. 8. Topological metric as a function of UAVs' mobility speed.



(a) Topological metric



(b) Routing-based metric

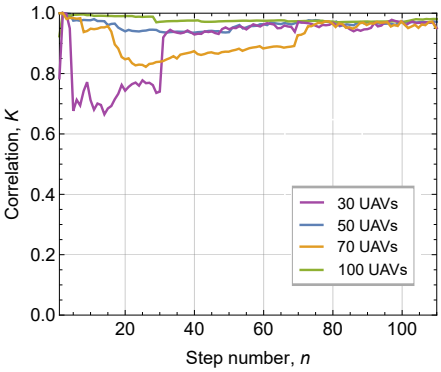
Fig. 9. Topological and routing metrics as a function of packet loss.

topological and routing-based metrics. As one may note, the suggested algorithm is more suitable for dense swarms. The correlation between topologies increases together with the number of UAVs. The same can be said about the routing-based metric – the metric shows the worst results for sparse swarms, while swarms with 70 and 100 UAVs provide better results.

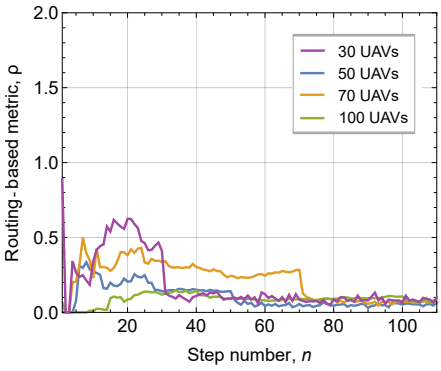
6.2 Merging Performance

We now proceed assessing the merging performance of the proposed algorithm. The assumed scenario is where two UAV swarms are gradually formed by adding one node at a time until the number of UAVs in each swarm reaches 25. Then, each swarm has 10 time steps to update their virtual coordinates, after which the merging process starts. Initially, each swarm, called cluster hereafter, has its own VCS, thereby the task is to translate the virtual coordinates of the two clusters into a unified system.

Fig. 11 presents the topological correlation metric before and after the merging of clusters for 1% and 10% positioning errors in Fig. 11(a) and Fig. 11(b), respectively. We consider the case of the merging of two clusters, each containing 25 nodes, swarm height of 5 m. Two cases are compared: positioning error of 1% and 10%. For the first 25 time steps, the nodes of the clusters exist separately, as reflected by the orange and blue curves. From 26 to 35 time steps the virtual coordinates are recalculated according to the proposed algorithm. Starting from 40th time step, one may observe the green curves which show the correlation metric for the joint cluster.



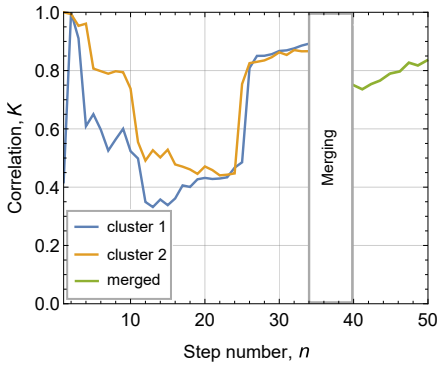
(a) Topological metric



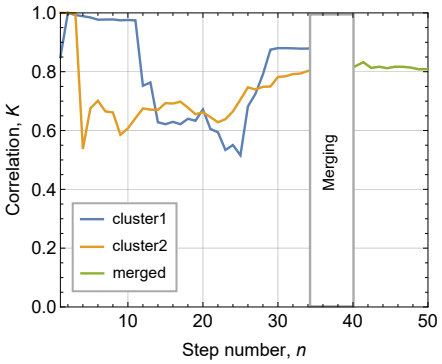
(b) Routing-based metric

Fig. 10. Topological and routing metrics as a function of UAVs' density.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3293034

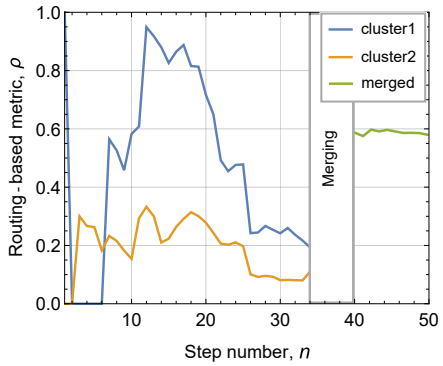


(a) Positioning error 1%

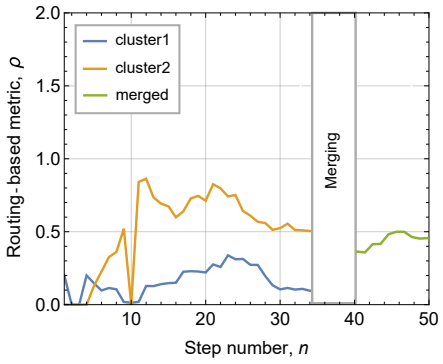


(b) Positioning error 10%

Fig. 11. Topological metric during the merging of two clusters.



(a) Positioning error 1%



(b) Positioning error 10%

Fig. 12. Routing metric during the merging of two clusters.

First of all, observe that there is no principal difference between 1% and 10% of positioning error which is in line with previous observations for single swarm performance. Further, the considered metric for initial clusters having 25 UAVs reaches approximately 0.5 and 0.6 for 1% and 10% of positioning error, respectively, and then, once all UAVs have joined the overlay, the correlation metric improves to 0.9. The merging process initially incurs imperfections into VCS address allocations that is reflected by the topological metric being smaller than 0.8 in both cases just after the merging is complete. However, immediately after it starts to increase reaching the value of 0.9 for 1% of positioning error already after 10 time steps. For 10% error, it remains almost intact. Thus, one may conclude that the merging process does not affect long-lasting performance degradation in the VCS operation.

Fig. 12 presents the routing metric before and after the merging process for the same set of chosen parameters. It is noticeable that this metric has greater values right after the merging but, nevertheless, they are well below 1. The rationale is that the topology changes and so do the routes, thus, further updates are required to improve the value.

7 Conclusions

To enable timely coordination and information exchange, the swarms of UAVs performing the missions in GNSS-denied environments such as deep woods, mountains, or indoors, need to be able to maintain their topology at all times. Motivated by this task, in this paper, we developed a set of algorithms for topology organization and maintenance without the use of external positioning information for UAV swarms. The developed algorithms are entirely distributed, utilize limited information about the proximity of UAVs that can be exchanged in link control messages such as "hello" beacons, and may tolerate inherent dynamics of UAV swarms in terms of loss of link synchronization and mobility. The developed algorithm is complemented with essential UAV swarms merging and disjoining functionality delivering the whole package for reliable UAV operation in hostile environments.

The performance of the proposed algorithm was investigated using both topological and routing performance metrics. Our results reveal that the algorithm is robust to the positioning error for directly connected UAV nodes and may tolerate up to 30% of error. Both mobility and synchronization loss events are handled with limited performance degradation. Finally, we showed that the merging functionality does not affect the algorithm performance and its impact lasts for 20-50 times steps which is equivalent to a few seconds or even less in real-time scenarios.

Several directions for future research are considered. Firstly, the algorithm's sensitivity to the loss of signal packets is to be investigated. The threshold of acceptable losses is of particular interest. Secondly, frequent disconnection and connection of drones results in significant overhead, which could potentially be avoided by applying some heuristics on top of the algorithm. Lastly, the algorithm should be tested in real-life settings to analyze its scalability and robustness in different environments with various numbers of UAVs and obstacles. The goal is to refine the algorithm's parameters and identify potential limitations.

Source Access

The source code used to produce results for this paper is available at https://github.com/gaydamanya/GAR.git.

REFERENCES

[1] A. López, J. M. Jurado, C. J. Ogayar, and F. R. Feito, "A framework for registering UAV-based imagery for crop-tracking in Precision Agriculture," *International Journal of Applied Earth Observation and Geoinformation*, vol. 97, p. 102274, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030324342030917X

[2] T. Hu, X. Sun, Y. Su, H. Guan, Q. Sun, M. Kelly, and Q. Guo, "Development and Performance Evaluation of a Very Low-Cost UAV-Lidar System for Forestry Applications," *Remote Sensing*, vol. 13, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/1/77

[3] C. Liu and T. Szirányi, "Real-time human detection and gesture recognition for on-board UAV rescue," *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/2180

[4] H. S. Munawar, H. Inam, F. Ullah, S. Qayyum, A. Z. Kouzani, and M. A. P. Mahmud, "Towards Smart Healthcare: UAV-Based Optimized Path Planning for Delivering COVID-19 Self-Testing Kits Using Cutting Edge Technologies," *Sustainability*, vol. 13, no. 18, 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/18/10426

[5] E. Montero, C. Rocha, H. Oliveira, E. Cerqueira, P. Mendes, A. Santos, and D. Rosário, "Proactive radio- and QoS-aware UAV as BS deployment to improve cellular operations," *Computer Networks*, vol. 200, p. 108486, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912862100431X

[6] Z. Xiao, H. Dong, L. Bai, D. O. Wu, and X.-G. Xia, "Unmanned Aerial Vehicle Base Station (UAV-BS) Deployment With Millimeter-Wave Beamforming," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1336–1349, 2020.

[7] S. K. Khan, U. Naseem, H. Siraj, I. Razzak, and M. Imran, "The role of unmanned aerial vehicles and mmWave in 5G: Recent advances and challenges," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, p. e4241, 2021.

[8] Q. Wu, J. Xu, Y. Zeng, D. W. K. Ng, N. Al-Dhahir, R. Schober, and A. L. Swindlehurst, "A comprehensive overview on 5G-and-beyond networks with UAVs: From communications to sensing and intelligence," *IEEE Journal on Selected Areas in Communications*, 2021.

[9] J. Wubben, F. Fabra, C. T. Calafate, J.-C. Cano, and P. Manzoni, "A novel resilient and reconfigurable swarm management scheme," *Computer Networks*, vol. 194, p. 108119, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912862100195X

[10] Y. Zhou, B. Rao, and W. Wang, "UAV swarm intelligence: Recent advances and future trends," *IEEE Access*, vol. 8, pp. 183 856–183 878, 2020.

[11] A. Sharma, S. Shoval, A. Sharma, and J. K. Pandey, "Path Planning for Multiple Targets Interception by the Swarm of UAVs based on Swarm Intelligence Algorithms: A Review," *IETE Technical Review*, pp. 1–23, 2021.

[12] A. Puente-Castro, D. Rivero, A. Pazos, and E. Fernandez-Blanco, "A review of artificial intelligence applied to path planning in UAV swarms," *Neural Computing and Applications*, pp. 1–18, 2021.

[13] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 243–254.

[14] S. Medjiah, T. Ahmed, and F. Krief, "AGEM: adaptive greedy-compass energy-aware multipath routing protocol for WMSNs," in *2010 7th IEEE Consumer Communications and Networking Conference*. IEEE, 2010, pp. 1–6.

[15] S.-J. Lee, M. Gerla, and C.-K. Toh, "A simulation study of table-driven and on-demand routing protocols for mobile ad hoc networks," *IEEE network*, vol. 13, no. 4, pp. 48–54, 1999.

[16] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.* IEEE, 2001, pp. 62–68.

[17] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," *IEEE Personal communications*, vol. 8, no. 1, pp. 16–28, 2001.

[18] H. Saha, S. Basu, S. Auddy, R. Dey, A. Nandy, D. Pal, N. Roy, S. Jasu, A. Saha, S. Chattopadhyay, and T. Maity, "A low cost fully autonomous GPS (global positioning system) based quad copter for disaster management," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 654–660.

[19] S.-s. Choi and E.-k. Kim, "Design and implementation of vision-based structural safety inspection system using small unmanned aircraft," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2015, pp. 562–567.

[20] M. Asadpour, K. A. Hummel, D. Giustiniano, and S. Draskovic, "Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 843–856, 2017.

[21] S. Ashraf, P. Aggarwal, P. Damacharla, H. Wang, A. Y. Javaid, and V. Devabhaktuni, "A low-cost solution for unmanned aerial vehicle navigation in a global positioning system–denied environment," *International Journal of Distributed Sensor Networks*, vol. 14, no. 6, p. 1550147718781750, 2018. [Online]. Available: https://doi.org/10.1177/1550147718781750

[22] W. Power, M. Pavlovski, D. Saranovic, I. Stojkovic, and Z. Obradovic, "Autonomous Navigation for Drone Swarms in GPS-Denied Environments Using Structured Learning," in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds. Cham: Springer International Publishing, 2020, pp. 219–231.

[23] M. Campion, P. Ranganathan, and S. Faruque, "UAV swarm communication and control architectures: a review," *Journal of Unmanned Vehicle Systems*, vol. 7, no. 2, pp. 93–106, 2019. [Online]. Available: https://doi.org/10.1139/juvs-2018-0009

[24] D. Mishra and E. Natalizio, "A survey on cellular-connected UAVs: Design challenges, enabling 5G/B5G innovations, and experimental advancements," *Computer Networks*, vol. 182, p. 107451, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620311324

[25] X. Chen, J. Tang, and S. Lao, "Review of unmanned aerial vehicle swarm communication architectures and routing protocols," *Applied Sciences*, vol. 10, p. 3661, 05 2020.

[26] A. Kohlbacher, J. Eliasson, K. Acres, H. Chung, and J. C. Barca, "A low cost omnidirectional relative localization sensor for swarm applications," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 694–699.

[27] S. Hameed, Q.-A. Minhas, S. Ahmad, F. Ullah, A. Khan, A. Khan, M. I. Uddin, and Q. Hua, "Connectivity of Drones in FANETs Using Biologically Inspired Dragonfly Algorithm (DA) through Machine Learning," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[28] W. Wang, P. Bai, Y. Zhou, X. Liang, and Y. Wang, "Optimal configuration analysis of AOA localization and optimal heading angles generation method for UAV swarms," *IEEE Access*, vol. 7, pp. 70 117–70 129, 2019.

[29] J. Zheng, R. Chen, T. Yang, X. Liu, H. Liu, T. Su, and L. Wan, "An efficient strategy for accurate detection and localization of UAV swarms," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 372–15 381, 2021.

[30] H. Qin, Z. Meng, W. Meng, X. Chen, H. Sun, F. Lin, and M. H. Ang, "Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1339–1350, 2019.

[31] S. Misra, B. Wang, K. Sundar, R. Sharma, and S. Rathinam, "Single vehicle localization and routing in gps-denied environments using range-only measurements," *IEEE Access*, vol. 8, pp. 31 004–31 017, 2020.

[32] F. She, Y. Zhang, D. Shi, H. Zhou, X. Ren, and T. Xu, "Enhanced relative localization based on persistent excitation for multi-uavs in gps-denied environments," *IEEE Access*, vol. 8, pp. 148 136–148 148, 2020.

[33] R. Moraes and E. Pignaton de Freitas, "Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services," *Journal of Intelligent & Robotic Systems*, vol. 92, 12 2018.

[34] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 96–108.

[35] T. Watteyne, I. Augé-Blum, M. Dohler, S. Ubéda, and D. Barthel, "Centroid virtual coordinates – a novel near-shortest path routing paradigm," *Computer Networks*, vol. 53, no. 10, pp. 1697–1711, 2009, autonomic and Self-Organising Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128608004325

[36] B. Leong, B. Liskov, and R. Morris, "Greedy virtual coordinates for geographic routing," in *2007 IEEE International Conference on Network Protocols*, 2007, pp. 71–80.

[37] A. Karima, B. Mohammed, and B. Azeddine, "New virtual coordinate system for improved routing efficiency in sensor network," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 3, p. 59, 2012.

[38] N. Filardi, A. Caruso, and S. Chessa, "Virtual naming and geographic routing on wireless sensor networks," in *2007 12th IEEE Symposium on Computers and Communications*. IEEE, 2007, pp. 609–614.

[39] J.-P. Sheu, K.-Y. Hsieh, and M.-L. Ding, "Routing with hexagonal virtual coordinates in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 9, no. 9, pp. 1206–1219, 2009.

[40] S. Ratnasamy, I. Stoica, and S. Shenker, "Routing algorithms for DHTs: Some open questions," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 45–52.

[41] R. Anwit, P. Kumar, and M. Singh, "Virtual Coordinates Routing Using VCP-M in Wireless Sensor Network," in *2014 International Conference on Computational Intelligence and Communication Networks*. IEEE, 2014, pp. 402–407.

[42] A. Awad, R. German, and F. Dressler, "Exploiting virtual coordinates for improved routing performance in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1214–1226, 2011.

[43] A. Samuylov, D. Moltchanov, R. Kovalchukov, A. Gaydamaka, A. Pyattaev, and Y. Koucheryavy, "GAR: Gradient assisted routing for topology self-organization in dynamic mesh networks," *Computer Communications*, vol. 190, pp. 10–23, 2022.

[44] S. Chahal and M. Singh, "An extensive literature review of various routing protocols in delay tolerant networks," *International Research Journal of Engineering and Technology*, vol. 4, no. 7, pp. 1309–1312, 2017.

[45] G. He, "Destination-sequenced distance vector (DSDV) protocol," *Networking Laboratory, Helsinki University of Technology*, vol. 135, pp. 1–9, 2002.

[46] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF, Tech. Rep., 2003.

[47] S. Mohapatra and P. Kanungo, "Performance analysis of AODV, DSR, OLSR and DSDV routing protocols using NS2 Simulator," *Procedia Engineering*, vol. 30, pp. 69–76, 2012.

[48] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, 1999, pp. 90–100.

[49] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," in *2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record*, vol. 1. IEEE, 2000, pp. 70–74.

[50] F. Cadger, K. Curran, J. Santos, and S. Moffett, "A survey of geographical routing in wireless ad-hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 621–653, 2013.

[51] L. A. L. da Costa, R. Kunst, and E. Pignaton de Freitas, "Q-fanet: Improved q-learning based routing protocol for fanets," *Computer Networks*, vol. 198, p. 108379, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621003595

[52] P. Nain, D. Towsley, B. Liu, and Z. Liu, "Properties of random direction models," in *IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, March 2005, pp. 1897–1907.

[53] S. Ruder, "An overview of gradient descent optimization algorithms," 2016. [Online]. Available: https://arxiv.org/abs/1609.04747

[54] C.-J. Hsieh and I. S. Dhillon, "Fast coordinate descent methods with variable selection for non-negative matrix factorization," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 1064–1072. [Online]. Available: https://doi.org/10.1145/2020408.2020577

[55] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 993–1000. [Online]. Available: https://doi.org/10.1145/1553374.1553501

Anna Gaydamaka received the M.Sc. degree in Computer Science, program Data Science and Business Informatics) from Universita di Pisa, Italy in 2021. Currently, she is pursuing a Ph.D. in Computing and Electrical Engineering and working as a Doctoral Researcher at Tampere University, Finland. Her research interests include resource planning of the fifth and sixth-generation wireless network, mathematical models for performance KPIs optimization, and machine learning techniques.

Andrey Samuylov received his Ms.C. in Applied Mathematics and Cand.Sc. in Physics and Mathematics from the RUDN University, Russia, in 2012 and 2015, respectively. Since 2015 he is working at Tampere University as a researcher, working on analytical performance analysis of various 5G wireless networks technologies. His research interests include P2P networks performance analysis, performance evaluation of wireless networks with enabled D2D communications, and mmWave-band communications.

Dmitri Moltchanov is a university lecturer at Tampere University. He received the M.Sc. (2000) and Cand.Sc. (2003) degrees from the St. Petersburg State University of Telecommunications, Russia, and the Ph.D. (2006) degree from TUT. He has authored more than 150 publications and has taught more than 60 courses on several topics in telecommunications. His current research interests include research and development of 5G/5G+ systems, industrial IoT applications, mission-critical V2V/V2X systems, and blockchain technologies.

Mateen Ashraf received PhD in wireless communication engineering in 2017. He is currently working as a research fellow at Tampere University. His current research interests include beamforming design for integrated sensing and communication (ISAC) systems for 6G networks and wireless energy harvesting systems. He has published several journal and conference papers in leading IEEE journals and conferences. He is an active reviewer of many leading journals and has acted as a TPC member for prestigious IEEE conferences.

Bo Tan received his PhD from the Institute for Digital Communications, The University of Edinburgh, U.K., in Nov 2013. From 2012 to 2016, he was a postdoctoral researcher at the University College London and the University of Bristol, UK, contributing to passive radar research and applications in healthcare and security. From 2017 to 2018, he was a lecturer at Coventry University, U.K.; since 2019, he has been a Tenure Track Assistant Professor at Tampere University, Finland. His research interests include radio signal processing, sensing and connectivity for intelligent machines. He is PI and coordinator of multiple Academy of Finland, Business Finland and Horizon European research projects on distributed machine learning, mmWave, positioning, surveillance, physical layer security, and radio sensing for healthcare. He is the reviewer of multiple IEEE/IET/ACM journals and conferences in wireless communications, radar, pervasive computing and sensing.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3293034

Yevgeni Koucheryavy received the Ph.D. degree from the Tampere University of Technology (TUT), Finland. He is currently a Professor at the Laboratory of Electronics and Communications Engineering, TUT. He is the author of numerous publications in the field of advanced wired and wireless networking and communications. His current research interests include various aspects in heterogeneous wireless communication networks and systems, the Internet of Things and its standardization, and nanocommunications. He is Associate Technical Editor of the IEEE Communications Magazine and an Editor of the IEEE Communications Surveys and Tutorials.