IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 22, 2019, accepted November 11, 2019, date of publication November 28, 2019,
date of current version December 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956724

INVITED PAPER

STFANET: SDN-Based Topology Management for Flying Ad Hoc Network

TULIO DAPPER E SILVA[1], CARLOS F. EMYGDIO DE MELO[1], PEDRO CUMINO[2], DENIS ROSÁRIO[2], EDUARDO CERQUEIRA[2], AND EDISON PIGNATON DE FREITAS[1]

[1]Graduate Program in Electrical Engineering, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre 90035-190, Brazil
[2]Computer Science Faculty, Federal University of Pará (UFPA), Belém 66075-110, Brazil

Corresponding author: Tulio Dapper e Silva (tuliodapper@gmail.com)

This work was supported in part by the National Council for Scientific and Technological Development (CNPq) and in part by the National Council for the Improvement of Higher Education (CAPES).

ABSTRACT In recent years, with the growth in the use of Unmanned Aerial Vehicles (UAVs), UAV-based systems have become popular in both military and civil applications. The lack of reliable communication infrastructure in these scenarios has motivated the use of UAVs to establish a network as flying nodes, also known as Flying Ad Hoc Networks (FANETs). However, the high mobility degree of flying and terrestrial users may be responsible for constant changes in the network topology, which makes more challenging to guarantee their communication during the operational time. In this context, this article presents a Software-defined networking (SDN) based Topology management for FANETs - called of STFANET -, which is a coordination protocol that englobes both an efficient SDN-based UAV communication and a set of topology management algorithms. The goal is to establish and maintain a FANET topology in order to provide a constant and reliable communication link among independent nodes - which are performing individual or collaborative missions - through relays units. Simulation results show the efficiency of the proposed protocol in order to provide communication in a dynamic scenario. Considering its use in a military setting, STFANET managed to achieve 25% of packet loss in transmitting data packets, 1.5ms of latency and 71% of connectivity on average.

INDEX TERMS FANET, topology management, relay node placement, SDN, communication protocol.

I. INTRODUCTION

The lack of reliable communication infrastructure in collaborative missions performed by multiple users, such as in post-disaster and warfare scenarios, is a notorious problem since decisions and actions might require shared information and a short time to be taken. In some cases, reliable and fast wireless communication among them through the network is critical, as time is precious when rescuing victims in a post-natural disaster scenario or soldiers on the battlefield support are under concern [1], [2]. In the past few years, collaborative applications of Unmanned Aerial Vehicles (UAVs) have been proposed to address such demands, by the so-called Flying Ad Hoc Network (FANET) [3]–[8], due to their capabilities of sensing, processing, moving and communicating [9], [10].

Particularly, nowadays warfare requires an adaptive, flexible and autonomous networks to rapidly adjust to different situations and missions [11]. In fact, FANETs have been already considered in very concrete military scenarios to support Command and Control (C2) systems in the battlefield [12]–[15], and teams of UAVs have been considered to search for victims in post-disaster situations [16]–[19].

In both civilian and military applications, multiple and independent nodes - which may be terrestrial or flying users - perform their own or collaborative mission by exploring, sensing, and disseminating information. As they may require fast data dissemination, a well-established FANET connection is a valuable asset in order to guarantee communication among them with high availability and reliability [14]. As a consequence, flying relay nodes may be considered being best allocated in order to provide desired connectivity among independent nodes as long and as stable as possible. In case of disconnection, the FANET must be able to adapt itself and reestablish the connection as swiftly as possible.

The controller and relay nodes' placement is referred to as topology management in this article and is generally

The associate editor coordinating the review of this manuscript and approving it for publication was Oussama Habachi.

a combination of two different phases: construction and adjustment [20], [21]. Firstly, there must have a solution for the initial nodes distribution, which is performed during the construction phase and is computationally expensive to achieve a feasible result in general. Therefore, further maintenance should be performed more lightly by adjustment algorithms. At some point, the adjustment solutions may not be particularly enough to maintain a satisfying network topology performance. Aiming to evaluate the network and then determine whether to execute the construction or the adjustment method, an integration algorithm is generally implemented, which is capable of evaluating the network current topology and deciding which method to consider (either the construction or the adjustment). As soon as a construction algorithm is performed, nodes should be assigned, which brings node allocation problem. In their work, Kim and Lee [21] proposed a set of algorithms in order to construct and maintain a FANET composed of Relay Units (RU), which were in charge of establishing a connection between Mobile Units (MU) and their ground station. In their solution, the authors did not consider the allocation of the controller node in the network, as well as it is aware of the nodes' current locations throughout the whole simulation time without relying on a multi-hop exchange of messages. As a consequence, their proposal by its own is still not feasible in a realistic scenario.

Observing the above-described landscape, a communication protocol must coordinate and sense the behavior of multiple UAVs in order to maximize the benefits of the FANET [22]. In this context, Software-Defined Networking (SDN) is a promising approach for the FANET's controlling task, since it introduces complete network programmability by separating the control plane and the data plane [23], [24]. As a consequence, SDN provides flexibility towards many aspects of the network management, such as dynamic routing, having been already used for monitoring and maintenance of the network by a centralized controller [23]. For instance, Zhao *et al.* [25] introduced an SDN architecture for FANET (SD-UAVNet), in which a controller node is capable of managing relay nodes' locations and their routing policies in order to disseminate video from a source to a destination node. Therefore, control packets are used towards the topology management in order to provide the best performance in data transmission. In their work, the topology does not change dynamically - *i.e.*, the source and the destination nodes are fixed in space -, which leads to a simple topology management methodology.

Although many works [14], [26]–[29] have proposed the use of FANET to establish the connection among different terrestrial and flying nodes, their proposals left room for further improvements, especially in reliability and connectivity maintenance [30]. Moreover, a careful review in the available literature reveals a lack of approaches that completely address the problem in setting up and maintaining the network connectivity of dynamic networks, such as FANETs, without relying on either anchor nodes (*e.g.*, a base station) or less flexible network topologies, which are setup offline. Zhao *et al.* [25] presents a realistic simulation, however with restricted network topology. On its turn, Kim and Lee [21] present a solution for topology management in a dynamic FANET formation, however without realistic simulations. Therefore, to the best of our knowledge, we have not found other works that have focused on a topology management solution, which considers the allocation of a controller and a set of relay nodes, in order to sustain a network composed by dynamic users and also have presented results from wireless network simulators.

To cover the gap found in the literature review, this article introduces a coordination protocol - which covers SDN-based UAV communication and topology management algorithms -, called STFANET. Thus, this article combines a UAV communication protocol based on Zhao *et al.* [25] with topology management algorithms based on Kim and Lee [21]. We consider a network composed of three types of nodes: a controller node, a set of relay nodes, and a set of independent nodes. Relay nodes aim to sustain communication links among independent nodes for as long as possible. Hence, the controller node is in charge of finding and setting up the best location for the relay nodes through topology management algorithms, as well as finding and setting up routing tables for the relay and independent nodes. From a software-defined point of view, the maintenance of the network is performed through the exchange of control packets. To execute this process, the controller periodically collects contextual information from all the nodes and informs their new location and routing configuration as needed in a multi-hop manner. Compared to the related work found in the literature, this article introduces the first realistic implementation of a FANET aimed to minimize the disruptions in a network composed of dynamic nodes, such as UAVs and terrestrial vehicles, as it manages to keep the entire network connected, with only short time disconnections.

In order to run the simulation, while considering modeled physical constraints, the proposed coordination protocol was implemented at the network layer using the OMNet++ simulator. The performed simulation considered a large area, such as a city, where independent nodes randomly move through the scenario. As most of the operations limit the number of relay node - as they introduce financial cost -, this article presents the FANET performance evaluation categorized by the number of relay nodes. Based on simulation results, STFANET maintains the independent nodes interconnected over 70% of the simulation time. Without considering re-transmission as delivered packets, the data packet loss rate is limited to 25% and the control packet loss rate is up to 10%. The performance gap is due to the capability of the controller to set the relay nodes' position in order to avoid being out of range, whilst the independent nodes are not over control of the controller node. The results also show that the latency is considerably reduced as a higher number of nodes is introduced in the network. Varying from 3 to 15 nodes, the latency can be reduced from 15% to 40%, depending on the number of hops needed for a single transmission.

The main contributions of this article are:

- The introduction of STFANET topology management algorithms - which include construction, adjustment, integration and node allocation - designed to establish and maintain communication links among mobile and independent nodes - also referred as users - by using a team of UAVs;
- The description of the STFANET communication protocol over the control plane enabling the controller to monitor other nodes' conditions and to control them based on the topology management algorithms;
- Implementation and evaluation of the STFANET coordination protocol in a network simulator in order to assess network performance metrics in terms of packet loss, latency, and connectivity by varying the number of relay nodes.

The rest of this article is organized as follows. Section II discusses the related works for the use of SDN concept in FANETs, highlighting their main drawbacks. Section III presents the architecture, which provides an overview of the nodes' resources and capabilities. One of them is the coordination protocol, part of the controller node and matter of discussion in Section IV. This section describes the SDN-based communication protocol implemented for the control plane, as well as, the topology management algorithms, along with the mathematical formulation used as the basis for its construction and maintenance. Section V describes the performed experiments to evaluate the proposed solution and discusses the acquired results. Finally, concluding remarks and possible future work are presented in Section VI.

II. RELATED WORKS

In the military domain, ground nodes may be partitioned due to the mission requirements. A fixed placement strategy of the communication structure is not wise as it would be probably be attacked and yet would not be able to serve marching troops. Basu *et al.* [15] proposed the use of UAVs which obey local flocking rules (like birds and insects) to adapt themselves to the motion of ground nodes and therefore maintain high connectivity among them. Hence, the authors focused their work to achieve a network connection among moving ground nodes in a distributed manner through a FANET. In their work, the authors aim to minimize the number of disconnected ground nodes, as well as minimize the standard deviation in the number of connected nodes by UAV. By the end, they have shown efficacy in using this approach rather than a static structure in a scenario which troops are marching.

Sánchez-García *et al.* [16] aimed their work to use a team of UAVs in post-disaster scenarios, where there is a need to effectively search for victims and provide them further assistance with data link communication. They proposed a distributed Particle Swarm Optimization (PSO) algorithm to perform the exploration mission by having the UAVs sharing their best candidate solutions with their neighbor nodes. Moreover, they presented a characterization of the proposed algorithm by using a different set of parameters' values. In comparison with a trajectory planning algorithm that sweeps the entire area - the Lawnmower Algorithm (LMA) -, their solution was able to find 25%, 50% and 75% of the victims faster than the LMA. In addition, they pointed out that their proposed algorithm presented a higher number of connections and a smaller amount of elapsed time among them in comparison to the LMA.

Focused on the Wireless Sensor Networks (WSN) context formed by static nodes, Magán-Carrión *et al.* [31] proposed a three-stage relay node placement strategy designed to maximize connectivity by measuring throughput and inter-node reachability. The three steps are: firstly, a set of potential locations for the relay nodes are identified (which may be higher in number than actually available). Then, the best locations are selected based on the Leave-One-Out (LOO) approach, which focuses on optimizing reachability. Finally, the solution is optimized in terms of throughput by using a PSO-based algorithm. In the simulation analyses, they demonstrated to have proposed an efficient method, although demanding long execution time. Regarding [31], Magán-Carrión *et al.* [20] continued focusing on the network deployment - *i.e.*, distribution of relay nodes -, however considering they capable of self-relocating. Having included such a feature, they proposed a multi-stage Dynamical RN placement Solution (DRNS), based on PSO algorithms and Model Predictive Control (MPC) techniques. Following a bi-objective optimization procedure, both network connectivity and throughput were jointly maximized.

Applications, such as surveillance operations, might require that information should be transmitted from a mobile device to the base station throughout the execution of the task. Having constraints related to the wireless transmission ranges and physical obstacles in the scene, relay chains formed by intermediate UAVs may be necessary. Burdakov *et al.* [32] focused their work on presenting a solution by maximizing transmission quality given a known target position and minimizing the number of relay nodes which are required. They presented two new algorithms: one uses label-correcting graph search to efficiently generate a set of optimal relay chains solutions - in which reveals a trade-off between the number of UAVs and the quality of the chain -, allowing ground operators to choose among them; the second uses a dual ascent technique to generate high-quality relay chain given a limited number of UAVs.

Kim and Lee [21] proposed a topology management methodology in order to sustain a FANET topology and therefore establish communication links between Mobile Units (MU) and their ground station, by using a set of Relay Units (RU). Through the execution of three algorithms - construction, adjustment, and integration -, they presented the efficacy of their approach in terms of the resultant distances among nodes. The strategy begins with the construction algorithm to determine a starting point for each node, which is computed based on Particle Swarm Optimization (PSO) approach. Consequently, this procedure is computationally overwhelming and moreover the result may differ

TABLE 1. Summary of topology management proposals.

Reference	Addressed Problem	Proposal	Centralized vs. Distributed	Mobile Independent Nodes	Control Plane Communication Protocol	Network Performance Evaluation	One-hop vs. Multi-hop to Controller
Sánchez-Garcia et. al. [16]	Discover and provide assistance to victims in a post-disaster scenario	UAV team follows a distributed PSO-based exploration algorithm	Distributed	No	No	No	Not applicable
Basu et. al. [15]	UAV placement and navigation strategies to provide connection among marching ground nodes	Adaption to the motion of ground nodes using local flocking rules	Distributed	Yes	No	No	Not applicable
Magán-Carrión et. al. [31]	Relay node placement for static scenarios	Three-stage placement procedure based on PSO and LOO algorithms	Centralized	No	No	No	One-hop
Burdakov et. al. [32]	Relay positioning for providing connection between UAV and base station with given obstacles	Label-correcting and dual ascent algorithms for relay chain generations	Centralized	No	No	No	One-hop
Kim et. al. [27]	FANET topology management for adapting to frequent and rapid fluctuations	Topology management based on construction and adjustment algorithms	Centralized	Yes	No	No	Not applicable
Magán-Carrión et. al. [20]	Relay node placement for dynamic scenarios	Multi-stage placement solution based on PSO and MPC techniques	Centralized	Yes	Partial	No	One-hop
Zhao et. al. [25]	Management of UAV network to guarantee satisfactory video quality	Software-defined UAV networking architecture (SD-UAVNet)	Centralized	No	Yes	Yes	Two-hop
STFANET This Proposal	FANET topology management for providing connection among mobile and independent nodes	SDN UAV controller performing topology management algorithms	Centralized	Yes	Yes	Yes	Multi-hop

significantly from the current nodes' location. In order to keep the connectivity of the network after its construction, a Gradient function algorithm is used to adjust the current formation. On its turn, this procedure is much lighter in terms of computational effort; however, at some point, the controller must be able to detect the lack of its efficiency. In this case, the formation is completely redesigned applying the construction algorithm, which may completely differ from the current one. Finally, the authors implemented a method to integrate both construction and adjustment procedures.

Zhao *et al.* [25] and Kirichek *et al.* [23] presented SDN-like solutions for FANET. Firstly, Zhao *et al.* [25] considered the allocation of relay nodes aiming to establish a communication link between a source and a destination node for video transmission. To achieve this goal, the controller node periodically gathers relevant information from each node, such as the location and the remaining energy. As soon as an event is identified, the controller selects and places relay and source nodes in strategical locations. After settling the first formation by managing their positions and filling their routing tables, the SDN controller keeps monitoring the condition of the nodes and may replace them if they inform low residual energy. Similarly, Kirichek *et al.* [23] also proposed an SDN-based solution for FANET in order to manage a group of UAVs. Flying nodes are used simply as switches in order to settle communication links between terrestrial segments and a flying data collector. An additional node is considered as the SDN controller to update the nodes' routing tables. Their work showed the efficiency of the SDN-based solution through simulation.

Orfanus *et al.* [14] proposed the use of Self-Organization (SO) paradigm to design a robust and efficient UAV relay network in order to support military operations, such as reconnaissance and area coverage. For robustness, they used Emergent Self-Organization (ESO), which is a decentralized type of SO. Simultaneously with ESO, they implemented a feedback-based system which applied positive feedback to expand area coverage and negative feedback to maintain nodes' connectivity. Their work considered the scenario where area coverage is the main goal, instead of keeping connectivity among nodes performing unknown missions.

Based on the analysis of the state-of-the-art, it is possible to conclude that there is a lack in the literature for realistic simulations of Centralized Self-Organizing (CSO) FANETs for accomplishing undetermined missions - carried by multiple and mobile independent nodes -, and which also include the controller in the network as a node. Therefore, the controller node must communicate with the others through a multi-hop wireless link. Moreover, there is a lack of network performance evaluations, such as packet loss and latency, in works that also propose solutions for the relay node placement problem. The coordination protocol, which considers both the control plane protocol and the topology management algorithms, must coordinate a fleet of UAVs aimed to avoid connection losses in a dynamic scenario. Very few works describe the communication protocol - including packet types and parameters - required over the control plane in order to perform such a task. Thus, so far, not all of these key features have been provided in a unified cooperative UAV scheme for enhancing the military and civil applications. Table 1 summarizes the main characteristics of existing works related to the topology management problem.

III. ARCHITECTURE

The architecture is composed of three types of nodes, namely, one Controller Node (CN), a set of Relay Nodes (RN) and a set of Independent Nodes (IN). The INs perform an arbitrary mission (such as the search for points of interest [28]) and thus their positions are defined exclusively by themselves. On its turn, the RNs must be allocated in order to provide the best possible link availability among the INs. The topology and routing management are centralized and performed by a flying node referred to as the CN. It periodically receives information from all the other nodes of the network, which contains contextual information, such as their position, trajectory, and speed. In this way, the CN is responsible for positioning the RNs and itself, and also for setting the routing tables of each node. By doing this, the CN aims to provide the connectivity among the INs, as well as between itself and all the other nodes. The descriptions of these three types of nodes, as well as their internal functionalities, are presented in the following:

- The RN is responsible for forwarding packets and, as a consequence, establishing the connection among INs (for data packet transmissions) and between all the nodes and the CN (for control packet transmissions on its turn). Every RN must periodically send its current state to the CN, so it is able to determine the positioning and routing scheme to extend or, at least, preserve the network connection;
- The CN is primarily responsible for monitoring nodes' locations, as well as setting up and maintaining the entire network. Hence, the CN updates the routing tables of every node (INs and RNs) and also determines the position of the RNs. Although the CN could serve as a relay node, it acts exclusively in collecting and transmitting control packets in order to focus its processing and radio capabilities in the coordination functionality;
- The IN is performing a mission which is unknown from the perspective of the network control. Hence, its location is not determined by the CN, however by the application itself. Similarly to the RNs, every IN must periodically provide its current state to the CN.

The Figure 1 presents the functionalities of the *Controller Node*, the *Relay Node* and the *Independent Node*. The links between the nodes are basically formed through the *Control Plane* and the *Data Plane*. All nodes exchange packets in *Control Plane*, so the CN is able to monitor and control the network through this plane. On the other hand, in the *Data Plane*, only INs can transmit data packets among each other through the RNs. Both INs and RNs contain a simple *Routing Protocol* that manages to receive data or control packets, process or relay them. In the *Control Plane*, both RNs and the INs transmit *Mobility Model*'s information (*i.e.*, current position, speed, and trajectory) to the CN and process its demands, altering its *Routing Table* or modifying its position through the *Mobility Manager*. The INs also have the *Application* layer implemented on it, which is responsible for sending and receiving data packets through the *Data Plane*. Instead of having a *Routing Protocol*, the CN contains the proposed *Coordination Protocol*, which has stored the content contained in the *Routing Table* of the entire network; the *Mobility Manager* in order to move itself; and the *Topology Manager* in order to process nodes' information and compute the entire network topology. This work focuses on the *Coordination Protocol*.

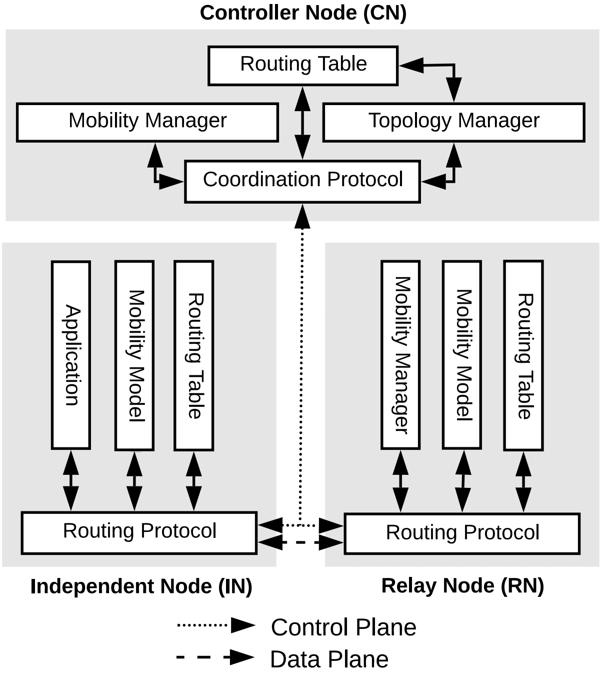

FIGURE 1. STFANET architecture.

IV. COORDINATION PROTOCOL

The coordination protocol is included in the network layer of the CN's communication stack. The protocol acts in the control plane, by monitoring and controlling the FANET behavior. In this section, the STFANET control plane protocol is presented, as well as the algorithms considered for construction and maintenance of the network topology based on the collected information.

TABLE 2. Control packet types exchanged by the CN, INs and RNs.

Packet Type	Source	Destination
Contextual information	IN and RN	CN
Set location	CN	RN
Set route	CN	IN and RN
Acknowledge	IN and RN	CN
Request route	IN and RN	CN

A. SDN-BASED COMMUNICATION PROTOCOL

The CN is responsible for routing and topology management through the exchange of control packets either from or to every node included in the FANET. In this work, it is considered the use of five types of messages, that are summarized in Table 2 and described in the following:

- Contextual information: information about nodes' mobility need to be periodically transmitted from each IN and RN to the CN informing their position, speed, and trajectory. This set of information is encapsulated and transmitted to the CN;

- Set location: after the CN has processed the topology management algorithms, it communicates the change of position to each node that is affected. The CN is able to alter the position exclusively of RNs;
- Set route: as soon as the CN alters an edge in the network graph, it communicates the change of routing to each affected node. In this case, both of the RNs and INs' routing tables need to be reconfigured by the CN;
- Acknowledge: whenever either an RN or an IN receives a control packet from the CN, it responds with an acknowledge message. If the CN does not receive any confirmation packet, it will re-transmit that original command in order to ensure its control over the FANET;
- Request route: in the case of an RN or an IN receives a packet with a destination that is not included in the routing table current rules, the node will notify the occurrence to the CN. The CN will then be able to transmit the required routing table rules to the affected nodes.

B. TOPOLOGY MANAGEMENT

Based on the collected mobility information from each node in the FANET, the CN is able to compute topology management algorithms to best allocate the available RNs in order to sustain the desired connectivity among the nodes. The execution of these algorithms is periodic, having a resulted configuration that is transmitted through control packets to the entire FANET.

In this sense, the main algorithms of the topology management are explained, which aim to solve the construction, adjustment, and integration of the topology formation, as well as the node allocation problem. As will be presented in this section, a PSO-based strategy is used in the construction algorithm. On its turn, a Gradient function is used towards the adjustment algorithm. Throughout the entire operation, both construction and adjustment algorithms are used independently to update nodes' positions, in a procedure referred to as the integration algorithm. Finally, an algorithm for solving the node allocation problem based on distance vectors is also presented.

1) SYSTEM MODEL

The architecture described in Section III considers: a set of INs $I = \{i_k\}_{k=1}^{|I|}$; a set of UAVs as RNs $R = \{r_k\}_{k=1}^{|R|}$; and one UAV as the CN c, as shown in Figure 2.

One of the main functionalities of the CN is to set the routing table rules of each node based on the graph of the network as a whole. It is considered two distinct routing tables. Firstly, there is one for the control packet transmissions P_c, in which the CN is either the source or the destination of the routes. Moreover, there is another for the data packet transmissions P_d, in which the CN is not included in the graph; for these routes, the source and the destination are the INs of the network. As a consequence, the CN does not participate as a router for data packets. This feature will avoid the CN of having data packets to be processed and relayed, saving its computational and energy resources as a consequence.

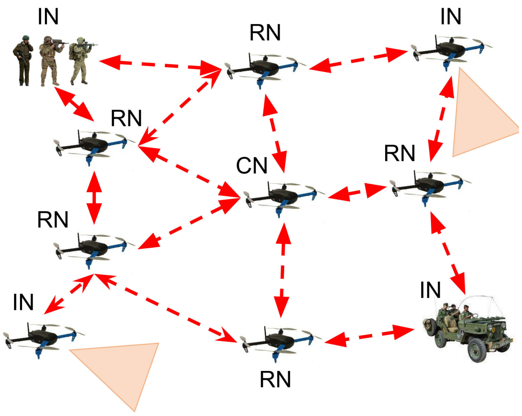

FIGURE 2. Architecture applied to a military application.

In order to manage the topology of the network, the CN needs to store the location and the routing information of each node. Firstly, its own location is defined by x_c. The set of RNs' location and the set of INs' location are defined by $X_R = \{x_v\}_{v \in R}$ and $X_I = \{x_v\}_{v \in I}$, respectively. Aiming to abbreviate further equations, X is a set containing all nodes' location, as shown in (1). In addition, the routing tables rules for data and control packet transmissions of each node are also stored at the CN. Firstly, (2) presents P_d as the set of routes p stored at the CN for data packet transmissions. On its turn, as shown in (3), P_c contains the routes p for the control plane messaging. The union of these set of routes P is shown in (4).

$$X = x_c \cup X_R \cup X_I \tag{1}$$

$$P_d = \{p_u^v\}_{u,v \in I, u \neq v} \tag{2}$$

$$P_c = \{p_u^v\}_{u=c, v \in R \cup I} \cup \{p_u^v\}_{u \in R \cup I, v=c} \tag{3}$$

$$P = \{p_k\}_{k=1}^{|P|} = P_d \cup P_c \tag{4}$$

In order to best place the RNs and itself, the CN takes into account many aspects. The first criteria to be considered is the minimum distance of d_s among the nodes to avoid their collision. In addition, an ideal link length d_r^* should also be respected in order to have proper communication through the wireless channel. Moreover, there is a maximum link length of d_r^{max} that two nodes may still be able to communicate among each other. The CN aims to have the nodes allocated between the safe distance and ideal link length - whenever there is an established route. Hence, the constant values d_s, d_r^* and d_r^{max} must obey (5):

$$d_s < d_r^* < d_r^{max} \tag{5}$$

Aiming to encounter a routing solution, the Dijkstra algorithm was adopted considering the length of each link as its cost. The algorithm is not detailed in this article, as it is a well-known algorithm and as it is not the focus of this work; it is represented by ρ, as shown in (6). This function is responsible for returning the set of routes P by having X, which contains the locations of the CN, the RNs and the INs, as well as the

TABLE 3. Priorities in order to compare solutions for the construction algorithm.

Priority	Description	Goal
1	Number of routes considering the maximum range of the link.	Have maximum number of routes.
2	Total number of hops used in each link.	Reduce total amount of hops aiming to have better quality of service.
3	Cost function of distances.	Better distribute the nodes through minimizing the cost function.

maximum link length to be considered d_r. At some point, it may be interesting to have the ideal - d_r^* - or the maximum - d_r^{max} - link length as the parameter d_r.

$$P = \rho(X, d_r) \tag{6}$$

Considering that it should have one route among each IN of the network, (7) defines the maximum - and ideal - number of data packet routes n_d^*. In addition, (8) presents the desired number of ideal number of control packet routes n_c^*, considering that it should have one route from the CN to all the other nodes and vice-versa. Finally, (9) defines n^* as the total amount of routes in an ideal scenario.

$$n_d^* = |I|^2 - |I| \tag{7}$$

$$n_c^* = 2(|I| + |R|) \tag{8}$$

$$n^* = n_d^* + n_c^* \tag{9}$$

2) CONSTRUCTION

STFANET considers a construction algorithm in order to build the topology first formation and redesign it whenever is needed. In this sense, STFANET takes into account the Particle Swarm Optimization (PSO) algorithm to find an optimal solution. Specifically, the aim is to define the best solution for node placement after a fixed number of random solutions have been interactively improved. Having it shortly, for each iteration and for each solution (a particle), the current one is compared to its best (known as the local best) and to the best of all other particles by the moment (known as the global best solution). The local and the global best solution are updated through the iterations. In the end, the final result is the best global solution that has been achieved.

In the PSO algorithm, particles - or solutions - need to be compared with the local and the global best solution at each iteration. Rather than using a single cost function, this article adds the consideration of a number of valuable properties prior to calculate and to compare the cost value. Some properties (*e.g.*, the number of connected nodes and the number of hops) are considered to be more relevant than other criteria. Hence, the comparison between different particles is performed following the criteria described in Table 3. Firstly, a particle will be selected rather than another if the first contains more active routes than the second. If there is no difference in this property, the second priority is analyzed: a reduced number of hops will result in lower latency and, consequently, a better quality of service for the user. In the end, the target is to minimize a cost function that considers the sum of the link length, and the maximum violations of the link length and safe distance among nodes. As a result, the proposed evaluation of the topology solutions takes into account the number of routes, the number of hops, the links' length and the distance among nodes.

Therefore, the topology strategy is mainly assessed through a cost function aimed to be minimized during the execution of the PSO algorithm. The cost function is presented in (10), which is the sum of (11), (12) and (13). In the following, $\delta(u, v)$ is considered to be the distance between nodes u and v. Firstly, (11) determines the sum of the link length being part of the topology, having σ as weight. Equation (12) considers the restriction of having link lengths greater than the ideal value (d_r^*). Therefore, this component is the sum of the distances between the nodes which do not respect the communication range distance and takes λ as weight. Finally, (13) considers the restriction of having the nodes closer than the admitted distance value. On its turn, this metric is defined as the square of the closest link that does not obey such restriction and takes μ as weight.

$$f(X, P) = f_1(X, P) + f_2(X, P) + f_3(X, P) \tag{10}$$

$$f_1(X, P) = \sigma \sum_{p \in P} \sum_{k=1,\ldots,|p|-1} \delta(p_k, p_{k+1}) \tag{11}$$

$$f_2(X, P) = \lambda \sum_{p \in P} \left(\max\left\{0, \max_{k=1,\ldots,|p|-1} \delta(p_k, p_{k+1}) - d_r^*\right\}\right)^2 \tag{12}$$

$$f_3(X, P) = \mu \left(\max\left\{0, d_s - \min_{u,v \in c \cup R \cup I, u \neq v} \delta(u, v)\right\}\right)^2 \tag{13}$$

In the following, the population Ψ is a set of particles ψ, which its size is $N_\Psi = |\Psi|$. Moreover, u_1 and u_2 are random variables uniformly distributed on vectors $[0, 1]^2$, and $\circ$ stands for the Hadamard product. During the iterations of the PSO algorithm, (14), (15), and (16) are used to determine the new particles' proprieties - *i.e.*, velocities and positions. Firstly, (14) determines the new velocity value V_ψ^{k+1}, which depends on the current value V_ψ^k and the weight w. In addition, the velocity value also depends on the current local best position $X_{\psi*}^k$ weighted by $c_1 \cdot u_1$ and the current global best solution X_{g*}^k weighted by $c_2 \cdot u_2$. In (15), after having determined the new velocity value V^{k+1} - based on the current state -, each element of $V_\psi^k = \{v_{\psi,j}^k\}_{j=1}^{|V|}$ is analysed and, if necessary, set to its limit V_{max}. Finally, the next position value is the sum of the current value and the velocity value recently determined, as seen in (16).

$$\begin{aligned} V_{c \cup R, \psi}^{k+1} &= w V_{c \cup R, \psi}^k \\ &\quad + c_1 u_1 \circ (X_{c \cup R, \psi^*}^k - X_{c \cup R, \psi}^k) \\ &\quad + c_2 u_2 \circ (X_{c \cup R, g}^k - X_{c \cup R, \psi}^k), \quad \forall \psi \in \Psi \end{aligned} \tag{14}$$

$$v^{k+1}_{\psi,j} = \begin{cases} v^{k+1}_{\psi,j}, & \text{if } v^{k+1}_{\psi,j} \in [-V_{max}, V_{max}]. \\ -V_{max}, & \text{if } v^{k+1}_{\psi,j} < -V_{max}. \\ V_{max}, & \text{otherwise.} \end{cases} \tag{15}$$

$$X^{k+1}_{c\cup R,\psi} = X^{k}_{c\cup R,\psi} + V^{k+1}_{c\cup R,\psi}, \quad \forall \psi \in \Psi \tag{16}$$

The procedures of the construction algorithm are presented in (1). As shown, the entries are the positions of the INs X_I and the number of available RNs $|R|$. The output is the CN and RNs' formation - respectively, x^*_c and X^*_R - resulted by the PSO computation. Firstly, the particles ψ are initialized with uniformly random RNs' positions within the INs' perimeter (line 2) and routes are formed (line 3). While the particles are initialized, their individual local best solution ψ^* are set to their initial values (line 4), as well as the global solution g is continuously updated among the initializations (line 5). Following that, the PSO algorithm is iteratively computed by altering the particle values (line 9 and 10), computing new routing policies (line 11), as well as choosing the best solution among the derived solutions (line 12 and 13). The iterations run until one of the two following conditions is satisfied: the global best solution does not suffer any change during a fixed number of iterations N^t_k or the maximum number of iterations N_k is achieved. In the end, the algorithm's output (x^*_c and X^*_R) is derived from the best global solution computed so far (line 15 and 16).

Algorithm 1 Construction Algorithm

```
   Input: $X_I$ and $|R|$.
   Output: $x^*_c$ and $X^*_R$.
1  for each particle $\psi \in \Psi$ do
2      Initialize $x_{c,\psi}$, $X_{R,\psi}$, $v_{c,\psi}$, and $V_{R,\psi}$.
3      $P_\psi \leftarrow \rho(x_{c,\psi}, X_{R,\psi}, X_I, d^*_r)$;
4      $\psi^* \leftarrow \psi$;
5      Update $g$ according to Table (3) and
       $f(x_{c,\psi^*}, X_{R,\psi^*}, X_I, P_{\psi^*})$.
6  end
7  repeat
8      for each particle $\psi \in \Psi$ do
9          Update $v_{c,\psi}$ and $V_{R,\psi}$ according to (14) and (15).
10         Update $x_{c,\psi}$ and $X_{R,\psi}$ according to (16).
11         $P_\psi \leftarrow \rho(x_{c,\psi}, X_{R,\psi}, X_M, d^*_r)$;
12         Update $\psi^*$ according to Table (3) and
           $f(x_{c,\psi}, X_{R,\psi}, X_I, P_\psi)$.
13         Update $g$ according to Table (3) and
           $f(x_{c,\psi^*}, X_{R,\psi^*}, X_I, P_{\psi^*})$.
14     end
15     $x^*_c \leftarrow x_{c,g}$;
16     $X^*_R \leftarrow X_{R,g}$;
17 until termination conditions are satisfied;
```

Equation 17 defines the construction algorithm. As can be seen, α represents the construction procedure that returns an unsorted RNs' location X'_R and the CN's location x_c, having the INs' location X_I and the number of RNs $|R|$ as entries.

$$\{x_c, X'_R\} = \alpha(X_I, |R|) \tag{17}$$

3) ADJUSTMENT

The incremental adjustment of the nodes' location is performed by monitoring the built topology. Basically, by measuring the distance between each node and its neighbors, a gradient function aims to evenly distribute them. In this way, the adjustment algorithm does not compute any other routing alternative in order to consider a better routing solution. Instead, it only alters nodes' position in order to best allocate them facing the current topology.

Both (18) and (19) are in charge of performing the topology adjustment. In (18), the resultant position x^*_n referred to the node n is derived from the current state x_n and gradient value Δ_{x_n}, which is on its turn determined based on its neighbors as shown in (19). In addition, there is a threshold value in order to consider a speed limit in the adjustment strength, where γ scales the gradient and γ_r defines the maximum adjustment value (speed limit). In order to determine the gradient value Δ_{x_n} referred to the node n, (19) takes into consideration each neighbour node v which belongs to one of the routes contained in P; $N_n(p)$ is the set of two neighbors of the node n belonged the route p. Finally, α defines the degree of agility (or response) for reacting to network topology changes.

$$\begin{cases} x_n - \gamma \cdot \Delta_{x_n}, & \text{if } ||\gamma \Delta_{x_n}|| \leq \gamma_r. \\ x_n - \gamma_r \frac{\Delta_{x_n}}{||\Delta_{x_n}||}, & \text{otherwise.} \end{cases} \tag{18}$$

$$\Delta_{x_n}(X, P) = \sum_{p \in P} \sum_{v \in N_n(p)} \left[\alpha ||x_n - x_v||^{\alpha-2}(x_n - x_v)\right] \tag{19}$$

The algorithm (2) describes the execution of the adjustment method. The input parameters are the nodes' positioning (x_c, X_R and X_I), and the current routing policy P. On its turn, the output of this algorithm is the desired positions of the CN x_c and the RNs X_R. As shown, the method will compute each position - considering the CN and the RNs - individually according to (18) and (19).

Algorithm 2 Adjustment Algorithm

```
Input: $x_c$, $X_R$, $X_I$, and $P$.
Output: $x^*_c$ and $X^*_R$.
for each node $n \in c \cup R$ do
    Find $x^*_n$ according to (18) and (19).
end
```

In this work, β represents the adjustment procedure, as shown in (20). The function returns the ideal position of the CN x^*_c and the RNs' location X^*_R, having the current nodes' location X and the active routes P as entries.

$$\{x^*_c, X^*_R\} = \beta(X, P) \tag{20}$$

4) NODES ALLOCATION

Towards the topology management, the first step for the CN is to construct the network topology. As seen earlier, after that, the topology is then periodically adapted by using the adjustment methodology. However, at some point, the CN may need to alter the topology by performing the construction algorithm once again. In both cases, after having constructed the topology, the current formation needs to be remodeled to the newest proposed formation. In order to perform that task, the RNs need to be assigned.

In this work, the selection policy takes into account the current nodes' location and the future formation, in a way that the closest node will be allocated. Aiming to perform the assignment, the strategy is to consider that $D_{|R| \times |R|}$ is a matrix that contains all combinations of distances from the current nodes to the desired locations. Considering that $||x_u - x_v||$ represents the distance from node v to node u, (21) depicts the content of matrix D.

$$D_{u,v} = \{||x_u - x_v||\}_{u,v \in R} \tag{21}$$

The idea is then to identify interactively the minimum value of the matrix D. The element which contains the minimum value reveals the closest relation from the current node to its future location. After finding it, both the column and the line of this element need to be discarded. For the next iterations, the algorithm will continue by searching for the minimum value of those elements that are still valid. As a consequence, the algorithm is able to identify the closest relation between current nodes' location and the desired formation.

Equation 22 presents λ as the function responsible for returning the ideal and sorted nodes' location X_R^* based on the current location X_R and the desired location X_R' - given in a unsorted sequence.

$$\{X_R^*\} = \lambda(X_R, X_R') \tag{22}$$

Moreover, after the node allocation algorithm has been processed, the CN is able to estimate how long should the RNs take in order to completely recompose the topology. The prediction is taken by the longest distance and the mobility model of the RN. The time needed in order to have all the RNs at the desired position is given by δ.

5) CLEANING ROUTES

In order to evaluate the current routing strategy having the nodes' location updates, the CN should be able to check whether the topology changes are not causing any damage for the network performance. Thus, the cleaning procedure aims to discard routes that might not be operating properly. The result of this method is not directly informed to the other nodes. This outcome is used to detect whether the CN should act in order to reestablish connectivity among the nodes.

Having a set of routes, the algorithm will select those ones that the distance between consecutive nodes are not greater than a threshold value given as parameter. Equation (23) represents this method considering X as the current position of all the nodes, P as the set of the routes, and d_r as the limit distance (threshold) for cleaning. The response will be a set of cleaned routes P'.

$$P' = \gamma(X, P, d_r) \tag{23}$$

6) INTEGRATION

The CN is responsible for combining both construction and adjustment algorithms in order to provide a communication link among the INs for as long as possible. As a consequence, the integration algorithm must regulate the network as fast as possible aiming to react quickly to the mobility of the INs.

Before performing either the construction or adjustment algorithm, the current nodes' positioning is estimated based on their last contextual information that was successfully received by the CN. Therefore, for each node, the CN uses the trajectory contained in the last received message along with the timestamp, in order to predict its current location. By doing that, even though some control packets have not been received, the CN is able to cover such missing information and the network is not significantly affected by that.

The integration algorithm is a combination of two main stages, which are: initialization and loop statement. The initialization is performed once, as soon as the controller node has just started. This stage is responsible for the initial topology formation, as well as defining its routing policies for forwarding both control and data packets. After that, the loop statement algorithm is a while-true operation, which is performed throughout the entire operation. This second phase is in charge of evaluating and correcting the current topology configuration.

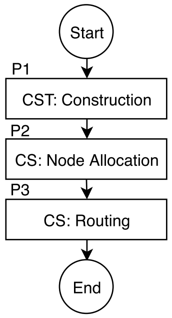

FIGURE 3. **Flowchart presenting the initialization phase of the integration algorithm including the descriptions of each step. In the flowchart, CS and CST stand for "Compute and Set" and "Compute and Store", respectively.**

Figures 3 and 4 present the flowchart which depicts the initialization algorithm. The first (Figure 3) describes the process, while the later (Figure 4) shows the equations regarding each step of the flowchart. Similarly, both Figures 5 and 6 present the flowchart of the loop statement algorithm including the description and the equations, respectively. The initialization and the loop statement are better described individually in the following.

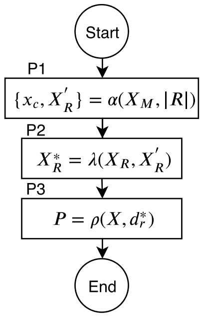

FIGURE 4. Flowchart presenting the initialization phase of the integration algorithm including the equations used in each step.

a: Initialization

The algorithm begins considering that every node is located in the range of the CN; even that they might not be in the range of each other. Firstly, the CN aims to plan the RNs' and its own position through the construction algorithm (P1). The RNs are then better allocated using the node allocation algorithm (P2). As soon as it is finished, the routing is computed using the routing strategy (P3). From this moment, the CN will keep monitoring the network in order to best update the routing strategies and RNs' location based on the collected information.

b: Loop statement

Once the topology is built and set, the CN is responsible for adjusting it according to the INs' movements. Knowing that the construction algorithm should be avoided, the adjustment algorithm is able to continuously adapt the network until there is a need to change the current topology.

The first aspect addressed in the loop statement is: has been passed the needed time δ since the last construction instant t_{last} until the present moment t_{now}? This question will guide the definition of which parameter should be used in order to accommodate the topology through the adjustment algorithm. In other words, this question will determine whether the adjustment algorithm will consider the current routing strategy (P5) or the one that is aimed when the nodes achieve at the desired position of the construction algorithm (P5 and P6).

After having just computed the adjustment of the topology, a checking procedure for evaluating the need for changing the network configuration is applied. The procedure starts at P7. At this point, the CN contains the current routing strategy P and the updated nodes' location X. The cleaning method is performed in order to internally discard the routes that have a link length greater than the ideal value of d^*_r. Having the cleaned routes P', the algorithm is able to evaluate whether the current nodes' locations have led the current solution to not reach the ideal number of routes n^* (Q2). In the case that the current solution is adequate, the network topology is not modified. However, if the current solution is not satisfactory, the algorithm attempts to solve the inefficacy by computing a routing strategy considering the current formation X (P8).

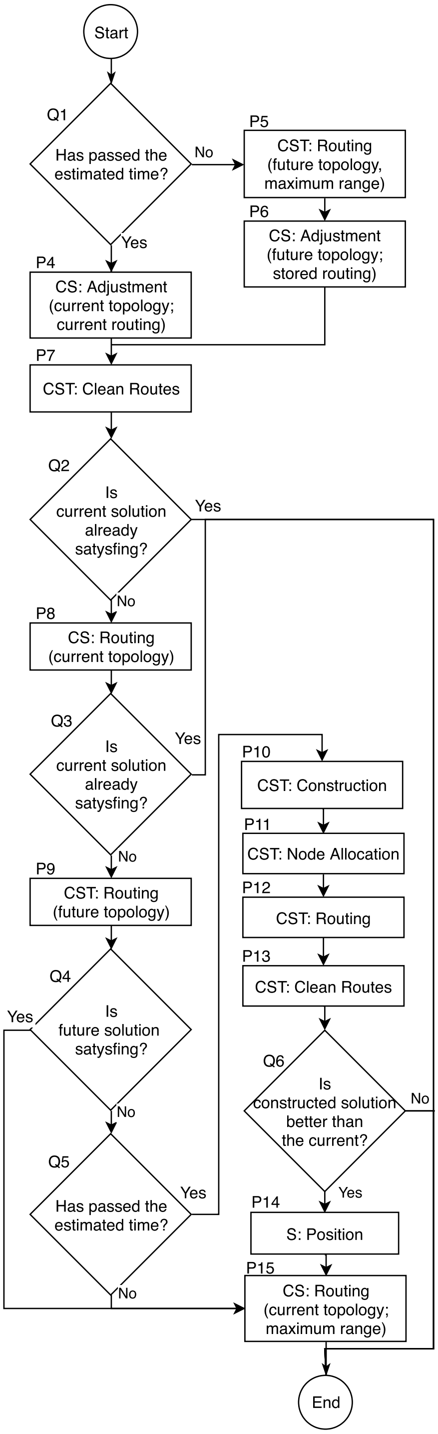

FIGURE 5. Flowchart presenting the loop statement phase of the integration algorithm including the descriptions of each step. In the flowchart, S, CS, and CST stand for "Set", "Compute and Set", and "Compute and Store", respectively.

The proposed routing solution needs to be evaluated. Thus, Q3 performs the same question as Q2, examining the current routing strategy P - which has been recently computed - with the ideal number of routes n^*. If the newest routing solution has solved the inefficacy, nothing else is either computed or modified.

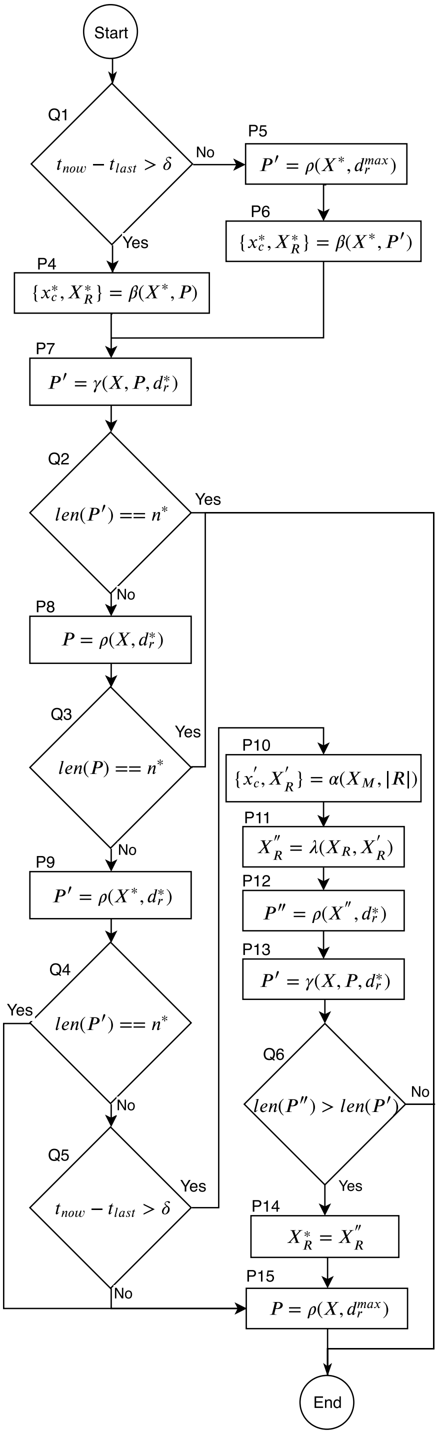

FIGURE 6. Flowchart presenting the loop statement phase of the integration algorithm including the equations used in each step.

The previous question Q3 may indicate that the current routing strategy may also not have been sufficient for having the desired connectivity. In this case, the algorithm should consider that the nodes may not have achieved their ideal location X^* by the moment. In order to evaluate the routing strategy considering that all nodes are hypothetically at their ideal locations, the routing algorithm is performed once again (P9). The resulting routing strategy P' is temporary and it is considered just to check its efficiency through question Q4. Having concluded that nodes' location targets lead to the desired connectivity, a temporary routing strategy is performed (P15). At this time, it is considered the current nodes' position X and the maximum link length d_r^{max} in order to have a temporary establishment of the network until the nodes do not reach their target position. On the other hand, the question Q5 may indicate that achieving the nodes' location targets does not lead to the desired network connectivity. In this case, the algorithm will start computing the construction algorithm (P10).

Similar to the initialization process, the computation algorithm is performed and the resulted topology is stored (P10). As soon as it is finished, having the current RNs' location X_R and the unsorted targets X_R^*, the algorithm needs to determine which RN is assigned to each desired location by setting a temporary solution X_R'' at P11. After completing this procedure, the CN needs to evaluate whether the recent computed result will overtake the current formation. As a consequence, the CN computes and stores the predicted routes having the desired positions at P12. In addition, the CN cleans the routes that the link length is greater than the ideal value of d_r^* at P13. The comparison between both solutions (Q5) will lead the CN to choose between keeping the current solution or setting the recent computed solution. Having the decision made to adopt the computed solution, the CN sets the RNs' target positions at P14, as well as set a temporary routing scheme until the RNs achieve their ideal positions (P15).

7) TIME COMPLEXITY ANALYSIS

The computational performance of both construction and adjustment algorithms can be evaluated through their time complexity analysis, which is presented in Table 4. In the following table, $|R|$ and $|I|$ stands for the number of relay and independent nodes, N_Ψ and N_k stands for the number of particles and iterations of the PSO algorithm and $|P|$ is the number of routes previously defined in Equation 4.

TABLE 4. Time complexity analysis of both construction and adjustment algorithms.

Algorithm	O(.)
Construction (PSO algorithm)	$O(N_k * N_\Psi * (\|I\| + \|R\|)^2)$
Adjustment (Gradient function)	$O(\|R\| * \|P\|)$

In O notation terms, the construction algorithm presented in Section IV-B2 can be expressed as $O(N_k * N_\Psi * (|I| + |R|)^2)$, which becomes $O(N^4)$. As it is computationally intensive and implies a high computation cost if performed frequently, a lighter algorithm is generally preferred. The adjustment algorithm, presented in Section IV-B3, can be defined as $O(|R| * |P|)$ in O notation, which becomes $O(N^2)$.

V. RESULTS

Simulations of the proposed STFANET coordination protocol were performed using the Mobile Multi-Media Wireless

Sensor Network (M3WSN) OMNet++ framework [33]. The OMNet++ is a network simulator for implementing and testing novel solutions. This work focus on the implementation of the STFANET in the network layer. Therefore, by using this framework, it is possible to gather valuable results considering the already modeled physical layer constraints. In addition, by being a discrete event simulator, there is no need to present the processing capabilities of the computers used for running the simulations. This means that the simulation processing times do not affect the results.

A campaign of 70 independent simulation runs was conducted, varying the INs' movements and transmissions. As evaluation metrics, packet loss, latency, and connectivity index were assessed, presenting the mean and the confidence interval of 95%. These metrics were evaluated by comparing the variation of the number of RNs contained in the FANET.

The following metrics are considered to evaluate the network reliability in terms of loss of packets, latency, and connectivity.

- Packet loss (%): rate of non-delivered packets to the final destination over the number of sent packets from the source.
- Latency (ms): period between the transmission of a packet from its source to its delivery in the final destination.
- Connectivity (%): rate of time that all nodes are capable of transmitting a message to any other node throughout the simulation time.

A. SCENARIO

This article considers a scenario with sets of 3, 6, 9, 12 and 15 RNs in order to provide connectivity among 5 INs moving randomly through a flat terrain of 10km x 10km. Each IN transmits a data packet to a random destination in intervals that varies from 100ms to 200ms. The INs are able to move from 5 to 10m/s, while the RNs and the CN are considered to be able to move at a maximum speed of 20m/s. The simulation was performed for 10 minutes long, which was considered to be enough in order to have the nodes sufficiently spread through the environment, as they start at the center of the scenario. All nodes can transmit data in a communication range of around 1000m. The coordination protocol also needs to be configured. The nodes are set to transmit their contextual information to the controller in intervals of 250ms. The CN adjusts or reconstructs the topology in intervals of 500ms. The simulation parameters were also set to allow wireless channel temporal variations, link asymmetry, and irregular radio ranges, as expected in a real FANET scenario. UAVs rely on the CSMA/CA MAC protocol, without using RTS/CTS messages and retransmissions. In case of buffer overflow, the UAVs consider a drop tail mechanism to drop packets.

The parameters referred to both scenario and STFANET coordination protocol are presented in Table 5. Those which are manipulated in mathematical expressions included in this paper are presented along with their respective symbols.

TABLE 5. Simulation parameters.

Simulation Parameters	Value
Scenario	
Time	10min
Dimension	10 x 10km
Number of independent nodes	5
Number of relay nodes	[3, 6, 9, 12, 15]
Independent nodes' speed	[5..10]m/s
Relay nodes' speed	20m/s
Protocol	
Beacon interval	250ms
Topology interval	500ms
Distances	
Minimum distance (d_s)	50m
Ideal link length (d_r^*)	700m
Maximum link length (d_r^{max})	1000m
Construction	
Number of particles (N_Ψ)	30
Number of iterations (N_k)	300
Stop threshold (N_k^t)	10
Inertia weight (w)	0.7
Cognitive parameter (c_1)	1.5
Social parameter (c_2)	1.5
Threshold for velocity clamping (V_{max})	20%
Sum of link length weight (σ)	0.5
Ideal link distance weight (λ)	0.3
Safety distance weight (μ)	0.3
Adjustment	
Positive step size (γ)	0.05
Maximum travel distance (γ_r)	2
Power of link distance (α)	5

In the following, the control plane includes messages being transmitted between each node and the CN for FANET maintenance proposes. On its turn, the data plane includes exclusively messages being transmitted among INs and originated from the application layer.

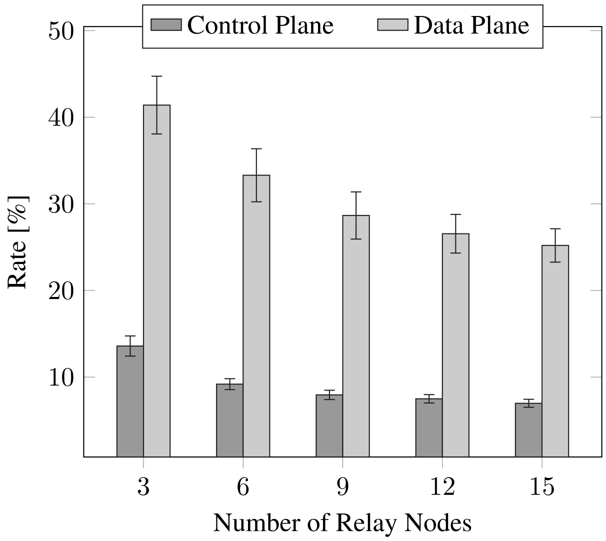

FIGURE 7. Packet loss for control and data plane.

B. PACKET LOSS

Figure 7 presents the mean and the confidence interval of the packet loss rate for the control and data plane. As expected, the increase of the number of RNs in the FANET resulted in a decrease of the packet loss in both planes. This is due to the greater number of connections and the smaller link length needed as more RNs are available. The rate dropped 7.7% (from 14% to 7%) in the control plane and 27.7% (from 42% to 25%) in the data plane. In addition, the overall best performance of the control plane against the data plane is due to the ability of the CN to address the RNs' positions, and therefore keep them connected even that the INs are not reachable.

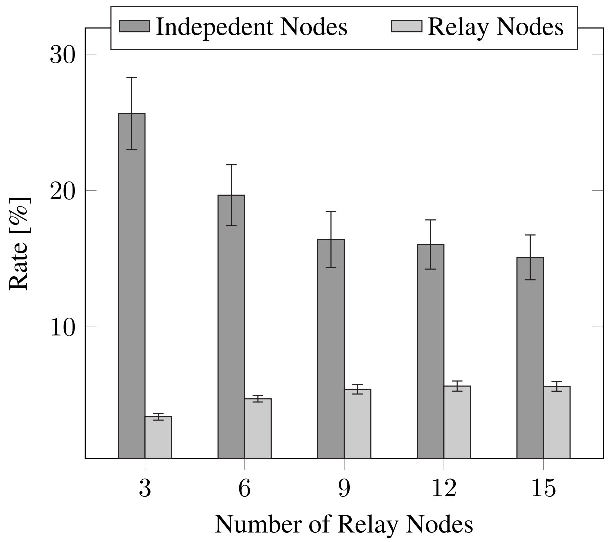

FIGURE 8. **Packet loss for control plane by relay and independent nodes.**

Specifically evaluating the control plane, Figure 8 categorizes the packet loss rate of the control plane by the CN is communicating either with an IN or an RN. The graph shows that the packet loss rate for RNs is consistently lower than for INs. As can be seen, there was a fall of 11% (from 26% to 15%) in the packet loss rate for the INs. As greater was the number of RNs, they were able to contribute better to form links between the INs and the CN. On the other hand, having control packets being delivered through one or more hops as increases the number of RNs in the FANET, there was a slight rise of 3% (from 3% to 6%) in the packet loss considering the RNs, exclusively.

In both Figure 7 and Figure 8, the interval confidence decreases with the increase in the number of RNs in the FANET. For instance, considering the messages being transmitted in the data plane, the confidence interval decreased by 42.2%. On its turn, in the control plane, it decreased by 60.2%. Hence, the results show that the increase in the number of RNs implies not only in the packet loss rate rise but also less variability among the simulations.

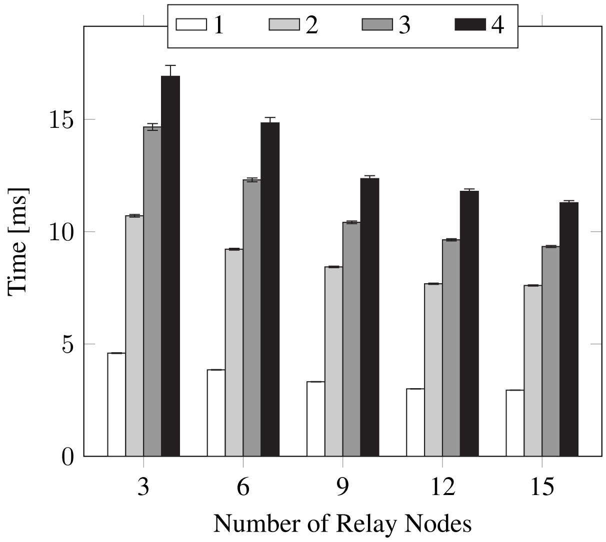

FIGURE 9. **Latency for control plane and number of hops.**

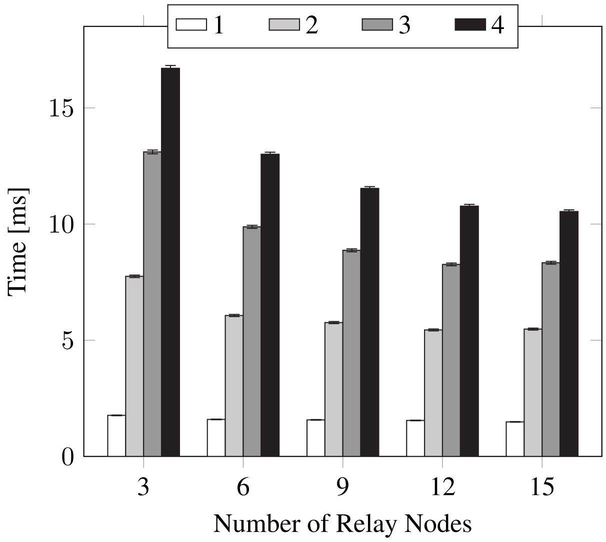

FIGURE 10. **Latency for data plane and number of hops.**

C. LATENCY

Figures 9 and 10 present the latency of delivered messages being transmitted through the data plane and the control plane, respectively. In addition, the figures categorize the measure according to the number of hops used for the delivery of the messages. For each number of RNs used in the simulation, it is presented the latency on average for the messages that used from 1 to 4 hops in order to be successfully transmitted. The following numbers of hops are not shown here, as there was no significant change in the behavior of the graphs.

As can be seen in Figures 9 and 10, there was a decrease of the latency for delivered packets as larger was the number of RNs in the FANET - either for the control plane or for the data plane. This behavior can be explained by the possibility of having the nodes even closer to each other as there was an increase in its quantity. As a consequence, the latency decreases significantly.

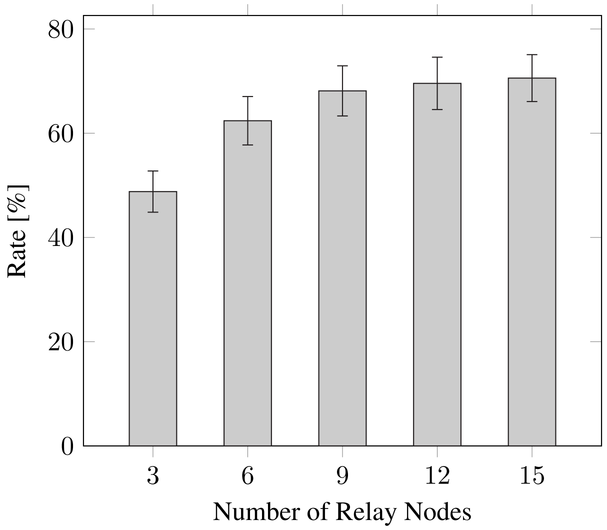

FIGURE 11. **Connected period ratio.**

D. CONNECTIVITY

The connectivity can be measured as the ratio of time that all the INs are able to communicate with each other. Having traced the routing tables' rules and nodes' positions through the simulation, it was possible to measure the period of time that all the INs were virtually connected - even that they were not in fact communicating. By doing this, as presented in Figure 11, the simulation showed that the connectivity rose 44.6% as it was from 49% with 3 RNs to 71% having 15 RNs available in the FANET. As expected, as greater was the number of available RNs, the network could keep the INs connected for a longer period of time.

VI. CONCLUSION AND FUTURE WORK

This article proposes a novel coordination protocol, which includes an SDN-based UAV communication for routing and topology management of flying ad hoc networks, named STFANET. Its primary goal is to establish the topology and to adjust it in order to keep the connection between the independent nodes (nodes accomplishing a given mission) through the relay nodes. STFANET passed through a comprehensive set of tests on OMNet++ to evaluate its efficiency.

As expected, the FANET has shown to provide higher network performance indicators - considering packet loss, latency, and connectivity -, as the number of relay nodes available is increased. Nevertheless, that raise in performance halts as soon as the number of relay nodes has passed a certain value. The aforementioned evaluation would be necessary for determining a sufficient and minimum quantity of relay nodes for providing the desired network performance in a specific scenario.

Considering the proposed STFANET, the FANET was also able to deliver over 90% of control packets. Since it was expected to meet those losses due to wireless limitations and nodes' frequent mobility changes, the controller's ability to predict nodes' positioning was also considered. Therefore, although control packets would not be successfully transmitted, the controller is able to predict nodes' location by taking their most recent transmission data. As a result, the proposed protocol was capable of maintaining the network connection for at least 75% of the time, considering the adopted experimental scenario. Such outcomes prove a promising direction of the solution for accomplishing its purposes.

As future work, it is planned to enhance the strategy used by the controller node to determine the relay nodes positioning and the nodes' routing policies, by using a more detailed set of information - for instance, energy resources and their individual capabilities. In addition, statistical inference would significantly improve the work, specifically in estimating nodes' positioning. Another possible direction for future work is to mix decentralized strategies of network connectivity control, which can be beneficial for some specific types of missions.

ACKNOWLEDGMENT

Firstly, the authors would like to offer special thanks to Dr. Mario Gerla, who, although no longer with us, continues to inspire by his example and dedication to the students he served over the course of his career. The authors would like also to thanks Maik Basso, MSc. for his valuable assistance that greatly improved this article. Finally, the authors credit Freepik for an image that was used to build Figure 2.

REFERENCES

[1] M. Półka, S. Ptak, and L. Kuziora, "The use of UAV's for search and rescue operations," *Procedia Eng.*, vol. 192, pp. 748–752, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877705817326759

[2] Y. Karaca, M. Cicek, O. Tatli, A. Sahin, S. Pasli, M. F. Beser, and S. Turedi, "The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations," *Amer. J. Emergency Med.*, vol. 36, no. 4, pp. 583–588, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0735675717307507

[3] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, J. Ashdown, and K. Turck, "A solution for dynamic spectrum management in mission-critical UAV networks," pp. 1–10, Apr. 2019, *arXiv:1904.07380*. [Online]. Available: http://arxiv.org/abs/1904.07380

[4] S. Vashist and S. Jain, "Location-aware network of drones for consumer applications: Supporting efficient management between multiple drones," *IEEE Consum. Electron. Mag.*, vol. 8, no. 3, pp. 68–73, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8684812/

[5] F. De Rango, G. Potrino, M. Tropea, A. Santamaria, and P. Fazio, "Scalable and ligthway bio-inspired coordination protocol for FANET in precision agriculture applications," *Comput. Elect. Eng.*, vol. 74, pp. 305–318, Mar. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0045790618323644, doi: 10.1016/j.compeleceng.2019.01.018.

[6] V. Sharma and R. Kumar, "G-FANET: An ambient network formation between ground and flying ad hoc networks," *Telecommun. Syst.*, vol. 65, no. 1, pp. 31–54, 2017.

[7] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2015.

[8] Z. M. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "A dynamic trajectory control algorithm for improving the communication throughput and delay in UAV-aided networks," *IEEE Netw.*, vol. 30, no. 1, pp. 100–105, Jan. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7389838/

[9] D.-Y. Kim and J.-W. Lee, "Joint mission assignment and location management for UAVs in mission-critical flying ad hoc networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 323–328. [Online]. Available: https://ieeexplore.ieee.org/document/8539355/

[10] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, 2013.

[11] A. Kott, A. Swami, and B. J. West, "The Internet of battle things," *Computer*, vol. 49, no. 12, pp. 70–75, Dec. 2016.

[12] G. Choudhary, V. Sharma, and I. You, "Sustainable and secure trajectories for the military Internet of Drones (IoD) through an efficient medium access control (MAC) protocol," *Comput. Elect. Eng.*, vol. 74, pp. 59–73, Mar. 2019.

[13] A. Kott and D. S. Alberts, "How do you command an army of intelligent things?" *Computer*, vol. 50, no. 12, pp. 96–100, Dec. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8220476/

[14] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military UAV relay networks," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 804–807, Apr. 2016.

[15] P. Basu, J. Redi, and V. Shurbanov, "Coordinated flocking of UAVs for improved connectivity of mobile ground nodes," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, vol. 3, Oct. 2005, pp. 1628–1634. [Online]. Available: http://ieeexplore.ieee.org/document/1495182/

[16] J. Sánchez-García, D. Reina, and S. Toral, "A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario," *Future Gener. Comput. Syst.*, vol. 90, pp. 129–148, Jan. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X18303649

[17] D. G. Reina, S. L. Toral, and H. Tawfik, "UAVs deployment in disaster scenarios based on global and local search optimization algorithms," in *Proc. 9th Int. Conf. Develop. eSyst. Eng. (DeSE)*, Liverpool, U.K., Aug. 2016, pp. 197–202.

[18] J. Scherer, B. Rinner, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, and H. Hellwagner, "An autonomous multi-UAV system for search and rescue," in *Proc. 1st Workshop Micro Aerial Vehicle Netw., Syst., Appl. Civilian Use (DroNet)*, 2015, pp. 33–38. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2750675.2750683

[19] S. Hauert, L. Winkler, J.-C. Zufferey, and D. Floreano, "Ant-based swarming with positionless micro air vehicles for communication relay," *Swarm Intell.*, vol. 2, nos. 2–4, pp. 167–188, 2008.

[20] R. Magán-Carrión, J. Camacho, P. García-Teodoro, E. F. Flushing, and G. A. Di Caro, "A dynamical relay node placement solution for MANETs," *Comput. Commun.*, vol. 114, pp. 36–50, Dec. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0140366417304619, doi: 10.1016/j.comcom.2017.10.012.

[21] D.-Y. Kim and J.-W. Lee, "Integrated topology management in flying ad hoc networks: Topology construction and adjustment," *IEEE Access*, vol. 6, pp. 61196–61211, 2018.

[22] P. Cumino, W. Lobato Junior, T. Tavares, H. Santos, D. Rosário, E. Cerqueira, L. Villas, and M. Gerla, "Cooperative UAV scheme for enhancing video transmission and global network energy efficiency," *Sensors*, vol. 18, no. 12, p. 4155, 2018.

[23] R. Kirichek, A. Vladyko, A. Paramonov, and A. Koucheryavy, "Software-defined architecture for flying ubiquitous sensor networking," in *Proc. 19th ICACT*, Feb. 2017, pp. 158–162.

[24] J. A. Wickboldt, W. P. D. Jesus, P. H. Isolani, C. B. Both, J. Rochol, and L. Z. Granville, "Software-defined networking: Management requirements and challenges," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 278–285, Jan. 2015.

[25] Z. Zhao, P. Cumino, A. Souza, D. Rosário, T. Braun, E. Cerqueira, and M. Gerla, "Software-defined unmanned aerial vehicles networking for video dissemination services," *Ad Hoc Netw.*, vol. 83, pp. 68–77, Feb. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1570870518306231

[26] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan. 2017.

[27] G.-H. Kim, J.-C. Nam, I. Mahmud, and Y.-Z. Cho, "Multi-drone control and network self-recovery for flying ad hoc networks," in *Proc. 8th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2016, pp. 148–150.

[28] R. S. Moraes and E. P. Freitas, "Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services," *J. Intell. Robot. Syst.*, vol. 92, pp. 645–656, Nov. 2017, doi: 10.1007/s10846-017-0726-z.

[29] S. Rosati, K. Kruželecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic routing for flying ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1690–1700, Mar. 2016.

[30] J.-J. Wang, C.-X. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, Sep. 2017.

[31] R. Magán-Carrión, R. A. Rodríguez-Gómez, J. Camacho, and P. García-Teodoro, "Optimal relay placement in multi-hop wireless networks," *Ad Hoc Netw.*, vol. 46, pp. 23–36, Aug. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1570870516300841

[32] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström, and P.-M. Olsson, "Relay positioning for unmanned aerial vehicle surveillance," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1069–1087, Jul. 2010. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364910369463

[33] D. Rosário, Z. Zhao, C. Silva, E. Cerqueira, and T. Braun, "An OMNeT++ framework to evaluate video transmission in mobile wireless multimedia sensor networks," in *Proc. 6th Int. ICST Conf. Simulation Tools Techn.*, 2013, pp. 277–284. [Online]. Available: http://dl.acm.org/citation.cfm?id=2512734.2512774

TULIO DAPPER E SILVA graduated in automation and control engineering from the Federal University of Rio Grande do Sul (UFRGS), Brazil, where he is currently pursuing the M.Sc. degree in electrical engineering. His research interests include the autonomous and intelligent systems, data science, wireless network, and software-defined networking.

CARLOS F. EMYGDIO DE MELO received the degree in computer engineering from the Federal University of Itajubá (UNIFEI), Brazil. He is currently pursuing the M.Sc. degree in electrical engineering with the Federal University of Rio Grande do Sul (UFRGS), Brazil. His current research interests include computer networks, wireless networks, flying ad hoc networks, cybersecurity, data analysis, and (multi)unmanned aerial vehicles.

PEDRO CUMINO graduated in computer engineering from the Federal University of Pará, Brazil, where he received the master's degree in computer science and is currently pursuing the Ph.D. degree in computer science. His current research interests include smart cities, flying ad hoc networks, quality of service, quality of experience, and software-defined networks.

DENIS ROSÁRIO received the Ph.D. degree in electrical engineering from the Federal University of Pará, Brazil, with a joint supervision undertaken by the Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland, in 2014. He is currently a Professor with the Federal University of Pará. His current research interests include multimedia adaptation, wireless networks, flying ad hoc networks, VANET, mobility, quality of experience, and software-defined networks.

EDUARDO CERQUEIRA received the Ph.D. degree in informatics engineering from the University of Coimbra, Portugal, in 2008. He is currently an Associate Professor with the Faculty of Computer Engineering, Federal University of Pará (UFPA), Brazil, and an Invited Researcher with the Network Research Lab, UCLA, USA, and the Centre for Informatics and Systems, University of Coimbra. His publications include five edited books, five book chapters, four patents, and more than 180 articles in national/international refereed journals/conferences. His research involves multimedia, the future Internet, quality of experience, mobility, and ubiquitous computing. He has been serving as a Guest Editor for six special issues of various peer-reviewed scholarly journals.

EDISON PIGNATON DE FREITAS received the Computer Engineering degree from the Military Institute of Engineering, in 2003, the M.Sc. degree in computer science from the Federal University of Rio Grande do Sul (UFRGS), in 2007, and the Ph.D. degree in computer science and engineering from Halmstad University, Sweden, in 2011. He is currently an Associate Professor with UFRGS, acting in the Graduate Programs on Computer Science (PPGC) and Electrical Engineering (PPGEE) developing research mainly in the areas of computer networks, real-time systems, and (multi)unmanned aerial vehicles.

• • •