Self-Organizing Aerial Mesh Networks for Emergency Communication

Marco Di Felice*, Angelo Trotta*, Luca Bedogni*, Kaushik Roy Chowdhury†, Luciano Bononi*
* Department of Computer Science and Engineering, University of Bologna, Italy
Emails: {difelice, trotta, lbedogni bononi}@cs.unibo.it
† Department of Electrical and Computer Engineering, Northeastern University, Boston, USA
Email: krc@ece.neu.edu

***Abstract*—Guaranteeing network connectivity in post-disaster scenarios is challenging yet crucial to save human lives and to coordinate the operations of first responders. In this paper, we investigate the utilization of low-altitude aerial mesh networks composed by Small Unmanned Aerial Vehicles (SUAVs) in order to re-enstablish connectivity among isolated end-user (EU) devices located on the ground. Aerial ad-hoc networks provide the advantage to be deployable also on critical scenarios where terrestrial mobile devices might not operate, however their implementation is challenging from the point of view of mobility management and of coverage lifetime. In this paper, we address both these issues with three novel research contributions. First, we propose a distributed mobility algorithm, based on the virtual spring model, through which the SUAV-based mesh node -called also Repairing Units (RUs) in this study- can self-organize into a mesh structure by guaranteeing Quality of Service (QoS) over the aerial link, and connecting the maximum number of EU devices. Second, we evaluate our scheme on a realistic 3D environment with buildings, and we demonstrate the effectiveness of the aerial deployment compared to a terrestrial one, in terms of coverage and wireless link reliability. Third, we address the problem of energy lifetime, and we propose a distributed charging scheduling scheme, through which a persistent coverage of RUs can be guaranteed over the emergency scenario.**

I. Introduction

In the aftermath of a large-scale emergency, the breakdown of communication infrastructure impacts the actions of the first responders, i.e. the dissemination of information to the general population [1], as demonstrated by recent catastrophic events worldwide (e.g. the earthquake in Italy in 2012). Thus, there is strong motivation towards the realization of backup communication systems that are able to quickly self-deploy in the aftermath of an emergency and ensure temporary network services in the affected area.

A recent report from FCC proposes the utilization of low-altitude, Deployable Aerial Communications (DAC) systems to support first response operations on post-disaster scenarios, due to their advantages over traditional terrestrial infrastructures [1]. DAC systems can guarantee higher coverage than ground wireless networks as aerial links are less affected from fading, and are more suited when road mobility has been compromised (e.g. flooding). Nowadays, deployment of low-altitude DAC systems is feasible thanks to the increasing availability and affordability of Small Unmanned Aerial Vehicles (SUAVs), such as quadcopters. However, when planning for the aerial coverage of large-scale emergency scenarios for at least the first 48 hours, coordinated mobility and energy issues must be taken into account [7]. Network mobility is known to be highly challenging in 3D environments, and only few works investigate the creation of flying self-organizing swarms, specially designed for rescue operations [2] [3] [4] [5] [6].

In this paper, we take into account both mobility and energy issues in the deployment of SUAV-based mesh networks for backup communication systems in post-disaster scenarios. More specifically, we consider an emergency scenario, where not all End-User (EU) devices are connected to each other, and the aerial mesh attempts to build the links between them. Three contributions are provided: (*i*) We extend our earlier distributed algorithm in [15] [16] in the context of a swarm of SUAVs, that allows them to self-organize into an aerial mesh to maximally connect the EUs on the ground. The mobility scheme is based on the Virtual Spring force model [14], and introduces channel-aware metrics in order to guarantee a minimum link quality on the air-to-ground and air-to-air links. (*ii*) Second, we model a realistic 3D urban environment in OMNET++, with shadowing effect caused by buildings, and investigate the benefits provided by an aerial mesh deployment compared to a terrestrial deployment (through mobile robots), in terms of coverage, link stability, and altitude. (*iii*) Third, we investigate approaches to maximize the lifetime of the aerial mesh by considering a scenario where SUAVs can recharge their batteries through contact with the ground station. A distributed scheduling algorithm is proposed that ensures a guaranteed coverage area by the SUAVs, while network lifetime is maximized.

II. Related Works

Recent studies have been focused on three research directions: characterization of aerial links, dynamic establishment of swarm flying structures, and task assignment in mission-critical scenarios [5]. Mathematical bounds for capacity, coverage and connectivity of nodes placed in 3D spaces are derived in [8]. However, this work does not consider several aspects of real scenarios, like the antenna orientation and the shadowing caused by buildings. For these reasons, testbeds have been used in [9] [10] to characterize the performance of two-hops aerial mesh networks. In [9], the path loss and the

978-1-4799-4912-0/14/$31.00 2014 IEEE

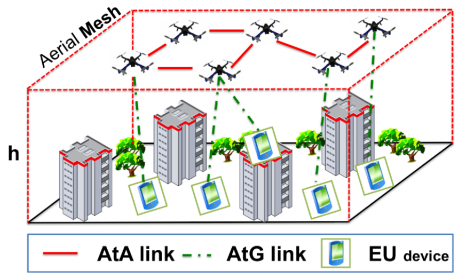


Fig. 1. The emergency scenario and the aerial mesh deployment.

small-scale fading on the air-to-ground links are investigated, and the advantages of multiple-antennas for 3D coverage are demonstrated [10]. When considering a fleet of autonomous SUAVs, swarm mobility is one of the main challenges. In [12] and [13], the authors propose network planning methodologies for aerial mesh networks (based on WI-MAX in [12] and on LTE in [13]), where the position of each SUAV is pre-computed on the basis of a coverage analysis. In [2], the authors compare the performance of three distributed mesh creation algorithms, considering the aspects of link reliability and the efficiency of spatial exploration. In [6], a channel-aware swarm mobility scheme is proposed using RSSI as the connectivity metric. In [4], the authors derive the mobility control laws through Lyapunov-like functions, and demonstrate the emergence of stable patterns of formations. In [7], the authors consider a scenario where fuel service stations are available on the ground, and formulate the problem of determining a persistent aerial coverage of SUAVs through orchestration of recharging operations, achieved through a centralized optimization algorithm.

III. System Model

We consider an urban, post-disaster scenario shown in Figure 1, with buildings of varying heights, and N End-User (EU) devices at outdoor/indoor locations. In the aftermath of the emergency, isolated partitions of EUs might emerge, who must be eventually connected through additional Repairing Units (RUs). Each RU is constituted by a SUAV (quadcopter), flying at an initial altitude of h meters, and equipped with a wireless radio transceiver to communicate with other RUs (air-to-air link) and to EU devices (air-to-ground link). Each RU is battery- powered, and consumes energy for flying and for radio communication. We assume RUs can recharge their batteries through fixed ground stations. Our work does not rely on specific wireless technologies, and for ease of exposition, we assume IEEE 802.11n wireless transceivers are used. The objective of this paper is to investigate energy-efficient distributed deployment strategies, so that RUs self-organize into an aerial mesh structure able to maximally connect the EU devices (shown in Figure 1). More formally, we say that two EU devices e_1 and e_2 are interconnected at time t whether there exists a path inside the aerial mesh from e_1 to e_2 passing through RUs. Let $E(t)$ the set of interconnected EU devices at time t. We define the Connectivity Index at time t (i.e. $CI(t)$) as $CI(t) = \frac{|E(t)|}{N}$. In practice, the CI metric quantifies the ability of the aerial mesh to re-establish the connectivity on the post-disaster scenarios.

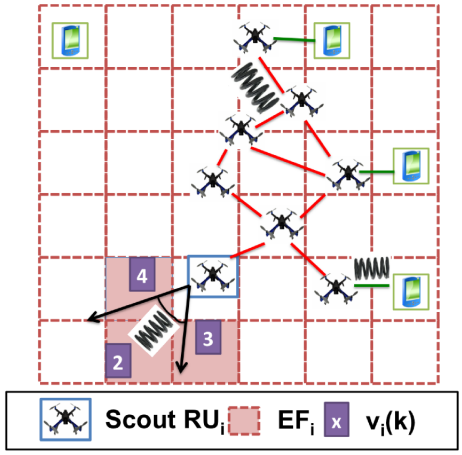


Fig. 2. The aerial mesh with virtual springs and scout RNs.

IV. Distributed Swarm Mobility

In our scenario, distributed mobility of RUs is subject to three rules: (*i*) it must preserve the connectivity of the aerial mesh, i.e. no partitions should occur, (*ii*) it must guarantee Quality of Service (QoS) over the air-to-air (AtA) and air-to-ground links (AtG), (*iii*) it must guarantee the CI metric (discussed earlier). Due to the challenges posed by 3D mobility [12] [13], we consider a simplified scenario in which all RUs move at the same known altitude (h), and thus we reduce the network deployment to a 2D case. In Section VII, we investigate the impact of the h parameter on system performance. The space is divided into a grid of GxG cells (Figure 2), and each RU is aware of its current cell through a GPS interface.

The distributed swarm mobility algorithm relies on the Virtual Spring Mesh scheme described in [14], and on the extension we provided in [15] [16] for the distributed creation and maintenance of dynamic (terrestrial) backbones. The algorithm in [15] [16] associates with each wireless link a virtual spring force, characterized by a natural length l_0 and a stiffness constant k, and acting according to the Hooke's law:

$$\vec{F} = -k \cdot (\vec{x} - l_0) \quad (1)$$

where $\vec{x}$ denotes the spring displacement. In our study, we consider three types of virtual spring forces that can act on each RU_i:

- *Air-to-Air* (AtA) Spring. The end-points of the spring are RU_i and another RU (e.g. RU_j) in the transmitting range of RU_i.
- *Air-to-Ground* (AtG) Spring. The end-points of the spring are RU_i and an EU device posed on the ground, which is thus connected to the aerial mesh through RU_i.
- *Air-to-Frontier* (AtF) Spring. The end-points of the spring are RU_i and the center of a cell in the 2D grid space. This force is introduced to enable space exploration by special RUs (called *scout* RUs), as better explained in Section IV-A.

We assume that each disconnected EU device periodically transmits a HELLO message, with its position and identifier in order to enable its localization from the RUs. Similarly, while flying, each RU_i broadcasts a BEACON message on the Common Control Channel (CCC) in the 2.4 GhZ band every T_f intervals containing its id, position, the number of EU devices currently connected to (n^i_{EU}), and the set of neighbor RUs of RU_i ($Neigh_i$).

In the following, we explain how the k, l and $\vec{x}$ parameters are defined for the AtA and AtG links. The AtF link case is described in Section IV-A. In [14], displacement of virtual spring is defined in terms of spatial distance among the end-points. Conversely, we propose a formulation of the displacement that reflects the communication quality of the AtA/AtG link in terms of Link Budget (LB). More specifically, once receiving a BEACON message from RU_j or an HELLO message from EU_j, RU_i computes the Link Budget of the link $i \leftrightarrow j$ (i.e. $LB(i,j)$) as follows:

$$LB(i,j) = Pr^i_j - RS^i_{thr} \tag{2}$$

where Pr^i_j is the receiving power at RU_i, and RS^i_{thr} its receiving sensitivity. The LB metric, also called fading margin, provides an indication of the communication reliability (i.e. it tells when the link is going to break), and at the same time indicates the maximum achievable rate that can be offered on that link. We introduce the requested link budget (LB_{req}) to express the Quality of Service (QoS) requirements which must be guaranteed on each link of the aerial mesh. Then, we formulate the displacement ($\vec{x} - l_0$) as a function of the requested and current LB on the $i-j$ link, which can be derived from propagation models as:

$$\delta = \sqrt[\alpha]{\frac{max(LB(i,j), LB_{req})}{min(LB(i,j), LB_{req})}} - 1 \tag{3}$$

Here, α is the propagation decay exponent (equal to 2 in our scenario). We observe for:

- *AtA link*. In this case k_{AtA} is a fixed parameter in the range [0:1] (0.5 in our experiments). If the spring displacement expresses the requested link quality, the value of k_{AtA} defines the system responsiveness, i.e. how quickly RUs will act in order to meet the QoS value (LB_{req}).
- *AtG link*. In this case, k_{AtG} is dynamically adjusted on the basis of number of EUs connected to RU_i (i.e. n^i_{EU}) as follows:

$$k_{AtG} = \frac{n^i_{EU}}{max(n^j_{EU})\ \forall j \in Neigh_i} \tag{4}$$

  In practice, the stiffness (and thus the force module) of the AtG spring is proportional to the number of connected EUs at the current location, and scaled on the basis of the CI of neighboring nodes. As a result, a RU connecting more EUs at the current location will oppose more resistance to move than its neighbors.

Based on its connection with other neighboring RUs, and discovered EU devices, multiple virtual forces $\vec{R_0}$, $\vec{R_1}$ ... $\vec{R_n}$ act on each RU. Every T_{DEC} intervals, each RU_i computes the resultant force $\vec{R}$ defined as $\vec{R} = \sum_{i=0}^{n} \vec{R_i}$, and moves in the direction indicated by $\vec{R}$, with constant speed. To avoid fluctuations we define two additional mobility mechanisms: (*i*) a minimum threshold R_{thr} is introduced, so that an RU will change its position only if $|\vec{R}| > R_{thr}$, and (*ii*) before moving on the direction indicated by $\vec{R}$, each RU verifies through the CI metric whether any breakage of AtA link will occur, and if so, it does not update its position.

A. Exploration phase

This phase allows the RUs to locate the EU devices and connect them to the mesh. While poor exploration might translate into suboptimal CI performance, uncoordinated mobility of RUs can lead to the partitioning of the aerial mesh. To address both the connectivity and performance issues, our mobility scheme delegates the exploration phase to a dynamically selected set of special nodes, called Scout RUs, which are placed on the edges of the aerial mesh. A virtual spring force (with length equal to 0, and thus attractive only) is used to drive RUs towards less explored cells of the grid. More specifically, the exploration procedure involves four steps:

- *Scout selection.* Every T_{SCOUT} time instants, each RU_i checks its position compared to its neighbour RUs. In case R_i does not detect any other RU in its visilbility zone (defined as a cone with sweep angle of θ centered in RU_i), then it self-elect as scout node, with probability equal to p_{SCOUT}.
- *Direction selection.* We assume that each RU_i keeps statistics about the number of times it has visited each cell j of the scenario (i.e. $v_i(j)$). The cell value $v_i(j)$ is incremented by 1 each time RU stands on cell j for a minimum duration interval (equal to 10 second in our experiments). We consider the exploration frontier of RU_i (EF_i) as the set of cells located inside the intersection of the square centered on the current cell, and of side equal to $2 \cdot h + 1$, and the visibility zone of R_i. In Figure 2, we depict the EF zone with h=1 and θ=30. In practice, the parameter h defines the horizon of the exploration phase, i.e. how far RU_i is looking at when deciding its next position. Then, RU_i selects the cell j that has been less visited in EF_i, i.e. $j = argmin\ v_i(k) | \forall k \in EF_i$.
- *Force computation.* Once the next cell j has been determined, a virtual AtF spring is built between RU_i and the center of cell j. The spring displacement is fixed and equal to a reference value LB_{min}. Vice versa, the stiffness constant k_{AtF} is adjusted on the basis of the amount of exploration performed on cell j, i.e.:

$$k_{AtF} = \left(1 - \frac{v_i(j)}{v_i^{Max}}\right)^{v_i^{min}+1} \tag{5}$$

where v_i^{Max} and v_i^{min} are respectively the maximum and minimum cell values for RU_i, considering all cells of the scenario, i.e. $v_i^{Max} = max(v_i(j)), \forall j \in G \times G$ and $v_i^{min} = min(v_i(j)), \forall j \in G \times G$. Through Equation 5, we express

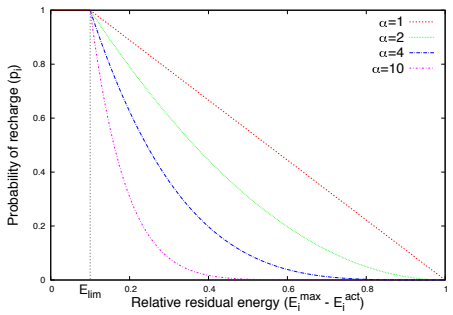


Fig. 3. The p_i^{sr} function for different values of α_i.

the intuitive concept that RU_i should be more attracted by cells that has been explored less than the average, and that the exploration force decreases when the discovery ratio of the full scenario (i.e. v^{min}) increases.

V. Distributed Charging Scheduling

In this section, we propose a distributed algorithm to let RUs autonomously decide when to recharge. The algorithm is designed based on the following requirements: (*i*) it attempts to preserve the CI, by giving precedence to RUs whose departure will not cause the partitioning of the aerial mesh, and (*ii*) it accounts for the recharging need of each RU based on its residual energy. To fulfill these requirements, we propose a probabilistic approach, in which each RU_i is assigned a probability p_i of recharging, defined as:

$$p_i(E_i^{act}, E_i^{max}) = \left(\frac{E_i^{max} - E_i^{act}}{E_i^{max}} \right)^{\alpha_i} \tag{6}$$

where E_i^{max} and E_i^{act} are, respectively, the maximum and the actual residual battery capacity of RU_i. The exponent $\alpha_i \geq 1$ modifies the probability p_i by reflecting the cost incurred by the departure of RU_i, as discussed below. To avoid the case of a RU having insufficient energy to reach the nearest charging station, we introduce a threshold value E^{lim} on the residual energy, below which the RU *must* go for recharging. Hence, the probability p_i is adjusted as follows:

$$p_i^{sr} = \begin{cases} p_i(E_i^{act} - E^{lim}, E_i^{max} - E^{lim}) & \text{if } E_i^{act} > E^{lim} \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

In Figure 3 we plot the p_i^{sr} values, for different configuration of α_i. It is easy to notice that p_i^{sr} decreases with the current state of charge of RU_i (E_i^{act}), and that -for the same value of E_i^{act}- the probability is lower for higher values of α_i. This latter is defined as follows:

$$\alpha_i = \alpha_i^{critical} + \alpha_i^{EU} + 1 \tag{8}$$

The first component $\alpha_i^{critical}$ reflects the connectivity degree of RU_i, and is defined as follows:

$$\alpha_i^{critical} = \alpha^{maxC} \cdot (1 - e^{-\#numclusters}) \tag{9}$$

Here, α^{maxC} bounds the maximum value of $\alpha_i^{critical}$ (equal to 12 in our experiments) and $\#numclusters$ is defined as the number of the potential clusters that might be formed if the RU_i disconnects from the mesh. This value is computed by determining the rank of the matrix A^{Neigh_i}, where A is the adjacency matrix of RU_i [1]. The second component α_i^{EU} is specific to the AtG link, and reflects the importance of RU_i in terms of EU devices currently interconnected:

$$\alpha_i^{EU} = \begin{cases} \alpha^{maxEU} \cdot k_{AtG} & \text{if } n_{EU}^i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where α^{maxEU} bounds the maximum value of α_i^{EU} (equal to 3 in our experiments) and k_{AtG} is the stiffness constant of the spring for AtG link defined by Equation 4. Basically, through (9) and (10), the probability p_i^{sr} is discounted by considering: (*i*) the potential clusters that might occur, and (*ii*) the isolated EU devices that might be left.

Every $T_{recharge}$ seconds, each RU_i decides with probability p_i^{sr} whether to go to recharge or not. We assume a linear charging model, i.e. the charging time CT_i is computed as: $CT^{r\_max} \cdot \left(\frac{E_i^{max} - E_i^{act}}{E_i^{max}} \right)$, where CT^{r_max} is the time required by a full battery recharge. After charging is completed, the RU resumes its operation.

VI. 3D Scenario Modeling

In this Section, we detail how the 3D scenario has been modeled through the Omnet++ tool. We consider rectangular-shape buildings, with varying dimensions on the 3 axes. Although in the evaluation (Section VII) we consider synthetic generated scenarios, realistic scenarios can also be modeled through our tool, by importing the XML maps provided by OpenStreetMaps with building information. Modeling the wireless propagation effects in 3D environments is highly-challenging, and owing to the complications of creating accurate ray-tracing models, we consider a simplified 3D propagation model that takes into account the attenuation effect caused by buildings on the line of sight. Although this model cannot capture the complex reflection/diffraction phenomena, its suitability to guarantee good approximation for large-scale network simulations has been demonstrated in [18]. More specifically, our algorithm works in three steps:

- First, we consider the Line-of-Sight (LOS), i.e. the straight line between the sender and the receiver (which can be two RUs or a RU and an EU device).
- Then, we determine all the points in which the straight line collides with an obstacle.
- Finally, we apply an attenuation factor to the received signal, based on the the length of the indoor path and on the number of intersected outdoor walls.

The pathloss (in dB) is modeled as $PL[d] = \alpha \cdot 10 \cdot log_{10}(d(i,j)) + \beta$, where $d(i,j)$ is the 3D distance between node i to node j, α is the propagation exponent (fixed to 2 in our case, i.e. a free-space model is considered) and β is a zero-mean Gaussian distributed random variable with standard

[1]The $\#numclusters$ value takes into account only the 2-hop neighbors of RU_i, and thus, it provides a local approximation of disconnected components. We omit details on how the $\#numclusters$ is computed.

deviation σ (in dB). By computing the intersection points between the LOS and the faces of a building b, we derive the length d'_b, in meters, in which the signal travels indoor. We thus model the signal attenuation as follows:

$$S_b = 2 \cdot k + d'_b \cdot \eta \tag{11}$$

where k is the attenuation factor due to the outside walls, and η is the attenuation related to indoor propagation through dry walls, furniture etc. The parameters k and η are set to 20dB and 1dB/mt, based on literature surveys. Finally, given B the set of buildings of the scenario, we compute the power of received signal as $P_{rx} = P_{tx} - PL[d] - \sum_{b=0}^{B} S_b$.

In order to further reduce the computation, we consider the cell-grid world depicted in Figure 2, and we reduce B to $B' \subseteq B$, i.e. the set of buildings that are on the cells intersected by the LOS between nodes i and j.

VII. Performance Evaluation

In this Section we evaluate the performance of the distributed mobility scheme (Section IV) and of the charging scheduling algorithm (Section V), on 3D scenarios modeled according the propagation model described in Section VI. We consider a scenario of 500m x 500m, with buildings of height 30m placed at random cells of a Manhattan grid scenario. 50 EU devices are randomly distributed at indoor/outdoor locations. In Figures 4(a) and 4(b) we depict the CI metric (Section III) when varying the number of RUs and the altitude from ground. More specifically, in Figure 4(a) we plot the CI values against the number of RUs, for three increasing building density configurations, namely Rural, Suburban, and Urban. Clearly, increasing the number of RUs has a beneficial effect regardless of the scenario considered, since it translates into the possibility to enlarge the exploration and coverage range of the aerial mesh network. Figure 4(a) demonstrates the fact that the attenuation caused by buildings might have a significant impact on the quality of AtA links, since RUs are forced to stay closer to guarantee the request QoS expressed in terms of minimum Link Budget LB_{req} (Equation 2). Also, frequent partitions may occur in the aerial mesh, caused by mobility of RUs. In Figure 4(b) we further investigate this issue, by varying the altitude h from ground of the aerial mesh. We consider a modified version of the Suburban scenario (medium building density) with building heights uniformly chosen in the range [10, 20] m. This result clearly demonstrates that the CI is affected by the altitude h more than the number of RUs. Indeed, CI value significantly increases when $h \geq$ 15m, i.e. when RUs are -on average- just over the buildings clutter height. The case with h=0 corresponds to the case in which mobile nodes move on the ground, and thus the mobility algorithm described Section IV is implemented by robot equipped with wheels. Thus, aerial communication performs much better than ground communication, and the CI with 5 RUs deployed at 20 m is much higher than the CI with 30 RUs on the ground (h=0).

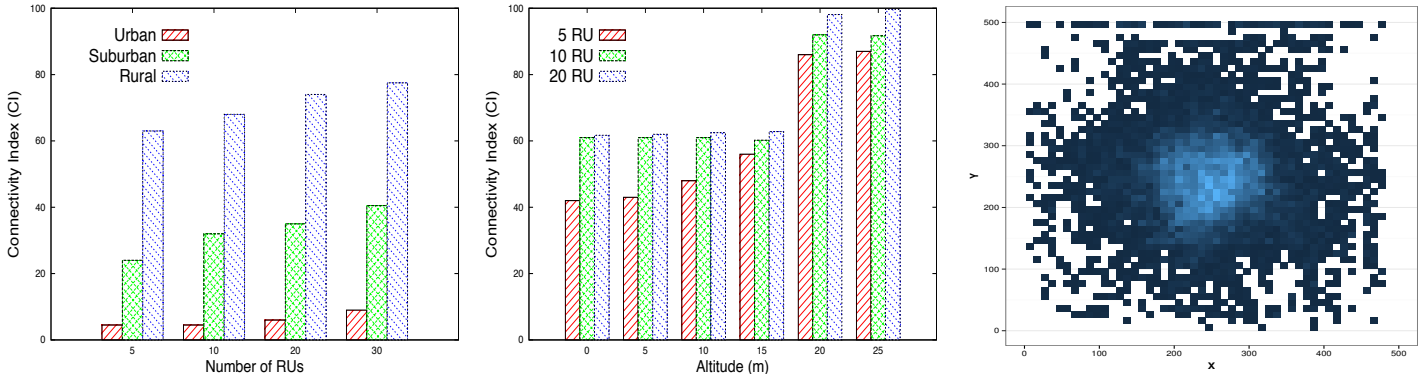
In Figures 4(c) and 5(a), we plot the coverage area with varying heights of 0m and 25m, when using the same number of RUs (i.e. 15). All the RUs are injected at the center of the scenario at the start of the simulation. The blue color gradients depict the probability that RUs will stop at a given position[2]. We see that RUs moving in the air are able to cover much wider area than terrestrial robots, while guaranteeing the same quality over the mesh link.

In Figures 5(b) and 5(c) we evaluate the effectiveness of the distributed charging scheme in guaranteeing a persistent coverage over the emergency area through four scheduling approaches: (*i*) Distributed, which corresponds to our solution described in Section V, (*ii*) Centralized (lowest), which corresponds to a centralized algorithm in which at each $T_{recharge}$ interval the RU_i with lowest residual energy E_i^{act} is selected for recharging, (*ii*) Centralized (lowest with $\alpha_i^{EU} = 0$) which works as the previous, but it chooses the lowest residual energy RU_i with $\alpha_i^{EU} = 0$, i.e. not connected to EU devices, (*ii*) Centralized (lowest with $\alpha_i^{critical} = 0$) which works as the previous, but it chooses the lowest residual energy RU_i with $\alpha_i^{critical} = 0$, i.e. not originating mesh partitions after its departure. Figures 5(c) shows the average CI for the four algorithms when varying the number of available RUs. This result demonstrates that selecting RUs on the basis of the energy factor only -i.e. like the Centralized (lowest) works- can cause frequent partitioning events within the mesh networks. This is also confirmed by Figure 5(c) where we depict the average number of isolated clusters that are originated during the simulation. The Centralized (lowest with $\alpha_i^{EU} = 0$) scheme attempts to maximize the number of connected EU devices on the AtG link, and for this reason, it outperforms the other two centralized schemes in terms of CI (Figure 5(c)). Vice versa, the Centralized (lowest with $\alpha_i^{critical} = 0$) scheme attempts to preserve connectivity among RUs on the AtA links, and for this reason it creates a reduced number of clusters compared to the other two centralized algorithms (Figure 5(c)). Our distributed algorithm takes into account both the connectivity of the aerial mesh and to EU ground devices through the two components of α_i (Equations 8 and 10), to give the best performance both in terms of CI and of number of cluster partitions.

VIII. Conclusion and Future Works

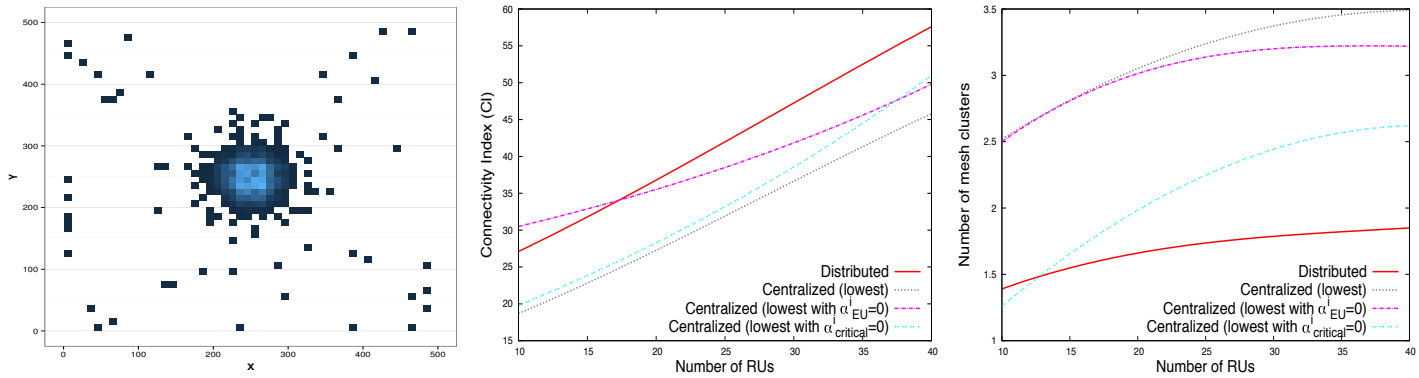
In this paper, we have explored ways to ensure connectivity among End User (EU) devices in post-disaster scenarios through the utilization of aerial mesh networks composed by Repairing Units (RUs). A swarm mobility algorithm, based on virtual spring model, has been proposed to allow the RUs explore the scene and self-organize into a multi-hop network. In addition, a distributed scheduling algorithm has been described to tackle the limited flying autonomy of RUs while guaranteeing a persistent coverage. We are further exploring implementation of the algorithm into a real testbed and analyzing the impact of antenna orientations on the network.

[2]Lighter colors indicate more frequent areas, darker colors indicate less frequent area, white color indicates areas with probability lower than 0.01.



(a) (b) (c)

Fig. 4. The CI metric as a function of the number of RUs and of the altitude h is shown in Figure 4(a) and 4(b), respectively. The average positioning of the RUs for $h = 25$ is shown in Figure 4(c).



(a) (b) (c)

Fig. 5. The average positioning of the RUs for $h = 0$ is shown in Figure 5(a). The average CI metric and the number of originated clusters for the four scheduling charging algorithms are depicted in Figures 5(b) and 5(c), respectively.

REFERENCES

[1] Federal Communication Commission (FCC). Deployable Aerial Communications Architecture in Emergency Communications. 2011.

[2] D. Behnke, K. Daniel and C. Wietfeld. Comparison of distributed ad-hoc network planning algorithms for autonomous flying robots. *Proc. of IEEE Globecom, Houston, USA, 2011.*

[3] J. Leonard, A. Savvaris, A. Tsourdos. Towards a fully autonomous swarm of unmanned aerial vehicles. *Proc. of IEEE UKACC, Cardiff, UK, 2012.*

[4] J. Vanualailal, A. Sharan and B. Sharma. A swarm model for planar formations of multiple autonomous unmanned aerial vehicles. *Proc. of IEEE ISIC, Hyderabad, India, 2013.*

[5] P. Dasgupta. A multiagent swarming systems for distributed automatic target recognition using unmanned aerial vehicles. *IEEE Transactions on Systems, Man and Cybernetics -part A: 38(3), pp. 549-563, 2008.*

[6] K. Daniel, S. Rohde, N. Goddemeier and C. Wietfield. Cognitive agent mobility for aerial sensor networks. *IEEE Sensors Journal: 11(11), pp. 2671-2682, 2013.*

[7] B. D. Song, J. Kim, J. Kim, H. Park, J. R. Morrison and D. H. Shim. Persistent UAV service: an improved scheduling formulation and prototypes of system components. *Journal of Intelligent & Robotic Systems: 74(1), pp. 221-232, 2014.*

[8] S. M. N. Alam and Z. J. Haas. Coverage and connectivity in three-dimensional networks. *Proc. of ACM Mobicom, Los Angeles, USA, 2006.*

[9] E. Yanmaz, R. Kuschnig, and C. Bettstetter. Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility. *Proc. of IEEE INFOCOM, Turin, Italy, 2013.*

[10] H. T. Kung, C. -K. Lin, T. -H. Lin, S. J. Tarsa, D. Vlah. Measuring diversity on a low-altitude UAV in a ground-to-air wireless 802.11 mesh network. *Proc. of IEEE Globecom, Miami, USA, 2010.*

[11] S. Morgenthaler, T. Braun, Z. Zhao, T. Staub and M. Anwander. UAVNet: a mobile wireless mesh network using unmanned aerial vehicles. *Proc. of IEEE Globecom, Anaheim, USA, 2012.*

[12] I. Dalmasso, I. Galletti, R. Giuliano and F. Mazzenga. WiMAX networks for emergency management based on UAVs. *Proc. of IEEE ESTEL, Rome, Italy, 2012.*

[13] A. Al-Hourani and S. Kandeepan. Cognitive relay nodes for airborne LTE emergency networks. *Proc. of IEEE ICSPCS, Gold Coast, Australia, 2013.*

[14] K. Derr and M. Manic. Extended virtual spring mesh (EVSM): The distributed self-organizing mobile ad hoc network for area exploration. *IEEE Trans. on Industrial Electronics*: 58(12), pp. 5424-5437, 2011.

[15] M. Di Felice, A. Trotta, L. Bedogni, L. Bononi, F. Panzieri, G. Ruggeri, V. Loscrı', P. Pace. STEM-Mesh: Self-Organizing Mobile Cognitive Radio Network for Disaster Recovery Operation. *Proc. of IEEE IWCMC*, Cagliari, Italy 2013.

[16] A. Trotta, M. Di Felice, L. Bedogni, L. Bononi. Re-establishing Network Connectivity in Post-Disaster Scenarios through Mobile Cognitive Radio Networks. *Proc. of IEEE MEDHOCNET*, Ajaccio, France, 2013.

[17] K. Daniel, S. Rohde, N. Goddemeier and C. Wietfield. A communication aware steering strategy avoiding self-separation of flying robot swarms. *Proc. of IEEE IS, London, UK, 2010.*

[18] C. Sommer, D. Eckhoff, R. German and F. Dressler. A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments. University of Erlangen, Technical Report, 2010.