

Topology-aware Path Planning for In-Transit Coverage of Aerial Post-Disaster Communication Assistance Systems

Julian Zobel, Benjamin Becker, Ralf Kundel, Patrick Lieser, Ralf Steinmetz
Multimedia Communications Lab

Technical University of Darmstadt, Germany

Email: {julian.zobel; benjamin.becker; ralf.kundel; patrick.lieser; ralf.steinmetz}@kom.tu-darmstadt.de

Abstract—Over the last years, natural disasters have shown to impair and destroy communication infrastructure. This results in an increased importance of infrastructure-independent ad hoc communication systems, such as delay-tolerant networks (DTNs). Research has shown that these networks are able to provide basic communication functionality for civilians. However, they are limited in their performance as the network topology is highly intermittent due to the human nature of clustering around important locations like shelters and moving in groups. Small Unmanned Aerial Vehicles (UAVs) have proven to be efficient data ferries between clusters due to their high mobility. This requires up-to-date knowledge about cluster locations to determine UAV flight paths. However, the shortest paths usually do not cover disconnected network nodes in transit between clusters, that will miss critical messages like evacuation notices or hazard warnings.

This paper provides two contributions for UAV-assisted post-disaster DTN communication. First, we present a novel approach to estimate dynamically changing cluster locations in a post-disaster scenario. Second, we introduce a topology-aware path planning approach for UAV data ferry flights, covering in-transit nodes in-between clusters. Our evaluation highlights the requirements on network topology knowledge for an efficient application of UAV data ferries. We furthermore demonstrate that our approach significantly reduces the number of disconnected in-transit nodes, which is especially important in the considered post-disaster scenario.

Index Terms—post-disaster communication, ad hoc communication, unmanned aerial vehicles, data ferry, topology-aware path planning

I. INTRODUCTION

In recent years, natural disasters like extreme weather conditions were more frequent and their occurrence as well as their devastation is expected to further increase in the future [27], [29]. These events often impair information and communication technologies (ICT) like the cellular grid, by either significantly damaging or destroying ICT infrastructure or by inhibiting critical infrastructure such as the power grid on which ICT infrastructure is reliant on. However, effective disaster relief efforts require communication especially for the civilian population, for example to provide information about save shelters or warnings of upcoming dangers. Thus, unavailable ICT stands in direct contrast to the requirements of disaster relief [13], [27].

In areas where the ICT infrastructure is unavailable, Disruption- or Delay-Tolerant Networks (DTNs) based on

everyday smart mobile devices such as smartphones are capable of providing basic, fast-deployable communication functionality for civilians [2], [11], [17]. In contrast to classical mobile ad hoc networks (MANET), DTNs use the *store-carry-forward* principle to cope with network partitions and constantly changing network topologies due to high device mobility in combination with short communication ranges of these devices. Especially as people in post-disaster scenarios tend to form groups and cluster around certain points-of-interest like shelters [2], applied DTNs are typically heavily intermittent with functional intra-cluster connectivity, but very sparse and infrequent inter-cluster communication. If inter-cluster distances—and therefore travel times—become too large in combination with certain message lifetimes after which they are dropped to ease the network load, communication among partitioned network clusters will eventually be impossible.

In this case, Aerial Post-Disaster Communication Assistance Systems (ALPACAS) comprising of one or more Unmanned Aerial Vehicles (UAV) are able to support inter-cluster communication by applying UAVs as quick and terrain-independent data ferries between network clusters [11], [19], [32] as depicted in Figure 1. Most notably, inexpensive commercial off-the-shelf UAVs are sufficient to be used for straightforward

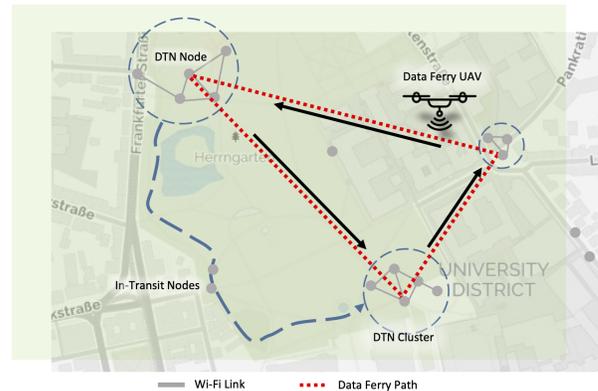


Fig. 1: Data ferry UAV connecting three clusters. Not all in-transit nodes are covered.

data ferry jobs [11]. However, the application of ALPACAS is especially challenging, as the network topology and cluster locations are usually unknown and also unsteady in disaster scenarios, in contrast to, e.g., static sensors in a wireless sensor network (WSN) [15]. Moreover, if topology information is available from the disaster area, data may be simplistic, error-prone, inaccurate, or incomplete. It is therefore of outmost interest to acquire knowledge on or estimate the locations of network clusters. Furthermore, people in transit between clusters are disconnected from the communication network during most of their transit time. Only a few disconnected devices may be tolerable, but with longer distances and more devices disconnected for longer times, the negative impact on the network communication performance could outgrow the costs of covering these devices. Especially when considering critical messages for civilians in the disaster area, like evacuation notices or hazard warnings, a thorough dissemination of these messages may be more important than disseminating less important messages.

In this paper, we approach the problem of estimating cluster locations within an unknown disaster scenario based on simplistic topology data and, if possible, adapt data ferry UAV flights to cover in-transit DTN nodes on the ground. Our approach enables a considerably higher coverage of DTN nodes with reasonable cost increase in terms of UAV flight time. In contrast to most approaches of related work, and current networking challenges in general, the presented approach considers the dynamicity of post-disaster DTNs and the special routing requirements of data ferry networks at the same time.

Specifically, we make the following contributions:

- We provide a concept to store and assess simplistic topology data for the estimation of partitioned DTN clusters in disaster scenarios. With a long-term assessment of the topology data, we are able to react to changing cluster locations as well as the formation and disbanding of clusters.
- Additionally, we provide an adaptive and topology-aware path planning approach for data ferry UAVs for in-transit coverage of DTN nodes based on available topology data.
- We evaluate the capabilities of our cluster estimation in different post-disaster scenarios. Furthermore, we highlight advantages and drawbacks as well as its influence of our topology-aware path planning approach on DTN communication performance in these scenarios.

The remainder of the paper is structured as follows. Related work and similar approaches are presented in Section II. Section III provides a detailed description of our approach for cluster estimation in dynamic post-disaster environments. Topology-aware path planning for in-transit node coverage is described in Section IV. In Section V, we evaluate our approach and its impact on communication performance. Eventually, Section VI concludes the paper.

II. RELATED WORK

Unmanned Aerial Vehicles (UAVs) are often deployed for aerial monitoring or object identification due to their high mobility and autonomy. This could, for example, include search-and-rescue missions, in which UAVs identify victims visually with their camera [12], [20], [25], or the fast localization of forest fires [4]. However, visual UAV exploration can be obstructed, e.g., by trees or may be impossible for urban areas where nodes are inside of buildings, and have the disadvantage of limited area coverage and additional weight of cameras [20]. Thus, researchers have proposed to track radio signals from devices, such as Wi-Fi communication or periodic ad hoc protocol beacons, to identify device locations without unobstructed line-of-sight and with larger tracking areas [20], [23]. When searching large areas, for example to detect nodes with Wi-Fi signals, the usage of UAV swarms was suggested and successfully applied by multiple researchers [8], [24]. By that, either the time for node recognition is decreased or larger areas can be covered in the same time, compared to a single-UAV system. Areas can be divided into smaller partitions, e.g., into a grid, where each cell is completely monitored by one or multiple UAVs in detail [8], or traversed by randomly moving UAVs which allows for a faster but not guaranteed identification of devices [24].

Utilizing the knowledge of the underlying topology, UAVs can be deployed to support or completely facilitate communication, for example in disaster scenarios with damaged or destroyed ICT infrastructure. In such cases, UAVs can be used as static aerial LTE or Wi-Fi base stations to restore cellular network coverage [22] or provide a Wi-Fi-based overlay mesh network to the disaster-stricken area [22], respectively. UAV positioning can, for example, be pre-computed to provide fixed locations for each UAV [22], or decided on the fly by a distributed swarm algorithm that optimizes static UAV locations to fit the current situation [28]. However, these approaches do not scale well for very large areas, as the number of UAVs required to provide the service increases with the area.

Nevertheless, UAVs can be applied as carriers or messengers, often called data mules or message ferries, distributing and collecting data between single devices or partitioned subnetworks [31]. This approach is especially useful when the number of available UAVs is not sufficient to maintain static relay networks and can be easily applied in cases when latencies in data distribution are manageable, such as in DTNs, WSNs, or Internet of Things (IoT) applications. For example, a single UAV can collect data from isolated communities to areas where Internet access is available, and vice versa, giving at least limited access to otherwise fully disconnected areas in disaster areas [3]. Calculating and optimizing UAV flight trajectories for data collection and distribution, however, is on the one side very scenario-dependent and on the other side often an even more complex variant of the NP-hard Traveling Salesman Problem (TSP) [3], [15], [18], [19].

Trajectory optimization can, for example, focus on data

collection from static sensor nodes. With the correct placement of UAVs, the lifetime of battery-powered IoT sensors can be significantly increased by reducing the power required to transmit data to the UAV [15]. On the other hand, calculating more energy-efficient trajectories for the UAVs can increase flight time, and therefore, result in higher coverage or shorter service intervals, overall increasing the efficiency of data distribution and collection [6], [15], [16], [30]. Similar approaches for trajectory optimization were presented in the context of wireless power transfer from UAVs to static sensor nodes [7]. Furthermore, cooperative trajectory optimization can be used to maximize the information gain of a multi-UAV system [6], for example when collecting topology information.

Most approaches for trajectory optimization assume static networks and often require perfect knowledge of the network topology, which is then used to solve the underlying TSP with different approaches like heuristics [19] or machine learning [18]. Currently, there exists no approach that includes factors such as dynamic network topologies, updates of available topology data, or adaptations to a changing network topology during runtime. Additionally, most approaches may not be suitable for scenarios with a mixture of static and mobile devices, and will omit mobile nodes most possibly, or they may not be suitable for large-scale scenarios. The related work has, nevertheless, shown that it is possible to detect mobile devices and gather topology data. However, interpreting and especially handling changes in the data over time is still an open issue.

Compared to these related approaches, our work focuses on adaptive cluster detection in dynamically changing environments while also considering the coverage of mobile in-transit devices when connecting network clusters.

III. CLUSTER DETECTION ON TOPOLOGY DATA

The network topology of DTNs in disaster scenarios is typically highly intermittent due to the human nature to form groups and cluster around points-of-interest [2]. Within these clusters, connectivity among the network nodes is good, whereas inter-cluster communication is sparse and infrequent. Aerial Post-Disaster Communication Assistance Systems (ALPACAS) can significantly increase inter-cluster communication performance by deploying highly mobile UAVs as data ferries.

However, knowledge of the cluster locations is a necessary requirement to deploy ALPACAS in the first place. Thus, topology information must be obtained to determine where communication support is required, i.e., by identifying network clusters. As discussed in Section II, acquiring topology information is feasible with several approaches. However, it is often neglected that topology information is changing over time and once optimal flight paths could become obsolete due to the dynamicity of the environment. Furthermore, this information may be provided by different sources at different times with different degrees of detail and accuracy. Topology information can comprise, for example, device locations and their time of measurement, but also further information such



Fig. 2: Grid overlay with location measurements in an inner-city environment. The grid size and its placement have a significant influence to which cells measurements are allocated to. Some groups are within a single cell, while others are spread over multiple cells.

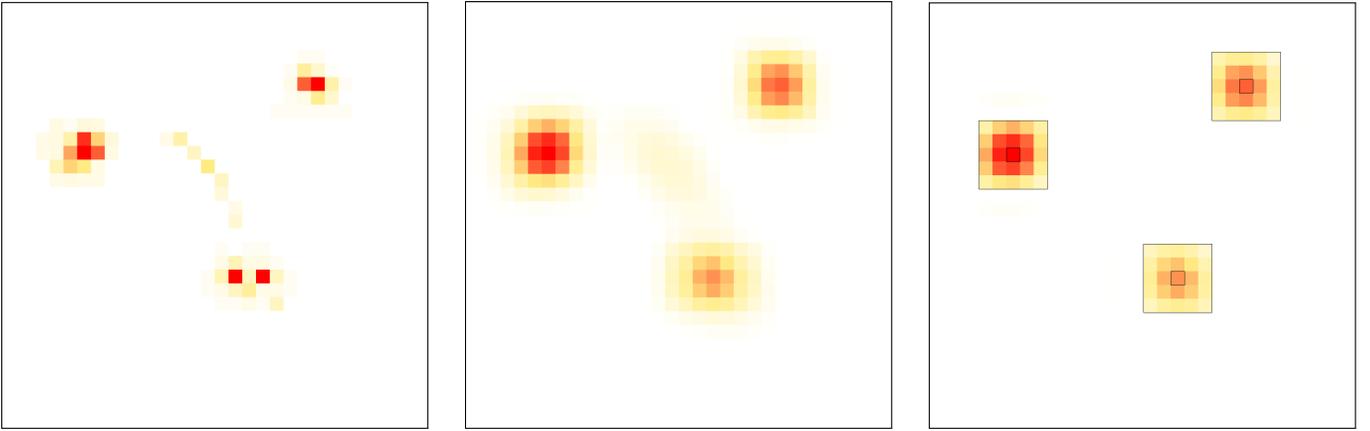
as unique device IDs, neighborhood topology information, movement directions, speeds, and more. Data could, however, also be derived from sources like satellite or aerial pictures and therefore only provide location information without further meta data. In order to deploy an ALPACAS in any unforeseen disaster situation, it therefore cannot be designed to rely on complex data.

Consequently, the following questions arise: (i) how simplistic data can be interpreted to detect or estimated network clusters, (ii) how updates of the topology data can be used to adapt and improve the detection or estimation over time, and (iii) how cluster detection and estimation can be robust against missing, incomplete, or erroneous data updates.

A. Collecting Topology Information: Heatmap

Assume we have a source of topology data, such as a topology monitoring UAS running in parallel to our ALPACAS. The provided simple topology data is in the form of a set of locations $P = \{p \mid p = (x_p, y_p)\}$ from the disaster area. Our system cannot influence the monitoring system and has no access to any further information in addition to the locations on the network topology. Potentially, the provided location data is already several minutes old and locations were measured with some error. Within a certain period Δt_{update} , the source will deliver a new set of locations as an update or addition to the last input, e.g., after a monitoring UAV returned with new information.

The most obvious approach would be to determine network clusters based on the latest data. However, if this data is incomplete or erroneous, we would potentially miss cluster locations completely. This would make the system less robust against real-world input data in disaster scenarios, hence,



(a) Grid counter heatmap. The lower cluster is suboptimal placed in the grid, resulting in two separate, non-adjacent high-density cells.

(b) The non-adjacent cells became merged by a Gaussian smoothing kernel convolution. In-transit nodes are mostly faded out.

(c) Detected hotspots and their center cells after high-pass filtering and hotspot detection.

Fig. 3: Three network clusters and sparse in-transit node movement in a $1500 \times 1500 \text{ m}^2$ area with cells of $d_{\text{cell}}=50\text{m}$. The used Gaussian kernel is 5×5 in size and $\sigma=50\text{m}$. After smoothing, the lower hotspot, which was distributed over two non-adjacent cells in (a) before, is detected as one cluster in (c) with its center between those two cells.

historic data should also be included in the detection. On the other side, determining clusters on all data ever collected is also infeasible, because (i) information may become outdated and less relevant, thus hinder correct cluster estimations and (ii) the amount of data in a long-term deployment will become unwieldy.

The determination of clusters could be done with well-known clustering algorithms such as k-Means [1] or DB-SCAN [5]. However, k-Means require knowledge of the number of clusters a priori or repeat it with multiple values for k and determining somehow the best k , which may become computationally expensive especially in large datasets [5]. Furthermore, usual properties like arbitrary shape detection and the clustering of all data points bear two significant disadvantages. First, close but distinct clusters may be wrongly detected as a single cluster. Second, it is hard or even impossible to detect outlines and sizes of clusters, if locations outside of clusters are seen to belong to it nonetheless. This is especially important, as the location data will have a significant amount of noise in the form of location measurements of in-transit devices.

Our approach is to aggregate the input data to cope with large amounts of location measurements and to increase robustness against problematic input data. For that, we divide the UAS mission area into a simple grid of equally-sized quadratic cells c of edge length d_{cell} as shown in Figure 2. Each location from the input data is then matched to their respective cell, each having a counter Ω_i representing the number of measured positions within the cell c_i . Thus, new data can easily be added on existing data, without the necessity to re-calculate or re-evaluate the whole topology. The resulting structure is a heatmap of the node distribution in the area.

Naturally, information may get outdated over time as

the network topology is subject to change. Within disaster scenarios, new clusters can form and existing clusters can dissolve, for example around temporary resource distributions centers [10]. Linearly increasing the counters at new measurements will result in a static topology assessment, which becomes more and more inaccurate. As the detection of and adaption to topology changes is one of the major issues for an efficient ALPACAS deployment, historical data must not overpower more recent measurements. Therefore, it is necessary that the value of information—that is each counter Ω_i —decreases over time. The intensity of this temporal decrease, however, is also a major influence factor on the overall system capability to adapt to topology changes. On the one hand, a slow decrease might lead to a loss in dynamicity and the system reacts too slow to changes. On the other hand, decreasing too fast could lead to instability in the topology assessments and a vulnerability to receiving incorrect or inaccurate data.

To address this challenge, we introduce a half-time period $T_{1/2}$ to the counters that we determine on the update interval such that $T_{1/2} = \Delta t_{\text{update}}$. With that, the counter Ω_i after a certain interval Δt is

$$\Omega_i(t + \Delta t) = \Omega_i(t) * e^{-\Delta t * \frac{\ln(2)}{T_{1/2}}} \quad (1)$$

such that Ω_i is only half of its original value after the counters half-time period. But as new measurements will be added again, cells with a stable amount of measurements will not change significantly. On the other hand, however, Ω_i is reduced exponentially without new measurements, which allows to quickly detect if a cluster has dissolved. Therefore, the system is robust against temporary errors in the topology measurement as historical data is included in the assessment,

but also flexible to changes as historical data is not paramount to new and updated data.

Another major influence on our approach is the choice of the grid cell size. In general, smaller cells have the advantage of depicting the area in greater detail, but the increased number of cells necessary within the same area are more complex to process. Furthermore, tiny cells will yield no significant advantage over the raw data, as data aggregation will be minimal. Bigger cells, on the contrary, are less complex, but also less accurate and detailed.

We propose to set the grid cell size to be within expected network communication ranges. With that, we have a high probability that nodes within a cell can communicate with each other and also that communication links to nodes in adjacent cells can exist. This connectivity is most important from our point-of-view, as it allows the assumption that adjacent cells form a single, connected cluster. A real-world test performed in an urban scenario [2] highlighted that most urban DTN connections can be expected to be within 50 to 100 meters, although they may reach up to 150 meters. Thus, cell sizes should be chosen from the range between 50 to 100 meters based on the encountered scenario, considering the required level of detail and accepted computation complexity.

B. Processing Topology Information: Cluster Detection

The naïve approach for the detection of clusters in the heatmap is to select the most populated cells. However, the major drawback of using a static data structure like this grid for hotspot detection is its subjection to the grid placement itself. That is, a group of nodes on the border between cells or spread across multiple cells will result in their location measurements being assigned different cells. Therefore, the measurement is distorted, which complicates the detection of this cluster. A coincidentally centered and high-density cluster, on the other hand, will result in a much clearer cluster detection. Especially with smaller cells, this problem increases in frequency, and thus, severity. Furthermore, we generally cannot assume clusters to have a certain size and a centered placement in the grid, so that further adaptations for the cluster detection are necessary.

To address this problem and reducing the severity of clusters distorted over multiple cells, exemplarily shown in Figure 3a, we propose to combine nearby cells together, while simultaneously reducing highly localized spikes in the heatmap. As depicted in Figure 3b, this can be achieved by applying a Gaussian smoothing filter on the heatmap. Similar to its application in digital image processing, location measurements are blurred and aggregated over multiple cells to obtain smoother values in general. Furthermore, it reduces the impact of localized peaks (noise) in the position measurements, as for example, a larger group of people in transit between clusters would produce, and therefore reduces false cluster detection. Recognizing this mobile group as a static hotspot would reduce effectiveness, as the UAV will approach a probably empty location.

The common approach for Gaussian smoothing in similar cases is to convolute the map with a pre-computed discretized

kernel. The entries of the two-dimensional kernel can be computed using

$$kernel[x][y] = \frac{1}{2 * \pi \sigma^2} * e^{-0.5 * ((\frac{x-mean}{\sigma})^2 + (\frac{y-mean}{\sigma})^2)} \quad (2)$$

where x and y are the indices in the kernel and mean is the center of the kernel. The kernel is typically symmetric and of odd size.

The σ determines the width of the Gaussian distribution, thus, in which extend adjacent cells will impact the smoothed result. Generally, a smaller σ results in a more narrow Gaussian distribution, thus adjacent cells will have less of an impact. In contrast, larger σ result in a wider distribution, incorporating neighbors more intensively, while attenuating the center cell more than a narrow distribution does. Again, we propose to use the expected average communication range as choice for σ , so that cells within that radius have the most impact on the smoothed value (around 68%), similar to the higher probability of communication links and therefore a higher probability of a connected cluster. The kernel should at least cover a radius of 2σ , so that it approximates the Gaussian distribution sufficiently (approx. 95.45%).

After smoothing the heatmap, we apply a high-pass filter to filter out small localized measurements, e.g., as seen in the center of Figure 3b, to prevent them from being falsely detected as clusters. However, the threshold of the high-pass filter is highly scenario-dependent. Using a big threshold it will negatively influence the detection of actual small clusters, whereas a small threshold will increase the amount of wrongly detected clusters. As of now, we chose to set a fixed border based on the minimal amount of nodes we want to serve on each cluster. Nevertheless, a dynamic or localized adaption of this high-pass filter could be a significant enhancement to the overall cluster detection and is left open for future research.

Figure 3c depicts the final step of the cluster detection, in which a local maximum search is performed such that the extend of a cluster match the size of the kernel. In the given example, we were able to detect all three clusters and can use, for example, their center cells as the locations UAVs should approach in each cluster.

With the gathered and processed information, we are now able to calculate the UAV flights between these center cells to bring our assistance support system in service.

IV. TOPOLOGY-AWARE IN-TRANSIT COVERAGE PATH PLANNING

Calculating an optimal path between found cluster locations, however, is no trivial task. Shortest path calculation over all cluster locations requires to solve the NP-hard Traveling Salesman Problem (TSP), thus, heuristics are usually applied. Additionally, a significant amount of turns on the path can negatively affect the energy consumption and with it the mission completion time of the UAV [16], [19], [24]. The shortest path may, therefore, not be the most optimal for a UAV. However, path optimization is still a significant research gap that is not within the scope of this work and we use a simple heuristic to determine the sequence of clusters to visit.

In this section, we specifically approach the problem to cover nodes that are in transit between clusters. Such nodes are usually required in DTNs to carry messages between clusters (*carry-store-forward principle*). Nevertheless, most DTNs exert a message lifetime to reduce overall network load, and with increasing distances between clusters the transit time will eventually exceed the lifetime. To overcome this problem of poor inter-cluster communication performance, we apply ALPACAS in the first place. Obviously, data ferry UAVs fly in straight lines between clusters to minimize message dissemination times.

However, in-transit nodes are thus not covered by the system if a data ferry UAV is not crossing the path of these nodes by coincidence. They are therefore disconnected from the ad hoc network for possibly the whole transit time. Because we specifically consider a post-disaster scenario, this is a significant problem as very important messages like evacuation notices or warnings of emerging threats may not be delivered in time. Our main objective is, therefore, to increase the amount of nodes that the UAVs have contact to on their path between clusters, which directly correlates to the amount of nodes that are notified timely.

A possible approach to increase the covered in-transit nodes is to use map data to adapt data ferry routes to the street layout. However, the question arises which streets should be traversed by UAVs, as (i) roads may be obstructed and not usable for ground nodes due to the disaster, (ii) shortest flights or biggest streets may not correlate to the routes taken by nodes, and (iii) nodes may not even use official streets at all but move cross-country. As this may be out of our knowledge, map data is not applicable to solve our problem.

Instead, we make use of the available topology data in the heatmap to calculate a flight path between two clusters. More specifically, we model our path planning problem as a shortest path problem [14] between a start and an end cluster. For that, visiting a cell c_i on that path provides a certain benefit

$$b(c_i) = \frac{\Omega_i}{\Omega_{max}} \quad (3)$$

that is dependent on the normalized counter of location measurements, so every measurement provides the same weight. Generally speaking, a longer path using more frequented cells should be favored over a shorter path using less frequented cells or cells without location measurements. Nevertheless, an increase of the path length will also increase the time a UAV requires to traverse all clusters, and thus, negatively influence overall message delivery delays. We therefore balance the tradeoff between fast message delivery and increased in-transit node coverage by scaling the benefit of each cell with an impact factor f . Thus, the weight ω for a directed edge between two adjacent cells c_i and c_j can be expressed as

$$\omega(c_i, c_j) = d(c_i, c_j) - b(c_j) * f \quad (4)$$

where $d(c_i, c_j)$ is the distance between these cells, i.e., $d(c_i, c_j) = 1.0$ for direct neighbors and $d(c_i, c_j) = \sqrt{2}$ for diagonal neighbors.

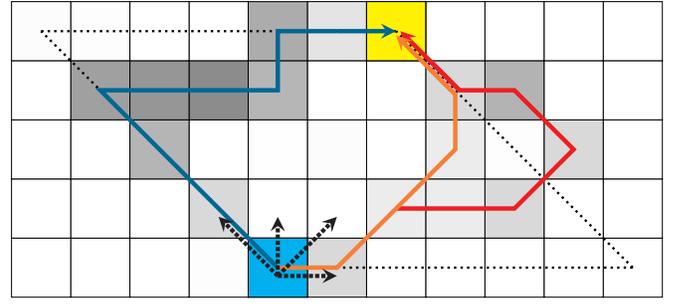


Fig. 4: In-transit coverage path planning with different possible paths. Paths are restricted to the dotted area based on the direction from the start at the bottom to the target on the top. If f is chosen sufficiently large, the longer path on the left with higher coverage is chosen over the shorter path with lower coverage in the center. The right path is outside of the restricted area and therefore invalid.

The overall route cost then is

$$\begin{aligned} \omega(c_0..c_n) &= \sum_{i=0}^{n-1} \omega(c_i, c_{i+1}) \\ &= \sum_{i=0}^{n-1} d(c_i, c_{i+1}) - f * \sum_{j=1}^n b(j) \end{aligned} \quad (5)$$

and the influence of f on the route becomes more graphic. For $f = 0$, the path planning algorithm chooses the shortest distance, but with $f > 0$ searches for available paths that have lower costs compared to the shortest distance path. As the graph may contain negative edge weights and negative cycles depending on f , using classical path planning algorithms like Dijkstra or Bellman-Ford is not applicable. Furthermore, finding shortest simple paths—paths without any loops—is an NP-hard problem. To remove negative cycles from the graph, we further restrict the path planning algorithm to use only four out of eight possible movement directions, based on the direction from source to target (c.f. Fig 4). The problem now reduces to a shortest path problem with negative edge weights without negative cycles, which we solve with dynamic programming [14]. This also restricts the path to be within the general direction of the target, avoiding excessive detours, and further balancing the path coverage tradeoff.

An example of our topology-aware path planning approach is given in Figure 4. Darker shades denote a higher Ω for that cell. When calculating the best path from the start (bottom) to the target (top), the planning area is restricted by the four grid directions in the direction of the target. In this case, as depicted by the arrows it is straight right, diagonal up and right, straight up, as well as diagonal left and up. The allowed area is framed by a dotted border. From the three shown paths, the right one is outside of the restricted area and therefore invalid. The left is considerably longer than the center path, but traverses more cells with higher node count on its way. Which path is chosen in the end depends on the size of f and Ω of the traversed cells. However, it becomes clear that the impact of large f

on the UAV flights may be significant, as even with the area restriction the maximum path length is $1 + \sqrt{2} \approx 2.41$ times the length of the shortest distance path on the grid, i.e., when $f = 0$. Thus, f has to be chosen reasonably.

V. EVALUATION

The evaluation of our approach was conducted within the SIMONSTRATOR [21] simulation platform using the PEERFACTSIM runtime environment [26]. Used simulation settings are listed in Table I. For analyzing the communication performance, we rely on the simple and robust HyperGossip [9] epidemic DTN protocol. Our approach was added to the simulation platform for Unmanned Aerial Systems as described in [11] and uses a single multicopter UAV as data ferry [32].

All simulations were performed within a $3 \times 3 \text{ km}^2$ inner-city area with 250 mobile nodes for 6 hours and 10 random seeds. The area size is chosen to be challenging to both the UAVs maximum flight range of approximately 9600 m (at a constant speed of $10 \frac{\text{m}}{\text{s}}$, cf. [32]) and the DTN communication performance. Node movement was restricted to streets and walkways accessible for pedestrians by using Open Street Map¹ (OSM) data. The used mobility model uses attraction points around which nodes move, and therefore clusters build up. Nodes may also move to another attraction point, thus becoming transit nodes.

We defined three different scenarios to assess both our hotspot detection approach as well as our adaptive in-transit coverage routing approach:

i) **STATIC**: 5 attraction points randomly placed within the simulation area with at least 200 m distance. Nodes move around the attraction points, but are not allowed to move between them. Thus, this scenario results in a very static topology with only changing cluster shapes.

ii) **MOBILITY**: Similar to **STATIC**, but nodes can move between attraction points. This results in a less static topology with frequent changes in both cluster sizes and shapes. Due to random node and target selection, the distribution of nodes over all attraction points may be highly imbalanced.

iii) **DYNAMIC**: For this scenario we have chosen 9 points-of-interests from OSM data within our inner-city area, such as hospitals and schools, that are important locations in a post-disaster scenario. Only three of these locations, however, are static for the whole time, the other will be created and again removed during the runtime. Therefore, node mobility is highly dynamic as new attraction points will result in changing movement patterns but also in a very unstable, and thus, challenging environment.

A. Cluster Detection

As already stated, we assume the cluster detection to be highly dependent on the cell size and the information update interval. A shorter interval should result in an increased currentness of the topology data, and thus, a more accurate cluster detection. Smaller cells should also provide a more

TABLE I: Simonstrator Environmental Settings

Scenario	Area	3000 m x 3000 m
	Map	Inner City, Post-Disaster
	Duration	1 h warmup, 5 h evaluation
	Nodes	250
	Node Speed	$0.8 - 1.5 \frac{\text{m}}{\text{s}}$
	Node Movement	Pedestrian OSM Map
	Attraction Points	random, OSM points-of-interest
Comm.	PHY	Wi-Fi
	Range	approx. 75 m
	Data Rate	5 Mbit/s
DTN	Protocol	epidemic, HyperGossip [9]
	Msg TTL	30 minutes
	Msg Size	600 Byte
	Msg Rate	6 per minute
UAS	UAV	1 quadrotor (cf. [32])
	Flight Range	approx. 9600 m at 10 m/s
	Battery Swap Time	60 seconds
Heatmap	Update Interval	5, 15, 30 minutes
	Cell Size	50 m, 75 m, 100 m
	σ Kernel Size	50 m

accurate view of the topology, and thus, a better performance than with bigger cells.

The cluster detection was assessed in the three scenarios for different grid cell sizes and information update intervals with the following metrics:

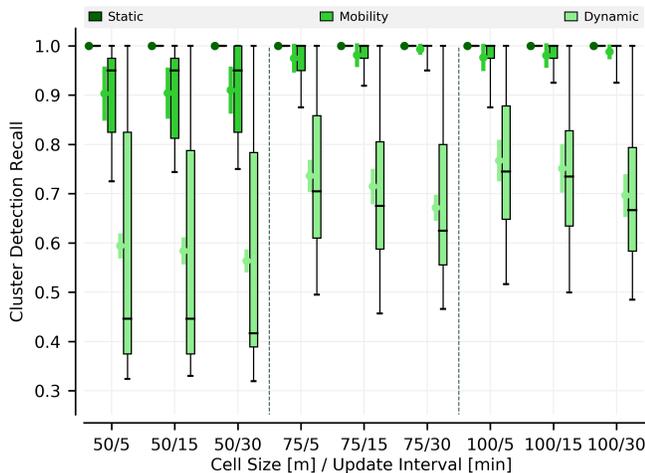
i) **RECALL** denotes the fraction of correctly estimated clusters among the set of all existing clusters.

ii) **PRECISION** denotes the fraction of correctly estimated clusters out of the set of estimated clusters.

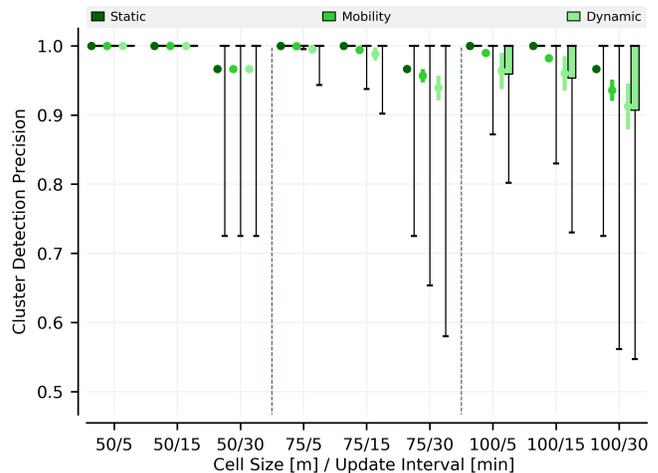
Figure 6 depicts the aggregated **RECALL** and **PRECISION** of ten simulation runs with different random seeds. Measurements for each scenario are grouped by cell size and therein shown with different update intervals. On each boxplot, the bold line shows the median, the colored box 25th and 75th percentiles, and the whiskers 2.5th and 97.5th percentiles, respectively. The colored dot and handles show the mean and their standard deviation, respectively. As the attraction points from the simulator's movement model indicate the locations where nodes will gather around, it makes them a reasonable choice to integrate them in our evaluation. Therefore, a cluster was counted as correctly detected if the estimated cluster area overlapped with an attraction point from the movement model, and a false estimation if there not.

As expected, the cluster detection performs very good for the **STATIC** scenario but worse for scenarios with movement. Especially the **DYNAMIC** scenario results in a high deviation and significantly lower correctly estimated clusters than in the other scenarios. This can be accounted for a highly imbalanced distribution of nodes over attraction points, such that small clusters cannot be detected. Interestingly, smaller cells do not yield better results for the cluster detection, in fact the opposite is the case. Although the false positive detection of clusters is better with smaller cells, correctly detecting clusters is much harder since then large clusters are spread over too many cells. With large cells, on the other hand, the false detection of

¹www.openstreetmap.org



(a) Cluster detection RECALL, i.e., the fraction of correctly estimated clusters out of the set of existing clusters.



(b) Cluster detection PRECISION, i.e., the fraction of correctly estimated clusters out of the set of estimated clusters.

Fig. 5: Cluster detection RECALL and PRECISION for cell sizes of 50, 75, and 100 meters as well as information update intervals of 5, 15, and 30 minutes, respectively. Smaller cells provide better PRECISION while larger cells provide better RECALL. The impact of the update interval is scenario-specific.

clusters is much higher, as groups of in-transit nodes could be taken for a cluster. We can determine a general trend that larger cells perform better in RECALL, while smaller cells result in a better PRECISION. Therefore, a cell size of 75 m performs best when maximizing both metrics in all three scenarios.

Overall, it becomes clear that the cell size has a more significant impact on the cluster detection performance than the update interval. For the STATIC and MOBILITY scenario, surprisingly, the RECALL deteriorates with smaller update intervals. The DYNAMIC scenario, on the other hand, perceives an increase in RECALL at the same time. This most probably resembles the scenario, as with rather static clusters in the first case a higher update frequency lead to a faster decrease of counters (cf. III) and with it a more unstable knowledge base on the whole topology. This is especially interesting as the system in this case is still able to operate the hotspot detection without very frequent updates. With a very dynamic topology, however, this shorter interval allows a higher flexibility and thus a better detection of changes, although longer update intervals still result in an acceptable cluster detection rate. Nevertheless, we must empathize that this is very scenario-specific and may not be correct for less-clustered application scenarios than our post-disaster scenario. To conclude, we can determine that our system is able to estimate clusters within our scenarios even with larger, more realistic topology information intervals.

B. In-Transit Coverage Path Planning

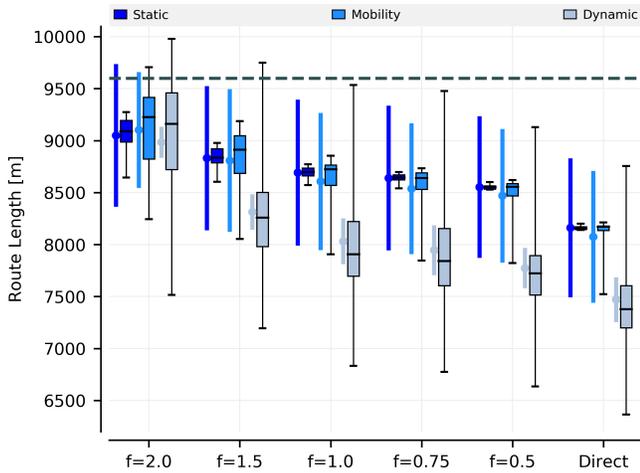
In Section IV, we discussed the influence of different values of f on the resulting path length. A higher f should result in longer paths and with that in a higher coverage of in-transit nodes. For the evaluation we used a cell size of 75 m and an

update interval of 15 min to resemble our system in action. All simulations were performed ten times with different random seeds on all three scenarios with different values for f .

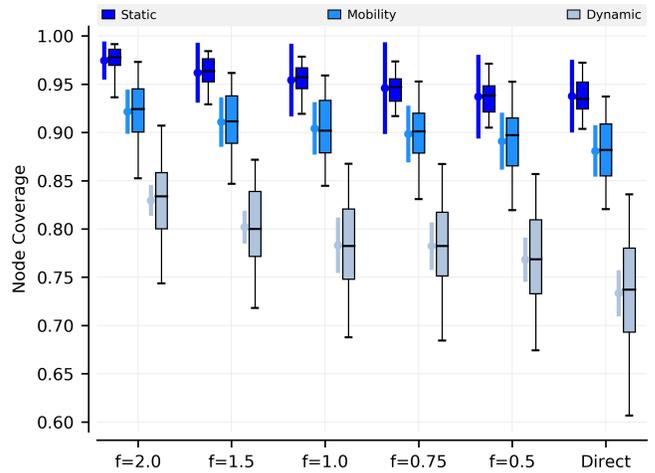
The results for path length variations are presented in Figure 6a. First of all, the high standard deviation for the STATIC and MOBILITY scenarios can be perceived. This is due to the random distribution of the attraction points in the simulation area, and therefore, differences in the general path lengths for different simulation runs. In contrast, the DYNAMIC scenario uses the same locations for each run, thus has a significantly smaller standard deviation. However, the overall deviation is much higher as the cluster detection performs worse, which affects the number of clusters and therefore the total path length. On the other extreme, path length deviations in the STATIC scenario are very small since all clusters are found and the length is mainly affected by the in-transit coverage adaption.

As expected, increasing f results in an increase in the path length in all scenarios. Though one may wonder why the length increases in STATIC despite there is no in-transit movement over which the UAV could be routed. However, a UAV can be routed over cells of another cluster depending on its location and the chosen f . This also result in more nodes for the UAV to interact with, if for example the nodes were out of reach when the UAV crossed the cluster the last time (cf. Fig. 6b).

The most significant increase happens within the DYNAMIC scenario, when the average path length increases by nearly 30% from direct paths towards $f = 2.0$. In that and other cases, the path length therefore exceed the range of UAV which then has to return to the base station before completing the cluster traversal. The most reasonable approach would



(a) Calculated UAV route lengths for different values of f compared to direct flight in the three scenarios. The dashed line denotes the maximum range of the UAV.



(b) With increasing value of f , the amount of nodes that the UAV has contacts with on its route increases significantly in comparison to the direct flight.

Fig. 6: UAV route lengths and node coverage for different values of f in comparison to direct routes.

be to calculate the best value for f for best coverage while not exceeding the UAV's range limit. This could extend the system capabilities to adapt to highly different scenarios and we consider it for future research.

When comparing the node coverage presented in Figure 6b with the path length in Figure 6a, we see a general trend to cover more nodes with longer paths, as expected. In the MOBILE scenario, approximately 12% of nodes are not covered by the direct path, while an average increase of 900m—or 11%—of the path reduces this amount down to approximately 7%. Therefore, the adapted topology-aware path covers around 42% of in-transit nodes formerly not covered. In the DYNAMIC scenario, around 27% of nodes are not covered. When increasing the path length by 9%, it is able to cover 26% of the in-transit nodes. However, it requires an even larger increase of 20% to further increase the in-transit coverage to 41%. Therefore, a constant increase in coverage cannot be assumed by just increasing the path length. Especially for highly dynamic scenarios, where nodes will take a multitude of different paths although the UAV can only cover one, it will be very challenging to further increase the in-transit node coverage.

At last, we assessed the impact of our approach on the general communication performance. For that, we tracked the fraction of message distribution, i.e., the spread of a message within the whole network, which is similar to RECALL. A message distribution of 1.0 thus denotes that every node in the network has received the message, whereas 0.0 denotes that no node other than the creator knows of the message. Furthermore, each message was restricted by a message lifetime of 30 minutes, after which the message is dropped and not further distributed by UAVs nor nodes.

Figure 7 shows the fraction of message distribution over

the message lifetime for $f = 1.0$ and $f = 2.0$ in comparison to the direct paths for the aggregated MOBILITY scenarios. Bold lines denote median values, dashed lines the 25th and 75th percentiles, outer lines 2.5th and 97.5th percentiles, respectively. In, general, we clearly see the tradeoff between covering in-transit nodes while increasing the UAV flight time. On the one hand, we have a perceivable gain seen as faster message distribution for upper percentiles on the left. But on the other hand, we also see that the lower percentiles can become worse than the direct flight. However, a positive impact of our in-transit coverage routing in general cannot be neglected.

More interestingly, a larger value for f may not result in a significant increase in performance compared to a smaller one. Comparable to the increase in network coverage in the MOBILITY scenario between $f = 1.0$ and $f = 2.0$ (approx. 3%) as shown in Figure 6b, the communication performance also receives only a slight increase for longer paths. However, we must denote that this again is specific for our post-disaster scenarios with relatively large numbers of nodes in clusters compared to in-transit nodes. In case of more in-transit nodes, we assume to also have a more apparent performance gain.

This evaluation showcases that our heatmap-based approach is able to estimate cluster locations even in challenging scenarios. For that, we only need simplistic topology information without prior knowledge of the topology or supplementary data. Additionally, we are able to further utilize this information to adapt our UAV flight paths to cover in-transit nodes. And although there is still room for improvement, we have shown that our approach could reduce the amount of nodes that are usually not covered by the data ferry UAVs by up to 42% with a 20% increase in UAV flights. Within the assumed post-disaster scenario, this allows significantly more nodes

to receive urgent messages like evacuation notices or hazard warnings. As the knowledge about dangers in the post-disaster area may prevent damages and injuries, every node covered is a success.

VI. CONCLUSION

Aerial Post-Disaster Communication Assistance Systems (ALPACAS) are an important supplement to post-disaster ad hoc networks, as they enable essential communication between isolated and distributed network clusters. However, their application requires knowledge of cluster locations. Furthermore, quick inter-cluster distribution is usually the single objective, therefore omitting nodes in transit between clusters.

At first, we presented an approach for adaptive cluster detection in dynamically changing environments based on simplistic topology data. We have shown that our approach can correctly estimate cluster locations even in challenging post-disaster scenarios with a low currentness of data. Furthermore, ALPACAS can react on changes in the topology, such as newly formed or dissolved clusters, at runtime.

Secondly, we introduced a novel topology-aware path planning approach to adapt UAV flight routes to areas where nodes have been in the past, such that the probability to cover in-transit nodes by UAVs is increased. Our evaluation results have shown that by considering these in-transit nodes, the number of disconnected nodes can be reduced by up to 42% while increasing the UAV path length by only 20% at the same time. Furthermore, we observed a positive impact on the message distribution with a large amount of messages spreading faster through the network, however with the tradeoff of disadvantaging other messages.

In future work we want to consider other data structures than grid-based heatmaps to improve the detection algorithm and overcome inadvertent splits of clusters over multiple cells. The benefit impact factor f used in path planning could be adapted depending on the scenario and also involve the technical constraints of available UAVs. Moreover, multi-UAV systems could further improve node coverage while decreasing message delivery delays.

With the presented approaches for cluster estimation and topology-aware path planning for in-transit node coverage, however, this paper provides valuable improvements to the successful application of Aerial Post-Disaster Communication Assistance Systems in unknown disaster areas.

ACKNOWLEDGMENT

This work has been co-funded by the LOEWE initiative (Hessen, Germany) within the *Nature 4.0 – Sensing Biodiversity* project and the *emergenCITY* centre, as well as the German Research Foundation (DFG) as part of the projects B1 and C2 within the Collaborative Research Center (CRC) 1053 MAKI – *Multi-Mechanisms Adaption for the Future Internet*.

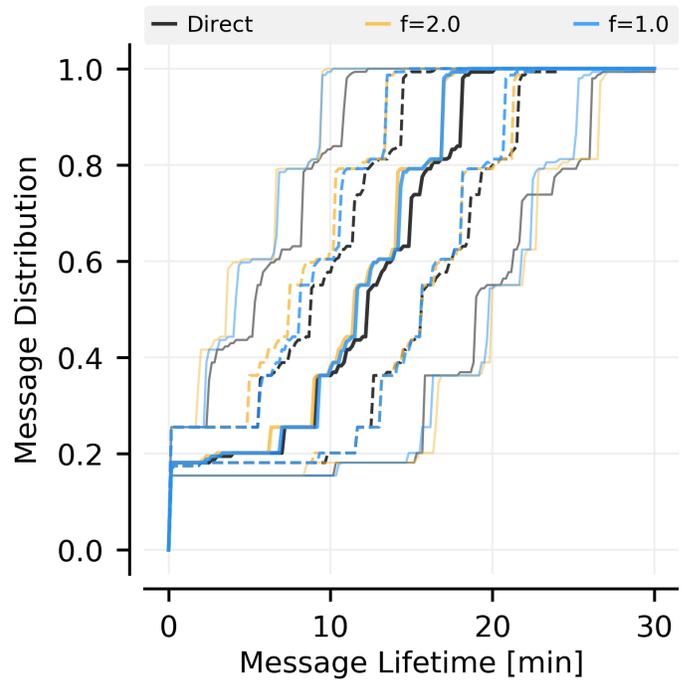


Fig. 7: Message distribution over 30 minutes message lifetime in the MOBILITY scenario. Bold lines denote the median, dashed lines 25th and 75th percentiles, outer lines 2.5th and 97.5th percentiles, respectively. Generally, messages are distributed faster with in-transit coverage. However, some messages are distributed slower than on direct paths due to the tradeoff between higher coverage and shorter flights.

REFERENCES

- [1] K. Alsabti, S. Ranka, and V. Singh, “An efficient k-means clustering algorithm,” 1997.
- [2] F. Álvarez, L. Almon, P. Lieser, T. Meuser, Y. Dylla, B. Richerzhagen, M. Hollick, and R. Steinmetz, “Conducting a Large-scale Field Test of a Smartphone-based Communication Network for Emergency Response,” in *Proceedings of the 13th Workshop on Challenged Networks*. ACM, 2018, pp. 3–10.
- [3] K. Anazawa, P. Li, T. Miyazaki, and S. Guo, “Trajectory and data planning for mobile relay to enable efficient internet access after disasters,” in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.
- [4] A. Belbachir, J. Escareno, E. Rubio, and H. Sossa, “Preliminary results on UAV-based forest fire localization based on decisional navigation,” in *Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2015, pp. 377–382.
- [5] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [6] J. Euler and O. von Stryk, “Optimized vehicle-specific trajectories for cooperative process estimation by sensor-equipped UAVs,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3397–3403.
- [7] Y. Hu, X. Yuan, J. Xu, and A. Schmeink, “Optimal 1d trajectory design for uav-enabled multiuser wireless power transfer,” *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5674–5688, 2019.
- [8] A. Khan, E. Yanmaz, and B. Rinner, “Information merging in multi-uav cooperative search,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 3122–3129.
- [9] A. Khelil, P. J. Marrón, C. Becker, and K. Rothenmel, “Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks,” *Ad Hoc Networks*, vol. 5, no. 5, pp. 531–546, 2007.

- [10] P. Lieser, N. Richerzhagen, T. Feuerbach, L. Nobach, D. Böhnstedt, and R. Steinmetz, "Take it or Leave it: Decentralized Resource Allocation in Mobile Networks," in *42nd IEEE Conference on Local Computer Networks (LCN)*, 2017.
- [11] P. Lieser, J. Zobel, B. Richerzhagen, and R. Steinmetz, "Simulation Platform for Unmanned Aerial Systems in Emergency Ad Hoc Networks," *Proceedings of the 16th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, 2019.
- [12] F. Mezghani and N. Mitton, "The potential of cooperative communications to speed up disaster relief operations," in *Information and Communication Technologies for Disaster Management (ICT-DM)*. Paris, France: HAL, 2019. [Online]. Available: <https://hal.inria.fr/hal-02375707>
- [13] A. Mori, H. Okada, K. Kobayashi, M. Katayama, and K. Mase, "Construction of a node-combined wireless network for large-scale disasters," in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. IEEE, 2015, pp. 219–224.
- [14] E. N. Mortensen, B. Morse, W. Barrett, and J. Udupa, "Adaptive boundary detection using "live-wire" two-dimensional dynamic programming," *Computers in Cardiology*, pp. 635–635, 1992.
- [15] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile Unmanned Aerial Vehicles (UAVs) for Energy-efficient Internet of Things Communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.
- [16] —, "Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8052–8066, 2017.
- [17] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: realizing multihop device-to-device communications," *IEEE Communications Magazine*, vol. 52, no. 4, 2014.
- [18] B. Pearre and T. X. Brown, "Model-free Trajectory Optimisation for Unmanned Aircraft Serving as Data Ferries for Widespread Sensors," *Remote Sensing*, vol. 4, no. 10, pp. 2971–3005, 2012.
- [19] Z. Qin, A. Li, C. Dong, H. Dai, and Z. Xu, "Completion time minimization for multi-uav information collection via trajectory planning," *Sensors*, vol. 19, no. 18, p. 4032, 2019.
- [20] G. Rémy, S.-M. Senouci, F. Jan, and Y. Gourhant, "SAR Drones: Drones for Advanced Search and Rescue Mission," *Journées Nationales des Communications dans les Transports*, 2013.
- [21] B. Richerzhagen, D. Stingl, J. Rückert, and R. Steinmetz, "Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications," *EAI International Conference on Simulation Tools and Techniques (SIMUTOOLS)*, pp. 1–6, 2015.
- [22] S. Rohde, M. Putzke, and C. Wietfeld, "Ad hoc self-healing of OFDMA networks using UAV-based relays," *Ad Hoc Networks*, vol. 11, no. 7, pp. 1893–1906, 2013.
- [23] A. Rubina, A. Artemenko, and A. Mitschele-Thiel, "Towards A Realistic Path Planning Evaluation Model for Micro Aerial Vehicle-Assisted Localization of Wi-Fi Nodes," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 169–176.
- [24] J. Sánchez-García, D. Reina, and S. Toral, "A distributed pso-based exploration algorithm for a uav network assisting a disaster scenario," *Future Generation Computer Systems*, vol. 90, pp. 129–148, 2019.
- [25] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An autonomous multi-UAV system for Search and Rescue," in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*. ACM, 2015, pp. 33–38.
- [26] D. Stingl, C. Gross, J. Rückert, L. Nobach, A. Kovacevic, and R. Steinmetz, "Peerfactsim.KOM: A simulation framework for peer-to-peer systems," in *Proc. High Performance Computing and Simulation (HPCS)*. IEEE, 2011, pp. 577–584.
- [27] H. Toya and M. Skidmore, "Cellular telephones and natural disaster vulnerability," *Sustainability*, vol. 10, no. 9, p. 2970, 2018.
- [28] A. Trotta, L. Montecchiari, M. Di Felice, and L. Bononi, "A GPS-Free Flocking Model for Aerial Mesh Deployments in Disaster-Recovery Scenarios," *IEEE Access*, vol. 8, pp. 91 558–91 573, 2020.
- [29] Université catholique de Louvain (UCL), "EM-DAT: The Emergency Events Database," 2019. [Online]. Available: www.emdat.be
- [30] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.
- [31] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2004, pp. 187–198.
- [32] J. Zobel, P. Lieser, B. Drescher, B. Freisleben, and R. Steinmetz, "Optimizing Inter-Cluster Flights of Post-Disaster Communication Support UAVs," in *44th IEEE Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 567–574.