DroneNet: Network Reconstruction through Sparse Connectivity Probing using Distributed UAVs

Dahee Jeong, So-Yeon Park, and HyungJune Lee[†]
Department of Computer Science and Engineering
Ewha Womans University, Seoul, South Korea
Email: {dahee.jeong, ewsy25}@i.ewha.ac.kr, hyungjune.lee@ewha.ac.kr

***Abstract*—In this paper, we consider a network reconstruction problem using UAVs where stationary ad-hoc networks are severely damaged in a post-disaster scenario. The main objective of this paper is to repair network by supplementing aerial wireless links into the stationary network to reconnect isolated ground networks each other with a limited number of UAVs. We propose a distributed motion planning that guarantees complete coverage to probe network connectivity from the air over stationary networks, while reducing duplicate coverage with other UAVs. Given the collected local connectivity information over region of interest, we deploy UAVs as relays into the locations of network holes to repair network-wide data delivery most effectively by formulating the problem into a binary integer program. Simulation results show that our network traversing algorithm outperforms a multi-agent exploration algorithm *Ants* in terms of complete coverage time, travel distance, and duplicate coverage. Also, our deployment optimization enhances network-wide routing performance compared to a practical baseline counterpart.**

I. Introduction

In catastrophic diaster scenarios, maintaining a reliable communication network through fast network repair is important for effectively sharing in-situ emergency information between victims and first responders. There have been efforts on utilizing autonomous unmanned terrestrial or aerial vehicles on a Region of Interest (ROI) [4]. These mobile vehicles can be used as effective communication resources to quickly reconnect isolated networks each other through the ad-hoc deployment.

An advantage of unmanned aerial vehicles (UAVs) compared to terrestrial vehicles is its physically less constrained movement for information gathering. The UAVs can retrieve data from the ground via the ground-to-air communication, relay them to other UAVs in the air-to-air communication, and send back to the ground via the air-to-ground communication. They can gather network collapse status with connectivity probing from the air, and also be deployed by themselves as communication relays if necessary.

We consider the major roles of UAVs as exploring network connection status over unknown ROI areas, and being deployed as ground-to-air and air-to-ground relays for autonomous network reconstruction. The challenges are 1) to design a distributed motion planning algorithm for sparse yet efficient connectivity probing over the damaged network, and 2) to locate network holes where the deployment of UAV relays helps to repair the damaged network.

There have been previous works to address the problem of multi-agent exploration in [6], [9], [11] mostly from robotics research community. In [9], [11], researchers propose simple distributed *Ants* algorithms simulating a colony exploration of ants while leaving pheromone traces during the environment traversing. Although [6], [7] present the *Brick&Mortar* algorithm and its variation that reduces duplicate coverage as opposed to the *Ants*, they suffer from somewhat computationally intensive loop closure procedures.

The problem of network hole detection and deployment has been studied in [1]–[3], [5], [10], [12] from network research community. [1], [3], [10] explore sensor deployment algorithms by finding network holes in sensor networks from more theoretical perspectives. Regarding the usage of aerial vehicles, aerial communication based on 802.11n performs poorly due to aerial link vulnerability [2], while some antenna extension can enhance the quality of aerial links [12]. Some researchers utilize UAVs to re-establish network connectivity with aerial deployment in [5]. However, network repair improvement with respect to network probing density and the optimal UAV deployment problem based on sparse connectivity information have not been investigated well.

In this paper, we aim to answer two key questions of 1) how to traverse a network efficiently with multiple UAVs in a distributed way without much duplicate coverage and 2) what the optimal UAV deployment algorithm based on tangible connectivity probing measurements should be to achieve a practical network recovery.

We propose a novel distributed motion planning algorithm based on independent and computationally light decisions among several pre-determined *zigzag* patterns. These patterns extend the local coverage as the UAVs are flying forward, while reducing duplicate coverage with other UAVs. Exploring the ROI area, the UAVs periodically probe network connectivity from the air toward stationary networks.

Once the network traversing procedure is completed, we find the optimal UAV relay positions that can repair network-wide data delivery most effectively. We formulate the problem into a binary integer program and obtain the optimal deployment strategy.

The rest of this paper is organized as follows: After describing system model in Sec. II, we present our network

[†]HyungJune Lee is the corresponding author.

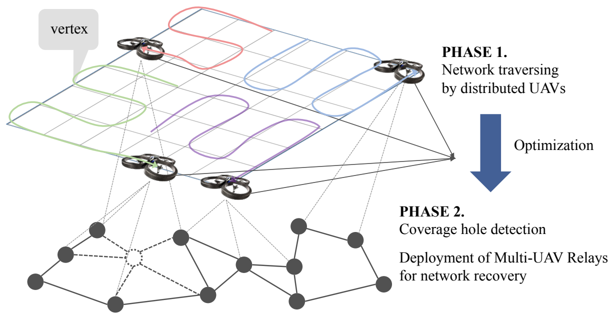
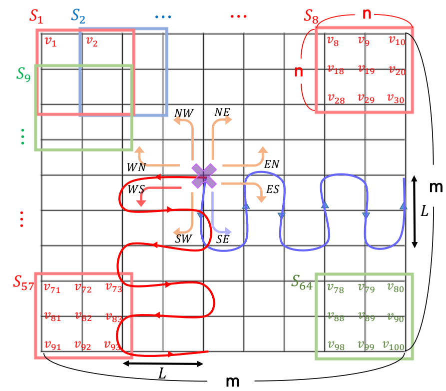
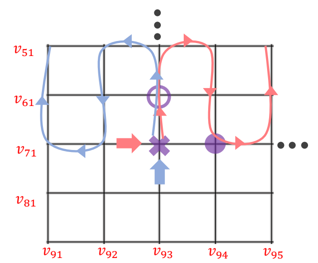


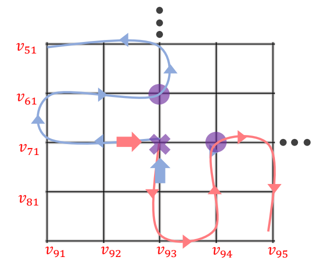
Fig. 1. Overall procedure of network traversing, coverage hole detection, and deployment by exploiting UAVs for autonomous network recovery.



(a) Logical grid coordinate consisting of vertexes, also showing eight pre-determined *zigzag* patterns for motion planning.



(b) Case 1: The East vertex toward the first quadrant with the longest length up to ROI is taken. Then, the pattern including the North vertex is chosen.



(c) Case 2: Both North and East vertexes toward the first quadrant with the longest length up to ROI are taken. Then, it finds a next longest pattern on another quadrant.

Fig. 2. Logical grid coordinate, *zigzag* movement trajectories, and future vertex visit decision rules.

traversing algorithm in Sec. III. In Sec. IV, we propose a UAV deployment algorithm based on binary integer optimization. After we evaluate our algorithms in a simulation environment in Sec. V, we finally conclude this paper in Sec. VI.

II. System Model

This work considers a network reconstruction problem using UAVs where stationary ad-hoc networks are severely damaged in a post-disaster scenario. Our goal is to repair network coverage by supplementing aerial wireless links into the stationary network to reconnect isolated ground networks each other with a limited number of UAVs.

We assume that UAVs are equipped with the same wireless radio as stationary nodes (e.g., 802.11 or 802.15.4). A UAV can communicate with a part of stationary nodes on the ground or other UAVs in the air as long as they are within radio range. It is also assumed that UAVs are aware of Region of Interest (ROI) to explore and can keep track of their relative position on ROI compared to its corresponding physical position. Any UAV control issues on moving from one location to another due to external environmental factors such as weather, obstacles, and collisions with other UAVs are out-of-scope in this paper. We consider UAVs to initially be fully charged and keep operating without recharging during a complete mission of network reconstruction.

The problem of network construction using UAVs can be divided into two sub-problems: 1) network connectivity probing from the air based on distributed motion planning of UAVs for the complete ROI coverage, while reducing duplicate coverage (refer to Sec. III), and 2) optimal UAV relay deployment for the most effective network recovery given a limited number of UAVs (refer to Sec. IV), as in Fig. 1.

III. Network Traversing

In this section, we propose our novel distributed motion planning algorithm of multiple UAVs and describe the procedure of network connectivity traversing of the UAVs. Multiple UAVs explore the network over Region of Interest (ROI) according to their own independent navigation decision. For an efficient distributed exploration on the ROI region, we define a frontier map that consists of square grids with $m \times m$ vertexes as in Fig. 2(a). Each UAV initiates its navigation at its currently visiting vertex or a designated vertex, continues its movement decision to the next vertex, and stops if it covers all of the vertexes on the ROI.

Each UAV initially generates a *future-vertex-visit-trajectory* with the longest length toward a certain direction up to the boundary of ROI among eight pre-determined *zigzag* patterns (North-East, North-West, South-East, South-West, East-North, East-South, West-North, West-South) as shown in Fig. 2(a). Whenever a UAV visits a vertex at a time, it adds the visited vertex ID to its *vertex-visit-list*. If two or more UAVs are within radio range, they share their own vertex-visit-list with others and merge them into its original vertex-visit-list.

When a UAV decides its next visiting vertex based on the future-vertex-visit-trajectory, it checks whether the anticipating visiting vertex has already been taken by other UAVs by searching it over the vertex-visit-list. In case that the anticipating vertex is already taken, the UAV lists up all available neighboring vertexes to move among (North, East, South, West), except the direction with the taken vertex, and randomly choose one direction for next move. In this way, a UAV is able to avoid duplicate exploration over the vertexes already visited by other UAVs in a distributed manner. It continues to generate a future-vertex-visit-trajectory with the longest length toward the boundary of ROI and execute its local visit decision

Algorithm 1 Distributed Multi-UAV Network Traversing

```
Input: CurrentVertexID
Output: NextVertexID
  // Part I: Motion planning
if (future-vertex-visit-trajectory == ∅) then
    Regenerate the future-vertex-visit-trajectory
      with the longest length that starts from an unvisited vertex;
    NextVertexID = future-vertex-visit-trajectory's first vertex ID;
    Move with one step to the next vertex;
else
    if future-vertex-visit-trajectory's next vertex is taken or null then
        future-vertex-visit-trajectory = ∅;
        if (any unvisited neighboring vertex in North, East, South, West)
        then
            NextVertexID = random-pick(unvisited neighboring vertexes);
            Invoke connectivity-probing();
            Move with one step to the next vertex;
        else
            if (there exists any unvisited vertex) then
                NextVertexID = the nearest vertex's ID on the grid coordinate
                from CurrentVertexID;
                Invoke connectivity-probing();
                Fly to the next vertex;
            else
                Terminate;
            end if
        end if
    else
        NextVertexID = future-vertex-visit-trajectory's next vertex ID;
        Invoke connectivity-probing();
        Move with one step to the next vertex;
    end if
end if
  // Part II: Connectivity probing
Function connectivity-probing()
    Broadcast hello packets;
    Receive response packets from neighboring stationary nodes;
    Calculate the average PRR for each responded stationary node;
    Update the PRR table for CurrentVertexID and StatonaryNodeId;
    if (any UAVs within radio range) then
        Exchange vertex-visit-list and PRR table, and update them
    end if
EndFunction
```

afterwards as illustrated in Figs. 2(b) and 2(c).

During each vertex visit, a UAV probes network connectivity with neighboring stationary nodes near the vertex. The UAV broadcasts hello packets with a periodic manner, and any neighboring stationary node that have received a hello packet replies back to the UAV with a response packet embedding its own node ID. Based on the collected response packets from connectable nodes for multiple hello packets, the UAV calculates the average Packet Reception Rate (PRR) for each responded node ID at the vertex position. As each UAV traverses over the network on the ROI, it continuously updates its PRR table for the attributes of visited vertex ID and stationary node ID and also exchanges its PRR table together with the vertex-visit-list if other UAVs are within radio range.

If a UAV checks that all of neighboring vertexes in the north, east, south, and west directions are taken, it compares its vertex-visit-list with the entire vertex list on ROI, and selects an unvisited vertex with the shortest distance on the grid coordinate for its next move. In this case, the UAV directly flies to the selected vertex. If there remains no vertex to visit, it finishes the network traversing procedure.

Our network traversing algorithm guarantees the complete coverage of vertexes with distributed motion planning of multiple UAVs and its successful termination without overlapping loops. The proofs are straightforward and omitted due to space constraints.

IV. UAV Relay Deployment

In this section, we present a UAV relay deployment algorithm that finds the best grid positions of multiple UAVs for the optimal network repair. Given the collected local connectivity information over Region of Interest (ROI), we find critical network holes that drastically undermine network-wide routing performance. We want to deploy a limited number of available UAVs as relays into the locations where local connection as well as end-to-end routing can significantly be improved.

Once the UAVs complete the network traversing procedure in Sec. III, we obtain the connectivity table consisting of Packet Reception Ratio (PRR) from stationary node ID k at vertex ID j, i.e., $\left[\ PRR\ \right]_{j,k}$ where $1 \leq j \leq M(= m^2)$ and $1 \leq k \leq N$. To find the network holes, we observe a set of vertexes that retain the weakest wireless links to the neighboring stationary nodes. Since the number of available UAVs is limited, we prioritize the network holes and select some of them as UAV relay deployment positions. It should be noted that UAVs should not be deployed into the network holes completely isolated by any neighboring stationary nodes because the deployed relay still remains unconnected to any of them.

To benefit the overall network from only few UAV relays for network reconstruction, we aim to minimize duplicate network coverage by prohibiting two or more UAVs from being deployed within communication range. Thus, we want each UAV to contribute to repairing its nearby network connectivity without partial or complete duplicate coverage for the overall network repair enhancement.

We formulate the problem of selecting grid positions of multiple UAVs for global network repair into a binary integer program. Our goal is to find a set of vertex regions that have the weakest non-zero PRRs averaged over neighboring stationary nodes, while avoiding duplicate coverage with any of other UAVs.

To formulate this setting, we first define a group of vertexes on a square sub-grid within the average radio range of a wireless interface as $S_i = \{v_{i_1}, v_{i_2}, v_{i_3}, \ldots, v_{i_{n^2}}\}$ where v_{i_l} $(1 \leq l \leq n^2)$ is a vertex element belonging to the set S_i, and n^2 is the total number of elements in set S_i, and $S_1 \cup S_2 \cup \cdots \cup S_K = \{v_1, v_2, \ldots, v_{m^2}\}$ as in Fig. 2(a). Given the P number of UAVs to deploy, the problem of selecting P grid positions of UAVs is to select the P number of sets with the lowest average PRRs over their corresponding belonging vertexes among $S_1, S_2, \ldots,$ and S_K, while any selected vertex sets should not share any vertex in common. Both S_i and S_j cannot be selected if $S_i \cap S_j \neq \emptyset$. For example, in Fig. 2(a), both S_1 and S_2 cannot be selected as deployment vertexes. This implies that we want to deploy a UAV into a group location of vertexes of which most or all suffer from similarly poor connection.

We introduce indicator functions J_i denoting the vertex group set S_i should be selected, and $I_{i,j}$ denoting whether the vertex group set S_i and its belonging vertex v_j should be selected.

Based on these notations, we define the objective function to minimize the summation of the average PRRs of the selected vertexes in the selected vertex set as follows.

$$\text{minimize} \quad \sum_{i,j \in S_i} \overline{PRR}_j \cdot I_{i,j} \tag{1}$$

$$\text{subject to} \quad \sum_{i} I_{i,j} \leq 1 \qquad \forall j \tag{2}$$

$$J_i = I_{i,i_1} = I_{i,i_2} = I_{i,i_3} = \cdots = I_{i,i_{n^2}} \quad \forall i \tag{3}$$

$$\sum_i J_i = P \tag{4}$$

where $\overline{PRR}_j$ is the average PRR over only the stationary nodes with non-zero PRRs at vertex v_j. In case that vertex v_j has no connection at all, i.e., $\overline{PRR}_j$ = 0, we force it to be 1 so that isolated vertexes should never be selected.

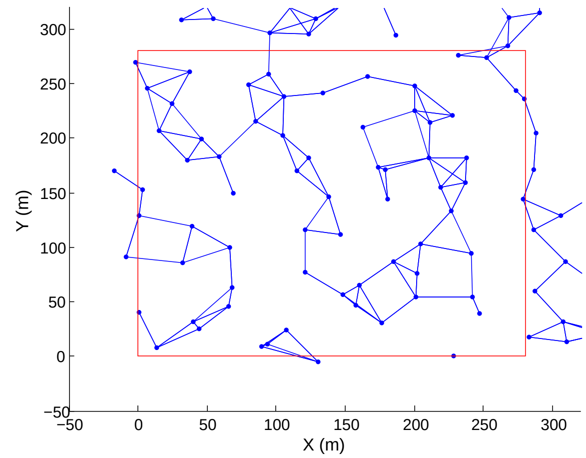
Constraint (2) ensures that any selected vertex sets should not share any vertex in common to avoid duplicate coverage. Constraint (3) enforces the condition that once a vertex group set S_i is selected, any belonging vertex $v_j \in S_i$ should be selected. The last constraint (4) requires the total number of selected vertex group sets to be the same number of UAVs.

By using MATLAB bintprog utility or AMPL/CPLEX solver, we can obtain the optimal sets of the most vulnerable vertex groups under critical link outage. Since each UAV ends up with the entire connectivity table for all the vertexes at the end of network traversing procedure, it calculates them for itself. Once each UAV tracks down to these sets, it determines one of sets according to the order of UAV ID, and flies directly to the center position of the selected vertex group for its self-deployment. These positions are exactly where the deployed UAVs can be used as crucial relay resources for starting repairing the broken network.

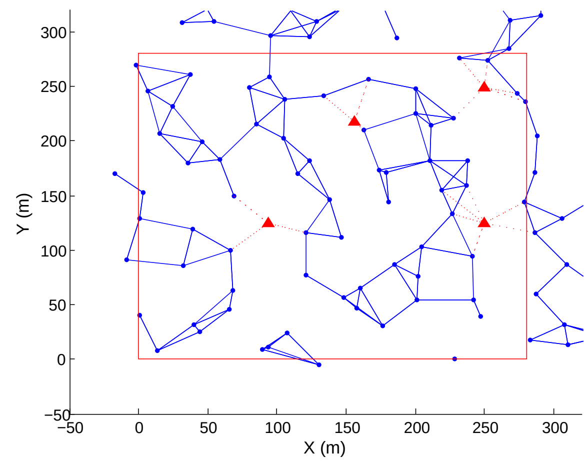
V. Evaluation

We validate our proposed scheme in a simulated damaged network of 218 stationary sensor nodes in a 500 × 500 m^2 area where the networks are sparsely connected each other with some degree of isolation. In particular, we focus on the ROI area in a 280 × 280 m^2 to which 65 stationary nodes belong as shown in Fig. 3(a). To simulate the radio propagation, we use a combined path-loss shadowing model with a path-loss exponent of 3, a reference loss of 46.67 dB, and an additive white Gaussian noise $N(0, 5^2)$ in dB [8].

We evaluate our network traversing algorithm compared to *Ants* algorithm [11] in terms of complete coverage time, travel distance, and duplicate coverage rate with respect to the number of UAVs in Sec. V-A. Then, we validate how our deployment algorithm improves the overall damaged network



(a) *Before* deployment.



(b) *After* deployment of 4 UAVs (denoted as red triangles) with *DroneNet*.

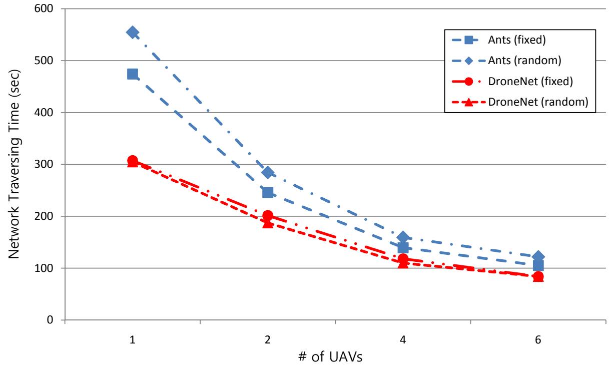
Fig. 3. Network connectivity of sensor nodes over ROI (within the red square boundary) in a simulated network (where wireless links are shown for PRR ≥ 75%).

in terms of end-to-end routing cost and routing hole fraction by varying the number of UAVs and the probing density of network traversing in Sec. V-B.

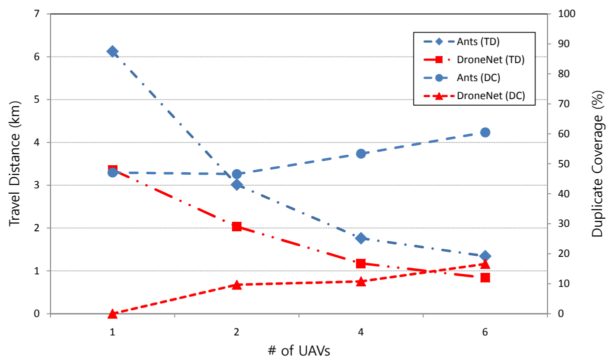
In our experiments, the total number of vertexes in grid over ROI, M is 100 where m = 10, and the total number of elements in vertex set S_i is 9 where n = 3 corresponding to the average radio range (as illustrated in Fig. 2(a)). 4 UAVs (i.e., P = 4) are deployed in the height of 9 m in the air. Each UAV calculates its PRR table based on 50 hello packets at a visited vertex, while $L = 1$ is used for the zigzag pattern width, unless otherwise noted.

A. Network Traversing

We investigate network exploration performance with respect to the number of UAVs. We measure network traversing time as complete coverage time by which all of UAVs terminate the traversing procedure, assuming that UAVs fly with the speed of 11.1 m/s (as per the specification of Parrot AR.Drone 2.0). As in Fig. 4(a), the complete coverage time

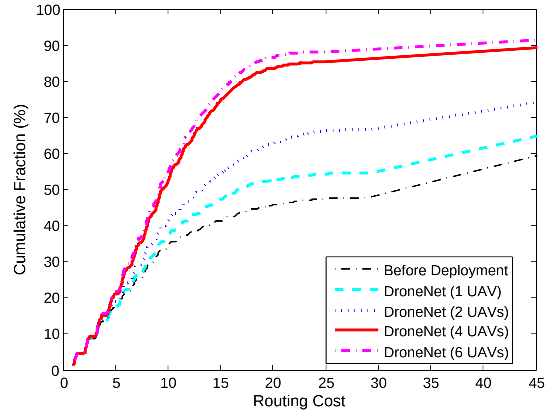


(a) Complete coverage time per UAV for network traversing.

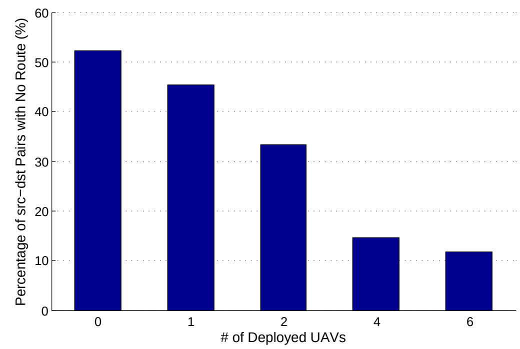


(b) Travel distance and duplicate coverage rate per UAV.

Fig. 4. Network exploration performance compared to *Ants* algorithm, with respect to the number of UAVs.



(a) Cumulative distribution of routing cost for *before* vs. *after* UAV deployment.



(b) Routing hole percentage with respect to the number of deployed UAVs.

Fig. 5. Network recovery performance in terms of end-to-end routing cost with respect to the number of deployed UAVs

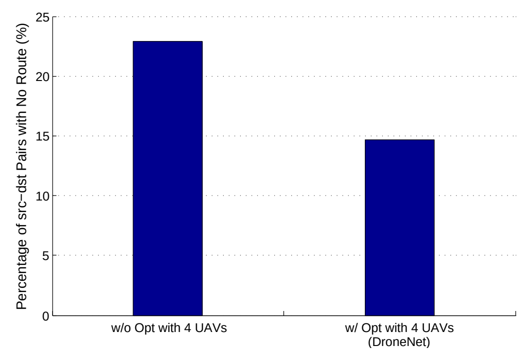


Fig. 6. Routing hole percentage for a heuristic algorithm without optimization vs. our *DroneNet* optimization algorithm.

of a UAV decreases as the number of UAVs increases for both algorithms, while our *DroneNet* outperforms *Ants* with a factor of up to 1.8. We also explore how the initial location of each UAV affects the performance. We let each UAV initially be located at a fixed vertex or a randomly selected vertex and launch its exploration. Our simulation result shows average performance over 10 runs. *DroneNet* shows very stable performance irrespective of the initial location of UAVs, whereas *Ants* is relatively more sensitive to where the UAVs initiate their exploration. We also measure the average travel distance of a UAV as in Fig. 4(b), showing a similar pattern as network traversing time.

To understand how *DroneNet* can result in small traversing time and distance, we measure how many vertexes each UAV has redundantly explored over the total number of visited vertexes as the duplicate coverage rate. As shown in Fig. 4(b), *DroneNet* keeps a small duplicate coverage rate around 10% with two and four UAVs, achieving less than 20% even with 6 UAVs, whereas *Ants* leads to almost 50% redundant coverage performance. This means that our motion planning algorithm minimizes the number of redundant vertex visits even if it is fully distributed, while guaranteeing complete coverage.

B. Network Recovery Performance

We investigate network recovery performance in terms of end-to-end routing cost with respect to the number of UAVs to deploy. The end-to-end routing cost is defined as the summation of per-hop link costs where the per-hop link cost is calculated as the expected number of transmissions on the link.

Fig. 5 shows that as a larger number of UAVs are deployed, the source-to-destination routing cost decreases. Given the simulated network topology, the most effective number of UAVs turns out to be 4. In other words, two more UAVs beyond 4 UAVs do not greatly improve routing performance

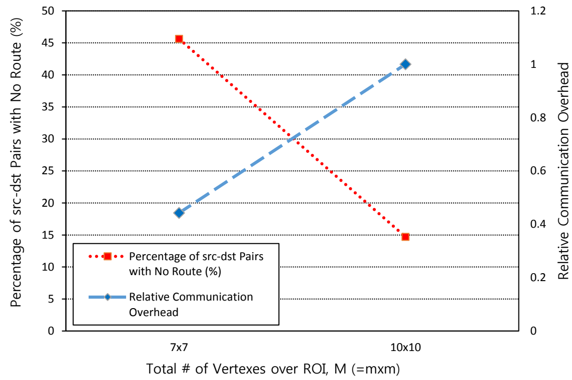


Fig. 7. Impact of sampling grid size on network communication overhead

further. This implies that depending on a sparsity of network topology, there exists an effective number of UAVs for the most influential routing enhancement.

We compare our deployment optimization algorithm against a heuristic deployment algorithm without optimization. For this comparison, both algorithms use the same PRR table achieved after network traversing. The only difference is that the heuristic algorithm deploys UAVs into the vertexes with the lowest PRRs excluding zero, whereas our algorithm applies a binary integer program-based optimization that reduces overlapped radio range among the deployed UAVs. As Fig. 6 demonstrates, our optimization technique lessens the number of source-to-destination pairs with no route (i.e., routing holes) with a factor of 1.6, compared to a heuristic approach.

Lastly, we would like to discuss how the probing density affects routing performance and communication overhead. We measure communication overhead as the accumulated packet transmissions for sending hello packets and response packets from each UAV, and exchanging the vertex-visit-list and the PRR table among UAVs. As shown in Fig. 7, as the probing density increases from 7×7 to 10×10, approximately by 2, we can achieve routing performance improvement with a factor of 3.1, while consuming more communication overhead with a factor of 2.3. This demonstrates that *DroneNet* can achieve a higher benefit of network-wide data delivery with a relatively smaller network overhead increase, showing an interesting trade-off relationship.

VI. Conclusion

We have presented a self-organizing UAV deployment algorithm based on sparse network status probing from the air along with a distributed motion planning. We have designed a network traversing algorithm based on a fully distributed local decision for its next movement, while minimizing the duplicate coverage and guaranteeing complete coverage. Then, we have formulated the problem of UAV relay deployment into a binary integer program that aims to spread them into the area under somewhat wide link outage, reducing overlapped coverage makeup.

Our experiment results indicate that our network traversing algorithm covers the complete ROI more effectively in terms of coverage time, travel distance, and duplicate coverage against a popularly used multi-agent exploration algorithm *Ants*. Also, our optimization technique finds an efficient deployment to repair the routing hole problem, while slightly increasing network overhead in return.

For future work, we would devise a UAV deployment algorithm that fundamentally infers the overall network topology based on only a few connectivity probing. By applying the compressive sensing signal processing theory with network tomography, we may obtain a more globally optimal solution for a drastic network reconstruction with only few UAVs. Also, the optimal motion planning of UAVs considering recharging in case of out-of-battery scenarios would be an interesting research direction.

Acknowledgment

This work was supported by the National Research Foundation of Korea funded by the Korean Ministry of Science, ICT, and Future Planning through the Basic Science Research Program under Grants NRF-2013R1A1A1009854, NRF-2013R1A2A2A04014772.

References

[1] N. Ahmed, S. S. Kanhere, and S. Jha. The holes problem in wireless sensor networks: a survey. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(2):4–18, 2005.

[2] M. Asadpour, D. Giustiniano, and K. A. Hummel. From ground to aerial communication: dissecting wlan 802.11 n for the drones. In *Proceedings of the 8th ACM international workshop on Wireless network testbeds, experimental evaluation & characterization*, pages 25–32. ACM, 2013.

[3] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus. Deploying sensor networks with guaranteed capacity and fault tolerance. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 309–319. ACM, 2005.

[4] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3602–3608. IEEE, 2004.

[5] M. Di Felice, A. Trotta, L. Bedogni, K. R. Chowdhury, and L. Bononi. Self-organizing aerial mesh networks for emergency communication. In *Proceedings of the 25th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2014 - Mobile and Wireless Networks)*, 2014.

[6] E. Ferranti, N. Trigoni, and M. Levene. Brick& mortar: an on-line multi-agent exploration algorithm. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 761–767. IEEE, 2007.

[7] E. Ferranti, N. Trigoni, and M. Levene. Rapid exploration of unknown areas through dynamic deployment of mobile and stationary sensor nodes. *Autonomous Agents and Multi-Agent Systems*, 19(2):210–243, 2009.

[8] A. Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

[9] N. Hazon, F. Mieli, and G. A. Kaminka. Towards robust on-line multi-robot coverage. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1710–1715. IEEE, 2006.

[10] V. Isler, S. Kannan, and K. Daniilidis. Sampling based sensor-network deployment. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 1780–1785. IEEE, 2004.

[11] J. Svennebring and S. Koenig. Building terrain-covering ant robots: A feasibility study. *Autonomous Robots*, 16(3):313–332, 2004.

[12] E. Yanmaz, R. Kuschnig, and C. Bettstetter. Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility. In *INFOCOM, 2013 Proceedings IEEE*, pages 120–124. IEEE, 2013.