Time-Dependent Ad-Hoc Routing Structure for Delivering Delay-Sensitive Data Using UAVs

JINYI YOON[1], SUNGJU DOH[1], OMPRAKASH GNAWALI[2](Member, IEEE), AND HYUNGJUNE LEE[1](Member, IEEE)
[1]Ewha Womans University, Seoul 03760, South Korea
[2]University of Houston, Houston, Texas 77204, USA

Corresponding author: HyungJune Lee (e-mail: hyungjune.lee@ewha.ac.kr)

ABSTRACT In disaster scenarios where communication networks have broken down, it is important to ensure a reliable data delivery from an emergency operations center to the local target nodes within its effective time limit. We propose a hybrid data delivery mechanism that exploits the *load-carry-and-delivery* by UAVs with a mixture of localized ad-hoc routing over partially connected terrestrial networks. We aim to achieve reliable on-time data delivery to the target nodes, while preserving the low routing cost. Our proposed routing methodology consists of three steps: 1) localized network construction, 2) network probing by UAVs, and 3) localized ad-hoc routing based on a dynamic depth routing tree depending on the data urgency. Here, we present an innovative cost-effective local data sharing structure called *localized minimal routing tree* that balances with the direct data delivery by UAVs. After the initial network setup and probing procedure, each UAV makes a series of near-optimal decisions of *which grid points to visit* considering its localized network topology and data urgency. Our time-dependent routing mechanism dynamically decides data recipient nodes to serve more urgent data delivery with a higher priority at a time. The simulation experiments validated our combined path planning and routing approach, achieving 71% higher reliability than the best possible performance by using only the network nodes and consuming 20% lower energy than the UAV-only approach, while maintaining high reliability. Thus, our work makes a strong case for systematically combining the two approaches.

INDEX TERMS Time-Dependent Routing, Delay Constraints, Load-Carry-and-Delivery, Vehicular Ad-hoc Networks, Unmanned Aerial Vehicles, Ad-hoc Data Delivery

I. INTRODUCTION

Emergency preparedness against the event of natural or man-made disasters has increasingly become one of the vital prerequisites for protecting civilian lives and urban facilities. Under these disaster situations, the telecommunications infrastructure is likely to have broken down because of the physical destruction of the network devices or network congestion [1]. The dynamically changing status or disaster response information tends to be both time-wise and location-wise sensitive. For example, the information for a highly damaged area needs to be delivered within a relatively tight deadline. With only the partially or completely collapsed network infrastructure, the distribution of critical information to users is a considerably challenging task with severely degraded node reachability.

To tackle this challenge, unmanned aerial vehicles (UAVs) can be leveraged as extraordinary information messengers, delivering data from an emergency operations center to the affected areas or users in isolated ad-hoc networks. In catastrophic disaster situations, where the timely data delivery should be reliable and stable, UAVs can contribute to relaying packets while constructing a flying ad-hoc network themselves [2], [3]. Moreover, UAVs can assist the existing ground networks as a substitute network on top of sparsely connected networks [4], [5]. Depending on the urgency degree and the location of the data to be delivered by a UAV, it can change its own navigation priority to maximize the percentage of successful data delivery given each distinct time constraint,

across a certain region of interest (RoI).

Regarding the utilization of UAVs as an additional network tier, several interesting works [6]–[8] have been proposed. The prior works consider UAVs as data ferries in a *load-carry-and-delivery* manner [9], [10]. By controlling their mobility to determine an efficient navigation path, they aim to deliver data to the corresponding targets at a reduced routing cost and with increased throughput. Although the load-carry-and-delivery paradigm frames the problem of data routing as *path planning*, it relies heavily on the direct data delivery from the UAVs to the local nodes, not fully utilizing the partially connected networks.

In this paper, we present a hybrid data delivery approach that not only considers the direct UAV data delivery but also takes advantage of the ad-hoc routing strategy in the locally connected networks. In a situation wherein various data need to be delivered from an emergency operations center to the designated target nodes within each distinct deadline, we aim to maximize the successful on-time data delivery rate to the local nodes, while preserving the low routing cost.

Without using any prior knowledge of the network topology over RoI, the proposed approach consists of two phases: 1) initial network probing and local routing tree construction, and 2) time-dependent routing execution. We let both the UAVs and the ad-hoc nodes perform the punctual data delivery task in a collaborative fashion. Thus, the UAVs can sparsely navigate the RoI area to drop off the data packets only to certain selected *delegate nodes*, which will be in charge of the local ad-hoc data delivery by using partially connected static networks.

The advantages of the proposed approach are two-fold: First, by decreasing the number of visiting places, the UAVs can reduce the data carrying (traveling) time, eventually earning more time to perform more urgent data delivery to other terrestrial nodes in need. Second, by using a time-dependent and cost-effective local data sharing structure called *localized minimal routing tree*, the chances of on-time data delivery to a designated target node would increase because of benefiting from the low network latency as compared to that in the case of direct *node-to-node* delivery.

During the initial setup phase, the terrestrial nodes construct their own neighbor node list by sending periodic or aperiodic broadcast packets. We extract the strongly connected nodes and group them as a virtual node called a *minimal graph* by using a *graph coarsening* procedure. After sharing each minimal graph with the neighboring minimal graphs up to M hops, we ensure that each node obtains an M-hop local network and finally builds an M-depth shortest-path tree having a minimal graph of the node as a root.

The UAVs initiate network probing to acquire all the possible localized M-depth trees from the terrestrial nodes via certain grid points by navigating over the grid-based RoI, instead of visiting each individual node. A UAV records the communicable nodes within its radio range at each visited grid point along with its own localized M-depth tree. To quickly complete the network probing with multiple UAVs,

we perform an almost equal distribution of the grid points to visit according to the number of UAVs. We apply the m-TSP genetic algorithm (*mTSP-GA*) [11] with some improvements to find each optimal fair path among multiple agents by iteratively trying out various path evolutions.

After the initial network probing, the proposed approach begins the routing execution where the UAVs deliver data with time constraints from an emergency operations center. Each UAV makes intelligent decisions of *which grid points to visit* considering the data urgency and the localized network topology to maximize the on-time data delivery in a fully distributed manner. When the UAV decides to visit a certain grid point, it composes the data that can be delivered via the communicable delegate nodes by using their localized tree. A delegate node constructs the localized routing tree by dynamically cutting some leaf or branch minimal graphs with a relatively large packet deadline to reduce the unnecessary data delivery with less urgency. The delegate node sends the node list on the finalized routing tree that will be served at this time back to the UAV so that it can reschedule the postponed data delivery for the nodes cut from the original M-depth tree, later through another delegate node at a future grid point visit.

Although UAV-based data delivery and ad-hoc routing were previously explored separately to improve data delivery in disrupted networks, the extent to which their simultaneous use improves the networking performance is unknown; however, the design of algorithms to systematically combine these two approaches is now feasible. To the best of our knowledge, this study is the first to exploit both the mobility of UAVs and the localized routing structure under time constraints for the problem of data delivery from UAVs. The proposed approach achieved 71% higher reliability than the best possible performance by using only the network nodes and consumed 20% lower energy than the UAV-only approach, while maintaining high reliability. Naive ways of combining the two approaches would result in a far lower gain as shown by our work. Thus, our work makes a strong case for systematically combining the two approaches.

Our main contributions can be summarized as follows:

- We introduce network graph coarsening to effectively extract the underlying routing skeletons from localized ad-hoc networks and generate a localized tree structure network for routing efficiency.
- We propose a novel routing framework such that UAVs only need to deliver data to a root node of the localized tree structure network, and then, the data can be effectively spread from the root node to its leaf nodes, instead of direct delivery to each node.
- We present a time-dependent ad-hoc routing structure that dynamically decides whom to be transferred and whom to be cut from a localized routing tree, depending on the data urgency for punctual data delivery over the network.
- Both UAV operations and communication require energy. We evaluated the tradeoffs between these two

Perspective	Reference	Problem	Method and Contribution
Ad-Hoc Routing	Gnawali *et al.* [12]	Design a routing layer to remain efficient, robust, and reliable in highly dynamic network topologies	Collection Tree Protocol (CTP): dynamically probes and validates the routing topology using data packets with fast recovery when the datapath finds a possible problem
	Xu *et al.* [13]	Design self-configuring networks that exploit redundancy to reduce energy consumption while preserving the fidelity in ad-hoc wireless networks	Geographical Adaptive Fidelity (GAF) : uses node deployment density and routing redundancy to increase the node duty cycle and to extend the network lifetime
	Taha *et al.* [14]	Highlight the energy consumption in MANET by applying the fitness function technique to optimize the energy consumption in AOMDV routing protocol	AOMDV with the fitness function (FF-AOMDV) : the fitness function is used to find the optimal path from source to destination to reduce the energy consumption in the multipath routing
UAV Message Ferrying	Cheng *et al.* [9]	Maximize throughput between two distant ground nodes using UAVs as message relays	Load-Carry-and-Delivery (LCAD) paradigm : UAV loads data from a source node, carries, and delivers the data to a destination node
	Oubbati *et al.* [15]	Detect incidents on the road, summon rescue team, and find the fastest path with UAVs under the road constraints	A robust routing scheme considering both the mobility and the energy constraints of UAVs
	Oubbati *et al.* [16]	Guarantee a communication stability by balancing the energy consumption of UAVs	A robust route detection considers the energy balancing, the link breakage prediction, and the connectivity degree of the discovered paths
	Harounabadi *et al.* [7]	Multiple mobile agents cooperate to deliver messages for disconnected wireless static nodes	A multi-hop data delivery is designed through efficient message forwarding and replication among UAVs
	Harounabadi *et al.* [17]	Path planning of multiple UAVs to deliver messages among isolated wireless nodes	The multiple TSP genetic algorithm optimizes the average weighted delay of message delivery
	Yoon *et al.* [8]	Find the optimal paths of UAVs to maximize the on-time data delivery to static nodes within each designated packet deadline	A distributed path planning determines the next visiting point considering travel time and remaining deadline, and a task division mechanism collaboratively distributes the unvisited grid points among encountered UAVs
Collaborative Routing with UAVs	Oubbati *et al.* [5]	Provide the reliable data delivery and guarantee paths by UAVs to ensure a durable connectivity under sparse networks	A greedy forwarding in VANETs is implemented. If disconnected, maintenance strategies using UAVs are employed
	Park *et al.* [18]	Repair the severely collapsed network by exploiting UAVs into the isolated ground networks	Perform a network probing to find out recovery points and deploy UAVs as intermediate relays nodes
	Ours	Ensure a reliable data delivery to multiple target nodes, while preserving the low routing cost under partially connected networks	A hybrid data delivery exploits the load-carry-and-delivery by UAVs with a mixture of localized ad-hoc routing

TABLE 1. A comparison of data delivery mechanisms

sources of energy expenditure to achieve a reliable delivery of data in disrupted networks.

II. RELATED WORK

The delay-sensitive data delivery problem in ad-hoc networks consisting of both static and mobile nodes has been studied mostly in two categories: by controlling the movement path of the mobile nodes for data delivery in the wireless networking community, or by finding efficient navigation paths of mobile agents in the robotics community.

Prior works on mobility control for data delivery use the *message ferry* paradigm [19], [20] that carries and directly delivers data to the designated target nodes or collects data from the local source nodes by finding optimal movement paths in terms of the coverage efficiency. In particular, UAVs have been used as flying message ferries based on load-carry-and-delivery (LCAD) [8], [9]. LCAD aims to maximize the throughput between two ground nodes by engaging UAVs to relay messages [9]. Inspired by the LCAD paradigm, a delay-sensitive data delivery to multiple targets using UAVs is proposed [8]. However, these works rely only on the direct door-to-door information delivery, lacking the possibility of utilizing partially connected ad-hoc networks.

Recently, there have been some studies on UAV-assisted routing [7], [10]. These works aim to solve the reliable data delivery problem in the UAV-assisted ground networks. Further, with an increasing interest in practical system issues, recent works have considered restricted energy capacity of UAVs [15], [16], [21].

The problem of finding the efficient navigation trajectories of mobile agents has been investigated as *path planning* by robotics and operation research communities [22]. Traditionally, prior works have been studied under the context of the traveling salesman problem (TSP) [23], [24] or the vehicle routing problem (VRP) [25], [26]. More closely related to our work requiring deadline constraints, deadline-TSP [27], [28] and VRP with time windows [29], [30] have been proposed by formulating the problem with time constraints into an optimization problem. Furthermore, motivated by evolutionary biology, some generic algorithm-based path planning mechanisms have been proposed in the UAV context [31], [32]. Although these works have provided strong theoretical results on space exploration, they may not be directly applicable to the ad-hoc routing problem with UAVs, missing the significant packet routing part.

However, not much work has been done on exploiting both UAV data ferries and static ad-hoc networks for achieving both routing agility and efficiency in the data delivery with

time constraints. Our work provides a novel hybrid path planning and routing strategy balancing between the load-carry-and-delivery by UAVs and the ad-hoc delivery by static nodes.

We summarize the problem and the methodology of previous approaches from the perspective of how much the UAV delivery agent is engaged, as in Table 1. We divide the data delivery schemes into three cases: 1) ad-hoc routing without using any UAV delivery agent; 2) UAV message ferrying; and 3) collaborative ad-hoc routing with UAVs.

III. SYSTEM OVERVIEW

We design a routing framework such that UAVs and local ad-hoc nodes collaborate with each other for ensuring on-time data delivery from one source location (e.g., an emergency operations center) to multiple nodes in the network. We consider a scenario in which the network in the RoI is partially connected, including numerous isolated sub-networks. We assume that there are a significant number of broken routing paths from one node to another in the terrestrial ad-hoc networks unless additional network devices such as UAVs are deployed. Under the severely disrupted networks, there is a limit to data delivery only with node-to-node routing strategies. The objective of utilizing UAVs under the disrupted network environment is to deliver timely data to target user without interruption. UAVs can be a way to deliver any kind of time-sensitive service information (e.g., urgent announcement with time expiry or evacuation plan in disaster scenarios).

We do not assume any prior knowledge of the network over the RoI, such as the node locations and the connection status with the neighbor nodes. We assume that the customized data for each node with its specific delivery deadline is ready to be picked up by the UAVs from a source node in one delivery round and that the packet deadline can range from tens of seconds to several minutes.

Both UAVs and target nodes are assumed to be equipped with the same wireless radio interface as the terrestrial ad-hoc nodes (e.g., 802.11 or 802.15.4) and the same wireless band or channel; therefore, a UAV can communicate with another UAV or the terrestrial ad-hoc nodes within its radio range. The UAVs can fly over a virtual grid topology on the RoI without experiencing any physical interference with the environments or other objects including other UAVs. Furthermore, we assume that a UAV is installed with a storage device sufficiently large to load the data for the terrestrial nodes in the RoI from the emergency operations center. Although UAV control-related issues such as obstacle, collision avoidance, or skewed movement are important factors and investigated [33]–[35], they are not mainly focused on, and out of scope in this paper.

Our goal is to find the optimal navigation trajectory of each UAV that maximizes the number of nodes with successful on-time data reception by reducing the duplicate coverage among the UAVs. From the network perspective, we aim to design a simple localized yet efficient routing structure that
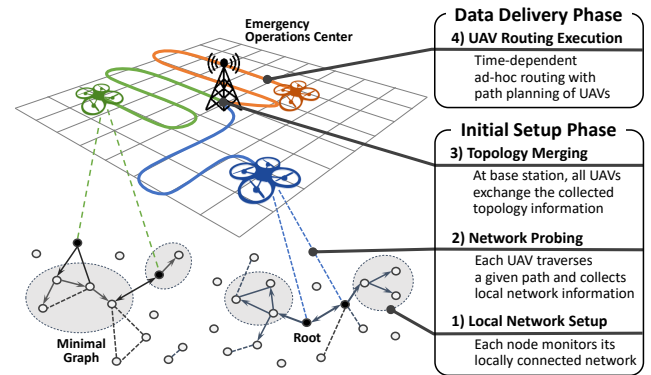

FIGURE 1. Overview of our time-dependent ad-hoc routing using UAVs

can reduce the overall delivery cost. The problem of data delivery that we aim to solve is to find *which grid points* to visit and then drop off a set of data that are supposed to be delivered to the selected target nodes within the local routing range, maximizing the punctual delivery rate. To this end, we seek a balanced routing decision between the direct UAV delivery and the network delivery with an optimized mixture under time constraints.

A. PROCEDURE

The proposed data delivery scheme using UAVs consists of three phases: local network setup, network probing, and UAV routing execution. A high-level illustration is provided in Figure 1.

1) Local Network Setup

During this setup process, each ad-hoc node constructs a brief neighborhood network by extracting groups of strongly connected nodes. First, on the basis of regular beaconing, each ad-hoc node maintains a neighbor node list where the node beyond a certain link quality threshold is considered a neighbor node. Second, a set of nodes with strong link quality is considered a single virtual node called the *minimal graph*, which is a fundamental routing skeleton. Third, each ad-hoc node shares its belonging minimal graph with its neighbors up to M hops and receives the neighboring nodes' minimal graphs from them. It finally forms its localized M-depth graph and tree structure as the root itself. In this procedure, each ad-hoc node maintains a simplified yet communication-wise and computation-wise efficient graph for effective data distribution toward its leaf nodes upon receiving the data from a UAV. More details can be found in Section IV-A.

2) Network Probing

In order for UAVs to collect the child node list at each local ad-hoc node and exploit it for effective path planning and the resulting data delivery, UAVs probe a given network by navigating a virtual grid topology. Upon visiting a specific grid point, a UAV records the connectable nodes and their own localized network information. All of the possible UAVs

start their own probing procedure from a center after being assigned the grid points to visit. To equally distribute the exploration area, a partitioning algorithm is executed to make all of the UAVs complete the data delivery mission as soon as possible. Refer to the detailed probing procedure in Section IV-B.

3) UAV Routing Execution

After the initial setup by the UAVs and the terrestrial ad-hoc nodes, the entire network can gain the benefit of the data delivery punctuality and efficiency from the collected local network information. All of the UAVs share the collected information about the communicable nodes at a grid point and their localized routing tree before starting the routing execution phase. Given this information, a UAV performs its own distributed navigation decision of which grid point to visit at a time for maximizing the on-time delivery service rate for all the target nodes. As a UAV cannot be aware of the progress status of the other UAVs unless they are encountered within the radio range, it makes its best effort to reduce duplicate coverage. Upon an encounter with another UAV, it shares the visited grid point list and performs collaborative task division for quick completion.

Once a UAV visits a specific grid point, it compiles all of the data that can be delivered via the grid point by checking the localized routing tree of the communicable (delegate) nodes and broadcasts them with one packet. Then, the delegate nodes run their own localized M-depth routing tree and dynamically trim certain nodes or branches over the routing path depending on the remaining delivery time. For example, if there exists a target node with a relatively large remaining packet delivery time, the current delegate node cuts the node from its routing tree at that time. There should be an opportunity later through another delegate node when its data delivery has become a little more urgent with lesser remaining time. The detailed procedure is presented in Section V.

Note that once the initial setup phase of the local network setup, network probing, and topology merging is executed, it does not necessarily take place again nor always with UAV routing, depending on the network dynamics.

IV. LOCALIZED ROUTING TREE WITH UAV PROBING

In an emergency situation where the communication network has (partially) collapsed, it is important to ensure a reliable communication channel from an emergency operations center to the target nodes in the affected areas. The information would likely be time-sensitive and location-sensitive: Each target node would need to receive the location-dependent data created at the center before the current information becomes outdated.

Direct data delivery from the center to the target nodes (or the nearby grid points) can reduce the energy consumption at the terrestrial ad-hoc nodes, while incurring high packet delivery time and possibly missing its deadline for the navigation.

We attempt to leverage the partially connected local networks with the UAV delivery at a few selected grid points. The use of local networks for data delivery may incur a slightly higher routing cost but improve the on-time packet delivery performance by reducing the number of visit points. We use this tradeoff relationship between the routing cost and the routing reliability in the problem of UAV data delivery.

In this section, we present the initial setup procedures for the local nodes and the UAVs. Each ad-hoc node builds a lightweight yet efficient local routing tree as the root based on its neighbor table sharing with its neighbors. UAVs perform network probing to capture the local network topology information of the connectable node list at each given grid point and the local routing trees for the nodes as the root. To quickly complete the probing with multiple UAVs, we want to almost equally partition the probing area into multiple sub-areas so that a UAV becomes in charge of an assigned sub-area.

A. LOCAL ROUTING TREE CONSTRUCTION

We construct a localized routing tree at each ad-hoc node that captures the essential *skeletons* of the routing topology. Each ad-hoc node monitors its vicinity by beaconing and discovers its connectable neighbor nodes that satisfy the minimum link quality threshold in terms of the packet reception ratio (PRR). The high-level representation of neighborhood connectivity can be illustrated as a graph as in Figure 2(a).

Data delivery based on the original connectivity graph often includes very detailed information and incurs a high computation and communication cost in the ad-hoc sensor networks.

To address this scalability issue, we abstract the original connectivity graph into a logically condensed graph called the *coarsened graph*. During this *coarsening* procedure, we apply a pioneering theoretical work of the multi-level partitioning algorithm [36] with some modifications in the edge cost calculation.

First, each node calculates the PRR of the communicable nodes. We denote $\overline{PRR}_{N_i \to N_j}$ as the PRR of the link from node i to node j. The directional link with a PRR value larger than or equal to the minimal threshold θ_{LK} is considered a valid communication link.

Second, we perform a graph coarsening procedure. Among the nodes, as shown in Figure 2(a), we partition a set of nodes that have the PRRs of both bi-directional links larger than or equal to θ_{MG} into one virtual node, called the minimal graph. The nodes belonging to a minimal graph can be considered as the *strongly connected nodes*. We denote MG_{N_i} as the minimal graph to which node i belongs, and $\eta_{MG_{N_i}}$ as a set of nodes belonging to the minimal graph as node i. For example, $\eta_{MG_{N_{n_1}}} = \{N_{n_1}, N_{n_2}, \ldots, N_{n_l}\}$ means that nodes $n_1, n_2, \ldots,$ and n_l belong to the same minimal graph $MG_{N_{n_1}}$ (or $MG_{N_{n_2}}$, etc.). If a node does not have any other strongly connected nodes, its minimal graph only embeds itself.
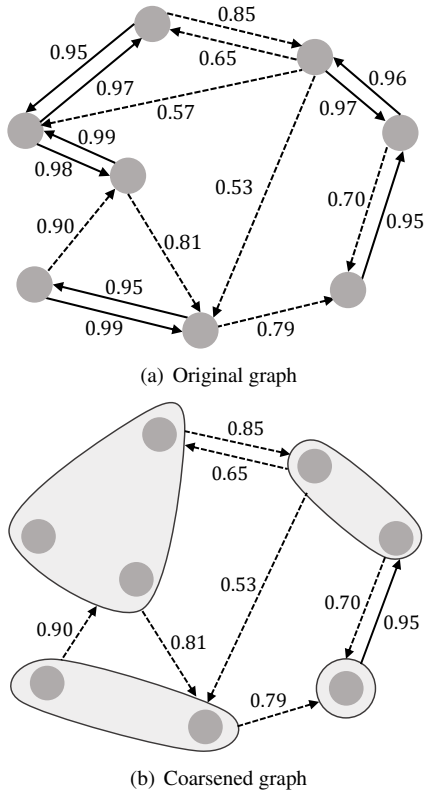
(a) Original graph

(b) Coarsened graph

FIGURE 2. Network graph coarsening from an original graph to its coarsened graph consisting of minimal graphs (where $\theta_{LK} = 0.5$ and $\theta_{MG} = 0.95$).

To determine the link status between the minimal graphs, we take the maximum PRR among the multiple directional links toward one minimal graph. For example, an original graph (Figure 2(a)) can be simplified as a coarsened graph consisting of minimal graphs in Figure 2(b).

Once each ad-hoc node constructs its minimal graph with the neighbor nodes, the minimal graph is propagated up to M hops (on the virtual node basis). After the constructed minimal graphs are shared among the other minimal graphs, node i that belongs to its minimal graph MG_{N_i} maintains an M-depth graph denoted as $MDG(MG_{N_i})$, which is a localized network of minimal graphs.

Finally, each ad-hoc node transforms its M-depth graph into a shortest-path tree by having itself as the root. We use the expected number of transmissions, i.e., $1/PRR$ as the weight of the directional edge. We complete the ad-hoc network setup by having M-depth local routing tree $SPT(MG_{N_i})$ for node i.

The detailed local routing tree construction algorithm is described in Algorithm 1. The complexity of coarsening a network graph is $O(|neighborNode(N_n)|)$ to find 1-hop strongly connected neighbors at node n, and $O(|\eta_{MG_{N_n}}| \cdot |neighborNode(N_s)|)$ to construct a 1-hop neighbor MG set at each 1-hop strongly connected neighbor node N_s, respectively.

Algorithm 1 Local Routing Tree Construction (by Each Node)

1: **Input:** *Its own node* n, θ_{LK}, θ_{MG}, M-depth
2: **Output:** *Local routing tree (having node n itself as root)*

  // I. Make 1-hop Neighbor Node List
3: $neighborNode(N_n) = \emptyset$; // 1-hop neighbor node set of node n
4: **for** node i that has ever sent ACK back **do**
5:    **if** $\overline{PRR}_{N_n \to N_i} \geq \theta_{LK}$ **then**
6:      $neighborNode(N_n) = \{N_i\} \cup neighborNode(N_n)$;
7:    **end if**
8: **end for**

  // II. Coarsen network graph
9: $\eta_{MG_{N_n}} = \{N_n\}$; // Initialize a minimal graph including itself
10: **for** node $i \in neighborNode(N_n)$ **do**
11:   **if** $\overline{PRR}_{N_n \to N_i} \geq \theta_{MG}$ && $\overline{PRR}_{N_i \to N_n} \geq \theta_{MG}$ **then**
12:     $\eta_{MG_{N_n}} = \eta_{MG_{N_i}} \cup \eta_{MG_{N_n}}$;
13:   **end if**
14: **end for**
15: $neighborMG(MG_{N_n}) = \emptyset$; // 1-hop neighbor MG set of MG_{N_n}
16: **for** node $s \in \eta_{MG_{N_n}}$ **do**
17:   **for** node $t \in neighborNode(N_s)$ **do**
18:     **if** $MG_{N_s} \neq MG_{N_t}$ **then**
19:       $neighborMG(MG_{N_n})$
         $= MG_{N_t} \cup neighborMG(MG_{N_n})$;
20:       $\overline{PRR}_{MG_{N_n} \to MG_{N_t}}$
         $= \max_{N_i \in \eta_{MG_{N_n}}, N_j \in \eta_{MG_{N_t}}} \overline{PRR}_{N_i \to N_j}$;
21:     **end if**
22:     $Weight_{MG_s \to MG_t} = 1/\overline{PRR}_{MG_s \to MG_t}$;
23:   **end for**
24: **end for**

  // III. Construct M-depth network and its shortest-path tree
25: $localNet(MG_{N_n}) = neighborMG(MG_{N_n})$;
26: **for** M-1 times **do**
27:   Send $localNet(MG_{N_n})$ to $MG_m \in neighborMG(MG_{N_n})$;
28:   Receive all $localNet(MG_m)$ from neighbor MGs and merge them;
29: **end for**
30: Make its own shortest-path tree $SPT(MG_{N_n})$ with MG_{N_n} as root from M-depth network;

B. NETWORK PROBING BY UAVS

Multiple UAVs collect the local connectivity information over terrestrial ad-hoc networks through network probing over the RoI. This network probing requires three steps: 1) partitioning the probing area, 2) detecting the local networks, and 3) merging the localized topology from all the UAVs into a complete network topology.

To quickly explore the terrestrial ad-hoc networks by using multiple UAVs, we partition all the grid points by the number of UAVs and assign each partitioned grid point list to a UAV. We apply *mTSP-GA* [11] to find the near-optimal shortest path for each UAV. The *mTSP-GA* tries out various path transitions such as flip, swap, or slide to find the most efficient path. We improve the original algorithm by replacing its randomly selected initial path with a *zigzag* path via K-means [37] and balanced K-means clustering [38]. For example, the network probing paths for five UAVs are initiated with K-means, as shown in Figure 4(b).

Once a UAV has its shortest grid point visit path after partitioning the probing area, it continues to visit each grid point from the path one by one. When the UAV visits a grid point, it broadcasts its presence towards the communicable

nodes for detecting the local networks. A node that receives the broadcast packet sends an ACK packet with its localized M-depth routing tree back to the UAV. The UAV calculates the PRR for the acknowledged nodes and records a set of nodes $\forall j$ with $\overline{PRR}_{G_i \to N_j}$ greater than or equal to θ_{LK} into its neighbor node list at grid point i denoted as G_i.

After all of the UAVs return to the starting point, which is the original source center, they share all the grid point-to-neighbor node lists and the collected localized M-depth child node list of each root node. Finally, all of the UAVs retain complete network connectivity by merging the partial local network topology.

The detailed network probing procedure by UAVs is described in Algorithm 2. The computation complexity of finding a grid point visiting path with mTSP-GA is $O(L \cdot S \cdot N^2)$, where L is the number of iterations, S is the population size at each generation, and N is the number of UAVs. This network probing procedure can be performed in a periodic or aperiodic manner to reflect the up-to-date network dynamics.

Algorithm 2 Network Probing

1: **Input:** *grid topology, # of UAVs: N, θ_{LK}*
2: **Output:** *grid point-to-neighbor node list, localized routing tree*

   // I. Find the grid point visiting path with *mTSP-GA*
3: [probing path for each UAV] = mTSP-GA(grid topology, N)

   // II. Network Detection
4: **for** UAV u = 1 to N **do**
5:    $networkInfo(u) = \emptyset$;
6:    **for** each grid point G_i in a given probing path **do**
7:       UAV u sends hello packets;
8:       **for** node j that has ever sent ACK back **do**
9:          **if** $\overline{PRR}_{G_i \to N_j} \geq \theta_{LK}$ **then**
10:             $networkInfo(u)$
             $= \{(G_i, N_j, localNet(MG_{N_j}))\} \cup networkInfo(u)$;
11:          **end if**
12:       **end for**
13:    **end for**
14: **end for**

15: **while** any of UAV still probing **do**
16:    Wait until all of UAVs finish and return back to the center;
17: **end while**
18: Merge all the collected local network information into $networkInfo$;

V. TIME-DEPENDENT AD-HOC ROUTING WITH UAVS

After the initial network setup and probing phase, UAVs become aware of a complete network snapshot that has been *stitched* with all the collected partial localized network topologies from the perspective of some representative ad-hoc nodes at certain grid points. This means that the UAVs can recognize all the possible delivery path candidates through a series of a certain grid point, its connectable *delegate* node, and the node's local routing tree that includes the target node.

We consider a scenario in which the UAVs need to eventually deliver all the different data to the corresponding ad-hoc nodes within the designated deadline. We assume that a certain data item that is collected and created only for a specific node has an effective time limit, e.g., some urgent evacuation information on a certain node location for the next

10 min. In the situation of data delivery toward numerous target nodes, the UAVs need to make intelligent path planning decisions for achieving reliable on-time delivery to as many destination nodes as possible.

As the data deadline range that we consider is not too tight but is comparable to the UAV traveling time, e.g., from tens of seconds to several minutes, the duplicate coverage over the grid points by multiple UAVs harms the on-time delivery performance for some parts of the target nodes and possibly the entire network. Moreover, a very large number of grid point visits by one UAV would lead to numerous out-of-time delivery outcomes. Therefore, the three elements of efficient path planning, fair (or collaborative) task division among UAVs, and some wise usage of locally connected networks along with the direct UAV delivery can contribute to maximizing the on-time data delivery performance. If a sufficient number of UAVs are used, we would rather rely more on the direct UAV delivery to reduce the network overhead for engaging local networks. If only some partial number of UAVs are available to serve the punctual delivery service, on the other hand, we want to push some responsibilities of the direct UAV delivery to the local ad-hoc routing.

We present a distributed UAV path planning algorithm for a UAV to determine the next grid point to visit on the basis of the data urgency and the local routing information collected at the grid points. With our path planning, the encountered UAVs within the radio range perform collaborative task division so that the remaining grid points to visit are distributed to them, and accordingly, each UAV ends up with an almost similar travel time. In our algorithm, a path planning and routing decision is not determined all at once, and rather at each time, a custodian UAV continues to make each moving decision online dynamically over time. This can help UAVs to compensate for some previous unexpected path visits (e.g., due to UAV control issues) in the middle of their mission.

Once a UAV visits a grid point to deliver data through its connectable delegate nodes, it compiles all the relevant data destined to all the target nodes deliverable via the delegate nodes into a packet, and broadcasts it toward the delegate nodes. To improve the broadcast packet delivery rate, our protocol allows the transmission of up to TX times until an acknowledgment is received. The minimal graph that a delegate node belongs to builds a dynamic depth routing tree from its local shortest-path tree on the basis of the remaining packet deadline for each target node. The root minimal graph and a series of intermediate minimal graphs on the tree determine whether to keep transferring data to the child minimal graphs or not, by checking the data urgency, data expiry, and duplicated delivery.

A. UAV PATH PLANNING WITH COLLABORATIVE TASK DIVISION

Our distributed path planning algorithm aims to minimize the traversal time of UAVs for packet delivery by visiting grid points and eventually maximize the on-time delivery service rate. At the same time, we exploit some available

local terrestrial ad-hoc networks at an increased routing cost to reduce the direct UAV data delivery time.

We consider three crucial parameters for designing an efficient path planning algorithm: 1) the travel time to a possible next grid point from the current one, 2) the number of target nodes serviceable through the delegate nodes at a grid point, and 3) the estimated data expiry status upon visiting a grid point. We extend an existing path planning algorithm called $Myopic$ [8], [39] by harnessing our local network reachability via a grid point.

We quantify the weighted sum measure considering the three aforementioned factors as *visit urgency* with respect to each future grid point to visit. In case a UAV visits grid point i later, we preprocess its measure beforehand. We list all the possible deliverable nodes set at grid point i on the basis of its directly communicable delegate nodes, denoted as $\eta_{G_i} = \{N_{n_1}, N_{n_2}, \ldots, N_{n_l}\}$, where N_{n_i} is a node whose data have not yet expired. Note that there exist certain nodes that are reachable via multiple grid points.

The visit urgency for all the remaining grid points to visit with respect to a grid point-to-node pair is calculated as follows:

$$w(G_i, N_j) = \alpha \cdot T_{N_j} + (1 - \alpha) \cdot d(G_{current}, G_i)/\bar{v}_{uav} \quad (1)$$

where $G_{current}$ is the grid point where the UAV is currently visiting, G_i is future grid point candidate i with $N_j \in \eta_{G_i}$, α is a tuning parameter depending on how much data urgency is stressed out compared to the UAV travel time, T_{N_j} is the estimated remaining data delivery time at the time of visiting grid point i, $d(G_a, G_b)$ is the physical distance between grid points a and b, and $\bar{v}_{uav}$ is the average UAV flying speed.

On the basis of the obtained visit urgency measure, the UAV selects k grid points with the k lowest values as the possible future grid points to visit. In case there are some candidates with the minimum value, we select the grid point closest to the current grid point. When these criteria do not rule out the ambiguities, we check the number of all the reachable nodes with a valid deadline at a grid point. Lastly, we use random selection and obtain k future grid point candidates.

To find the most efficient future trajectory from the set of k selected grid points, we try out all the possible $k!$ permutation paths and find the best trajectory according to the number of serviceable nodes normalized with the routing cost. This criterion implies the cost-effective number of serviceable nodes for the decision. If there still remain multiple candidates even after this criterion, we perform the following priorities of the estimated travel time over the trajectory, the closest first visiting grid point from the current one, and the number of serviceable nodes, which is defined above, of the first visiting grid point. If these priorities can still not rule out the ambiguities, we randomly select one out of the k trajectories.

Each UAV continues to determine the next grid point to visit according to the above procedure until there exists no remaining node to service. Upon encountering another UAV on the fly within the radio range, the UAVs exchange their
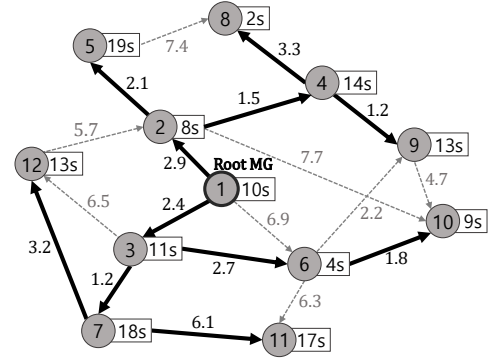
(a) M-depth coarsened graph to its shortest-path tree (with solid line)
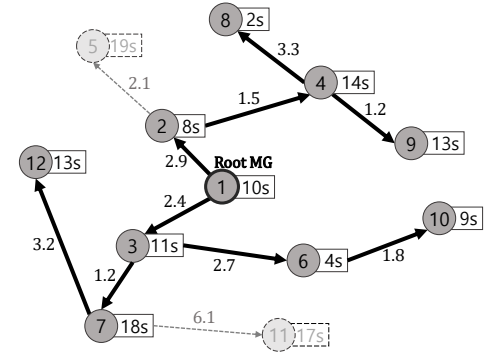
(b) Shortest-path tree to dynamic depth tree (with solid line)

FIGURE 3. Constructing a dynamic depth tree from the M-depth coarsened graph marked with (MG ID, remaining deadline) for each MG (where $\delta = 3$)

already-visited grid point list and update their remaining grid points to visit. To fairly distribute the grid points to visit according to the geographical distribution of the remaining grid points and quickly complete the entire data delivery mission, we use the well-known existing clustering techniques such as K-means clustering and balanced K-means clustering.

B. TIME-DEPENDENT ROUTING IN LOCAL AD-HOC NETWORKS

When a UAV visits a grid point, it informs the data deadline for the reachable target nodes of its connectable delegate nodes. Using this information, each root minimal graph to which a delegate node belongs builds a dynamic depth tree from its shortest-path tree by cutting the leaf nodes or branches with a relatively large remaining time or with data that have already expired.

Each minimal graph m calculates its representative remaining delivery time T_{MG_m} by taking the minimum value among all the remaining time values for its nodes. If the root minimal graph receives data from the UAV, it checks whether the remaining delivery time for any child minimal graphs is not too high or larger than a certain threshold, which is $T_{MG_{root}} + \delta$ (where $\delta = |median_{\forall MG_i \in localNet(MG_{root})}(T_{MG_i} - T_{MG_{root}})|$, as a minimal margin). For the child minimal graphs that have lesser remaining delivery time than the threshold, the root minimal graph distributes the data down to its child minimal graphs.

Algorithm 3 Time-Dependent Ad-Hoc Routing with UAVs

1: **while** there exist new data at data center **do**
2:     Retrieve target node and their deadline list $\{N_i, T_{N_i}\}$;
3:     $\Delta = 0$; // Reset relative time clock

4:     **while** there exist feasible nodes **do**
5:         // I. Path planning of UAV
6:         Select k candidate grid points according to criteria;
7:         Find the best trajectory out of $k!$ permutations of k grid points;
8:         Move to the next grid point of the trajectory;

9:         // II. Time-dependent ad-hoc routing
10:        **for** node i s.t. $\{G_{cur}, N_i\} \in networkInfo$ **do**
11:            $childNodeList$
             $= time\text{-}dependent\text{-}routing(MG_{N_i}, T_{MG_{N_i}} - \Delta)$;
12:            Delete all $N_i \in childNodeList$ from target node list;
13:        **end for**

14:        // III. Task division among multiple UAVs
15:        **if** multiple UAVs encountered **then**
16:            Exchange their own visited grid point list;
17:            Perform task division for the remaining grid points;
18:        **end if**
19:     **end while**
20:     Go back to the data center;
21: **end while**

22: **Function** $time\text{-}dependent\text{-}routing(MG_{root}, T_{MG_{root}})$
23:     Initialize dynamic depth tree $DDT(MG_{root}) = SPT(MG_{root})$;
24:     **for** $MG_m \in localNet(MG_{root})$ **do**
25:         Define $T_{MG_m} = min_{N_i \in \eta_{MG_m}} T_{N_i}$ as deadline of MG_m;
26:         **if** $T_{MG_m} > T_{MG_{root}} + \delta$ **then**
27:            Remove MG_m from $DDT(MG_{root})$;
28:         **end if**
29:         **if** any descendant of MG_m included in $DDT(MG_{root})$ **then**
30:            Insert MG_m and all upward parent MGs into $DDT(MG_{root})$;
31:         **end if**
32:     **end for**
33: **end Function**

Otherwise, the child minimal graphs that have a larger deadline than the threshold are discarded from the routing tree. The minimal graph cut from the routing tree still has a chance to receive its data later through the UAV's future grid point visit and one of its delegate node as long as the corresponding codes are still in the UAV's to-do-list. However, in case a minimal graph needs to be cut on the basis of this remaining time threshold, but its downward child minimal graph has satisfied the threshold condition, we do not cut the branch including the original minimal graph that violated the condition from the routing tree for on-time delivery to the child minimal graph in need.

For example, as illustrated in Figure 3(a), we obtain the shortest-path tree from the M-depth coarsened graph. Then, we extract the dynamic depth tree from the shortest-path tree by cutting the nodes (e.g., nodes 5 and 11), whose deadline is greater than $T_{MG_{root}} + \delta = 10\ s + 3\ s = 13\ s$, as shown in Figure 3(b). Although node 7 also fails to meet the condition, like nodes 5 and 11, we let node 7 remain in the dynamic depth tree because it includes one of its child minimal graphs, node 12, whose deadline is within the threshold.

Once the root minimal graph finalizes its dynamic depth tree by trimming the unnecessary minimal graphs and their branches, it reports the valid target node list that is supposed to be delivered in this round through the tree back to the UAV. The UAV erases the target nodes from its remaining target node list to deliver the data. On the basis of the remaining target node list to deliver the data, the UAV runs its path planning algorithm to determine which next grid point to move to and moves to this point.

The detailed routing steps are described in Algorithm 3. Regarding the computation complexity of the path planning, it takes $O(n \cdot log(n))$ to sort the n remaining target grid points, and select the top k candidate grid points. To find out the best trajectory based on the selected k candidate grid points, it takes $O(k! \cdot k)$, where $O(k!)$ is to find all possible permutation order among the k candidates, and $O(k)$ is to traverse over the k grid points.

VI. EVALUATION

We validated the proposed scheme in a simulated network of 300 randomly distributed nodes over the RoI of $300\ m \times 300\ m$ with a 16×16 virtual grid topology, as shown in Figure 4(a). This is the network size where a single UAV takes $441\ s$ for visiting all the grid points along one of the most efficient paths based on *mTSP-GA*. We simulated a damaged network where 83.4% of the src-to-dst pairs had no valid routes. We implemented a packet-level simulator in MATLAB for the algorithm-level validation. As a radio propagation model, we used a combined path-loss shadowing model with a path-loss exponent of 4, a reference loss of 46.68 dB, and an additive white Gaussian noise $N(0, 5^2)$ in decibels. In our experiments, we calculated the PRR by taking the average rate over past 50 transmissions. We used the PRR values of 0.5 and 0.95 for the valid link and minimal graph requirements. Next, we chose a suitable grid size smaller than the communication range. The parameters used in the simulation are listed in Table 2.

We randomly generated the uniformly distributed packet delivery deadline for each distinct node with the interval of $[120\ s, 180\ s]$ for an urgent scenario. The flying speed $\bar{v}_{uav}$ of the UAVs was set to $12\ m/s$. The packet transmission time was negligible as compared to the UAV traveling time. The UAVs initiated and ended their network probing and data delivery services at a data center, currently located at the center of the RoI.

We evaluated our data delivery performance in terms of the on-time serviced node percentage as the routing reliability, transmission cost as the routing efficiency, and the travel time of the UAVs as the path planning efficiency. We measured the on-time serviced node percentage as the target node percentage of the successfully received corresponding data from the center node. We quantified the transmission cost as the sum cost of the expected number of transmissions for each link with PRR for all the broadcast and ACK packets. Our time-dependent routing used up to five transmissions. We showed the travel time of the UAVs by taking the average values among all the used UAVs for a complete traversal over the grid points. We run 10 different experiments with randomly distributed time deadlines destined to all the nodes
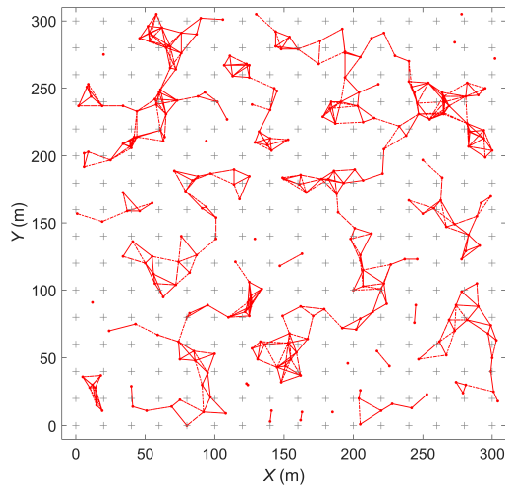
(a) Network topology
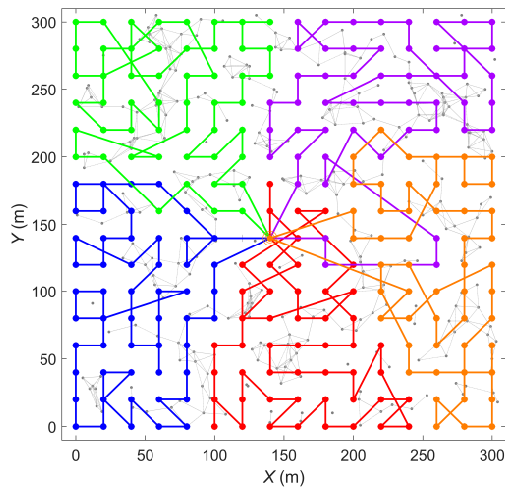
(b) Network probing paths
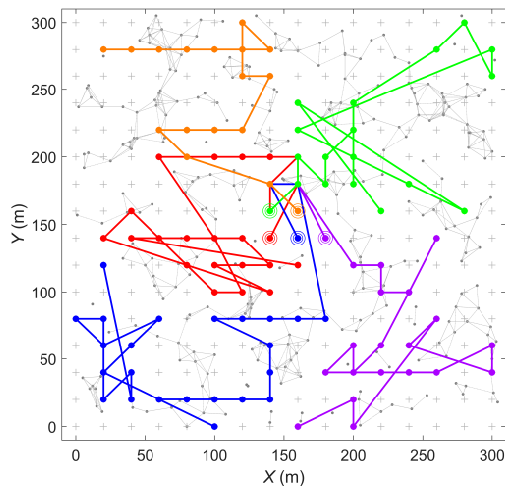
(c) Data delivery paths

FIGURE 4. Simulated network topology with 83.4% of the source-to-destination pairs having no valid route (a solid line denotes a strongly connected link, and a dotted line represents a neighbor link) and 16×16 grid points (with a gray "+" marker) and an example of the network probing paths using balanced K-means clustering and the data delivery paths with five UAVs over the network area

Simulation Environment	
Territory area	$300\ m \times 300\ m$
# of nodes	300
Grid size	$20\ m$
Packet delivery deadline	$0 \sim 530\ s$
UAV speed	$12\ m/s$
UAV flying height	$7\ m$
Radio Propagation Model	
Path-loss exponent	4
Reference loss	46.68 dB
Additive white Gaussian noise	$N(0, 5^2)$ in dB
Simulation Parameter	
# of UAV	$1 \sim 6$
Weight parameter α	0.2
Permutation parameter k	2
Valid link PRR θ_{LK}	0.5
Strongly connected link PRR θ_{MG}	0.95

TABLE 2. Simulation environment and parameters

and reported the average values with the standard deviation. Initially, one UAV was used to purely validate the routing performance, while avoiding the task division effect on the path planning. The weight parameter $\alpha = 0.2$ and the permutation parameter $k = 2$ for the UAV path planning were tuned during the experiments, as in [8]. The maximum depth $M = 3$ for a dynamic depth tree and the deadline range of $[120\ s, 180\ s]$ for on-time delivery to the target nodes were used, unless otherwise noted.

In our proposed system, during the network probing phase, the probing paths for N UAVs are generated at the source center, and each UAV travels along the assigned path independently, as shown in Figure 4(b) for an experiment with five UAVs. We quantified the initial probing efficiency in terms of the travel time of each UAV in Section VI-A. Once the initial setup is completed, during the actual data routing phase, all UAVs attempt to load data from the center and start their data delivery tasks. Considering data urgency and travel cost, each UAV consecutively runs its own path planning to determine its next visiting grid point in a distributed online manner, without any knowledge of other UAVs, as shown in Figure 4(c) for an experiment with five UAVs. We investigated the data delivery performance in Section VI-B.

A. INITIAL SETUP
During the initial setup, our scheme converted an original network graph into its coarsened network graph (according to Section IV-A). To check the coarsening outcome in this simulation setup, we measured the number of vertices, edges, and logical hops among all the connected vertex pairs for each network graph. A vertex indicated an original node in the original network graph, while indicating a minimal graph in the coarsened one. The number of logical hops among the nodes within the same minimal graph was counted as 0. After the coarsening procedure, the number of vertices decreased from 300 to 140 with a 53.3% reduction, the number of edges decreased from 1,125 to 265 with a 76.4% reduction, and the number of logical hops decreased from 141,684 to 86,466 with a 61.03% reduction. This implied
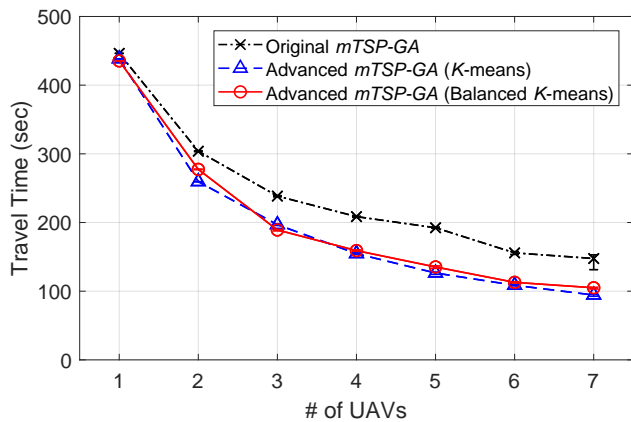
FIGURE 5. Travel time of UAVs according to our network probing path generation with respect to the number of UAVs

that the network coarsening helped to reduce the unnecessary computation and communication overhead on both the UAV and the node tiers.

We investigated the initial network probing efficiency in terms of the travel time of the UAVs, as shown in Figure 5. We compared three cases that we used as the initial values for the iterations: 1) the original random values, 2) K-means clustering, and 3) balanced K-means clustering. We measured the minimum, the average, and the maximum travel time among the UAVs. As the number of UAVs increased, the travel time for all the cases decreased as expected. The usage of the initial values obtained after both K-means and balanced K-means algorithms generated the most efficient paths with up to 36% reduced travel time. Further, regarding the time gap between the first completed UAV and the last completed UAV, the original case had a time gap of up to $22\ s$, whereas K-means and balanced K-means clustering resulted in a time gap of at most $3\ s$ and $4\ s$, respectively. This result implied that the use of efficient initial paths on the basis of the clustering techniques led to a relatively large number of global optimal points for generating more equally distributed paths of multiple UAVs in the initial network setup.

B. TIME-DEPENDENT ROUTING STRUCTURE

We examined our time-dependent routing performance in the routing execution phase. To understand how the parameter selection of the maximum depth M in our M-depth local routing tree and the delay margin threshold δ for the dynamic depth tree affected the routing efficiency as well as the routing overhead, we varied M and the delay threshold, as shown in Figure 6.

We first investigated how the maximum depth M affected the performance by using a single UAV, as shown in Figure 6(a). In a tight deadline case of $[120\ s, 180\ s]$, the on-time data delivery rate improved with an increase in M, while increasing the network overhead as a tradeoff. This implied that under very tight time constraints, the proposed algorithm found a cost-effective way (e.g., using the M depth of 3) for distributing time-sensitive data by utilizing some parts of
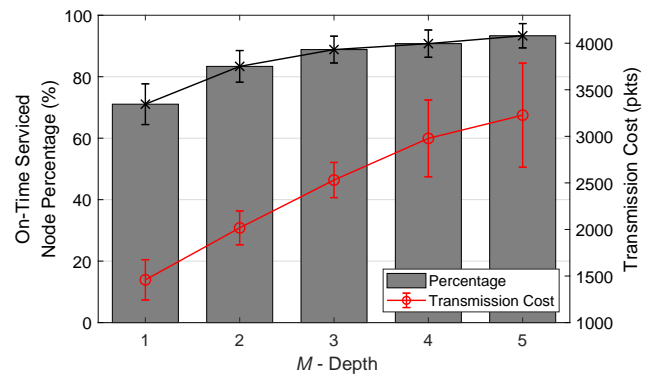
(a) Maximum depth M
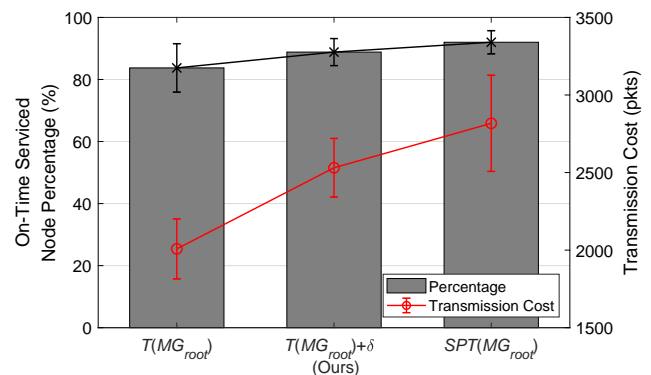

(b) Deadline threshold

FIGURE 6. On-time serviced node percentage and transmission cost with respect to the maximum depth M and the deadline threshold of the dynamic depth tree with the deadline of $[120\ s, 180\ s]$ using one UAV

a local tree structure, thereby reducing the number of grid points for a UAV to visit.

We validated our dynamic depth tree construction by varying our internal delay margin threshold, as shown in Figure 6(b). The use of a certain threshold margin δ upon the formation of the final dynamic depth tree provided a great balance point that had a high on-time serviced node percentage of up to 88.8% with a relatively medium network overhead. Our time-dependent routing scheme dynamically changed a routing tree in accordance with the urgency of packet delivery with high adaptability toward dynamic networks.

As the proposed approach co-optimized both the UAV path planning and the local network resources, we compared each component with other counterpart algorithms. To evaluate our routing-based path planning called *RT-Path*, we compared it against a method proposed in a previous study [8], called *Myopic*, where UAVs collaboratively traverse grid points and perform direct data delivery just to a 1-hop neighbor node from a visiting grid point without using any locally connected network.

To validate our time-dependent routing structure called *TD Routing*, we compared it with a simple *flooding* mechanism that broadcasts the information twice (resulting in a steady-state delivery performance) with a duplicate check until all of the connectable nodes are reached. We also compared it with
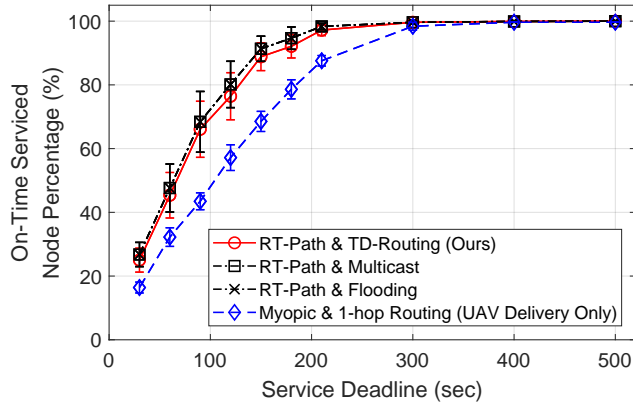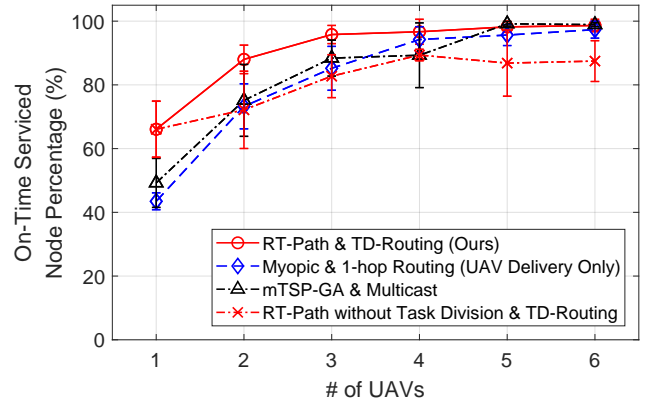
(a) On-time serviced node percentage
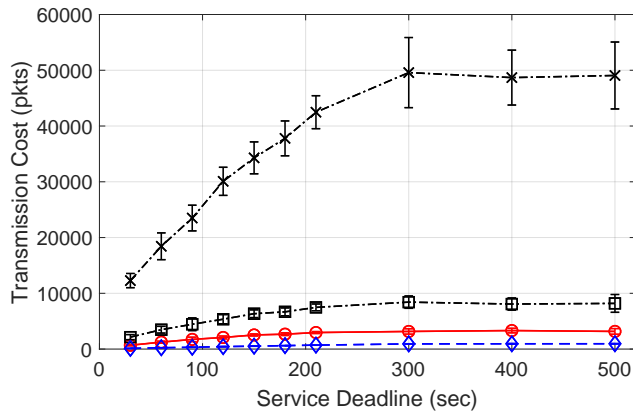
(b) Transmission cost

FIGURE 7. On-time serviced node percentage and transmission cost with respect to service deadline using one UAV with $M = 3$
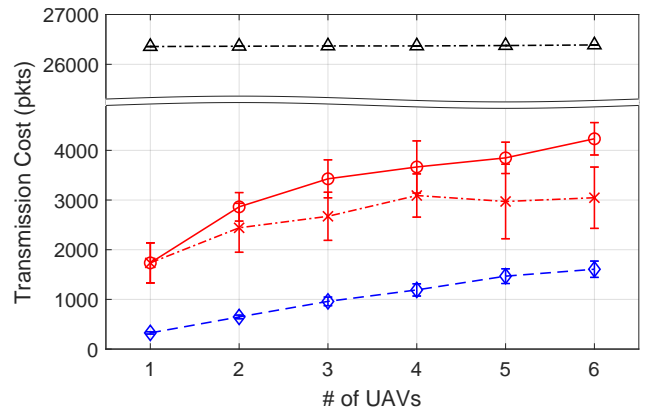

(a) On-time serviced node percentage

(b) Transmission cost

FIGURE 8. On-time serviced node percentage and transmission cost with respect to the number of UAVs with the deadline of $[60\ s, 120\ s]$ with $M = 3$

Multicast along the shortest paths to multiple target nodes, allowing the transmission of only up to five times until it received *ACK* (which was optimized to reach the similar data delivery performance to that of *flooding*).

We explicitly quantified the number of grid points that a UAV visited according to our *RT-Path* and *Myopic*. Our algorithm visited 63.0 grid points on average, while *Myopic* visited 84.1 grid points. This implied that our algorithm significantly reduced the number of visits by 25.1% with an efficient exploitation of the local networks, contributing to higher on-time delivery performance. We also measured the travel distance between two consecutive grid points. A UAV with our algorithm flew further (35.46 *m*) than that with *Myopic* (26.54 *m*), on average. This showed that our algorithm let the nearby local network involved with routing, and made a UAV move away to a distant territory.

We varied the service deadline from 30 *s* to 500 *s* with the uniformly random range of ±30 *s*, as shown in Figure 7. We combined our routing-dependent path planning algorithm *RT-Path* with our own routing protocol *TD-Routing* and other counterpart routing algorithms of *Multicast* and *Flooding*, while also comparing it against *Myopic*. In the range of service deadlines from 30 *s* to 210 *s*, which were relatively tight, our path planning algorithm with all the routing algo-

rithms started to achieve very high on-time serviced node percentage as the service deadline was relaxed, while outperforming *Myopic* and its one-hop routing algorithm. From the perspective of the routing overhead, only the *TD-Routing* algorithm among all the routing algorithms with *RT-Path* maintained a very low transmission cost comparable to that of only the 1-hop data sharing with *Myopic*. The proposed path planning and its coupled routing algorithm provided a very cost-effective performance, whereas *Myopic* & 1-hop delivery scheme performed poorly, particularly especially with some tight service deadline ranges. *Myopic* often misses the service deadline for visiting more grid points because of using only 1-hop transmission at a grid point without using any local networks. Moreover, note that although *Multicast* and *Flooding* exhibit high routing performance, their transmission cost is significantly higher than that of the proposed algorithm with a factor ranging from 2.7 (for *Multicast*) to 15.7 (for *Flooding*). This implied that a local routing strategy coupled with its direct load-and-carry delivery strategy under time constraints should be based not on a static routing structure but on a dynamic one considering the up-to-date packet urgency.

We examined how the routing accuracy and overhead performance were affected by the number of UAVs in a tight
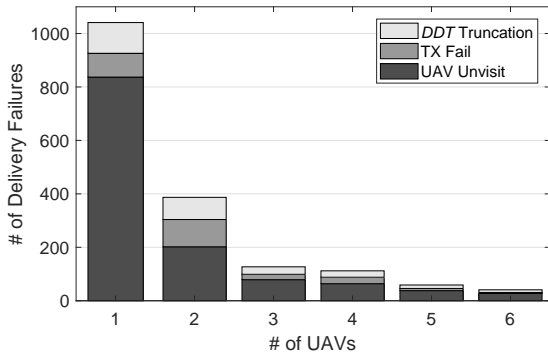
FIGURE 9. Delivery failure performance according to UAV resource with the deadline of $[60\ s, 120\ s]$ and $M = 3$.
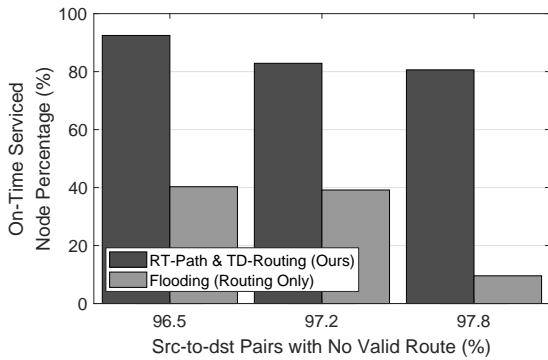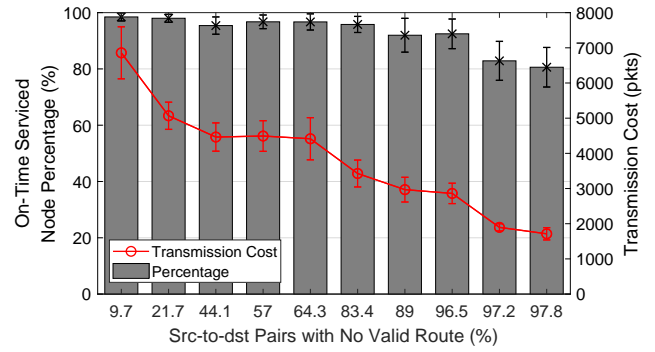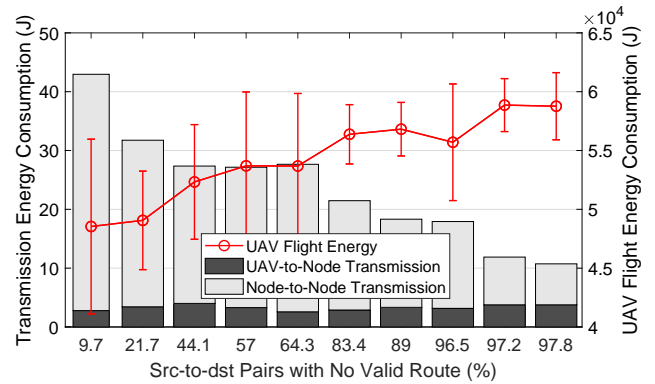

FIGURE 10. On-time serviced node percentage w/ and w/o using UAVs under severely disrupted networks with the deadline of $[60\ s, 120s]$ (3 UAVs and $M = 3$ used in our algorithm)

(a) On-time serviced node percentage and transmission cost

(b) Energy consumption

FIGURE 11. On-time serviced node percentage, transmission cost, and energy consumption in various networks with the deadline of $[60\ s, 120\ s]$, $M = 3$, and 3 UAVs

deadline case of $[60\ s, 120\ s]$, as shown in Figure 8. Our path planning and routing algorithm, i.e., *RT-Path* and *TD-Routing*, was compared against *Myopic* and 1-hop routing [8] and *mTSP-GA* and *Multicast*. Our algorithm achieved the highest on-time data delivery, while maintaining a lower routing overhead than *Multicast* with *mTSP-GA*. Even under scarce UAV resource with very few UAVs, our *RT-Path* and *TD-Routing* kept relatively stable timely delivery performance without much decline. It is due to the fact that UAVs hover the RoI more extensively by effectively arranging the given data delivery tasks to the local network routing rather than missing the packet deadline. Upon comparing our algorithm with the one without task division, we found that collaborative task division among UAVs is a necessary step to pre-plan their future visiting paths that can arrange more urgent packet delivery with a higher priority, achieving up to a 13% higher delivery rate. As the use of more UAV resources means that each UAV has a relatively large deadline through task division and the on-time data delivery performance improves. Note that the optimal number of UAVs should be selected considering both the network impairedness and the delivery urgency.

We analyzed where the on-time data delivery failure came from: 1) unvisited cell by UAV, 2) transmission failure, and 3) dynamic depth tree truncation. We counted the number of failures belonging to each failure case. As more UAV resources can be applied, the total number of delivery failures

significantly drops, as shown in Figure 9. It is interesting to see that the dominant delivery failures come from UAV's unvisit to cells. This makes sense with our original hypothesis that the use of mobile agents for solving the problem of on-time data delivery can secure valid routing paths even under critically damaged network environments. This implies that as more UAVs divide cells to visit in a collaborative manner, they are capable of using the saved time for on-time data delivery to nodes with a higher priority.

We investigated how our algorithm based on a balanced mixture of direct UAV data delivery and static ad-hoc routing advanced the pure static ad-hoc routing without any UAVs in severely disconnected networks. As shown in Figure 10, we compared against *Flooding* considered the upper bound static node-based routing scheme in terms of data delivery reliability. As the network becomes very severely damaged, the on-time data delivery performance of our algorithm is well-sustained, whereas the routing-only scheme degrades severely to a delivery ratio of only 9.6%. This means that in a severely disrupted network situation, the use of controlled mobile agents can considerably recover the node reachability.

We validated the resilience of our work under different network environments with different network outage degrees, as shown in Figure 11. For a different network environment, we quantified the percentage of src-to-dst pairs with no valid route. As shown in Figure 11(a), our routing protocol

provides very stable routing reliability with at least 80.6% (even under the worst network). One interesting observation is that as the network gets more severely broken, the routing overhead is rather significantly reduced. This is because the depth of the localized network itself becomes lower because of the more partitioned network status, and thus, the resulting transmission cost over the static network becomes lower.

Lastly, we examined how much energy was consumed by the three different operations: 1) UAV flight movement, 2) UAV-to-node transmission, and 3) node-to-node transmission, as shown in Figure 11(b). We applied a real-world drone energy consumption unit, which was $178.4\ W$ as per the DJI Phantom 4 Pro V2.0 specification for UAV flight energy consumption [40]. The energy consumption by wireless transmission was calculated on the basis of the TelosB CC2420 specification with a maximum of $100\text{-}ms$ transmission time per packet [41]. As the network got more disconnected, UAVs played a more dominant role in door-to-door data delivery and consumed more energy for UAV flights with three orders of magnitude as compared to the transmission energy consumption. With respect to the energy consumption by wireless transmission, the energy consumption by the node-to-node transmission decreased because of the lower utilization of static nodes for routing, while the UAV-to-node transmission consumed slightly more energy for more active UAV utilization for routing. Thus, there was an interesting tradeoff between energy consumption and on-time data delivery performance.

VII. CONCLUSION

We have presented a time-dependent ad-hoc routing structure for delivering delay-sensitive data using UAVs. The proposed algorithm exploits an optimal mixture of direct UAV data delivery to certain spots and a localized cost-effective data sharing structure from there to the local nodes. Our work uses a tradeoff relationship between routing reliability and efficiency in the problem of data delivery using UAVs.

We have incorporated the *minimal graph* structure for extracting the strongly connected nodes via a *graph coarsening* procedure. By preparing a localized cost-effective routing structure at the terrestrial ad-hoc nodes, UAVs performed their own distributed optimal path planning to determine the grid points to visit on the basis of the data urgency and the local network status. We have demonstrated that our algorithm considerably improved the on-time data delivery performance, while reducing the routing cost as compared to some counterpart algorithms.

In the future, we may consider UAV data delivery scenarios from multiple data center locations to generalize path planning and its dependent routing architecture. Moreover, as the battery outage for traveling over a selected trajectory may affect its original movement, it would be interesting to optimize the UAV's navigation path for visiting charging stations. In case of having each different data value, its corresponding priority should also be considered to select next visiting points in the path planning phase.

REFERENCES

[1] A. Townsend and M. Moss, Tellecommunications Infrastructure in Disasters: Preparing Cities for Crisis Communication. New York University, 2005.

[2] P. Li, T. Miyazaki, K. Wang, S. Guo, and W. Zhuang, "Vehicle-assist resilient information and network system for disaster management," IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 3, pp. 438–448, 2017.

[3] N. Zhao, W. Lu, M. Sheng, Y. Chen, J. Tang, F. R. Yu, and K.-K. Wong, "Uav-assisted emergency networks in disasters," IEEE Wireless Communications, vol. 26, no. 1, pp. 45–51, 2019.

[4] O. Sami Oubbati, N. Chaib, A. Lakas, S. Bitam, and P. Lorenz, "U2rv: Uav-assisted reactive routing protocol for vanets," International Journal of Communication Systems, p. e4104, 2019.

[5] O. S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, and A. Rachedi, "Uav-assisted supporting services connectivity in urban vanets," IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3944–3951, 2019.

[6] T. He, K. Lee, and A. Swami, "Flying in the dark: Controlling autonomous data ferries with partial observations," in Proceedings of ACM Mobi-Hoc'10, 2010.

[7] M. Harounabadi and A. Mitschele-Thiel, "Applying message forwarding and replication to multi-uav message ferry networks," Mobile Networks and Applications, vol. 23, no. 5, pp. 1337–1346, 2018.

[8] J. Yoon, Y. Jin, N. Batsoyol, and H. Lee, "Adaptive path planning of uavs for delivering delay-sensitive information to ad-hoc nodes," in IEEE WCNC'17, March 2017.

[9] C. Cheng, P. Hsiao, H. T. Kung, and D. Vlah, "Maximizing throughput of uav-relaying networks with the load-carry-and-deliver paradigm," in 2007 IEEE Wireless Communications and Networking Conference, March 2007, pp. 4417–4424.

[10] O. S. Oubbati, A. Lakas, N. Lagraa, and M. B. Yagoubi, "Uvar: An intersection uav-assisted vanet routing protocol," in 2016 IEEE Wireless Communications and Networking Conference. IEEE, 2016, pp. 1–6.

[11] J. Kirk, "Multiple traveling salesmen problem - genetic algorithm," 2014. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/19049-multiple-traveling-salesmen-problem-genetic-algorithm?

[12] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in Proceedings of the 7th ACM conference on embedded networked sensor systems. ACM, 2009, pp. 1–14.

[13] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in Proceedings of the 7th annual international conference on Mobile computing and networking. ACM, 2001, pp. 70–84.

[14] A. Taha, R. Alsaqour, M. Uddin, M. Abdelhaq, and T. Saba, "Energy efficient multipath routing protocol for mobile ad-hoc network using the fitness function," IEEE access, vol. 5, pp. 10 369–10 381, 2017.

[15] O. S. Oubbati, A. Lakas, P. Lorenz, M. Atiquzzaman, and A. Jamalipour, "Leveraging communicating uavs for emergency vehicle guidance in urban areas," IEEE Transactions on Emerging Topics in Computing, 2019.

[16] O. S. Oubbati, M. Mozaffari, N. Chaib, P. Lorenz, M. Atiquzzaman, and A. Jamalipour, "Ecad: Energy-efficient routing in flying ad hoc networks," International Journal of Communication Systems, 2019.

[17] M. Harounabadi, M. Bocksberger, and A. Mitschele-Thiel, "Evolutionary path planning for multiple uavs in message ferry networks applying genetic algorithm," in 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). IEEE, 2018, pp. 1–7.

[18] S.-Y. Park, C. S. Shin, D. Jeong, and H. Lee, "Dronenetx: Network reconstruction through connectivity probing and relay deployment by multiple uavs in ad hoc networks," IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 11 192–11 207, 2018.

[19] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in Proceedings of ACM MobiHoc'04, 2004.

[20] M. Bin Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in Proceedings of ACM MobiHoc'06, 2006.

[21] F. Khelifi, A. Bradai, K. Singh, and M. Atri, "Localization and energy-efficient data routing for unmanned aerial vehicles: Fuzzy-logic-based approach," IEEE Communications Magazine, vol. 56, no. 4, pp. 129–133, 2018.

[22] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," Robotics and Autonomous systems, vol. 61, no. 12, pp. 1258–1276, 2013.

[23] G. Reinelt, "Tsplib - traveling salesman problem library," ORSA Journal on Computing, vol. 3, no. 4, pp. 376–384, 1991.

[24] K. Hoffman, M. Padberg, and G. Rinaldi, "Traveling salesman problem," in Encyclopedia of Operations Research and Management Science. Springer, 2013.

[25] B. Golden, S. Raghavan, and E. Wasil, The vehicle routing problem: latest advances and new challenges. Springer Science & Business Media, 2008, vol. 43.

[26] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," European Journal of Operational Research, vol. 59, no. 3, 1992.

[27] D. Taş, M. Gendreau, O. Jabali, and G. Laporte, "The traveling salesman problem with time-dependent service times," European Journal of Operational Research, vol. 248, no. 2, pp. 372 – 383, 2016.

[28] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff, "Approximation algorithms for orienteering and discounted-reward tsp," SIAM Journal on Computing, vol. 37, no. 2, pp. 653–670, 2007.

[29] N. Bansal, A. Blum, S. Chawla, and A. Meyerson, "Approximation algorithms for deadline-tsp and vehicle routing with time-windows," in Proceedings of the Thirty-sixth Annual ACM STOC'04, 2004, pp. 166–174.

[30] O. Polat, C. Kalayci, O. Kulak, and H.-O. Günther, "A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit," European Journal of Operational Research, vol. 242, no. 2, pp. 369 – 382, 2015.

[31] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning," IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 132–141, Feb 2013.

[32] M. Arantes, J. Arantes, C. Toledo, and B. Williams, "A hybrid multi-population genetic algorithm for uav path planning," in Proceedings of the Genetic and Evolutionary Computation Conference, ser. GECCO'16, 2016.

[33] A. Budiyanto, A. Cahyadi, T. B. Adji, and O. Wahyunggoro, "Uav obstacle avoidance using potential field under dynamic environment," in 2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC). IEEE, 2015, pp. 187–192.

[34] Y. Lin and S. Saripalli, "Sampling-based path planning for uav collision avoidance," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 11, pp. 3179–3192, 2017.

[35] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1123–1152, 2015.

[36] B. Hendrickson and R. Leland, "A multi-level algorithm for partitioning graphs," in SC Conference(SC), vol. 00, 12 1995, p. 28. [Online]. Available: doi.ieeecomputersociety.org/10.1109/SUPERC.1995.3

[37] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881–892, 2002.

[38] R. He, W. Xu, J. Sun, and B. Zu, "Balanced k-means algorithm for partitioning areas in large-scale vehicle routing problem," in International Symposium on Intelligent Information Technology Application, vol. 3, 2009, pp. 87–90.

[39] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in 25th IEEE International Real-Time Systems Symposium, Dec 2004, pp. 296–305.

[40] D. Phantom, "pro/pro+ user manual. 2018," 4.

[41] A. Prayati, C. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos, "A modeling approach on the telosb wsn platform power consumption," Journal of Systems and Software, vol. 83, no. 8, pp. 1355–1363, 2010.

JINYI YOON received the B.S. and M.S degrees in Computer Science and Engineering from Ewha Womans University, Seoul, South Korea, in 2017 and 2019, respectively. She is currently working toward a Ph.D. degree at Ewha Womans University. Her current research interests include wireless ad-hoc networks, vehicular networks, IoT security system, and deep learning.

SUNGJU DOH received the B.S. degree in Computer Science and Engineering from Ewha Womans University, Seoul, South Korea in 2020. She has worked on wireless ad-hoc networking, as an undergraduate researcher at Intelligent Networked Systems Laboratory, Ewha Womans University. She will be joining Google in Munich, Germany, in June 2020, as a software engineer.

OMPRAKASH GNAWALI is an Associate Professor at the Computer Science Department of the University of Houston. He does research on IoT, with a focus on wireless and sensing technologies. Other areas of interest include cybersecurity, data analytics, mobile systems, and technology for the emerging markets. He received his SB and MEng from MIT, PhD from USC, and was a postdoc at Stanford.

HYUNGJUNE LEE is an Associate Professor in the Department of Computer Science and Engineering at Ewha Womans University, Seoul, Republic of Korea. He received his B.S. degree in Electrical Engineering from Seoul National University, Republic of Korea in 2001. He obtained his M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 2006 and 2010, respectively. He joined Broadcom as Sr. Staff Scientist for working on research and development of 60GHz 802.11ad SoC MAC. Also, he worked for AT&T Labs as Principal Member of Technical Staff with the involvement of LTE overload estimation and LTE-WiFi interworking. His current research interests include future wireless networks on IoT, fog computing, VANET, and machine learning-driven network system design.