Rendezvous: Opportunistic Data Delivery to Mobile Users by UAVs Through Target Trajectory Prediction

JinYi Yoon, A-Hyun Lee, and HyungJune Lee, *Member, IEEE*

***Abstract*—Reliable and timely delivery of data to mobile targets is a challenging problem in mobile ad-hoc networks, because of the opportunistic and unpredictable nature of the problem. Finding a complete series of mobile-to-mobile contacts and forwarding data toward targets within a designated deadline is even more difficult. We leverage unmanned aerial vehicles (UAVs) as message ferries to travel over highly probable rendezvous points for over-the-air delivery to target users. We propose *Rendezvous*, an opportunistic yet disciplined data delivery scheme based on trajectory prediction of users and UAV path planning. During the offline learning phase, we identify both temporal and spatial regularities of mobile users from real-world trajectories using sequence-wise clustering, and construct a compact yet well-summarized cluster signature that enables efficient search. In the data delivery phase, given recent movement history of users, UAVs perform their own distributed path planning collaboratively over time. Each UAV finds a chronological sequence of future visiting points where it can make consecutive data delivery efforts to users to overcome the uncertainty in mobility and ensure timely delivery. Real-world trace-driven simulation experiments demonstrate that *Rendezvous* achieves reliable and punctual data delivery using only few UAVs compared with existing algorithms.**

***Index Terms*—Opportunistic data delivery, trajectory prediction, path planning, mobile Ad-Hoc Network (MANET), unmanned aerial vehicles (UAVs).**

I. Introduction

In an era where mobile devices are ubiquitous, the construction of reliable yet efficient mobile ad-hoc networks (MANETs) is an important aspect of wireless networking. In MANETs, data are delivered from a source to a destination through a series of opportunistic mobile-to-mobile user wireless contacts based on human mobility in infrastructure-less environments. Although these kinds of networks are very flexible for use in ad-hoc situations, they have a high risk of failure of data delivery. Due to their inherent dependency on intermittent mobile contacts, and the existence of a dynamically changing node topology, data delivery tends to fail or suffer from latencies, which may be up to a few days.

In most practical scenarios, data need to be delivered to a user within a designated deadline [1]. If delivery takes anything from a few hours to several days from the time the data are requested, the data are likely to be already obsolete by the time of delivery. Aggravating this situation, MANETs can be partitioned to a greater of lesser degree, producing situations in which source-to-destination routing paths may be rare [2]. Under these circumstances, a certain group of users may always be isolated from the community unless some occasional intended door-to-door visits are scheduled for them. To tackle the innate data delivery issues incurred by network isolation and latency, a message ferry with controlled movement has been introduced, to deliver data from a base station or an administrative center to a local user or user group. In this context, harnessing unmanned aerial vehicles (UAVs) as message ferries is a very promising approach, since the UAVs can control themselves, and are relatively free from environmental constraints [3], [4].

With the recent significant growth in the number of mobile devices in use in the general population, node mobility has been recognized as one of the key factors in opportunistic data delivery. Due to the dynamic movement of users, however, it is still very challenging to identify patterns of movement from recorded trajectories and embed them into the design of practical systems for data delivery. Previous work has produced feasible approaches to dealing with uncertainty in user movement, by capturing a high degree of temporal and spatial regularity [5], [6]. Some predictive knowledge of users' future movements can be utilized by UAVs to determine their navigation paths. It may be possible to predict points at which specific users may pass, and therefore data packets can be delivered to them from the air via UAVs.

In this paper, a traffic scenario is considered in which some intermittent data deliveries are necessary for on-site mobile users. Examples could include the distribution of advertisement information with expiration dates in an infrastructure-less environment. In this situation, the coordination of mobile agents such as UAVs is key to the delivery of consecutive data to multiple users, under time constraints.

We propose *Rendezvous*, an opportunistic yet disciplined data delivery scheme incorporating multiple UAVs to deliver data to mobile users within designated deadlines. We leverage mobility pattern modeling and prediction techniques to infer highly probable future trajectories of target users. We capture a few highly possible trajectory patterns based on a cell-based grid topology

Manuscript received April 5, 2019; revised October 11, 2019; accepted December 10, 2019. Date of publication December 27, 2019; date of current version February 12, 2020. This work was supported by Samsung Research Funding Center of Samsung Electronics under Project SRFC-IT1803-00. The review of this article was coordinated by Dr. F. Tang. *(Corresponding author: HyungJune Lee.)*

J. Yoon and H. Lee are with the Department of Computer Science and Engineering, Ewha Womans University, Seoul, Republic of Korea (e-mail: yjin3012@gmail.com; hyungjune.lee@ewha.ac.kr).

A-H. Lee is with the School of Computer Science and Engineering, Seoul National University, Seoul, Republic of Korea (e-mail: ahyun.lee004@gmail.com).

Digital Object Identifier 10.1109/TVT.2019.2962391

in which real-world movement traces are classified with some degree of regularity. With the help of the mobility trajectory model constructed from these data, we derive a consecutive visiting sequence of each UAV during the actual data delivery phase, at positions which are expected to be highly probable rendezvous points for the UAV to reach out to a target user for data delivery.

The main challenges in this problem are two-fold: 1) predicting both a sequence of locations, and the expected passing time of mobile users; and 2) finding the optimal navigation paths which UAVs can use to deliver data within time constraints. To serve multiple users given limited UAV resources, the problem becomes even more complicated for attaining a chronological sequence of visiting points; it is essential to determine *where, when, and how long* each UAV should visit and stay over time.

To the best of our knowledge, this paper is the first to describe an over-the-air data delivery system using mobile agents to deliver data to mobile users under time constraints, by combining trajectory prediction and path planning. Existing techniques can be unreliable, since they rely on accidental encounters, potentially incurring very high latency and thus making them impractical. To remedy this data delivery problem with open delivery time, we apply machine learning-based approaches for the otherwise nearly impossible timely data delivery in MANET. Our main contributions can be summarized as follows:

- We extract both temporal and spatial regularity data regarding mobile users from real-world trajectories, using sequence-wise clustering to predict a series of probability-based preceding visiting positions.
- We design a compact cluster signature that summarizes essential features inferred from training trajectories belonging to specific representative trajectory clusters by filtering out irregular noisy movement traces.
- We propose a probabilistic path planning scheme in which multiple UAVs identify a chronological sequence of future visiting points at which they can make consecutive data delivery efforts to target users, to overcome the uncertainty introduced by user mobility and ensure timely delivery.

II. Related Work

This work is related to both opportunistic data delivery to mobile nodes, and path planning of a mobile agent in order to effectively cover a designated area and exploit it as a message ferry, based on the load-carry-and-delivery paradigm.

The problem of data delivery to mobile users has been explored under the name of DTN using mobile ad-hoc nodes as described in [7]–[9]. One of the main challenges in this area is to find a valid forwarding path for ensuring data delivery from one user to another based on replication [10]–[12], contact history [13]–[16], predefined interest group [17], [18], or social relationships among users [13], [19], [20]. However, most of the current approaches suffer from high latency due to their heavy reliance on the inherent intermittent connection between mobile users.

To tackle this problem, recent work has attempted to apply predictions about mobile users to data delivery. It has been shown that the dynamics of mobile users can be learned using mobility pattern modeling and can therefore be predicted with some level of certainty [6], [21]. Existing work has used the observed history traces to predict probable future trajectories, and estimates the probability of future motions by modeling their probability distribution [22], [23]. Also, some approaches have considered not only the underlying regularity, but also the social-based knowledge, such as social conformity [24] or living habits [25]. Recent work has applied a deep learning approach of Generative Adversarial Network (GAN) to predict the future motions in sequence [26].

More closely related to our work on the data delivery problem, prior research has addressed the issue of the prediction of the future movements of mobile users, and have used these predictions to store data packets in advance at nodes which are likely to be visited [5]. Although these approaches require pre-installed static nodes to capture and learn mobility patterns of users and to be used as infrastructure-based ad-hoc networks, this research has provided concrete instances of the way in which predictive knowledge about user activity can be used to produce timely data delivery. However, there remains a key question: how can timely delivery of data to mobile users be achieved even in *infrastructure-less* environments?

Mobile agents can be introduced for data delivery in an infrastructure-less network environment. To alleviate the intermittent connectivity problem, mobile agents can load data from a mobile user or an administrative center, carry the data, and provide them to another user or group of users [27], [28]. Considerable work based on this *message ferry* (MF) scheme has been carried out [29]–[31]. This work has focused on designing efficient navigation paths for mobile agents distributing or collecting data in MANETs or VANETs [32]–[35].

The problem of finding optimal traveling paths has been studied under the name of *path planning* in the robotics and operations research communities. The traditional *path planning* problems include the traveling salesman problem (TSP) [36], [37] and vehicle routing problem (VRP) [38]–[40]. Solutions to these problems usually suffer from high computational complexity, since they involve linear programming [41]. Taking into account time constraints, deadline-TSP [42], [43] and VRP with time windows [44]–[46] have been proposed.

The use of UAVs as message ferries, based on the *load-carry-and-delivery* (LCAD) paradigm has been proposed [47]. The LCAD paradigm divides the data delivery problem to three tasks of *loading* from a source node, *carrying* from source to destination, and *delivering* to the destination node by utilizing multiple UAVs. Some researchers have exploited UAVs as proactive relays, where UAVs are dedicated nodes for communication [48], [49]. Moreover, the path planning for detection and tracking multiple objects has been proposed [50]. Recently, UAVs acting as an additional network tier have been investigated [51], [52]. In particular, *AC-POCA* [51] has solved a challenging problem of assigning the radio channels for the UAV network construction, which is crucial in real-world deployments.

Interestingly, an *Adaptive* path planning [49] has solved the problem of data delivery to multiple nodes from a single source under delivery deadline constraints. It finds the next visiting

place considering the remaining deadline and the travel time in a heuristic manner. In Deadline Triggered Pigeon with TSP with Deadlines (DTP-TSP-D) [48], each UAV is dedicated to a single cluster of nodes, and it selects the next ground with the maximum timely delivered messages using TSP genetic algorithms. However, the prior work has mostly focused on the delivery to non-mobile nodes, lacking a some more challenging part of delivery to dynamically changing mobile nodes.

There has not been much work on data delivery to mobile users under time constraints in an infrastructure-less environment. Unless pre-installed storage is available on-site for temporary data stashing, it is very challenging to deliver *on-time* and *on-site* data to moving targets. To tackle the problem, we exploit mobile agents using spatio-temporally planned delivery in mobile networks. Our work provides a novel timely data delivery mechanism for MANETs using UAVs as message ferries, incorporating trajectory-based learning and prediction. We suggest a new approach to timely data delivery to multiple mobile users using only mobile UAV agents, by predicting the probable future trajectories of target users and estimating when and where to deliver data.

III. System Overview

In our scenario, it is assumed that there is a very minimal infrastructure that can detect a user's presence at certain spots (such as a booth in a fair event, or a well-known place in a city) via wireless tagging. An administrative center that wants to deliver some useful data to each user may use information about their past location to predict their likely future movement paths. Along the probable visiting paths, a UAV attempts to drop relevant packets at some identified probable spots to the target user from the air, within a delay constraint.

We assume that the UAVs can communicate with mobile ad-hoc nodes or with other UAVs using a wireless radio, such as IEEE 802.11. UAVs are equipped with global positioning system technology (GPS) and fly over a virtual cell-based topology. While traversing at a designated altitude over a region of interest (RoI), UAVs are assumed to be free from collisions and navigation control issues. UAVs are fully charged and have local storage, so they can complete their delivery mission before either one is out of capacity. We ignore transmission delay from UAVs to users, since it is negligible compared to the traversal time of the UAVs.

We assume that mobile users carry mobile devices – wearable devices or smart phones – equipped with GPS that keep recording their own positions. The trace information collected from all participating users is used in an offline learning phase. During this phase, it should be noted that we do not use any user-specific information, such as user ID, or geographical features such as buildings or roads.

Our goal is for each UAV to find a chronological visiting sequence whereby a UAV can encounter as many mobile users as possible, and deliver data to target users within each given deadline in each cell. We take a prediction-based approach by inferring some possible future trajectories of users based on their past movement patterns and using these movements to identify cells that each UAV should visit consecutively, in order to achieve the best possible on-time delivery performance.

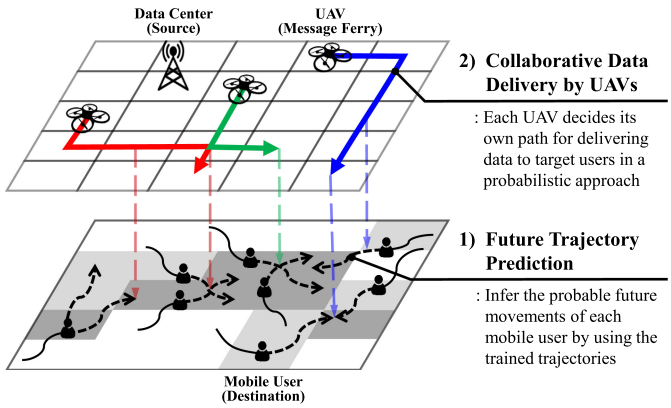


Fig. 1. Overall system overview of our data delivery mechanism to mobile users through mobility prediction and path planning by cooperative UAVs.

Our opportunistic yet strategic data delivery scheme using mobility prediction can be divided into three parts: 1) mobility pattern modeling; 2) future trajectory prediction; and 3) cooperative path planning of UAVs. (Fig. 1)

A. Procedure

1) Mobility Pattern Modeling: We build a mobility pattern model by learning and capturing some more-or-less regular movement patterns from real movement traces of users in an offline learning phase. To extract high-level meaningful movement patterns from the raw GPS traces of users, we convert each location record to its corresponding cell in a pre-defined virtual cell topology. Given complete cell-based trajectories, we classify them into several trajectory clusters, where a trajectory cluster may be considered as a complete and unique representative trip that a user can take from a source location to a destination location. The details can be found in Sec. IV.

2) Future Trajectory Prediction: In the data delivery execution phase, we use mobile users' recent partial trajectories to run our prediction algorithm that infers possible future cell-based trajectories. First, we search several well-matching trajectory clusters over the mobility pattern model built in the prior phase. Then, we identify which parts are overlap strongly with the user's current movement trace within the matching clusters. Finally, cells that come after the overlapped part are considered to be the future cell trajectory of a user. After finding the probable future visiting points, we compute the probability of the user appearing in each cell. The detailed prediction procedure is described in Sec. V-A.

3) Cooperative Path Planning of UAVs: Once all possible future cells where some target users may appear with a certain probability are identified, UAVs collaborate in data delivery. Since a UAV cannot be aware of the presence of other UAVs unless they are within radio range, it assumes the need to visit all possible cells, to deliver data to all possible target users. Each UAV continually computes the next cell to visit, in which the most users are predicted to be present within the deadline. Since

one-shot successful data delivery cannot happen in practice, due to the inherent imperfection of prediction, our mechanism lets a UAV make several attempts to deliver data, while maximizing efficiency. When UAVs come within radio range, data delivery loads for visiting cells are distributed among the encountered UAVs according to our task balancing algorithm. The path planning algorithm is presented in Sec. V-B.

IV. Mobility Pattern Modeling

The movements of mobile users are restricted due to physical structures, such as buildings, corridors, roads, or other natural environments. In some special-use facilities such as university campuses, users tend to have specific starting points and destinations. If it is possible to extract an underlying regular(-like) structure regarding human mobility, we may use this structure to understand how a mobile user will be likely to move within a geographical space in the near future. It would be useful to develop an effective method for capturing some principal regularity features by learning from *symptoms* that a mobile user has generated toward the environment while moving in the past.

In this section, we propose a method of constructing a mobility pattern model that extracts some regular movements from all possible past moving paths in the offline learning phase. Ultimately, we use the model to predict representative future trajectories in the online application phase.

A mobile device continuously records its spatial location at a specific time interval while traveling within a constrained region. Given past movement information collected from all mobile users, we represent one travel path from a starting point to a destination point as a *trajectory*. Using all trajectories of the mobile users, we classify similar trajectories into a single group, using sequence matching and clustering techniques, and finally construct a unique mobility pattern model. To efficiently search through these models in the application phase, we synthesize individual training trajectories into a condensed *cluster signature* for each cluster, enabling lightweight cluster signature-wise matching.

A. Clustering Similar Trajectories

We abstract the location of users at a grid cell level, instead of using the raw location, to capture the underlying essential movement patterns while suppressing unnecessary details. We define a fixed-size square sub-region in the RoI as a single virtual cell and transform a location trajectory into a sequence of virtual cell IDs. We thus represent a user's movement in the space of cell trajectory and could use sequence matching and clustering techniques for classifying a set of similar cell trajectories into a unique trajectory cluster.

1) Representing the Paths of Mobile Users: We represent a single journey of a mobile user by its spatial position at each time interval. We do not require any high-level user-specific or geography-specific information such as user identification, building name, or road name. We define a *trajectory* as a chronological sequence of virtual cell IDs, where the constrained area consists of square cells, as illustrated in Fig. 2. In case where the original raw position is $(18, 19)$, we map it into cell 16.

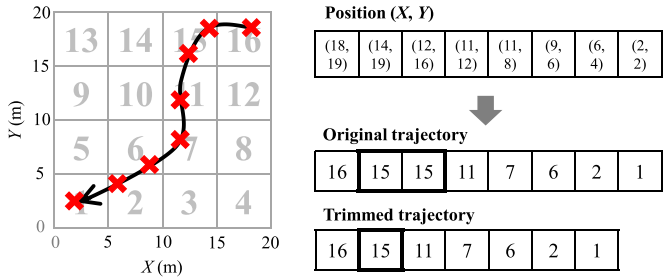


Fig. 2. Representing a series of spatial positions labeled in the form of (X, Y) (marked with red '×') for a moving path (marked with black full line) of a mobile user into a trajectory, which is a sequence of cell IDs with the cell size of 5 m.

It should be noted that a suitable cell size should be selected depending on the degree of mobility pattern regularity and the size of the RoI. If the cell size is too small, one trajectory can be too specific to be classified into a similar group. If the cell size is too large, one trajectory may lose its unique transitional features and patterns, making it indistinguishable from other similar yet distinct trajectories.

We can represent a trajectory in two ways: the *original* trajectory, which records a location cell at every time interval; and a *trimmed* trajectory that records only unique cell transitions at consecutive sampling times. Since the original trajectory is formed using time information, time-related cell visiting patterns may be investigated. The trimmed trajectory, on the other hand, may focus more on only the spatial distribution. For example, in Fig. 2, a moving path has an original trajectory of $\{16, 15, 15, 11, 7, 6, 2, 1\}$, and a trimmed trajectory of $\{16, 15, 11, 7, 6, 2, 1\}$.

2) Hierarchical Clustering Based on a Similarity Measure: We calculate a *similarity* measure for every pair of unique (trimmed) trajectories collected at the learning phase. We borrow the underlying idea from the edit distance [53], as in [5]. Calculation of the edit distance involves counting the number of operations required to covert from one sequence to another. Edit distance penalizes dissimilarity, and our similarity measure also rewards similarity in a complementary manner, when identifying nearby cells within the communication range with the highest similarity. Given two trajectories $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$, we can quantify the similarity of X and Y as follows:

$$S_{X,Y}(i,j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ \max \begin{cases} 0 \\ S_{X,Y}(i-1,j) + d \\ S_{X,Y}(i,j-1) + d \\ S_{X,Y}(i-1,j-1) + t(x_i, y_j) \end{cases} & \text{otherwise} \end{cases} \tag{1}$$

We solve this equation using dynamic programming. In our experiments, we consider only identical cells or adjacent cells within the communication range. If two cells, x_i and y_j, are identical or within the communication range, $t(x_i, y_j) = 1$. Otherwise, $t(x_i, y_j) = 0$. When we calculate the distance between

TABLE I
CALCULATING SIMILARITY $S_{X,Y}(i,j)$ OF $X = \{16, 15, 11, 7, 6, 2, 1\}$ AND $Y = \{15, 11, 10, 6, 5, 9\}$ BY DYNAMIC PROGRAMMING

	X		16	15	11	7	6	2	1
Y	j \ i	0	1	2	3	4	5	6	7
	0	0	0	0	0	0	0	0	0
15	1	0	1	1	1	1	1	1	1
11	2	0	1	2	2	2	2	2	2
10	3	0	1	2	3	3	3	3	3
6	4	0	1	2	3	4	4	4	4
5	5	0	1	2	3	4	5	5	5
9	6	0	1	2	3	4	5	5	5

cells, we quantify the farthest inter-cell distance. We consider the required insertion and deletion cases as dissimilar, that is, $d = 0$.

For example, let us consider two trajectories $X = \{16, 15, 11, 7, 6, 2, 1\}$ and $Y = \{15, 11, 10, 6, 5, 9\}$ as shown in Table I, on the grid topology of Fig. 2 with a communication range of 10 m. Initially, two cells, x_1 and y_1 are within radio range, and the similarity measure $S_{X,Y}(1,1) = 1$. Following Eq. 1, the similarity is calculated to be $S_{X,Y}(7,6) = 5$.

In long trajectories, the overall similarity measure tends to be higher than in shorter trajectories. Even if a short trajectory is perfectly matched to another trajectory, the similarity measure depends on the trajectory length, essentially penalizing short trajectories. We therefore normalize the similarity measure as follows:

$$S_{norm}(X,Y) = \frac{S_{X,Y}(m,n)}{\min(m,n)} \tag{2}$$

where m is the length of trajectory X and n is the length of trajectory Y. For example, we normalize $S_{X,Y}(7,6) = 5$ by the length of $\min(|X|, |Y|)$ where $|X| = 7$ and $|Y| = 6$. Finally, we produce the net similarity measure $S_{norm}(X,Y) = 5/6$. In our parameter setting, $S_{norm}(X,Y)$ should be less than or equal to 1 since the $S_{X,Y}(m,n)$ is always less than or equal to the length of the shorter trajectory.

We calculate the normalized similarity measure $S_{norm}(X,Y)$ for every pair of trajectories collected in the learning phase. We then apply a classic hierarchical clustering algorithm [54] that arranges the trajectories pairwise in a hierarchical manner, based on similarity. Each trajectory starts as a single cluster, and is fc then merged with other trajectories according to similarity, in a recursive manner. We use the *average linkage* [55] between two clusters in the hierarchical clustering, a measure of the average pairwise similarity of all trajectories across different clusters. For example, given two clusters $Clst_A = \{A_1, A_2, \ldots, A_k\}$ and $Clst_B = \{B_1, B_2, \ldots, B_l\}$, the *average linkage* of $Clst_A$ and $Clst_B$ is calculated as follows:

$$\overline{S(Clst_A, Clst_B)} = \frac{1}{k \cdot l} \sum_{i=1}^{k} \sum_{j=1}^{l} S_{norm}(A_i, B_j) \tag{3}$$

To measure the degree of similarity within a cluster, we introduce the intra-cluster similarity as:

$$\overline{S(Clst_A)} = \frac{2}{k(k+1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} S_{norm}(A_i, A_j) \tag{4}$$

The optimal number, N_{clst}, of clusters must be selected. Since we keep grouping clusters with a larger similarity, the intra-cluster similarity is an important criterion for cluster merging. Thus, we decide the desired number of clusters at the peak of the intra-cluster similarity regarding the number of clusters.

B. Adaptive Cluster Representation

Given an observed past trajectory of a target user, we aim to predict the future trajectory by matching it with training trajectories. Since it is not feasible to compare every single sequence over all clusters, we construct a compact signature that summarizes the essential features of the training trajectories belonging to a certain cluster. By creating a condensed probabilistic *cluster signature* for each cluster, the predicted trajectory can be calculated at the cluster level.

We design our cluster representation scheme based on a well-known concept, the *longest common subsequence (LCS)* [56]. As the first step, we identify the LCS common to all sequences within a cluster. To incorporate time information along the trajectory, we use the original trajectory. The signature of $Clst_A = \{A_1, A_2, \ldots, A_k\}$ is denoted as $sign_A(k)$.

However, the length of training trajectories within a cluster varies, and most trajectories are from incomplete trips, and have different start and end points. For these reasons, the longest common subsequences are calculated in a way which keeps only the common elements and may lose some front and rear elements, which do not always appear in common across trajectories, but are essential parts of a complete representative trip. For example, let us suppose that there are four trajectories in the same cluster, $\{1, 3, 7, 9\}$, $\{3, 5, 7\}$, $\{3, 7, 11\}$, and $\{3, 7, 9, 11\}$. The LCS of the cluster ignores the border parts of the trajectories, and extracts just the common elements, $\{3, 7\}$. To tackle this problem, we develop a relaxed LCS, in which the cluster signature can include the front and rear parts from the longer sequence. Our adaptive cluster signature is calculated in a recursive way, as follows:

$$sign_A(i) = relaxedLCS(sign_A(i-1),\ A_i) \tag{5}$$

In the above example, $sign_A(2) = \{1, 3, 7, 9\}$, $sign_A(3) = \{1, 3, 7\}$ (because the operation between 9 and 11 is a substitution), and $sign_A(4) = \{1, 3, 7, 9, 11\}$.

Lastly, we treat nearby cells within communication range as the same, because these cells are in same the communication territory. To take into account wireless coverage among cells, we introduce a higher-level group cell GC_{g_k}, to which cell x_i and cell y_j belong if the maximum distance between two cells is within the communication range. Initially, each cell forms its own group cell. A group cell can be merged with another group cell if all of the cells from the two group cells are within communication range. We apply the relaxed LCS (in Eq. 5) at the group cell level to obtain our final cluster signature.

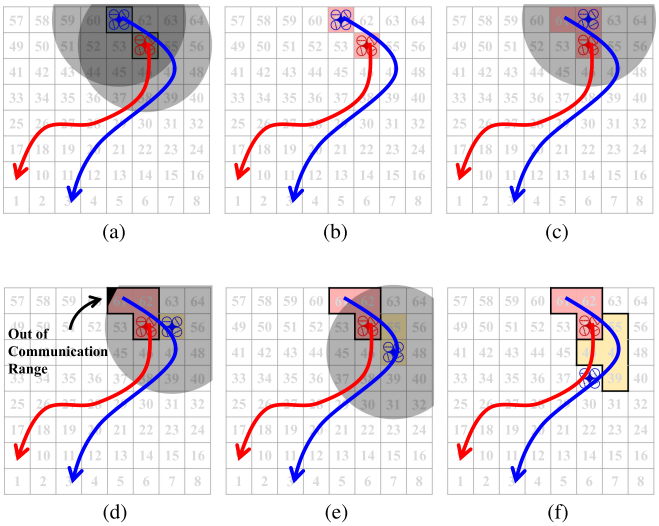


Fig. 3. Procedure of constructing a hierarchical cell structure over training trajectories for cluster representation. (a) Step 1. (b) Step 2. (c) Step 3. (d) Step 4. (e) Step 5. (f) Step 6.

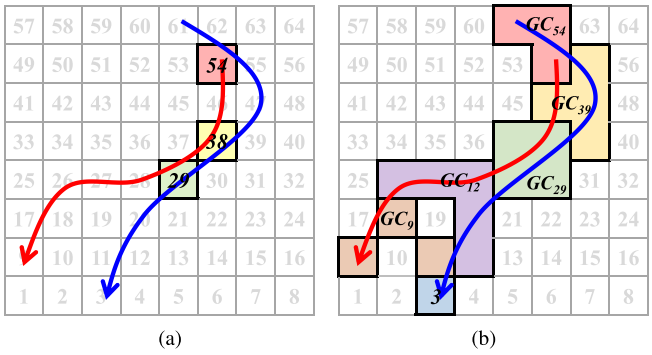


Fig. 4. Examples of cluster signature, which is generated by the original LCS and our adaptive LCS, respectively. (a) Original LCS. (b) Our adaptive LCS.

For example, given trajectories $X = \{61, 62, 54, 55, 47, 39, 38, 30, 29, 20, 12, 11, 3\}$ and $Y = \{54, 46, 38, 37, 29, 28, 27, 26, 18, 17, 9\}$ as in Fig. 3, we group cell 61 in trajectory X and cell 54 in trajectory Y into a group cell while assigning the lower cell ID to its group ID, since the two cells are within radio range. However, the group cell to which cells 54 and 61 belong cannot be merged with cell 55 since its inner cell 61 is out of radio range of cell 55, even though another inner cell, 54, is within radio range. Accordingly, our adaptive LCS between two trajectories becomes $\{GC_{54}, GC_{39}, GC_{29}, GC_{12}, GC_9, GC_3\}$ as shown in Fig. 4(b), whereas the original LCS is $\{54, 38, 29\}$ as shown in Fig. 4(a).

V. Cooperative Data Delivery by UAVs

In our scenario, users roam around a certain RoI area, and their presence can be tracked using wireless tagging via Bluetooth BLE beacon or some pre-installed wireless access points. The location-tagged information of a mobile user in the RoI is recorded at a server. Let us consider a situation in which an administrative center has a mission to deliver time-sensitive information, such as emergency evacuation information, announcements, or customized short-period advertisements to each mobile user who is currently moving around the RoI. We can investigate whether it is possible to find a more efficient way to deliver data using UAVs to mobile users currently in the RoI if their past presence information collected from local devices can be utilized as clues for predicting their current and future presence. To tackle this question, we first take a prediction-based approach by inferring some possible future trajectories of mobile users based on their past trajectory information.

Let us suppose that an administrative center needs to deliver data to 50 target users within the next three minutes. Even if their future trajectory information can be leveraged, it is not a trivial problem for UAVs to find their mission paths for data delivery within a short period of time. It would be necessary to find a *crowded zone* in which many target users will be likely to roam around, and attempt to deliver data near the zone to as many users as possible.

In this section, we present a procedure for cooperative data delivery by UAVs which involves two steps: 1) predicting a set of future trajectories for each user; and 2) generating each UAV's navigation path in such a way as to cover as many users as possible. Using the known past presence, we try to obtain the most similar clusters by matching past trajectories with cluster signatures. From the best-matched clusters, we aggregate all the future cells and their probability of being visited for all users. Based on the inferred knowledge about users' future movements, we present a distributed path planning algorithm for UAVs to find a promising point to come in contact for data delivery to more users, or with higher probability. We also present a task-balancing algorithm that distributes data delivery loads evenly among UAVs encountered within radio range.

A. Future Trajectory Prediction

After constructing the mobility database, we want to infer the future movement of each user using the training trajectories. Firstly, we calculate the similarities between a user's recent presence and each cluster signature, and select the best-matching clusters . Then, we align the recent user trajectory to all training trajectories in the selected clusters, and finally identify the cells visited thereafter as probable future visiting cells.

1) Finding Matching Clusters: Given the previous visiting location of users with a history size of L, we convert the location trajectory of user into its corresponding cell-based test trajectory X_{test}. We calculate the similarity measure $S_{norm}(X_{test_i}, sign_A(n_A))$ between X_{test_i} and each cluster signature $sign_A(n_A)$ $(= \{GC_1, GC_2, \ldots, GC_{n_A}\})$. To calculate $t(x_i, GC_j)$, we let $t(x_i, GC_j) = 1$ when x_i can communicate with all the cells belonging to GC_j. After calculating the similarity measure over all N_{clst} clusters, we identify the K best matching clusters with the highest similarity measures. The selection of a suitable K value needs to be determined empirically.

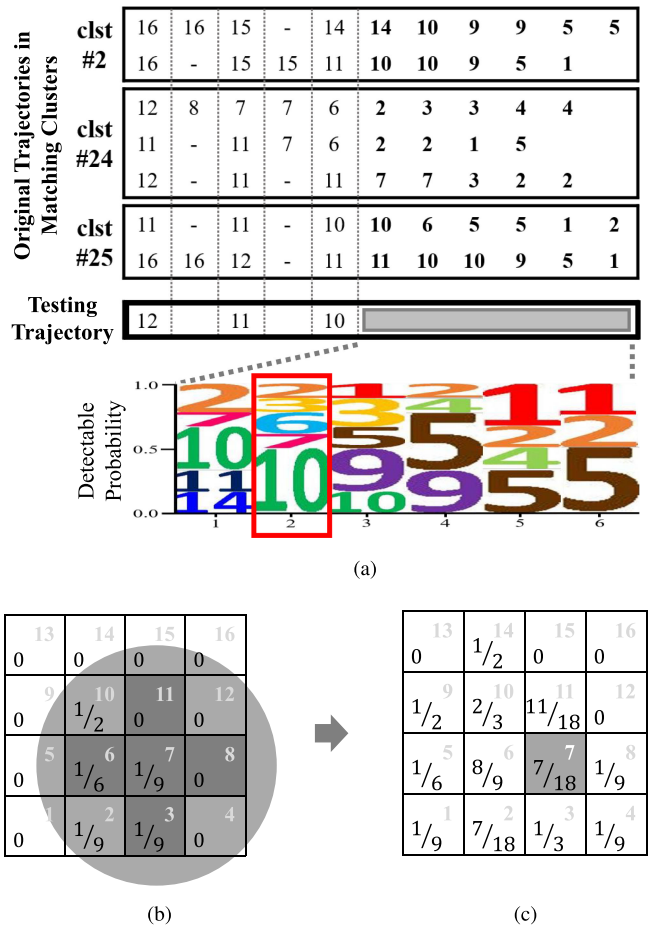


Fig. 5. Procedure of extracting predicted cells and calculating the probability for a user to appear at cells. (a) Probabilistic representation of predicted trajectories. (b) Detectable probability $P_t(U,C)$ at $t = t_0 + 2\Delta$. (c) Extended detectable probability $\overline{P}_t(U,C)$ at $t = t_0 + 2\Delta$.

After selecting the matching clusters, we perform an alignment between the *original* test trajectory, which records the original visiting cell at each sampling time, and each *original* training trajectory from the clusters. We use the *original* trajectory at this time because we want to extract the preceding cells followed by the matched cell parts from the as-is trajectory. We apply the similarity calculation based on dynamic programming, as shown in Eq. 1, and choose the trajectory index with the maximum similarity . The corresponding cells located at the following index are used as the future trajectory cells of the user. As illustrated in Fig. 5, given a test trajectory of $\{12, 11, 10\}$, we can select the best matching three clusters . Using the above procedure of comparing the test trajectory against original training trajectories within the matching clusters, the test trajectory is aligned with all original trajectories. The following cells over the next six time slots are identified as the future visiting cells of the user.

We can calculate the computation overhead for finding the matching clusters over a cluster signature and for finding the matching index over raw training trajectories within the matching clusters. Assuming that the upper bound of the similarity calculation complexity is C, the entire computation complexity is approximately given by $O\big((N_{clst} + K \cdot N_{trainTraj}/N_{clst}) \cdot C\big)$ where $N_{trainTraj}$ is the total number of training trajectories.

2) Probabilistic Representation of Potential Trajectories: After predicting the cells in which a user might appear based upon their past trajectory, we calculate a *detectable* probability at each future time slot for the target user . For a future time t, we first check the effective number of matching clusters. If we use K matching clusters, we usually apply the weight of $1/K$ to the probability calculation. If there is no predicted cell belonging to a cluster at time t, we use the adjusted weight $1/K_t$, so that only the effective number of matching clusters, K_t, should be counted. Then, we reflect the chance of cell appearance from the predicted trajectory out of all trajectories within the cluster. Thus, the probability of occurence of cell C at time t in a predicted trajectory X from cluster i for user U, $P_{X_t}(U,C)$ is given by $1/(K_t \cdot N_{clstTraj_{i,t}})$ where $N_{clstTraj_{i,t}}$ is the effective number of trajectories including a predicted cell at time t within cluster i. In the example of Fig. 5(a), at the sixth future time slot, there are only two effective matching clusters, and only one cell, cell ID 5, is included within cluster 2, and two cells, cell IDs 1 and 2, are included in cluster 25. The detectable probabilities at cells 1, 2, and 5 is calculated as $1/(2 \cdot 2)$, $1/(2 \cdot 2)$, and $1/(2 \cdot 1)$, respectively. Then, we accumulate the detectable probability at time t over predicted trajectories as follows:

$$P_t(U,C) = \Sigma_{xid \in predictedTrajs}\, P_{xid_t}(U,C) \tag{6}$$

Two distinct cells within the radio range, are treated as qualitatively the same cells taking into account the wireless communication effect. We incorporate this consideration when calculating the detectable probability as an extended version, $\overline{P}_t(U,C)$, as follows:

$$\overline{P}_t(U,C) = \Sigma_{cid \in nearby(C)}\, P_t(U,cid) \tag{7}$$

where $nearby(C)$ is a set of nearby cells that are within communication range of cell C.

From the example in Fig. 5(a), at the next second time slot, the detectable probability at cells for each predicted trajectory is accumulated over all the predicted trajectories as $P_t(U,C)$ in Fig. 5(b). Based on the accumulated detectable probability over all possible future trajectories, we finally obtain the extended detectable probability $\overline{P}_t(U,C)$ as in Fig. 5(c). Thus, our prediction algorithm infers the future cells which are visited from the previous behavior of a user.

B. UAV Path Planning With Task Balancing

Given the inferred future cells visited by users, we want only a limited number of UAVs to carry time-sensitive data to each of them. In a situation in which users roam around the region, it is very challenging to find a cell visiting schedule so that at least one UAV can arrive at a right location at a right time and deliver data to a user. However, since there is a high chance that the UAV does not encounter the target user in only one trial in practice, a UAV's path planning and data delivery attempts should be contiguous yet efficient until a successful data delivery to the target user is achieved within the original deadline.

Algorithm 1: Probabilistic Path Planning by UAVs.

Input: current time: T, current cell: $cell_T$, $targetUserList$, $RoICellList$, $\overline{P}_t(U,C)$, exploration range: R_E, $W_{penalty}$, $finalFlag_T$
Output: next cell ID to move: $cell_{T+1}$, $finalFlag_{T+1}$
// I. Deliver data at current cell and assign a penalty to the mismatched trajectories of users who were expected, but are actually not detected at this cell
Deliver data to target users at $cell_T$;
if $finalFlag_T == TRUE$ **then**
  for user $U \in targetUserList$ not at $cell_T$ && $P_{X_T}(U, cell_T) > 0$ **do**
    for time $t = T+1, T+2, \ldots$ **do**
      $P_{X_t}(U,C) = P_{X_t}(U,C) \times W_{penalty}$;
    end for
  end for
  Re-calculate $\overline{P}_t(U,C)$;
end if
// II. Calculating the accumulated probability for each cell
for cell $C \in RoICellList$ && C in R_E **do**
  $\overline{P}_{T+1}(C) = 0$;
  for user $U \in targetUserList$ **do**
    $\overline{P}_{T+1}(C)$ += $\overline{P}_{T+1}(U,C)$;
  end for
end for
// III. Task balancing among encountered UAVs
if there are communicable UAVs **then**
  Exchange the delivery history and update $targetUserList$;
  Select $masterUAV$;
  Call *task-balancing*($\overline{P}_t(C)$) for $t = T+1, T+2, \ldots$;
end if
// IV. Finding the most probable cell
$maxCell = arg\,max_{cid \in RoICellList}(\overline{P}_{T+1}(cid))$;
if $maxCell$ can be arrived in $(T+1)$ **then**
  $cell_{T+1} = maxCell$;
  $finalFlag_{T+1} = TRUE$;
else
  $cell_{T+1}$ = the farthest cell reachable at $t = T$+1 toward $maxCell$;
  $finalFlag_{T+1} = FALSE$;
end if

While a UAV can generate its own distributed path planning based on knowledge inferred about users, we consider some more optimization in case of multiple UAVs, by reducing duplicate coverage and performing load balancing among UAVs. Improvement of efficiency of path planning for UAVs would produce more time for eventual timely data delivery. We develop a task balancing scheme which is performed when a UAV encounters other UAVs within radio range.

1) Path Planning of UAVs: Each UAV performs its own distributed navigation based on the inferred future cells visited by users and their probabilities of occupation. When there are a large number of users, it is inefficient for a UAV to decide its future trajectory based on data from only one target user, because no prediction can be guaranteed to be accurate. To address this issue, we calculate the probability of detection for all users as an *accumulated* detectable probability, so that UAVs could classify cells as being part of a *crowded zone* and *reliable zone* . We calculate the accumulated detectable probability by summing up the extended detectable probability for each cell C, over all possible users, at time t as follows:

$$\overline{P}_t(C) = \Sigma_{uid \in targetUsers}\, \overline{P}_t(uid, C) \tag{8}$$

A UAV moves to the most probable cell out of these cell candidates that satisfy $\overline{P}_t(C) > 0$. However, if we consider only cells reachable from the current cell by the next time slot, the UAV tends to move around only a limited local region, getting trapped in local optima. To escape from local optima, we relax the cell candidate qualification. Some distant important cells may need to be included, so that the UAV strives to move toward them, even if they are not reachable by the next time point. Thus, we include cells with non-zero detectable probability with $\overline{P}_t(C) > 0$ within an exploration range window R_E. If the UAV cannot arrive at the selected cell by the next time slot, the UAV continues to run its visiting cell selection for the next time slot. When there are multiple cells with the same maximum probability, the UAV selects the closest cell among them and flies to it by the next time slot. In the case where there are no cells with non-zero detectable probability, it stays at the current cell for the next time period.

Once the UAV arrives at a cell, it sends data to all communicable target users, including the ones expected to be encountered. If a target user successfully receives its data within its designated deadline, it sends an acknowledgment to the UAV so that the UAV can maintain an up-to-date target user list. If the UAV fails to deliver within the deadline, it erases the expired user from the target user list. To reflect on-going prediction-based delivery performance for precedent prediction with improved accuracy, we penalize delivery failure due to incorrect predictions. If the UAV visits a cell with the highest detectable probability, but fails to encounter the expected target user at the cell, we give a penalty weight $W_{penalty}$ to the corresponding cell from the predicted trajectory X for user U from the next time slot $t = T+1, T+2, \ldots$ as follows:

$$P_{X_t}(U,C) \leftarrow P_{X_t}(U,C) \times W_{penalty} \tag{9}$$

The full UAV path planning is described in Algorithm 1.

2) Task Balancing Among Encountered UAVs: When multiple UAVs meet, they perform task balancing by dividing cells to visit evenly amongst them. They exchange information about the target users to which each UAV has completed data delivery, and the latest updated probability information, $P_{X_t}(U,C)$, of the remaining target users over the next time slot t. UAVs merge this information, and execute task balancing amongst all cells remaining to be visited, considering their geographical distribution, user-specific and cell-specific probability, and remaining delivery deadline .

Algorithm 2: Task Balancing among Encountered UAVs.

Function *task-balancing* (encountered UAV IDs, current locations of encountered UAVs, $accumProb$)
  if $isMasterUAV == TRUE$ **then**
    K-$meansClustering$(# of encountered UAVs);
    Calculate $sumProb(clst)$;
    $stdSumProbClst = std(sumProb(clst))$;
    while $endFlag == TRUE$ **do**
      $maxClst = max(sumProb(clst))$;
      for cell $C \in maxClst$ **do**
        if $isBoundaryCell == TRUE$ $\&\&min_{C \in maxClst}(accumProb(C))$ **then**
          Assign cell C to its nearest cluster;
        end if
      end for
      Re-calculate $sumProb(clst)$;
      $stdSumProbClst = std(sumProb(clst))$;
      if $stdSumProbClst$ decreased **then**
        $endFlag = FALSE$;
      else
        $endFlag = TRUE$;
      end if
    end while
    Assign each cluster to the closest UAV;
  else
    Receive task balancing result from $MasterUAV$;
  end if
end Function

We devise a variant of the K-means clustering algorithm [57] for task division. We adjust the acquired clustering outcomes to reflect per-cell probability, so that cells to visit can be more evenly distributed . At initial setup, the UAV which can directly reach more UAVs than the others is chosen as the master UAV, and plays the main role in clustering. It should be noted that some more advanced methods for master UAV selection can be devised based on their past role history. The master UAV performs an original K-means clustering to classify geographically close cells into the same cluster. Then, a fine adjustment procedure is followed. We calculate the per-cluster detectable probability by aggregating the detectable probability across member cells and future time slots as follows:

$$\overline{P}(clst) = \Sigma_{C \in clst} \Sigma_{\forall t} \overline{P}_t(C) \tag{10}$$

We select one cluster with the highest value of $\overline{P}(clst)$, which is anticipated to conduct relatively more crucial tasks than the other clusters. To mitigate congestion in one or several clusters, we adjust cells on the boundary of the selected cluster and finally reach a balanced point where the probability discrepancy among clusters is minimized. Among boundary cells of which at least one adjacent cell belongs to a different cluster, we try to move one cell with the lowest probability from the selected cluster to its nearest cluster and update the probability of the affected clusters. We continue this cluster adjustment procedure until the probability deviation among clusters no longer decreases.

TABLE II
SIMULATION ENVIRONMENT AND PARAMETERS

Simulation Environment	
Territory area	$600 \times 1000\ m^2$
Cell size	$5\ m$
# of target users	$5 \sim 60$
Packet delivery deadline	$0\ s \sim 180\ s$
UAV speed	$12\ m/s$
Communication radio range	$30\ m$
Time interval	1 s
Simulation Parameter	
# of UAV	$1 \sim 5$
# of clusters (N_{clst})	64
# of matching clusters (K)	3
History size (L)	$180\ s$
Exploration range (R_E)	$120\ m$
Penalty weight ($W_{penalty}$)	0.3

After finishing the clustering adjustment, we assign each cluster to the closest UAV from its cluster center using the Hungarian method [58]. Each UAV then runs its path planning according to the procedure described in Sec. V-B1. The detailed task balancing procedure is described in Algorithm 2.

VI. EVALUATION

We validated *Rendezvous* using real-world trace-driven simulation. We collected 172 unique real-world mobile user traces from students over a period of two months, using an RoI of $600 \times 1000\,\mathrm{m}^2$ in a university campus at Ewha Womans University, Seoul, South Korea. GPS positions with latitude and longitude were recorded with every second, and were then converted into the cartesian coordinate system. The trace data were sanitized after using the reversed trajectories, splitting some trajectories with missing parts of longer than five seconds, and filling out with the latest records in other cases. The average moving speed of mobile users was measured at $1.3\,\mathrm{m/s}$. A cell granularity of $5 \times 5\,\mathrm{m}^2$ was used to convert a set of location sequences to a set of cell trajectories (Fig. 6). We used 80% of the collected trajectories for training and 20% for testing.

A network layer was simulated with UAVs where the flying speed of the UAVs was $12\,\mathrm{m/s}$ and the communication radio range among UAVs and mobile devices was 30 m for algorithm-level validation (Table II). We varied the packet delivery deadline from the most stringent case of $[0, 60]$s up to the most relaxed case of $[120, 180]$ s. One time slot unit was 1 s, and the number of mobile users varied from 5 to 60. The number of matching clusters, K, was set to three, the history size, L, was 180 s, the exploration range, R_E, was 120 m, and the weight penalty, $W_{penalty}$, was 0.3 unless otherwise noted.

A. Mobility Pattern Modeling

During the mobility pattern modeling phase, we used 881 training trajectories to calculate the similarity measures and perform the hierarchical clustering. To empirically determine an optimal number of clusters, the intra-cluster similarity measure was quantified by varying the number of clusters from 881 down to 1 (Fig. 7). We selected 64 clusters with the highest intra-cluster similarity of 0.83 out of 1.0. The average number of trajectories per cluster was $881/64 \approx 13.8$.

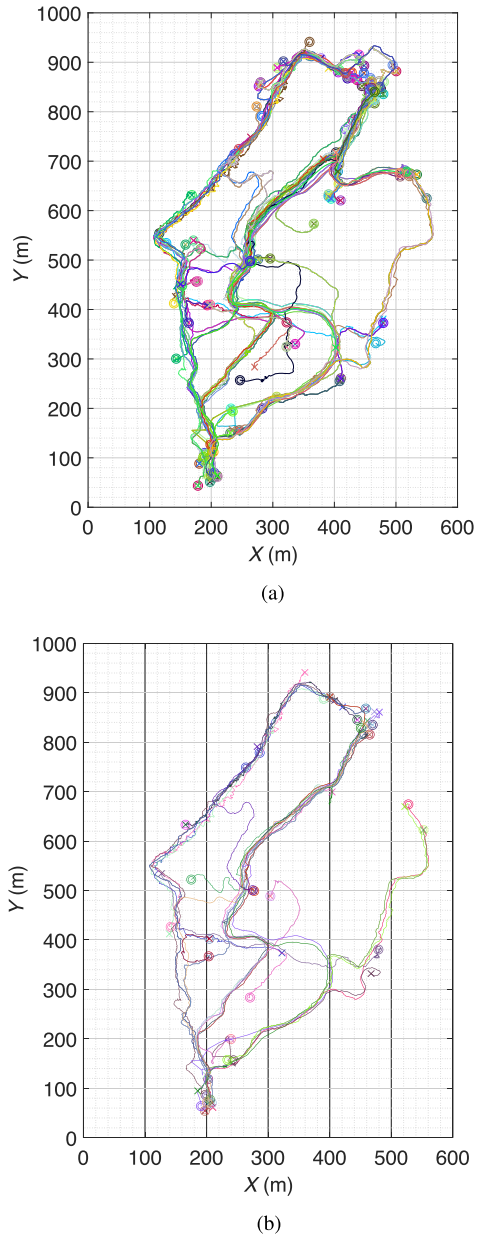


Fig. 6. A real-world university campus trace dataset of 138 training trajectories and 34 testing trajectories with starting points (marked with '◎') and destinations (marked with '×'). (a) Training trajectories. (b) Testing trajectories.

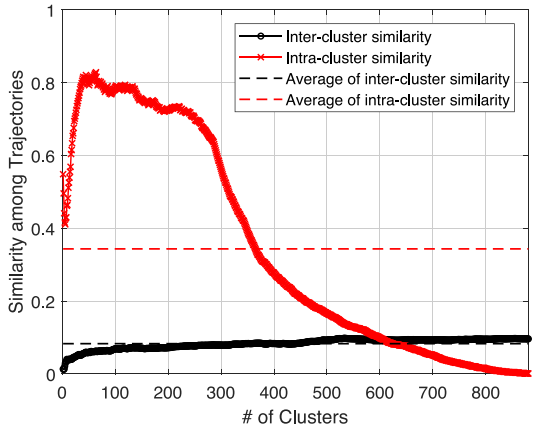


Fig. 7. Intra-cluster and inter-cluster similarities depending on the selected number of clusters.

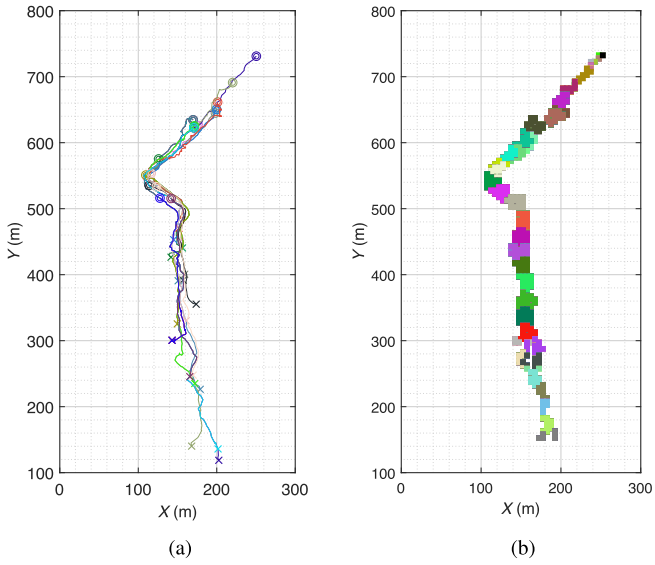


Fig. 8. A trained cluster and its adaptive signature. (a) 21 training trajectories belonging to cluster #13 where the original LCS is null with no common cells at all. (b) Our adaptive cluster signature of cluster #13 where a group cell is represented with the same color.

We constructed a mobility pattern model using 64 mobile trajectory clusters; a representative cluster in terms of trajectory distribution is shown in Fig. 8. As plotted in Fig. 8(a), 21 trajectories were classified into cluster 13, and they share some common patterns. If the original LCS as a cluster signature is applied to these trajectories, it loses all crucial intermediate cells that should have been captured as common. When our adaptive LCS signature is used, as in Fig. 8(b), the cluster signature summarizes the overall high-level movement behavior well. We verified with other clusters and their cluster signatures that our mobility pattern model with a mixture of dynamic programming and hierarchical clustering works well with real-world training trajectories, and training trajectories were categorized into clusters with distinctive summary signatures after successfully filtering out temporal and spatial deviations.

B. UAV Data Delivery

We examined the data delivery performance of UAVs using our mobility prediction and path planning algorithms in the UAV data delivery phase. We aimed to deliver each data item to 45 target users within different deadlines, which were determined using a uniform random distribution over $[120, 180]$s for each run, unless otherwise noted.

We measured what percentage of target users successfully received data within the appropriate deadline, as the on-time serviced user percentage. To validate more in-depth prediction performance, we quantified the percentage of target users who successfully received data based on our prediction decision, as

the *intended delivery ratio*. We separately measured the percentage of target users who received at a cell for which there was no probability estimated for that user to appear as the *unintended delivery ratio*.

In our experiments, UAVs started at randomly distributed points for 10 different runs, and we took the average and standard deviation .

1) Parameter Settings: Before verifying the efficacy of our system, we investigated how the parameter settings affected the overall on-time data delivery performance using prediction and path planning (Fig. 9). Starting with one UAV, we varied the number of matching clusters, K, and history size, L . We also varied the penalty weight, $W_{penalty}$, and exploration range, R_E.

We first examined the effect of the number of matching clusters K on the on-time serviced delivery percentage and the intended delivery ratio. As shown in Fig. 9(a), both performance metrics improved up to 15% as K increased from 1 to 3. After three clusters, the performance degraded slightly. This observation implies that the use of several well-matching candidates rather than only the single best matching candidate contributes to the generation of strategies which are not too specific, and achieving on-time delivery with high fidelity.

As we varied the history size, L, on the test trajectory, the prediction performance and the overall on-time serviced user percentage were affected (Fig. 9(b)). Our results showed that using the history of the previous 180 cells visited corresponding to the last three minutes of a user's behavior, produced the most accurate prediction, achieving higher on-time data delivery.

To compensate for delivery failure due to wrong predictions, subsequent path planning should also play an important backup role. We varied both penalty weight and exploration range, parameters which affect the cell visiting priority of the UAVs (Figures 9(c) and 9(d)). If the path planning algorithm penalizes a wrong visiting at a cell scaling the probability by a factor of 0.3, a 13.8% increase in on-time data delivery was achieved (Fig. 9(c)). When the cell selection range was extended from 12 m, which is the UAV travel distance for one slot time of 1 s, to 120 m, some cells in crucial directions were included as candidates for visiting . This extension meant that the UAV would not be stuck in its local area, but could move toward globally optimal cells, producing an improvement in delivery performance of up to 39.8%.

The parameter values were selected on the basis of the experiments reported in Fig. 9. Having chosen suitable values, we focused on investigating the performance of how our algorithm with respect to each different environmental factor.

2) Reliability Under Various Conditions: We increased the number of UAVs to four, and investigated how the on-time data delivery was affected by different service deadlines selected from a uniform random distribution with 60 second intervals (Fig. 10(a)). Under the tightest deadline case of $[0, 60]$ seconds, on-time data delivery was successful for only 43.1% of intended deliveries, with correct prediction measured at 37.3%. As the service deadline was relaxed, the on-time delivery increased significantly, with a success rate of 85.1% at a service deadline

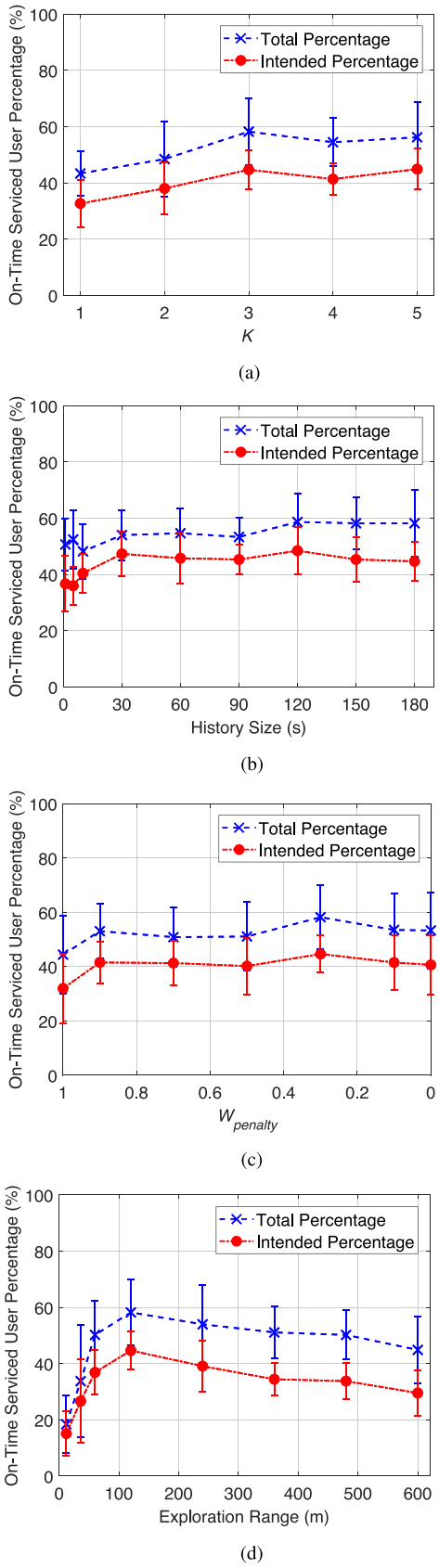
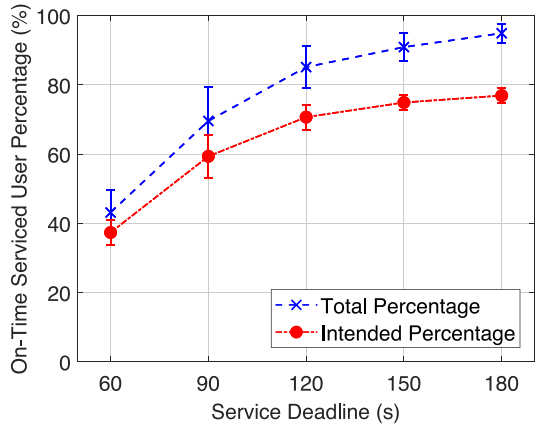
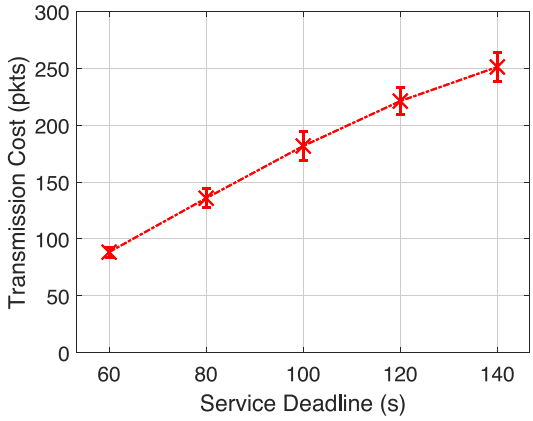


Fig. 9. On-time serviced user percentage with respect to parameters using one UAV. (a) Effect of # of matching clusters K. (b) History size L. (c) Effect of $W_{penalty}$. (d) Effect of the exploration range R_E.



(a)



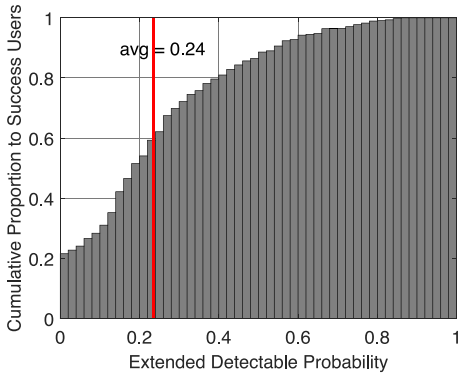
(b)

Fig. 10. On-time serviced user percentage and transmission cost by varying service deadline using 4 UAVs. (a) On-time data delivery ratio. (b) Transmission cost.

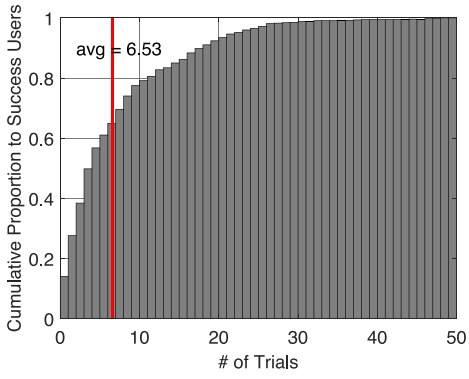
of $[60, 120]$ seconds . After this point, the intended delivery rate remained constant, while the total delivery rate continued to improve slowly.

We also examined the transmission cost of sending broadcast packets from UAVs and acknowledgment packets from target users to UAVs (Fig. 10(b)). As the service deadline increased, the UAVs had more opportunities to deliver data to more target users in a linearly increasing fashion.

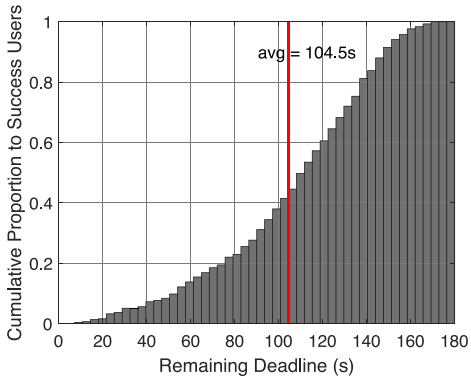
We analyzed the way in which different factors affected successful on-time data delivery (Fig. 11). We generated a cumulative distribution of the actual extended detectable probability, the number of trials till successful data delivery, and the remaining service deadline at the time of successful delivery among all 747 successful data delivery cases over 10 different runs. Fig. 11(a) shows that successful data delivery performance results from extended detectable probabilities. Even with the somewhat low prediction accuracy of 0.24 on average, our path planning algorithm ran with successive efficient delivery attempts of 6.53 on average. Both trajectory prediction and path planning are important for achieving high on-time data delivery performance. As shown in Fig. 11(c), the average remaining deadline was 104.65 seconds, where the cumulative success



(a)



(b)



(c)

Fig. 11. Cumulative distributions of successful data delivery with respect to prediction probability, the number of trials, and remaining deadline till successful delivery using 4 UAVs. (a) Extended detectable probability. (b) # of trials till successful delivery. (c) Remaining deadline.

percentage reached 45.11%. The distribution is almost linear, meaning that our data delivery effort works evenly over time.

We evaluated how our scheme performed with an increasing number of target users, using the same resources of service deadline and UAVs. As shown in Fig. 12, the on-time serviced user percentage is not significantly affected and the total percentage remained high, while the intended data delivery performance was slightly degraded . This observation indicates that our path planning algorithm came into play by compensating for some degraded prediction quality for the increased load, maintaining a

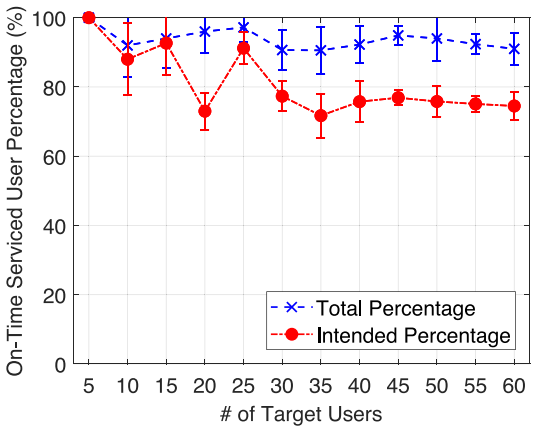


Fig. 12. On-time serviced user percentage with respect to target user load under 4 UAVs.

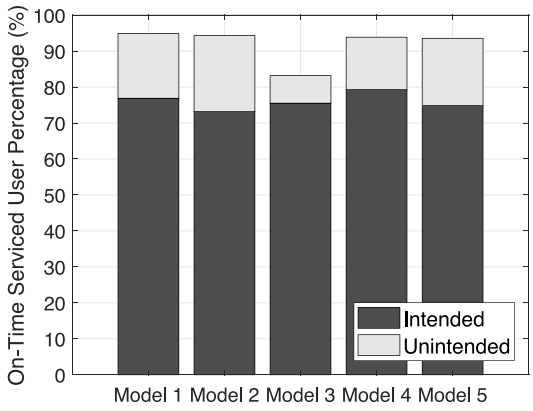


Fig. 13. On-time serviced user percentage with respect to model variation based on different choice of training and testing trajectories under 4 UAVs.

delivery performance of over 90% even for delivery to 60 target users.

To validate the stability of our mobility pattern modeling approach, we construct four different models by randomly splitting the trajectories collected into training and testing trajectories compared to the model that was previously used (denoted Model 1). As shown in Fig. 13, using four UAVs, each model constructed using different training trajectories with the same modeling approach showed very similar performance.

3) Comparative Data Delivery Performance: We compared *Rendezvous* against different path planning algorithms: *Levy's Flight (Levy)* [59], *Adaptive Path Planning (Adaptive)* [49], and *Maximum Probability First (MPF)*, and different prediction algorithms, *Direct* and *Perfect* (Table III). We devised a path planning algorithm, *MPF*, which chooses the next visiting cell based only on the maximum detectable probability, in a greedy manner without applying the exploration range. We implemented the *Levy*'s path generation algorithm based on random walks without using any prior knowledge of target users. We also compared our UAV data delivery scheme without prediction that tried to send data to the last known location of users without prediction of their current location. Using the assumption that the future

TABLE III
COUNTERPART ALGORITHMS FOR PERFORMANCE COMPARISON

Prediction \ Path	Levy [36]	Adaptive [56]	MPF	Ours
No	**Levy**	**Adaptive**	-	**Direct**
Ours	-	-	**MPF**	**Rendezvous**
Perfect	-	-	-	**Perfect**

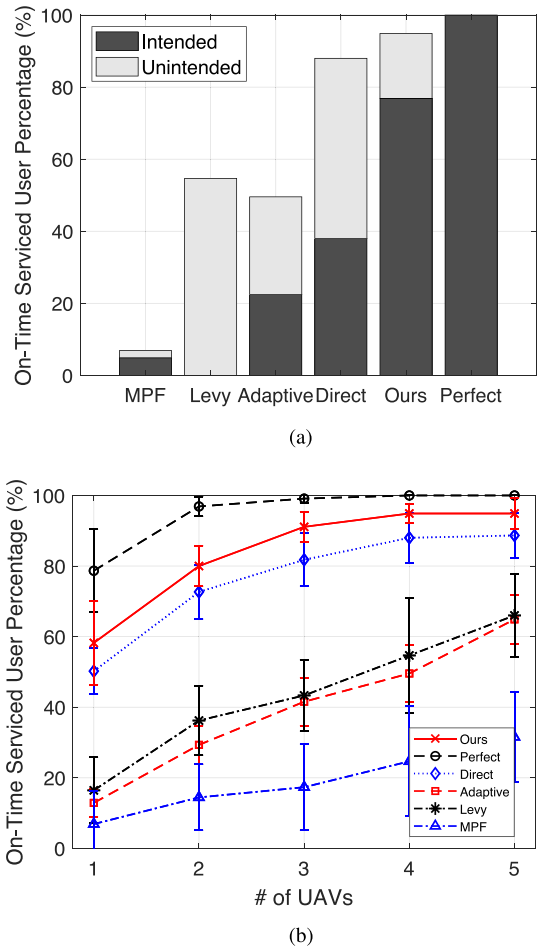


Fig. 14. On-time serviced user percentage with respect to intended data delivery and UAV resource compared to other path planning and prediction algorithms. (a) Proportion of intended data delivery under 4 UAVs. (b) Effect of UAV resource.

trajectory is perfectly known, we applied this perfect future knowledge to our path planning algorithm as *Perfect*, serving as the upper bound of our scheme. Lastly, we compared *Adaptive* without prediction, a path planning algorithm for multiple UAVs to deliver data to each static node under time constraints. Each UAV finds the next visiting place considering the remaining packet deadline and the travel time.

We quantified the overall on-time serviced user percentage and the intended data delivery ratio across *MPF*, *Levy*, *Adaptive*, *Direct*, and *Perfect* with *Rendezvous* (Fig. 14). Fig. 14(a) shows that *Perfect* reaches 100% in both metrics. This means that since UAVs are fully aware of the future visiting cells of mobile users,

the overall performance is affected by the path planning , and our path planning algorithm with task balancing performed reliably. *Rendezvous* achieved almost 95% delivery, of which 77% arose from prediction and 18% from opportunistic encounters, producing a relatively small performance gap from *Perfect*. If predictive knowledge was not exploited in data delivery, *Direct* had an overall 88% delivery performance, whereas its intended delivery was achieved only 38% of the time. Almost 50% of successful delivery cases came from opportunistic encounters between UAVs and users.

Regarding path planning , *MPF* had the worst performance, with a 25% success rate, due to its continuous myopic path decisions, traveling around locally-optimal cells. In our evaluation setting, *Levy* could deliver data with a success rate of 55% purely based on opportunistic encounters between UAVs and users. Since both *Direct* and *Levy* do not exploit any prediction knowledge for data delivery decisions, their data delivery success due to opportunistic encounter is similar, in the range of 50–55%. *Adaptive* reaches almost 50% with 27% of the intended deliveries , outperforming *Levy* thanks to more cost-effective path planning. However, it still has large performance gaps compared to our algorithm in both intended delivery and overall delivery rates. These results indicate that our work shows comparatively stable performance even under very erratic real-world situations, because our approach infers the future trajectories of mobile users, and exploits time for predictive data delivery with advanced path planning.

Finally, we evaluated on-time data delivery performance by varying the number of UAVs (Fig. 14(b)). As the number of UAVs used as message ferries increases, all of algorithms exhibit improved performance, since all of them have relatively more time for data delivery using more UAV resources for the same time deadline. Starting from one UAV, the performance difference between *Rendezvous* and *Perfect* shows a relatively large gap of 30% , whereas the gap is reduced to 5% based on efficient utilization of UAV resources made in our approach. *Levy* and *Adaptive* still lacked coverage of the area even with an increased number of UAVs. Leveraging the regular movement patterns of users and predicting future movements are therefore key components to support opportunistic data delivery. Most schemes showed similar delivery performance under the relaxed deadline range and when using a sufficient number of UAVs. However, our approach performed effectively under even considerably more limited environmental conditions.

VII. Conclusion

We have presented *Rendezvous*, an opportunistic approach to data delivery to mobile users within designated deadlines, using UAVs as message ferries, and incorporating the prediction of the mobility of targets. To learn the probable trajectories, we constructed a mobility pattern model by clustering similar trajectories, and represented each cluster with a summary adaptive signature to enable effective matching and prediction for the data delivery application.

In the data delivery phase, UAVs decide their own navigation paths according to our probabilistic path planning algorithm, based on inferred future trajectories of target users, searching over the mobility pattern model. We have demonstrated that our proposed system achieves reliable on-time data delivery in opportunistic mobile networks, through mobile trajectory-driven predictive delivery schedules by UAVs.

In future work, we will address the issue of further optimization of probability based task balancing for UAVs, to collaboratively reduce the duplicate data delivery. We will also consider the application of the approach to more practical settings, such as periodic or aperiodic data delivery to target users over dynamic runs. It would also be interesting to apply a social-based data delivery approach, by delegating some data delivery tasks for a set of target users to users and predicted to be highly socially valued.

References

[1] I. Chuang, T. Lin, C. Lo, and Y. Kuo, "Time-sensitive data dissemination in opportunistic networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2014, pp. 2420–2425.

[2] J. Whitbeck and V. Conan, "Hymad: Hybrid dtn-manet routing for dense and highly dynamic wireless networks," *Comput. Commun.*, vol. 33, no. 13, pp. 1483–1492, 2010.

[3] M. Karrer, M. Agarwal, M. Kamel, R. Siegwart, and M. Chli, "Collaborative 6DoF relative pose estimation for two UAVs with overlapping fields of view," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 6688–6693.

[4] E. Larsen, L. Landmark, and Ø. Kure, "The effects of a UAV on a terrestrial MANET," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2018, pp. 816–821.

[5] H. Lee, M. Wicke, B. Kusy, O. Gnawali, and L. Guibas, "Predictive data delivery to mobile users through mobility learning in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5831–5849, Dec. 2015.

[6] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[7] S. R. Azzuhri, H. Ahmad, M. Portmann, I. Ahmedy, and R. Pathak, "An efficient hybrid MANET-DTN routing scheme for OLSR," *Wireless Pers. Commun.*, vol. 89, no. 4, pp. 1335–1354, 2016.

[8] S. Burleigh *et al.*, "Delay-tolerant networking: An approach to interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003.

[9] A. Galati, *Delay Tolerant Network*. Germany: LAP Lambert Academic Publishing, 2010.

[10] M. Harounabadi and A. Mitschele-Thiel, "Applying message forwarding and replication to multi-UAV message ferry networks," *Mobile Netw. Appl.*, vol. 23, no. 5, pp. 1337–1346, 2018.

[11] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 109–120, Aug. 2005. [Online]. Available: http://doi.acm.org/10.1145/1090191.1080106

[12] S. K. Pandey and A. K. Singh, "Efficient prophet with buffer management for multicasting in DTN," in *Proc. IEEE Int. Conf. Inventive Res. Comput. Appl.*, 2018, pp. 1200–1205.

[13] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETS," in *Proc. 8th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, ser. MobiHoc '07. New York, NY, USA, 2007, pp. 32–40. [Online]. Available: http://doi.acm.org/10.1145/1288107.1288113

[14] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Proc. Int. Workshop Service Assurance with Partial Intermittent Res.*, 2004, pp. 239–254.

[15] J. Papaj, L. Dobos, and A. Cizmar, "Hybrid MANET-DTN and a new algorithm for relay nodes selection," *Wireless Pers. Commun.*, vol. 96, no. 4, pp. 5145–5166, 2017.

[16] H. Zhou, V. C. Leung, C. Zhu, S. Xu, and J. Fan, "Predicting temporal social contact patterns for data forwarding in opportunistic mobile networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10 372–10 383, Nov. 2017.

[17] B. R. Karki, A. Hämäläinen, and J. Porras, "Social networking on mobile environment," in *Proc. ACM/IFIP/USENIX Middleware Conf. Companion*, ser. Companion '08. New York, NY, USA, 2008, pp. 93–94. [Online]. Available: http://doi.acm.org/10.1145/1462735.1462760
[18] I. G. A. S. Negara, L. V. Yovita, and T. A. Wibowo, "Performance analysis of social-aware content-based opportunistic routing protocol on MANET based on DTN," in *Proc. IEEE Int. Conf. Control, Electron., Renewable Energy Commun.*, 2016, pp. 47–53.
[19] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE RAP: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.
[20] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "MobiClique: Middleware for mobile social networking," in *Proc. 2nd ACM Workshop Online Soc. Netw.*, ser. WOSN '09. New York, NY, USA, 2009, pp. 49–54. [Online]. Available: http://doi.acm.org/10.1145/1592665.1592678
[21] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
[22] T. M. T. Do, O. Dousse, M. Miettinen, and D. Gatica-Perez, "A probabilistic kernel method for human mobility prediction with smartphones," *Pervasive Mobile Comput.*, vol. 20, pp. 13–28, 2015.
[23] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *Proc. IEEE Intell. Vehicles Symp.*, 2012, pp. 141–146.
[24] Y. Wang *et al.*, "Regularity and conformity: Location prediction using heterogeneous mobility data," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1275–1284.
[25] Q. Lv, Y. Qiao, N. Ansari, J. Liu, and J. Yang, "Big data driven hidden markov model based individual mobility prediction at points of interest," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5204–5216, Jun. 2017.
[26] P. Kothari and A. Alahi, "Human trajectory prediction using adversarial loss," in *Proc. 19th Swiss Transp. Res. Conf.*, Apr. 2019.
[27] I. Jawhar, N. Mohamed, J. Al-Jaroodi, D. P. Agrawal, and S. Zhang, "Communication and networking of UAV-based systems: Classification and associated architectures," *J. Netw. Comput. Appl.*, vol. 84, pp. 93–108, 2017.
[28] K. P. Valavanis, *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*, vol. 33. New York, NY, USA: Springer Science & Business Media, 2008.
[29] M. C. Chuah and W. Ma, "Integrated buffer and route management in a DTN with message ferry," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2006, pp. 1–7.
[30] M. Harounabadi, M. Bocksberger, and A. Mitschele-Thiel, "Study on the network architectures for message ferry networks with multiple UAVs," in *Proc. IEEE 10th Int. Conf. Ubiquitous Future Netw.*, 2018, pp. 59–63.
[31] S. Roy, D. Tomasi, M. Conti, S. Bhusal, A. Roy, and J. Li, "Optimizing message ferry scheduling in a DTN," in *Proc. 16th ACM Int. Symp. Mobility Manage. Wireless Access*, 2018, pp. 113–117.
[32] M. M. Bin Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse Ad Hoc Networks with mobile nodes," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, ser. MobiHoc '06. New York, NY, USA, 2006, pp. 37–48. [Online]. Available: http://doi.acm.org/10.1145/1132905.1132910
[33] Y. Liu, H. Wu, Y. Xia, Y. Wang, F. Li, and P. Yang, "Optimal online data dissemination for resource constrained mobile opportunistic networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5301–5315, Jun. 2017.
[34] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile Ad Hoc Networks," in *Proc. 5th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, ser. MobiHoc, 2004, pp. 187–198. [Online]. Available: http://doi.acm.org/10.1145/989459.989483
[35] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies.*, Mar. 2005, vol. 2, pp. 1407–1418.
[36] K. L. Hoffman, M. Padberg, and G. Rinaldi, *Traveling Salesman Problem*. New York, NY, USA: Springer, 2013, pp. 1573–1578.
[37] S. Kim and I. Moon, "Traveling salesman problem with a drone station," *IEEE Trans. Syst., Man, Cybernet.: Syst.*, vol. 49, no. 1, pp. 42–52, Jan. 2019.
[38] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Comput. Ind. Eng.*, vol. 99, pp. 300–313, 2016.
[39] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan, "Rich vehicle routing problem: Survey," *ACM Comput. Surv. (CSUR)*, vol. 47, no. 2, Jan. 2015, Art. no. 32.
[40] B. L. Golden, S. Raghavan, and E. A. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*, vol. 43. New York, NY, USA: Springer Science & Business Media, 2008.
[41] P. Kitjacharoenchai, M. Ventresca, M. Moshref-Javadi, S. Lee, J. M. Tanchoco, and P. A. Brunese, "Multiple traveling salesman problem with drones: Mathematical model and heuristic approach," *Comput. Ind. Eng.*, vol. 129, pp. 14–30, 2019.
[42] N. Bansal, A. Blum, S. Chawla, and A. Meyerson, "Approximation algorithms for deadline-TSP and vehicle routing with time-windows," in *Proc. 36th Annu. ACM Symp. Theory Comput.*, ser. STOC '04. New York, NY, USA, 2004, pp. 166–174. [Online]. Available: http://doi.acm.org/10.1145/1007352.1007385
[43] D. Taş, M. Gendreau, O. Jabali, and G. Laporte, "The traveling salesman problem with time-dependent service times," *Euro. J. Oper. Res.*, vol. 248, no. 2, pp. 372–383, 2016.
[44] O. Polat, C. B. 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," *Eur. J. Oper. Res.*, vol. 242, no. 2, pp. 369–382, 2015.
[45] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Res.*, vol. 35, no. 2, pp. 254–265, 1987.
[46] P. Toth and D. Vigo, *The Vehicle Routing Problem*. Philadelphia, PA, USA: SIAM, 2002.
[47] C. Cheng, P. Hsiao, H. T. Kung, and D. Vlah, "Maximizing throughput of UAV-relaying networks with the load-carry-and-deliver paradigm," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2007, pp. 4417–4424.
[48] C. Barroca, A. Grilo, and P. R. Pereira, "Improving message delivery in UAV-based delay tolerant networks," in *Proc. IEEE 16th Int. Conf. Intell. Transp. Syst. Telecommun.*, 2018, pp. 1–7.
[49] J. Yoon, Y. Jin, N. Batsoyol, and H. Lee, "Adaptive path planning of UAVs for delivering delay-sensitive information to Ad-Hoc nodes," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2017, pp. 1–6.
[50] H. Van Nguyen, H. Rezatofighi, B.-N. Vo, and D. C. Ranasinghe, "Online UAV path planning for joint detection and tracking of multiple radio-tagged objects," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5365–5379, Oct. 2019.
[51] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "AC-POCA: Anticoordination game based partially overlapping channels assignment in combined UAV and D2D-based networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1672–1683, Feb. 2018.
[52] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, F. Ono, and R. Miura, "On a novel adaptive UAV-mounted cloudlet-aided recommendation system for LBSNs," *IEEE Trans. Emer. Topics Comput.*, vol. 7, no. 4, pp. 565–577, Oct.–Dec. 2019.
[53] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Phys. Doklady*, vol. 10, no. 8, pp. 707–710, 1966.
[54] F. Corpet, "Multiple sequence alignment with hierarchical clustering," *Nucleic Acids Res.*, vol. 16, no. 22, pp. 10 881–10 890, 1988.
[55] R. R. Sokal, "A statistical method for evaluating systematic relationship," *Univ. Kansas Sci. Bull.*, vol. 28, pp. 1409–1438, 1958.
[56] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *J. ACM*, vol. 21, no. 1, pp. 168–173, Jan. 1974. [Online]. Available: http://doi.acm.org/10.1145/321796.321811
[57] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
[58] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, no. 1-2, pp. 83–97, 1955.
[59] B. B. Mandelbrot, *The Fractal Geometry of Nature*, vol. 2. New York: WH Freeman, 1982.

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 the Ph.D. degree with Ewha Womans University. Her current research interests include wireless ad-hoc networks, vehicular networks, IoT security system, and deep learning.

A-Hyun Lee received the B.S. degree in computer science and engineering from Ewha Womans University, Seoul, South Korea, in 2019. She is currently working toward the M.S. degree with the School of Computer Science and Engineering with Seoul National University, Seoul, South Korea. Her research interests include wireless networks, Internet of Things, localization and deep reinforcement learning.

HyungJune Lee received the B.S. degree in electrical engineering from Seoul National University, Republic of Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, in 2006 and 2010, respectively. He is an Associate Professor with the Department of Computer Science and Engineering with Ewha Womans University, Seoul, Republic of Korea. He joined Broadcom as a Sr. Staff Scientist for working on research and development of 60 GHz 802.11ad SoC MAC. He also, 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.