Efficient Data Transmission Scheme for Massive High Altitude Platform Networks

Yuanyuan Wang, Chi Zhang, *Member, IEEE*, Taiheng Ge, and Miao Pan, *Senior Member, IEEE*

***Abstract*—Due to the wide coverage and station-keeping feature of high altitude platforms (HAPs), a large-area sensor network above remote regions can be achieved by the use of a mobile HAP ad hoc network. To realize efficient data transmission in the HAP network, we propose two schemes at the network layer and MAC layer, respectively. At the network layer, we propose a break and reconstitution (B&R) algorithm to construct an aggregation tree with minimum convergecast delay (MCD) for collecting data generated by each HAP. This scheme is beneficial in many monitoring systems, especially for time-critical and safety-critical tasks. In addition, a rapid parent selection (RPS) algorithm is presented to deal with the dynamic changes in the HAP network. At the MAC layer, we utilize a particle swarm optimization algorithm to optimize transmission and reception beamwidths to maximize the minimum transmission data rate. This scheme is beneficial for supporting tremendous data in the HAP network. Finally, we evaluate the performance of our proposed schemes through extensive experiments with real wind data and results show that they outperform some baseline approaches.**

***Index Terms*—High altitude platform (HAP), minimum convergecast delay (MCD), particle swarm optimization, tree construction.**

I. Introduction

With the technological innovations in terms of array antennas, aeronautical facilities, and battery energy, the usage of high altitude platforms (HAPs) in broadband coverage, natural disaster recovery, and environment monitoring has attracted much interest from both academia and industry [1], [2]. HAPs are high altitude balloons (HABs) and airships operating in the stratosphere approximately 20-30 km above the Earth's surface [3]. Compared with unmanned aerial vehicles, HAPs have the outstanding advantages of large load capacities and coverage area. The maximum coverage of an HAP can reach 60 km [4]. Since a variety of communication devices and sensors can be equipped on the HAP, mobile HAP ad hoc network can be utilized to realize a large-area sensor network above the remote regions, such as ocean and desert.

Mobile HAP ad hoc network has some unique characters. Due to the low water content in the stratosphere, there are practically no weather phenomena. Communication between HAPs cannot be effected by cloud, rain, and dust. Hence the negligible atmospheric effects make the line-of-sight communication channel between HAPs has a high level of quality. In addition, the stratosphere is dynamically stable with the mildest winds and there is no or little turbulence. The wind speed at altitudes between 20 km and 30 km is 15–20 m/s [3]. This allows the attitude of the HAP to be in a quasi-stationary state and the moving speed of the HAP to be slow. Hence the mobile HAP ad hoc network does not face frequent drastic changes. However, it is difficult to remotely control the movement of HAPs since they are drifted by the variations of wind fields. This will lead to some challenges at the network layer and MAC layer, respectively.

At the network layer, our proposed mobile HAP ad hoc network differs from traditional flying ad hoc networks. Due to the large-area coverage of the mobile HAP ad hoc network, we deploy several ground control stations (GCSs), each of which manages a part of all HAPs. The data generated by the sensors on each HAP contain its current longitude, latitude, altitude, 3D flight speed, energy state, and environmental observations. Each HAP will transmit these data to its related GCS. Only a fraction of all HAPs can directly access GCSs, and the others can connect to GCSs by the use of multihop transmission. Conventional routing protocols utilized in the flying ad hoc network have high control overheads and memory overheads since each HAP must maintain a routing table. Considering the major traffic pattern in the mobile HAP ad hoc network is the data flow from HAPs to the GCSs, it is efficient to construct multiple aggregation trees in which each GCS is regarded as a sink node. Utilizing the tree-based routing policy can reduce the overhead of maintaining a routing table at each HAP and improve the transmission efficiency from HAPs to GCSs. In addition, the load of each GCS depends on the number of connected HAPs and the amount of data generated by these HAPs. Unbalanced load among GCSs will lead to traffic congestion and influence data transmission efficiency. Therefore, it is necessary to divide the whole coverage area into several subdomains by balancing

Manuscript received 17 May 2023; revised 23 August 2023; accepted 27 December 2023. Date of publication 18 January 2024; date of current version 30 April 2024. This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA17040517, and in part by the National Natural Science Foundation of China under Grants 61871362 and U19B2023. The work of Miao Pan was supported by U.S. National Science Foundation under Grants US CNS-1801925 and CNS-2029569. An earlier version of this paper was presented at the IEEE International Conference on Communications (ICC), Montreal, Canada. [DOI: 10.1109/ICC42927.2021.9500650] Recommended for acceptance by Prof. Haijun Zhang. *(Corresponding author: Chi Zhang.)*

Yuanyuan Wang and Chi Zhang are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: yywang95@mail.ustc.edu.cn; chizhang@ustc.edu.cn).

Taiheng Ge is with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: gth@mail.ustc.edu.cn).

Miao Pan is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204 USA (e-mail: mpan2@uh.edu).

Digital Object Identifier 10.1109/TNSE.2024.3350560

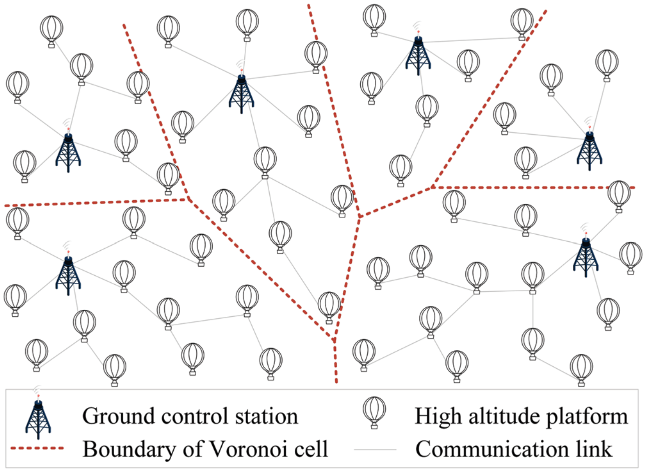


Fig. 1. GCS assignment based on geographic proximity (Voronoi diagram).

the load of each GCS. And each subdomain is assigned with a GCS to manage the HAPs within the subdomain.

At the MAC layer, tremendous data should be transmitted in the mobile HAP ad hoc network. Therefore, how to increase the network capacity becomes a significant factor to improve the performance of the mobile HAP ad hoc network. International telecommunications union (ITU) has already provided HAP communication systems with a dedicated 47/48 GHz band to shoulder such huge burden [5]. In addition, each HAP is equipped with a directional antenna to perform beamforming technology, which can concentrate transmission power on a narrow beam to achieve long-distance communication between HAPs. Furthermore, the transmission beam and reception beam must be accurately aligned to establish a reliable communication link between HAPs. Since the movement of HAPs are not controlled, each HAP must perform beam searching continuously to realize beam alignment. The alignment of transmission beam and reception beam introduces an alignment overhead-throughput tradeoff. A narrower beamwidth provides a larger directivity gain, while it leads to a higher alignment overhead, since more directions have to be searched. Therefore, selecting proper transmission and reception beamwidths is helpful to improve the transmission data rate in the mobile HAP ad hoc network.

In this paper, we address the challenges described at the network layer and MAC layer. To balance the traffic load among GCSs, weighted Voronoi diagram (WVD) is utilized to adjust the boundaries of subdomains. Then we propose two schemes to improve the performance of HAP ad hoc network at the network layer and MAC layer, respectively. In summary, the contributions of this paper are summarized as follows:

- We propose a break and reconstitution (B&R) algorithm to construct an aggregation tree with minimum convergecast delay (MCD) for collecting data in each subdomain. Then a rapid parent selection (RPS) algorithm is presented to solve with the changes of boundaries and communication interruptions between HAPs.
- We optimize the transmission and reception beamwidths to maximize the minimum transmission data rate in the HAP network. According to the time slot scheduling in the aggregation tree, we can obtain communication pairs at each slot. The transmission and reception beamwidths are optimized by a particle swarm optimization (PSO) algorithm.
- We perform a comprehensive set of experiments to analyze the performance of our proposed schemes. We not only explore the quantitative performance with different settings and parameters, but also compare with several baselines.

The rest of this paper is organized as follows. Section II introduces the system model in the mobile HAP ad hoc network. The scheme for handling HAP mobility at the network layer is presented in Section III. Section IV presents the scheme for handling HAP mobility at the MAC layer. In Section V, we evaluate the performance of the proposed schemes with diverse setups of the considered scenario. Section VI provides the related work. Finally, conclusions are drawn in Section VII.

II. System Model

In this section, we introduce the network model and antenna model utilized in the mobile HAP network.

A. Network Model

We consider a target area where there exist a fixed number of geographically distributed GCSs. The HAPs flying above the area and the GCSs constitute a large-scale network which can realize seamless coverage for the area. Each HAP in the network will transmit its collected data through HAP-to-HAP link and HAP-to-Ground link to a selected GCS. To balance the traffic load among all GCSs, a trivial method to solve the GCS assignment problem is illustrated in Fig. 1. Each HAP is assigned to the geographically nearest GCS. Assuming that there are $\mathcal{A}$ Voronoi cells, each Voronoi cell $\mathcal{V}_\alpha$ (for $\alpha \in \{1, 2, \ldots, \mathcal{A}\}$) describes the subarea managed by the GCS s_α. The set of HAPs within $\mathcal{V}_\alpha$ is denoted by $V(s_\alpha)$ and $\mathbb{V} = \cup_\alpha V(s_\alpha)$ represents the set of all HAPs. Whenever an HAP flies across a cell boundary, from $\mathcal{V}_i$ to $\mathcal{V}_j$, the HAP will switch its associated GCS from s_i to s_j. However, there are two important aspects ignored in the proximity criterion. First, at any time, the amount of aggregated data from all HAPs in a Voronoi cell may differ greatly among different cells. Second, a richly connected GCS should aggregate a larger amount of data, which will increase the load of the GCS.

A simple method to solve these two problems together is to jointly consider the distances between HAP and GSCs as well as the traffic load of GCSs. To be specific, let $f(s_\alpha)$ represent the amount of data traffic (in Gbps) collected from all HAPs in the Voronoi cell $\mathcal{V}_\alpha$. For each HAP $\upsilon \in \mathbb{V}$, we define its load distance to GCS s_α as

$$\Delta(\upsilon, s_\alpha) = d_{\upsilon,s_\alpha} \cdot (1 + f(s_\alpha) + \mathbb{1}(\upsilon \notin V(s_\alpha)) \cdot \beta_\upsilon f_\upsilon), \quad (1)$$

where d_{υ,s_α} means the Euclidean distance (in km) between the HAP υ and the GCS s_α. The notation f_υ denotes the amount of data traffic generated by HAP υ and β_υ denotes the aggregation ratio of data from HAP υ. The indicator function $\mathbb{1}(\upsilon \notin V(s_\alpha)) = 1$ if $\upsilon \notin V(s_\alpha)$, and $\mathbb{1}(\upsilon \notin V(s_\alpha)) = 0$ otherwise. Whenever $f(s_\alpha)$ varies, the boundaries of Voronoi cells will be modified according to the load distance. The improved

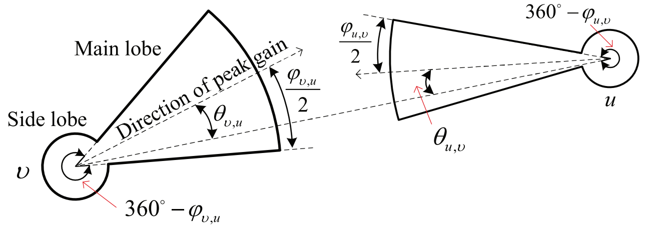

Main lobe
Direction of peak gain
Side lobe
υ
$\theta_{\upsilon,u}$
$\frac{\varphi_{\upsilon,u}}{2}$
$360^\circ - \varphi_{\upsilon,u}$
$\frac{\varphi_{u,\upsilon}}{2}$
$360^\circ - \varphi_{u,\upsilon}$
u
$\theta_{u,\upsilon}$

Fig. 2. Parameters of the ideal sectored antenna model.

diagram is called as weighted Voronoi diagram. Due to the ultra-wide coverage of geostationary (GEO) satellite, we assume that there exists a GEO satellite which can monitor a global view of the Voronoi diagram. The GEO satellite computes for each HAP $\upsilon \in \mathbb{V}$ (currently associated with GCS ς):

- its current load distance $\Delta(\upsilon, \varsigma)$;
- the GCS $\hat{s}$ at minimum load distance, i.e., satisfying $\Delta(\upsilon, \hat{s}) = \min_{s \in S} \Delta(\upsilon, s)$, where S is the set of GCSs, i.e., $S = \{s_\alpha, \forall \alpha \in \{1, \ldots, \mathcal{A}\}\}$.

If $\varsigma = \hat{s}$, $\forall \upsilon$, no switch is required and the boundaries of Voronoi cells remain unchanged. Otherwise, the HAP $\nu \in \mathbb{V}$ with the greatest metric ratio, i.e., $\nu = \arg\max_{\upsilon \in \mathbb{V}} \left\{ \frac{\Delta(\upsilon,\varsigma)}{\Delta(\upsilon,\hat{s})} \right\}$, performs a switch from GCS ς to GCS $\hat{s}$. Therefore, the GEO satellite periodically checks whether any HAP may enjoy a shorter load distance to a connected GCS, according to the current geographic distribution of the network and the current load situation of the GCSs.

For each Voronoi cell $\mathcal{V}_\alpha$ with $\alpha \in \{1, \ldots, \mathcal{A}\}$, let $G_\alpha = (V_\alpha, E_\alpha)$ denote a directed graph. The vertex set $V_\alpha = V(s_\alpha) \cup s_\alpha$ is composed of all nodes in the cell $\mathcal{V}_\alpha$. The feasibility of the communication link between any two nodes in the set V_α requires that the link should fulfill a minimum signal to noise ratio (SNR) γ_0. For a maximum transmission power $P_{\max}$ and free space signal propagation, the maximum communication distance is determined as

$$d_{\max} = \frac{\lambda}{4\pi} \sqrt{\frac{P_{\max}}{k_B k_T \gamma_0}}, \tag{2}$$

where λ denotes the wavelength of the signal. Let k_B and k_T represent the Boltzmann constant and the receiver temperature, respectively. Thus, link (u, υ) from node u to node υ exists if distance $d_{u,\upsilon}$ between the nodes is less than $d_{\max}$. For the convenience of discussion, assuming that all HAPs have the same maximum communication range, the edge set E_α consists of all possible communication links, i.e., $E_\alpha = \{(u, \upsilon) | u, \upsilon \in V_\alpha, d_{u,\upsilon} < d_{\max}\}$.

B. Antenna Model

All nodes are equipped with a single-beam directional antenna to communicate with others on the same carrier frequency. For the sake of tractability, we utilize an ideal sectored antenna model to approximate the directional antenna patterns as illustrated in Fig. 2. This model includes four features relevant to the radiation pattern, i.e., the half-power beamwidth, the boresight direction, and the directivity gains of the mainlobe and sidelobe. Let $g_{u,\upsilon}$ denote the antenna gain of node u when it communicates with node υ. The value of $g_{u,\upsilon}$ can be expressed as [6]

$$g_{u,\upsilon} = \begin{cases} \frac{2\pi - (2\pi - \varphi_{u,\upsilon}) g_\triangleleft}{\varphi_{u,\upsilon}}, & \text{if } |\theta_{u,\upsilon}| \leq \varphi_{u,\upsilon}/2, \\ g_\triangleleft, & \text{otherwise,} \end{cases}$$

where $\theta_{u,\upsilon}$ denotes the alignment error between node u and node υ relative to their boresight directions. Let $\varphi_{u,\upsilon}$ represent the half-power beamwidths of node u. The non-negligible gain in the sidelobe is denoted by $0 \leq g_\triangleleft \ll 1$.

III. Scheme for Handling HAP Mobility at the Network Layer

Once the weighted Voronoi diagram is determined, each HAP merges its own data and the received data from multiple senders, and then transmits the aggregated data to the neighbor which is geographically closest to the GCS. It is important to reduce the time needed for collecting data at GCS in many monitoring systems, especially for time-critical and safety-critical tasks. Hence, we propose a B&R algorithm to build an MCD tree and introduce an RPS algorithm to solve the dynamic changes in the mobile HAP ad hoc network.

A. Minimum Convergecast Delay Computation

For Voronoi cell $\mathcal{V}_\alpha$, the constructed tree based on the graph G_α is represented by $\mathbb{T}_\alpha$. Let $N(\upsilon)$, $C(\upsilon)$, and $P(\upsilon)$ denote the neighbors, children, and parent of υ, respectively. The neighbors $N(\upsilon)$ of node υ are those nodes which have communication links connected to node υ, i.e., $N(\upsilon) = \{u \in V_\alpha | (\upsilon, u) \in E_\alpha\}$. The minimum convergecast delay of node υ is denoted by MCD(υ). Each HAP in the cell $\mathcal{V}_\alpha$ can only select one parent node and the transmission schedule should satisfy the following condition: HAP υ spends at least MCD(υ) to aggregate data from its subtree. In addition, the aggregated data can be completely transmitted during one time slot. Considering the single-beam antenna, the children $C(\upsilon)$ must transmit data to HAP υ in sequence.

Given a randomly initialized tree $\mathbb{T}_\alpha$ in Voronoi cell $\mathcal{V}_\alpha$, the MCDs of leaf nodes are 0 since they have no children. The MCD of each non-leaf node relies on the MCDs of its children. For node υ, let $C(\upsilon) = [u_1, u_2, \ldots, u_n]$ denote the ordered children of υ which satisfy

$$\text{MCD}(u_i) \leq \text{MCD}(u_j),\ 1 \leq i < j \leq n. \tag{3}$$

For a feasible transmission schedule, the transmission time slot of υ is denoted by $t(\upsilon)$. Then there exists following relationship between $t(\upsilon)$ and MCD(υ):

$$t(\upsilon) > \text{MCD}(\upsilon),\ \upsilon \in V(s_\alpha). \tag{4}$$

In addition, due to the limitation of single-beam directional antenna, a parent can receive data from one child at any time. Thus the children's transmission time slots should be different from each other, i.e.,

$$t(u_i) \neq t(u_j), 1 \leq i \neq j \leq n. \tag{5}$$

Algorithm 1: Minimum Convergecast Delay Computation.

```
Input: Node set $V_\alpha$, an initial tree $\mathbb{T}_\alpha$
Output: MCDs of all nodes in the tree $\mathbb{T}_\alpha$
Initialize: $V^{'} \leftarrow V_\alpha$, $\text{MCD}(\upsilon) = 0, \forall \upsilon \in V_\alpha$
while $V^{'} \neq \varnothing$ do
    $Q = \{\upsilon | \upsilon \in V^{'};\ C(\upsilon) \cap V^{'} = \varnothing\}$;
    for $\upsilon \in Q$ do
        $\text{MCD}(\upsilon) = \max\{\text{MCD}(u_i) + |C(\upsilon)| - i + 1 | 1 \leq i \leq |C(\upsilon)|, u_i \in C(\upsilon)\}$;
        $V^{'} = V^{'} \backslash \{\upsilon\}$;
    end
end
```

Based on such constraints, the relationship between the transmission time of a parent and the MCDs of its children is formulated as follows.

Lemma 1: Let $\hat{t}(\upsilon) = \max\{t(u_i) | u_i \in C(\upsilon)\}$, $\forall \upsilon \in V(s_\alpha)$. Then $\hat{t}(\upsilon)$ satisfies

$$\hat{t}(\upsilon) \geq \mathit{MCD}(u_i) + n - i + 1,\ 1 \leq i \leq n. \tag{6}$$

Proof: For any child pair u_i and u_j, according to inequalities (3) and (4), we have

$$t(u_j) > \text{MCD}(u_j) \geq \text{MCD}(u_i),\ 1 \leq i \leq j \leq n.$$

According to the definition of $\hat{t}(\upsilon)$, we can derive

$$\hat{t}(\upsilon) \geq t(u_j) > \text{MCD}(u_i),\ i \leq j \leq n. \tag{7}$$

Since there are $(n - i + 1)$ inequalities corresponding to inequality (7) for child u_i, the inequality (6) holds. □

Theorem 1: The minimum convergecast delay of node $\upsilon \in V_\alpha$ is calculated as

$$\mathit{MCD}(\upsilon) = \max\{\mathit{MCD}(u_i) + n - i + 1 | 1 \leq i \leq n\}.$$

Proof: According to the definition of $\hat{t}(\upsilon)$, it can be utilized to describe the convergecast delay of υ. From inequality (6), the minimum value of $\hat{t}(\upsilon)$ can be regarded as $\text{MCD}(\upsilon)$ which satisfies

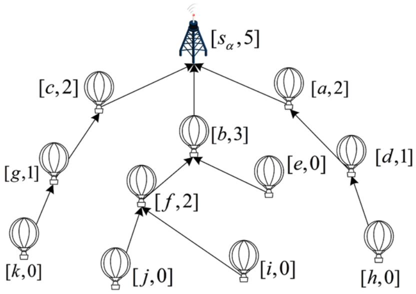
$$\text{MCD}(\upsilon) \geq \text{MCD}(u_i) + n - i + 1,\ 1 \leq i \leq n. \tag{8}$$

Then, we select the largest value of the right side in such inequalities as the $\text{MCD}(\upsilon)$. We consider the following assignment of transmission time slot,

$$t(u_i) = \text{MCD}(\upsilon) - n + i,\ 1 \leq i \leq n. \tag{9}$$

It ensures that each child is assigned a different time slot, which satisfies inequality (5). And we have $t(u_i) \geq \text{MCD}(u_i) + 1$ according to inequality (8) and (9), which satisfies inequality (4). □

For a given $\mathbb{T}_\alpha$, the MCDs of all nodes are calculated according to Algorithm 1. Firstly, the MCDs of all nodes in $\mathbb{T}_\alpha$ are initialized to 0. Next the MCD of each node is computed in a bottom-up manner. The node set V', initiated by set V_α, can be utilized to store the nodes whose MCDs have not been updated. At each iteration, the algorithm selects the leaf nodes in V' and calculates MCDs for them. The node whose MCD has been updated must be removed from V'. Then the algorithm goes to the next iteration until the MCD of the root node is updated, i.e., the set V' is empty. The computational complexity of Algorithm 1 is $\mathcal{O}(|V_\alpha|W)$ where W denotes the maximal number of children of nodes in the tree $\mathbb{T}_\alpha$, i.e., $W = \max_{\upsilon \in \mathbb{T}_\alpha} |C(\upsilon)|$. Fig. 3 illustrates a randomly initialized tree in which all nodes have calculated their MCDs and corresponding data transmission schedule. The arrowed line represents the transmission direction of aggregated data.



Time slot	Data Transmission
0	$k \to g, j \to f,$ $e \to b, h \to d$
1	$g \to c, i \to f, d \to a$
2	$c \to s_\alpha, f \to b$
3	$b \to s_\alpha$
4	$a \to s_\alpha$

Fig. 3. Tree with MCD calculated according to Algorithm 1. The notation $[a, 2]$ represents node a with $\text{MCD}(a) = 2$.

B. Break and Reconstitution Algorithm

According to the definition of MCD, we can find the following implications:

- After $\text{MCD}(\upsilon)$, HAP υ may not transmit data to its parent node immediately until waiting for a certain additional time. We call the time that an HAP must wait before transmitting data to its parent as *waiting time*.
- Many HAPs in the tree may spend a part of the waiting time in receiving data. Thus these HAPs may stay idle during the remaining time slots.
- If some HAP receives data from a new child during the idle waiting time slots, the waiting time of the HAP remains unchanged.
- If some HAP changes parent, the former parent of the HAP potentially reduces its waiting time, which may decrease the MCD of the root node.

To further decrease the MCD of each GCS s_α, we propose a B&R method to optimize the tree structure in each Voronoi cell. Considering a tree with a feasible transmission schedule, a B&R operation performed on node υ and its neighbor κ consists of two actions: node υ breaks its connection with its current parent, and selects κ as its new parent. The former parent may decrease its MCD if υ performs break action. This is because the former parent has a smaller number of children. If the transmission time of κ does not increase after it adopts υ as its new child, the B&R operation brings a chance to reduce the MCD of the root node. To ensure the successful progress of the B&R operation, it is necessary to avoid any circle in the resulting tree. First, we define the following changing-parent rule which does not increase the transmission time of the new parent.

Theorem 2: Considering a tree $\mathbb{T}_\alpha$ with a feasible transmission schedule, for node $\kappa \in \mathbb{T}_\alpha$, node $\upsilon \in N(\kappa) \backslash C(\kappa)$ satisfies $t(\upsilon) < t(\kappa) - |C(\kappa)|$. The parent of υ changes to κ, which can keep the tree topology of the consequent graph and does not increase $t(\kappa)$.

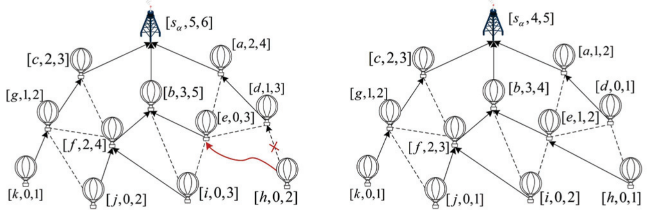
Proof: We first prove that the changing-parent operation does not result in any circle in the resulting tree. The conclusion is

Algorithm 2: Latest Transmission Time Computation.

```
   Input: An initial tree $\mathbb{T}_\alpha$
   Output: LTSs of all nodes in the tree $\mathbb{T}_\alpha$
1  Initialize: $Q = \{s_\alpha\}$, $\text{LTS}(s_\alpha) = \text{MCD}(s_\alpha) + 1$, $B = \varnothing$
2  while $Q \neq \varnothing$ do
3      $\upsilon \leftarrow$ pop first element in $Q$;
4      $B = C(\upsilon)$;
5      while $B \neq \varnothing$ do
6          $w = \arg\min_{\omega \in B} \text{MCD}(\omega)$;
7          $\text{LTS}(w) = \text{LTS}(\upsilon) - |B|$;
8          $B = B \backslash \{w\}$;
9          $Q = Q \cup \{w\}$;
10     end
11 end
```

Algorithm 3: Break and Reconstitution.

```
   Input: An initial tree $\mathbb{T}_\alpha$
   Output: A B&R tree $\mathbb{T}_\alpha^*$
1  Initialize: $Q = \{s_\alpha\}$, $idx(\upsilon) = 0, \forall \upsilon \in V_\alpha$
2  while $Q \neq \varnothing$ do
3      $\kappa \leftarrow$ pop first element in $Q$;
4      $B = \{\upsilon | \upsilon \in N(\kappa) \backslash C(\kappa);\ \text{LTS}(\kappa) - |C(\kappa)| > \text{LTS}(\upsilon)\}$;
5      $\omega = \arg\max_{\upsilon \in B} \{\text{MCD}(\upsilon) | idx(\upsilon) = 0\}$;
6      $C(\kappa) = C(\kappa) \cup \{\omega\}$, $P(\omega) = \kappa$, $idx(\omega) = 1$;
7      Update MCD and LTS of nodes in the paths
         including the old and new parents;
8      $J = C(\kappa)$;
9      while $J \neq \varnothing$ do
10         $\omega = \arg\max_{\omega \in J} \text{LTS}(\omega)$;
11         $Q = Q \cup \{\omega\}$;
12         $J = J \backslash \omega$;
13     end
14 end
```



(a) Breaking the link between h and d, and selecting e as a new parent of h.

(b) Updating the MCDs and LTSs of the remaining nodes.

Fig. 4. Example of a specific B&R operation. The notation $[c, 2, 3]$ represents node c with $\text{MCD}(c) = 2$ and $\text{LTS}(c) = 3$.

proved by contradiction. A tree topology can be described by a directed graph in which there exists only one outgoing link from each node to its parent. After node υ breaks connection with its current parent $P(\upsilon)$ and selects node κ as its new parent, the outgoing link of υ connects to κ. Supposing that there is a cycle if υ changes parent, the circle can be represented as an ordered node list $\{\upsilon, \kappa, \omega_1, \omega_2, \ldots, \omega_h, \upsilon\}$. Because the transmission time of a parent must be later than that of each child, we have $t(\upsilon) < t(\kappa) < t(\omega_1) < \cdots < t(\omega_h) < t(\upsilon)$. The result contradicts with the definition of the transmission time. Therefore, there exists no cycle in the resulting tree after changing parent.

Next, we demonstrate that there exists an unoccupied time slot in the range $[t(\upsilon), t(\kappa))$ which has not been assigned to any child of κ. Node υ can transmit data to the new parent κ at such a time slot, which will not increase $t(\kappa)$. The children of κ must transmit data to κ at different time slots. Since $t(\upsilon) < t(\kappa) - |C(\kappa)|$, there are $t(\kappa) - t(\upsilon) \geq |C(\kappa)| + 1$ time slots in the range $[t(\upsilon), t(\kappa))$. Thus we can find at least one slot which is represented as the transmission time of υ. The theorem has been proved. □

Theorem 2 suggests that node κ with higher $t(\kappa)$ and smaller $|C(\kappa)|$ will be selected as a new parent of node υ with a bigger possibility. To maximize the possibility, the latest transmission time slot (LTS) is defined as follows. We let $\text{LTS}(\kappa)$ represent the LTS of node κ, which can be computed by Algorithm 2. Initially, the LTS of the root node s_α is $\text{LTS}(s_\alpha) = \text{MCD}(s_\alpha) + 1$ because s_α does not transmit data. Next, the value of LTS is calculated in a top-down manner. At each iteration, the child of υ with the i-th largest MCD takes the value $\text{LTS}(\upsilon) - i$ as its LTS. In the while-loop of Algorithm 2, there are $|V_\alpha|$ loops because all nodes in the tree are added into the queue Q. Given a node υ with its LTS, when calculating LTSs for the children of node υ, the algorithm first sorts the children in the ascending order of MCD and then calculates LTS for each child. The time complexity of such procedure (line 5 to line 10 in Algorithm 2) is $\mathcal{O}(W \log W)$. Hence, the computational complexity of Algorithm 2 is $\mathcal{O}(|V_\alpha| W \log W)$.

Theorem 3 can be derived from Theorem 2 by replacing the transmission time slot with LTS. Changing-parent operation based on Theorem 3 can obtain a valid tree topology.

Theorem 3: Considering a tree $\mathbb{T}_\alpha$ and a feasible transmission schedule corresponding to the LTS, for node $\kappa \in \mathbb{T}_\alpha$, node $\upsilon \in N(\kappa) \backslash C(\kappa)$ satisfies $LTS(\upsilon) < LTS(\kappa) - |C(\kappa)|$. The parent of υ changes to κ, which can keep the tree topology of the consequent graph and does not increase $LTS(\kappa)$.

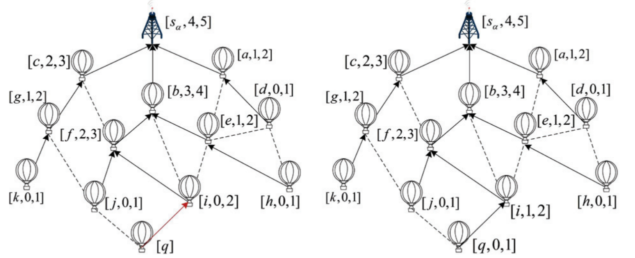
Proof: Since Theorem 2 is proved for any feasible transmission schedule, it is obviously valid for the feasible transmission schedule corresponding to the LTS. □

Algorithm 3 performs B&R operations in a top-down manner. The binary variable $idx(\upsilon) = 0$ means that υ has never changed its parent (line 1). The larger LTS the node has, the higher priority it has to adopt a new child. As for a selected node $\kappa \in Q$, we regard nodes satisfying Theorem 3 as the candidates B (line 4). Parent κ will adopt the node w with the largest MCD in B and the value of $idx(w)$ is updated (lines 5 to 6). Next, the children of κ will be added to Q in the decreasing order of LTS (lines 9 to 13). The algorithm does not stop until all nodes in the tree are checked once. In the parent-changing operations of Algorithm 3, each node in the tree has one chance to adopt a new child, so the algorithm should iterate $|V_\alpha|$ times. When adopting a new child, each node checks the neighbors and then selects the node that has the highest MCD and satisfies Theorem 3. The time complexity of such procedure is $\mathcal{O}(W)$. Next, the algorithm updates the MCDs of the nodes in the path from the former parent and the adopter to the sink node. Such operation takes at most $\mathcal{O}(|V_\alpha| W)$ if we need to recalculate MCDs for all nodes in the tree $\mathcal{T}_\alpha$. Hence the computational complexity of Algorithm 3 is $\mathcal{O}(|V_\alpha|^2 W)$.

Fig. 4 illustrates a specific B&R operation. Any two nodes connected by a dotted line are within the communication range

Algorithm 4: Rapid Parent Selection.

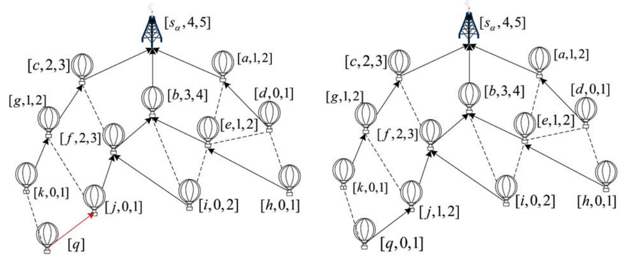
```
   Input: The tree $\mathbb{T}_\alpha$, the neighbor of HAP $q$, i.e., $N(q)$,
          and the descendant nodes of HAP $q$, i.e., $D(q)$
   Output: Parent node of HAP $q$
1  Initialize: $Q = \varnothing$, $J = N(q) \backslash D(q)$. Build $|J|$ paths, each
     of which starts with each node in the set $J$.
2  while $Q == \varnothing$ do
3      $ind1 = \max\{\text{LTS}(j), \forall j \in J\}$;
4      $B = \{j \mid \text{LTS}(j) = ind1, \forall j \in J\}$;
5      if $|B| == 1$ then
6          $Q \leftarrow$ the first node in the path which ends with
             $B$;
7      else
8          Add parent $P(b)$ of each node $b \in B$ into the
             path which ends with $b$;
9          $J = \{P(b), \forall b \in B\}$;
10     end
11 end
12 Output the parent node $Q$;
```



(a) Selecting HAP i with larger LTS as the parent of HAP q.

(b) Updating MCD of the selected parent i.

Fig. 5. Parent selection for HAP q and the MCD updates of nodes in the path from q to s_α when the LTSs of neighbors of HAP q are different.



(a) Selecting HAP j whose parent has larger LTS as the parent of HAP q.

(b) Updating MCD of the selected parent and LTSs of related nodes.

Fig. 6. Parent selection for HAP q and the MCD updates of nodes in the path from q to s_α when the LTSs of neighbors of HAP q are same.

of each other. As shown in Fig. 4(a), since node e has no child and $\text{LTS}(e) = 3$, it can adopt an additional child. Among the neighbors of e, node h satisfies Theroem 3 because $\text{LTS}(h) < \text{LTS}(e) - 0$. Then node h breaks the link connected to d, and selects node e as its new parent. Finally, we update the MCDs and LTSs of remaining nodes in the resulting tree as in Fig. 4(b). The MCD of the root node in the current tree is $\text{MCD}(s_\alpha) = 4$, which is smaller than 5 in the previous tree before changing parent.

C. Rapid Parent Selection Algorithm

Due to the movement of HAPs and the adjustment of boundaries of Voronoi cells, it is possible that some transmission links in the tree topology $\mathbb{T}_\alpha$ may be interrupted and some HAPs may be reallocated to the other Voronoi cell. To keep the tree topology and MCD of the root node unchanged as much as possible, we propose a rapid parent selection algorithm to choose a node in the tree to collect data transmitted from the HAP which does not have parent node currently.

Assuming that there exists any kind of network change in Voronoi cell $\mathcal{V}_\alpha$, the GEO satellite executes Algorithm 4 to select a new parent for HAP q which does not have any communication link with its current parent. Let $D(q)$ denote the descendant nodes of HAP q, i.e., the nodes in the subtree with HAP q as the root node. To avoid any circle when selecting a parent node for HAP q, the neighbors which are also the descendant nodes of HAP q are out of consideration. Hence the algorithm initializes $|N(q) \backslash D(q)|$ searching paths, each of which is assigned with a different beginning node in the set $N(q) \backslash D(q)$ (line 1). Algorithm 4 selects a new parent for HAP q in a bottom-up manner. To be specific, we find the alternative node set B which consists of the nodes with the largest LTS among the set J (line 3 to 4). If there exists only one node in the set B, the first node in the path ending with B will be selected as the optimal parent of the HAP q (line 6). Otherwise, the parent of each node in the set B will be added into the corresponding path. Meanwhile, the algorithm updates the set J and turns to the next iteration. The computational complexity of Algorithm 4 is $\mathcal{O}(|N(q) \backslash D(q)| \cdot M)$, where M is the maximum number of nodes on the paths from nodes in the set $N(q) \backslash D(q)$ to the root node s_α in the tree $\mathbb{T}_\alpha$.

Figs. 5 and 6 illustrate the specific RPS operation. We assume that HAP q is reassigned to the Voronoi cell $\mathcal{V}_\alpha$. For example in Fig. 5(a), the neighbors of q (i.e., HAP i and HAP j) have different LTSs. Since the LTS of HAP i is larger than that of HAP j, we select HAP i as the parent of HAP q and update the MCDs and LTSs of nodes in the tree $\mathbb{T}_\alpha$. As we can see in Fig. 5(b), only the MCD of HAP i should be updated. If more than one neighbor of HAP q has the same maximum LTS as illustrated in Fig. 6(a), i.e., $\text{LTS}(j) = \text{LTS}(k)$, we will turn to their parents. Since the LTS of HAP f is larger than that of HAP g, we select HAP j as the parent of HAP q and update the MCDs and LTSs of nodes in the tree $\mathbb{T}_\alpha$. As we can see in Fig. 6(b), only the MCD and LTS of HAP j should be updated.

IV. Scheme for Handling HAP Mobility at the MAC Layer

To realize long-distance communication between HAPs, we utilize directional antennas to establish wireless links. Since the movement of HAPs are not controlled, continuous beam searching for achieving beam alignment will introduce a significant alignment overhead. Not only the alignment overhead but also throughput is related to the beamwidth. Improving the transmission data rate between HAPs is beneficial for building an information backbone HAP network which can support tremendous data. Hence, we utilize a PSO-based scheme to optimize the transmission beamwidth and reception beamwidth.

A. Alignment Delay and Transmission Rate

Based on the tree topology defined in Section III, we can obtain the MCD and LTS of each HAP in set $V(s_\alpha)$. The set of transmission time slot is denoted by $\mathcal{T}_t = \{0, 1, \ldots, \mathrm{MCD}(s_\alpha)\}$. For the sake of discussion, the transmission time slot of node $\upsilon \in V(s_\alpha)$ is equal to $\mathrm{LTS}(\upsilon)$. Since relative movement between HAPs leads to a varying tree topology, misalignment may occur at the transmitter and the receiver of any communication pair. It is necessary for each HAP to perform beam alignment before transmitting or receiving data. Similar to [7], a transmission time slot consists of an alignment phase and a data transmission phase.

In this paper, we present a two-staged beam alignment technology, simplified from the beam codebook-bassed method in [8]. To be specific, the two stages include a series of pilot transmissions and a trial-and-error approach, after which the optimal steering for beams at transmitter and receiver are executed. Without loss of generality, we assume that for the transmitter and receiver of any link in the tree topology, a coarse sector-level alignment has detected the best sectors for them. Then, communication pairs perform a fine-grained beam-level alignment. In such two-staged beam alignment method, the well-known tradeoff between alignment delay and throughput [7] is revealed: the narrower beamwidth a node selects, the longer training overheads it will suffer, which reduces effective transmission time. For the directional link (u, υ), the alignment time penalty is expressed as

$$\tau_{u,\upsilon} \triangleq \tau_{u,\upsilon}\left(\varphi_{u,\upsilon}, \varphi_{\upsilon,u}\right) = \frac{\psi_{u,\upsilon} \cdot \psi_{\upsilon,u}}{\varphi_{u,\upsilon} \cdot \varphi_{\upsilon,u}} T_p, \tag{10}$$

where $\psi_{u,\upsilon}$ and $\psi_{\upsilon,u}$ represent the sector-level beamwidths of HAP u and HAP υ, respectively. The time duration of a pilot transmission is denoted by T_p. Considering the fact that $\tau_{u,\upsilon}$ cannot exceed the duration of a transmission slot T_t, the values of $\varphi_{u,\upsilon}$ and $\varphi_{\upsilon,u}$ should satisfy

$$\varphi_{u,\upsilon} \cdot \varphi_{\upsilon,u} \geq \frac{T_p}{T_t} \cdot \psi_{u,\upsilon} \cdot \psi_{\upsilon,u}. \tag{11}$$

In addition, since beam alignment performs within the sector-level beamwidth, the half-power beamwidths of u and υ should satisfy $\varphi_{u,\upsilon} \leq \psi_{u,\upsilon}$ and $\varphi_{\upsilon,u} \leq \psi_{\upsilon,u}$, respectively.

Once the beams of transmitter and receiver have been aligned, the achievable data rate $r_{u,\upsilon}$ between u and υ depends on the remaining effective transmission time and the measured SINR at the receiver υ. The date rate $r_{u,\upsilon}$ is given by

$$r_{u,\upsilon} = \left(1 - \frac{\tau_{u,\upsilon}}{T_t}\right) \cdot B \cdot \log_2\left(1 + \mathrm{SINR}_\upsilon\right), \tag{12}$$

where B denotes the bandwidth allocated to the HAP. Due to the directivity of beam, the interferences from other transmitting HAPs are out of consideration. Hence SINR_υ is degraded into the signal to noise ratio SNR_υ which is expressed as

$$\mathrm{SNR}_\upsilon = \frac{p_u g_{u,\upsilon} g_{\upsilon,u}}{k_B k_T B} \left(\frac{c}{4\pi f_c d_{u,\upsilon}}\right)^2, \tag{13}$$

where p_u is the transmission power of u. Let c and f_c denote the light speed and carrier frequency, respectively. Since the HAPs which are floated in the same wind layer move with the same direction and speed, the channel quality between adjacent HAPs is relatively steady. It is straightforward to find that the data rate $r_{u,\upsilon}$ increases when no beam alignment is executed, as per (12) if $\tau_{u,\upsilon} = 0$.

B. PSO-Based Beamwidth Optimization

Equation (13) indicates that selecting narrower transmission and reception beamwidths may lead to higher directivity gains and higher antenna gains. However, these gains are enhanced at the expense of more alignment time, which reduces available time for data transmission. This exposes an alignment delay versus effective data rate tradeoff. We let $E(\mathbb{T}_\alpha)$ denote the set of wireless links in the tree $\mathbb{T}_\alpha$. In this paper, we formulate a problem of beamwidth selection to maximize the minimum transmission data rate as follows

$$\max_{\boldsymbol{\varphi}} \min_{(u,\upsilon)\in E} r_{u,\upsilon} \tag{14a}$$

$$\text{s.t.} \quad \varphi_{u,\upsilon} \cdot \varphi_{\upsilon,u} \geq \frac{T_p}{T_t} \cdot \psi_{u,\upsilon} \cdot \psi_{\upsilon,u}, \quad \forall (u,\upsilon) \in E, \tag{14b}$$

$$\varphi_{u,\upsilon} \leq \psi_{u,\upsilon}, \qquad \forall (u,\upsilon) \in E, \tag{14c}$$

$$\varphi_{\upsilon,u} \leq \psi_{\upsilon,u}, \qquad \forall (u,\upsilon) \in E, \tag{14d}$$

where $\boldsymbol{\varphi} = (\varphi_{u,\upsilon}, \varphi_{\upsilon,u}) \in \mathbb{R}^{2\times|E|}$ with $E = \cup_\alpha E(\mathbb{T}_\alpha)$. Here, $|\cot|$ denotes the cardinality of a set. Inequalities (14b) through (14d) represent the limitations imposed on the beamwidths. The optimization problem is intractable to solve analytically.

Recently, particle swarm optimization (PSO [9]) based technique has been successfully used in the design of antenna components [10], [11]. PSO is similar with genetic algorithm and evolutionary algorithms in some ways, but needs less computational workload and fewer lines of code, having the advantages that the basic method is easy to understand and implement. The core of PSO is computationally light agents or particles which can interact with each other according to collectively intelligent behavioural rules. Such rules simulate social activity patterns operated in bird flocks when they respond to a new environment. These species utilize the adaptive collective behavior to discover optimum area within search range based on the measure of global fitness.

PSO-based beamwidth optimization scheme iteratively updates a pool consisting of H candidate solutions $\{\boldsymbol{W}_h\}_{h=1}^{H}$. Each candidate can be expressed as $\boldsymbol{W}_h = [\boldsymbol{\varphi}_h]$ with $h \in \{1, \ldots, H\}$ and $A \triangleq |\boldsymbol{W}_h| = 2 \times |E|$. The PSO optimization procedure starts by defining an initial set of particles which denote the beamwidths of the transmitter and receiver for all directional links and setting a velocity vector $\boldsymbol{\upsilon}_h = \{\upsilon_h^1, \ldots, \upsilon_h^A\}$.

All beamwidths are initially assigned a fixed value 3°, and each element of the corresponding velocity vector $\boldsymbol{v}_h$ is drawn uniformly at random in the range [0°, 90°]. According to the objection in (14a), the fitness function $F(\boldsymbol{W}_h)$ of the h-th candidate is given by

$$F(\boldsymbol{W}_h) = \min_{(u,v)\in E} r_{u,v}. \tag{15}$$

Then we select the global optimal fitness value $F(\boldsymbol{W}^{\triangleright})$ with the global optimal particle location $\boldsymbol{W}^{\triangleright}$. The optimal individual location $\boldsymbol{W}_h^*$ for particle $\boldsymbol{W}_h$ is recomputed and kept during the algorithm procedure. The algorithm continues by iteratively meliorating the velocity vector according to its current velocity, the optimal individual location, and the global optimal particle location as follows

$$\boldsymbol{v}_h \leftarrow \eta \cdot \boldsymbol{v}_h + \varepsilon \cdot \boldsymbol{r}_\varepsilon \left(\boldsymbol{W}_h^* - \boldsymbol{W}_h\right) + \xi \cdot \boldsymbol{r}_\xi \left(\boldsymbol{W}^{\triangleright} - \boldsymbol{W}_h\right).$$

The second term in the righthand expression is the difference between the current location and the optimal location of particle h. The third summand denotes the difference between the current location of h and the global optimal location in the whole population of all particles. We will update the value of each candidate solution $\boldsymbol{W}_h$ as $\boldsymbol{W}_h \leftarrow \boldsymbol{W}_h + \boldsymbol{v}_h$ when the velocity vector changes. Next, the optimal individual location $\boldsymbol{W}_h^*$ and the global optimal location $\boldsymbol{W}_{\triangleright}$ are updated if necessary. Parameters η (inertia), ε and ξ (learning factors) are utilized to control the search behaviour of particle, whereas $\boldsymbol{r}_\varepsilon$ and $\boldsymbol{r}_\xi$ are two $1 \times A$ vectors with elements distributed in the range $[0,\ 1]$ uniformly at random. This process does not stop until the maximum number of iterations $K_{\max}$ is reached. Finally, the global optimal location $\boldsymbol{W}_{\triangleright}$ is our selected solution for problem (14). The computational complexity of the PSO-based beamwidth optimization scheme is $\mathcal{O}(|E| \cdot H \cdot K_{\max})$.

V. Simulation Settings and Results

In this section, we conduct a large number of comprehensive computer experiments to evaluate the network performance of the proposed schemes.

A. Simulation Settings

To simulate a real mobile HAP ad hoc network, we focus on the stratosphere above the area (45°N to 65°N latitude and 5°W to 60°W longitude). We denote the side length (in km) of the simulation area as l. According to the guidelines proposed by ITU, we assume that all HAPs are uniformly distributed above the simulation area initially and fly at the same altitude of 20 km with the time-varing wind. Our simulations utilize the wind data from March 20, 2016 at 14:00 UTC, which is obtained from the United States' National Oceanic and Atmospheric Adminstration [12]. Each HAP collects its information, receives data from other HAPs, and then transmits the aggregated data to a selected GCS. The carrier frequency and bandwidth for inter-HAP communication are set to 47 GHz and 600 MHz. Assuming that there are n HAPs in the network, to describe the topological characteristic of the multi-HAP network, we define two parameters as follows

TABLE I
Simulation Parameters

Parameters	Value
Network density (D)	[15, 20, 25, 30, 35, 40]
Network side length (L)	[1, 2, 3, 4, 5, 6, 7, 8]
Bandwidth (B)	600 MHz
Carrier frequency	47 GHz
Transmission power of HAP (p_u)	10 W
Noise spectral density (N_0)	−174 dBm/Hz
Path loss exponent (α)	2
Sector-level beamwidth ($\psi_{u,v}, \psi_{v,u}$)	45°
Peak transmit/slot time (T_p/T_t)	0.01

$$D = n\pi r^2/l^2, \tag{16}$$

$$L = l/r, \tag{17}$$

where r denotes the maximum communication radius (in km) of an HAP. The network density D can be represented as the average number of HAPs within a circle area of radius r. The parameter L describes the ratio of the side length to the maximum communication radius. For the convenience of discussion, the value of r is normalized to 1. Hence the value of L is equal to that of the side length l. We let L represent the side length of the square area below. The value of each parameter employed in the simulations is illustrated in Table I.

The range of D is from 15 to 40 by the step of 5 and the range of L is from 1 to 8 by the step of 1. We set the minimum value of D to be 15, which can guarantee the network connectivity. The minimum value of L is set to 1, which means that the side length equals the communication radius. With the combination of different network density D and different network side length L, we have various simulation scenarios. For each case, we operate 200 runs of the experiment and exhibit the average result to analyze.

We perform the B&R algorithm on the two schemes of constructing initial aggregation tree, namely:

- Shortest path tree (SPT), in which each node establishes a shortest path from it to the root node.
- Minimum lower bound spanning tree (MLST) [13], in which a transmission link with the minimum cost defined by the receiver's hop distance to the root node and the number of children will be added to the MLST.

Let SPT-B&R and MLST-B&R represent the resulting trees of SPT and MLST after performing B&R algorithm.

Once an HAP is reassigned to a new Voronoi cell or interrupts the communication link with its current parent, the RPS algorithm is performed to find a new parent for the HAP in the cell. We compare our proposed RPS algorithm with the following two mechanisms.

- Rerunning SPT-B&R. As for the Voronoi cell adopting a new HAP or in which there exist some interrupted communication links, we perform SPT-B&R again to obtain a new tree structure which may be completely different from the previous one.

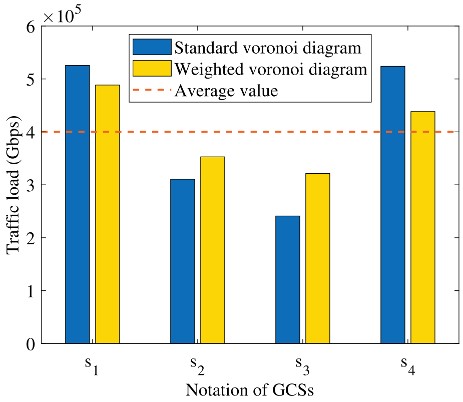


Fig. 7. Traffic load of the four GCSs.

- Selecting the closest neighbor (CN) as the parent. The HAP selects the closest neighbor in the Voronoi cell as its parent, which is the simplest method.

B. Results and Discussion

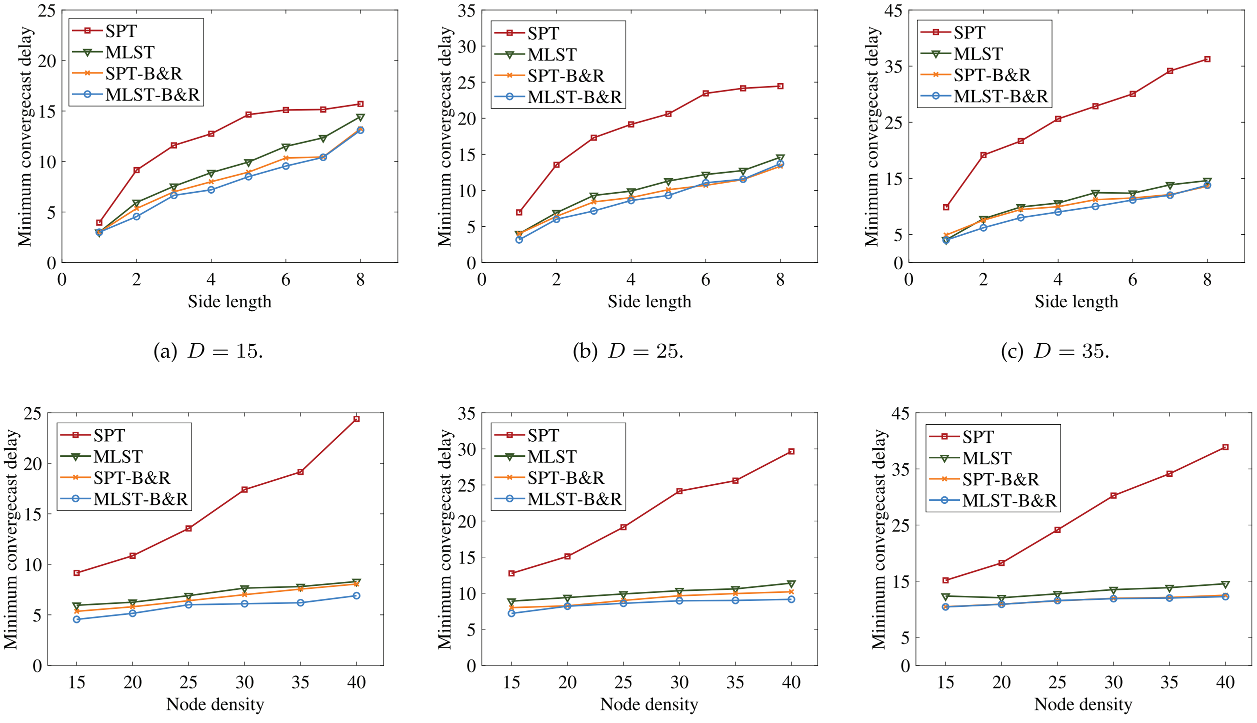
Fig. 7 illustrates the traffic load of each GCSs in the weighted Voronoi diagram and standard Voronoi diagram. We deploy 4 GCSs randomly located at the edge of the simulation area with the side length of 8. Assuming that the network density is set to be 15, the traffic flow of each HAP is randomly selected in the range $[0, 10]$ Gbps. In the Fig. 7, the dotted line represents the average value of traffic load ideally. As we can see, the traffic load of each GCS in the weighted Voronoi diagram is closer to the average value than that of each GCS in the standard Voronoi diagram. This is because some HAPs near the boundaries of the Voronoi cell $\mathcal{V}_1$ and $\mathcal{V}_4$ are reassigned to the Voronoi cell $\mathcal{V}_2$ and $\mathcal{V}_3$.

Fig. 8(a), (b), (c) illustrate the minimum convergecast delay as a function of the side length, with D fixed at 15, 25, and 35. With the increasing side length, the depth of the constructed tree must grow up. Hence the minimum convergecast delay of SPT, MLST, SPT-B&R, and MLST-B&R increase accordingly. When the side length is small, i.e., $L = 1$, almost all HAPs are 1-hop away from their selected GCSs. Hence the convergecast delay is equal to the number of neighbors of the root node. As for the SPT, the larger density is, the larger comparative advantage SPT-B&R can have. For example in Fig. 8(a) with a small density of 15, the convergecast delay of SPT-B&R is 16%–41% smaller than that of SPT. In Fig. 8(c) with a large density of 35, the convergecast delay of SPT-B&R is 50%–62% smaller than that of SPT. Compared with the SPT, the convergecast delay of the MLST is smaller because MLST permits transmission among the nodes which have the same hop distance away from the root node. The characteristic of the MLST is beneficial to alleviate the transmission bottleneck at the root node. Note that performing B&R on the SPT and MLST leads to a very close tree quality, which suggests that any initial tree does not affect the performance of the B&R algorithm and the quality of the final constructed tree.

Fig. 8(d), (e), (f) show the minimum convergecast delay as a function of the network density, with L fixed at 2, 4, and 7. With the increasing density, the number of nodes within an unit area becomes larger. Hence the minimum convergecast delay of SPT, MLST, SPT-B&R, and MLST-B&R increase accordingly. As for the MLST, the smaller side length is, the larger comparative advantage MLST-B&R can have. For example in Fig. 8(f) when $L = 2$, the convergecast delay of MLST-B&R is 13%-23% shorter than that of MLST. As illustrated in Fig. 8(f) when $L = 7$, the convergecast delay of MLST-B&R is 9%-15% shorter than that of MLST. As we can see, when the side length becomes large, the difference between the convergecast delay of SPT-B&R and that of MLST-B&R turns to small. It suggests that the B&R algorithm performs more efficiently on the SPT in a larger network.

Fig. 9(a) plots the computation time of executing the SPT, MLST, and B&R algorithms as a function of the side length with D being 25. With the increase of the side length, the computation time of executing each algorithm increases. We can observe that the computation time of establishing an SPT is always smaller than that of establishing an MLST. Since the MLST has a similar performance of the MCD compared with MLST-B&R, it always takes smaller time to perform B&R algorithm on the MLST than the SPT. When the side length is 8, the total computation time spent in performing the SPT algorithm and performing the B&R algorithm on the SPT is larger than that in performing the MLST algorithm and performing the B&R algorithm on the MLST. Fig. 9(b) plots the computation time of executing the SPT, MLST, and B&R algorithms as a function of the node density with L being 4. The computation time of executing each algorithm increases with the increase of node density. When the node density is 40, the total computation time spent in performing the SPT algorithm and performing the B&R algorithm on the SPT is larger than that in performing the MLST algorithm and performing the B&R algorithm on the MLST. Hence, it is better to choose the MLST algorithm for establishing the initial tree for a smaller total computation time when the network side length and the node density are large.

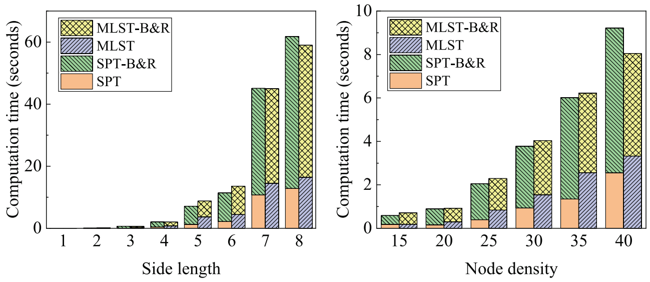
Fig. 10 illustrates the performance of our proposed RPS algorithm. The network side length is set to be 8. Fig. 10(a) shows the number of changing MCD as a function of network density. It can be observed that rerunning SPT-B&R changes more number of MCD than the other two mechanisms. This is because rerunning SPT-B&R may change the previous tree structure in the Voronoi cell $\mathcal{V}_2$ significantly, further altering the MCD of each node. When the network density is in the range $[15, 30]$, the performance of selecting the closest neighbor as parent is close to that of RPS algorithm. As the density approaches 40, the advantage of the RPS algorithm is clear. Fig. 10(b) illustrates the MCD of GCS s_2 as a function of network density. With the increasing density, the MCD of s_2 for each algorithm presents a non-decreasing trend. As we can see, the MCD of rerunning SPT-B&R is always no more than that of the other two mechanisms. This is because rerunning SPT-B&R maintains the optimal minimum convergecast delay of s_2 at the cost of breaking up the previous tree structure. In most cases, the MCD of RPS can be consistent with that of rerunning SPT-B&R.



(a) $D = 15$. (b) $D = 25$. (c) $D = 35$.

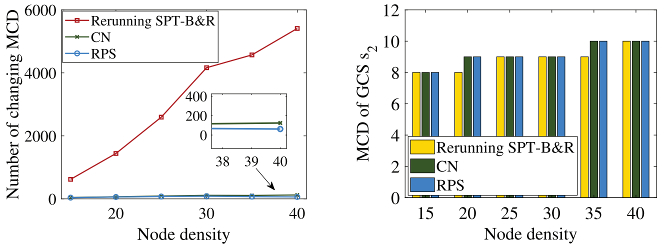
(d) $L = 2$. (e) $L = 4$. (f) $L = 7$.

Fig. 8. Performance of B&R algorithm operated on the SPT and MLST. Minimum convergecast delay versus side length with $D \in \{15,\ 25,\ 35\}$ in (a), (b), and (c). Minimum convergecast delay versus node density with $L \in \{2,\ 4,\ 7\}$ in (d), (e), and (f).



(a) Node density $D = 25$. (b) Side length $L = 4$.

Fig. 9. Computation time of SPT, MLST, and B&R algorithms.



(a) The number of changing MCD. (b) The MCD of GCS s_2.

Fig. 10. Performance comparison after performing the three parent selection mechanisms.

Fig. 11 illustrates the average MCD gap of all GCSs between the RPS algorithm and rerunning SPT-B&R. The network side length is set to be 8 and the node density is set to be 25. During the 2 hours time window, each HAP will move in the circular range with radius 1. The average MCD gap is always less than a threshold 0.5, which indicates that the performance of the RPS algorithm is similar to the performance of rerunning SPT-B&R.

The convergence of the PSO-based beamwidth optimization is illustrated in Fig. 12(a). We select the value of side length and node density from the set $\{4, 8\}$ and $\{15,\ 25,\ 35\}$, respectively. The evolution of the transmission data rate validates that Swarm Intelligence is capable of efficiently optimizing the network performance. As the side length and node density increase, the decreasing convergence speed of the PSO-based beamwidth optimization is acceptable.

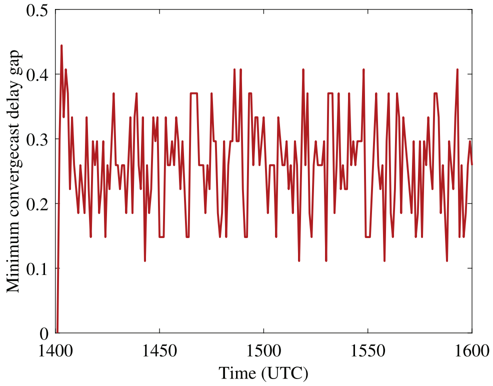
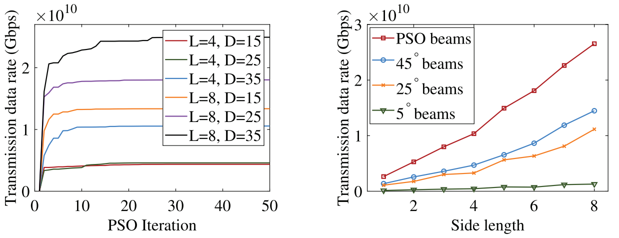


Fig. 11. Average MCD gap between the RPS algorithm and rerunning SPT-B&R.



(a) The convergence of the PSO-based beamwidth optimization.

(b) The performance comparison after performing the four beams.

Fig. 12. Performance of the PSO-based beamwidth optimization.

Fig. 12(b) shows the transmission data rate as a function of the network side length. The values of fixed beamwidth are set to 5°, 25°, and 45°. It can be concluded that PSO-based beams render better network performance than any other fixed beamwidths. This outperforming behavior is significant with the increasing side length. As for the narrowest beamwidth, it performs worse than the others and the transmission data rate improves slightly as the side length increases. This is because the narrowest beamwidth not only reduces the available time for data transmission, but also brings huge interference to some receiving HAPs.

VI. Related Work

Some recent works have studied a number of ideas related to the multi-hop routing in wireless flying ad-hoc networks. Currently, there are four classes of routing protocols, namely static routing protocol, proactive routing protocol, reactive routing protocol, and geographic routing protocols. Static routing protocols, such as load carry and deliver routing (LCAD) [14], multilevel hierarchical routing (MLH) [15], and data centric routing (DCR) [16], have been studied for multiple UAV-assisted networks. However, static routing table is unsuitable for a scenario with dynamic changes. Proactive routing protocols, such as destination-sequenced distance vector routing (DSDV) [17] and optimized link state routing (OLSR) [18], suffer masses of additional message overhead which are exchanged between nodes for updating routing table. Considering the limited bandwidth, the proactive routing protocols are not suitable for dynamic multi-HAP network. Reactive routing protocols including dynamic source routing (DSR) [19] and ad hoc on-demand distance vector (AODV) [20], have been proved to be incompetent because of scalability issues. Geographic routing protocols, such as greedy perimeter stateless routing (GPSR) [21] and geographic position mobility oriented routing (GPMOR) [22], make routing decisions depending on the position mobility information.

The HAP ad hoc network has many differences compared with common flying ad hoc networks, such as sparsity, wide coverage, and slow variation. The Loon project operated the HAB mesh network above three continents and developed a temporospatial software defined network (TS-SDN) architecture which aimed to orchestrate the current and future state of the HAB mesh network by forecasting the performance of the physical layer [23]. Over the past years, the TS-SDN system has been able to control the physical wireless topology and routing according to the high-fidelity modeling of the relationships, constraints, and accessibility of wireless links in the network [24]. By predicting the near-term trajectory of HABs, the SDN routing application in the system can anticipate topology changes and route breakages before they occur. The work in [25] proposed a novel HAB enabled maritime network (HABMN) architecture. The architecture utilized a space-time graph to construct a holistic view of the time-varying network due to the movement of HAB. Then a flow reconfiguration mechanism was presented to perform the route update in the HABMN. However, to the best of our knowledge, none of these existing routing algorithms are designed with the specific task of multi-HAP network in mind and therefore they cannot be applied in our scenario.

The existing literatures have studied the directional communication of HAP. To realize spatial spectrum reuse efficiently, each HAP can form multiple cells by use of advanced beamforming technologies. In [26], Sudheesh et al. utilized beamforming technology to realize interference alignment with the goal of maximizing sum rate of HAP communications. An intelligent beamforming strategy was proposed to enlarge the coverage and capacity of HAP communications in [27]. Xu et al. [28] presented a low-complexity location-assisted beamforming scheme to improve the spectral efficiency in HAP downlink communication. In [29], an aerial reconfigurable intelligent surface (IRS) was introduced to assist the downlink transmission of HAP when the direct link is blocked. The work in [30] minimized the transmit power of HAP under necessary constraints in a IRS-enhanced satellite and HAP integrated network. Note that these works typically focus on the communication between platforms and terrestrial users. Different from the above works, we focus on how to maximize the transmission data rate of inter-HAP directional links.

VII. Conclusion

In this paper, an HAP ad hoc network is envisioned for realizing a large-area sensor network in the remote regions. To efficiently transmit a large amount of data from HAPs to GCSs, we propose a cross-layer dynamic tree-based routing and beamwidth optimization scheme. In the routing policy, we first utilize the weighted Voronoi diagram to divide the whole target area into several subdomains with the goal of balancing the traffic load among GCSs. Then we present a B&R algorithm to construct a tree topology with minimum convergecast delay to aggregate data in each subdomain. Moreover, an RPS algorithm is introduced to deal with the dynamic changes in the multi-HAP network. Based on the transmission schedule in the routing policy, a PSO algorithm is proposed to optimize transmission and reception beamwidths to maximize the minimum transmission data rate. Numerical results in the networks with different sizes show that our proposed schemes can effectively improve the network performance in comparison with some baseline approaches.

References

[1] Y. Wang, C. Zhang, and M. Pan, "Optimizing data transmission in high altitude balloon networks with multi-beam directional antennas," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[2] G. Avdikos, G. Papadakis, and N. Dimitriou, "Overview of the application of high altitude platform (HAP) systems in future telecommunication networks," in *Proc. IEEE 10th Int. Workshop Signal Process.*, 2008, pp. 1–6.

[3] A. Aragon-Zavala, J. L. Cuevas-Ruíz, and J. A. Delgado-Penín, *High-Altitude Platforms for Wireless Communications*, vol. 5. Hoboken, NJ, USA: Wiley, 2008.

[4] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, Jul. 2017.

[5] ITU Radiocommunication Assembly, Preferred Characteristics of Systems in the Fixed Service Using High Altitude Platforms Operating in the Bands 47.2–47.5 GHz and 47.9–48.2 GHz, Rec. ITU-R F, 1500, International Telecommunication Union, Geneva, Switzerland, 2000.

[6] J. Wildman, P. H. J. Nardelli, M. Latva-aho, and S. Weber, "On the joint impact of beamwidth and orientation error on throughput in directional wireless poisson networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 12, pp. 7072–7085, Dec. 2014.

[7] H. Shokri-Ghadikolaei, L. Gkatzikis, and C. Fischione, "Beam-searching and transmission scheduling in millimeter wave communications," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 1292–1297.

[8] J. Wang et al., "Beam codebook based beamforming protocol for multi-Gbps millimeter-wave WPAN systems," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 8, pp. 1390–1399, Oct. 2009.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. 4, pp. 1942–1948.

[10] C. Perfecto, J. Del Ser, and M. Bennis, "Millimeter-wave V2V communications: Distributed association and beam alignment," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2148–2162, Sep. 2017.

[11] Y. Liu, X. Fang, and M. Xiao, "Joint transmission reception point selection and resource allocation for energy-efficient millimeter-wave communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 412–428, Jan. 2021.

[12] "National oceanic and atmospheric administration," NOAA Operational Model Archive and Distribution Systems, 2024. [Online]. Available: http://nomads.ncep.noaa.gov/

[13] C. Pan and H. Zhang, "A time efficient aggregation convergecast scheduling algorithm for wireless sensor networks," *Wireless Netw.*, vol. 22, no. 7, pp. 2469–2483, Aug. 2016.

[14] C.-M. Cheng, P.-H. 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.*, 2007, pp. 4417–4424.

[15] G. B. Lamont, J. N. Slear, and K. Melendez, "UAV swarm mission planning and routing using multi-objective evolutionary algorithms," in *Proc. IEEE Symp. Comput. Intell. Multicriteria Decis. Mak.*, 2007, pp. 10–20.

[16] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," in *Proc. IEEE Infocom Conf.*, 2002, vol. 2, pp. 39–44.

[17] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. Conf. Commun. Archit., Protoc. Appl.*, 1994, pp. 234–244.

[18] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proc. IEEE lnt. Multi Topic Conf.*, 2001, pp. 62–68.

[19] D. B. Johnson et al., "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 139–172, 2001.

[20] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. IEEE 2nd Workshop Mobile Comput. Syst. Appl.*, 1999, pp. 90–100.

[21] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 243–254.

[22] L. Lin, Q. Sun, J. Li, and F. Yang, "A novel geographic position mobility oriented routing strategy for UAVs," *J. Comput. Inf. Syst.*, vol. 8, no. 2, pp. 709–716, Feb. 2012.

[23] F. Uyeda et al., "SDN in the stratosphere: Loon's aerospace mesh network," in *Proc. ACM SIGCOMM Conf.*, 2022, pp. 264–280.

[24] B. Barritt and V. Cerf, "Loon SDN: Applicability to NASA's next-generation space communications architecture," in *Proc. IEEE Aerosp. Conf.*, 2018, pp. 1–9.

[25] T. Ge, Y. Wang, C. Zhang, and Y. Fang, "Reconfiguration in maritime networks integrated with dynamic high altitude balloons," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[26] P. G. Sudheesh, M. Mozaffari, M. Magarini, W. Saad, and P. Muthuchidambaranathan, "Sum-rate analysis for high altitude platform (HAP) drones with tethered balloon relay," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1240–1243, Jun. 2018.

[27] M. D. Zakaria, D. Grace, and P. D. Mitchell, "Antenna array beamforming strategies for high altitude platform and terrestrial coexistence using k-means clustering," in *Proc. IEEE 13th Malaysia Int. Conf. Commun.*, 2017, pp. 259–264.

[28] Z. Xu, G. White, and Y. Zakharov, "Optimisation of beam pattern of high-altitude platform antenna using conventional beamforming," *IEE Proc. Commun.*, vol. 153, no. 6, pp. 865–870, Dec. 2006.

[29] N. Gao, S. Jin, X. Li, and M. Matthaiou, "Aerial RIS-assisted high altitude platform communications," *IEEE Wireless Commun. Lett.*, vol. 10, no. 10, pp. 2096–2100, Oct. 2021.

[30] S. Xu, J. Liu, T. K. Rodrigues, and N. Kato, "Robust multi-user beamforming for IRS-enhanced near-space downlink communications coexisting with satellite system," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14900–14912, Aug. 2022.

Yuanyuan Wang received the B.E. degree from the Hefei University of Technology, Hefei, China, in 2017. She is currently working toward the Ph.D. degree with the University of Science and Technology of China, Hefei. Her research interests include wireless ad hoc network, network optimization, and edge computing.

Chi Zhang (Member, IEEE) received the B.E. and M.E. degrees in electrical and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2011. In 2011, he joined the School of Information Science and Technology, University of Science and Technology of China, Hefei, China, as an Associate Professor. His research interests include network protocol design and performance analysis and network security, particularly for wireless networks and social networks.

Taiheng Ge received the B.E. degree from the Anhui University of Technology, Anhui, China, in 2013. He is currently working toward the Ph.D. degree with the University of Science and Technology of China, Hefei, China. His research interests include wireless communications and network optimization.

Miao Pan (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the Dalian University of Technology, Dalian, China, in 2004, the M.A.Sc. degree in electrical and computer engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2012. From 2012 to 2015, he was an Assistant Professor of computer science with Texas Southern University, Houston, TX, USA. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX. His research interests include cognitive radio networks, cyber-physical systems, and cybersecurity. He was the recipient of the best paper awards in Globecom 2017 and Globecom 2015, respectively. He is currently an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL.