ESTA: An Efficient Spatial-Temporal Range Aggregation Query Processing Algorithm for AAV Networks

Liang Liu, Wenbin Zhai, Xin Li, Youwei Ding, Wanying Lu, and Ran Wang, *Senior Member, IEEE*

***Abstract*—Autonomous Aerial Vehicle (AAV) networks are increasingly deployed in military and civilian applications, serving as critical platforms for data collection. Users frequently require aggregated statistical information derived from historical sensory data within specific spatial and temporal boundaries. To address this, users submit aggregation query requests with spatial-temporal constraints to target AAVs that store the relevant data. These AAVs process and return the query results, which can be aggregated within the network during transmission to conserve energy and bandwidth-resources that are inherently limited in AAV networks. However, the dynamic topology caused by AAV mobility, coupled with these resource constraints, makes efficient in-network aggregation challenging without compromising user query delay. To the best of our knowledge, existing research has yet to adequately explore spatial-temporal range aggregation queries in the context of AAV networks. In this paper, we propose ESTA, an Efficient Spatial-Temporal range Aggregation query processing algorithm tailored for AAV networks. ESTA leverages pre-planned AAV trajectories to construct a topology change graph that models the network's evolving connectivity. It then employs an efficient shortest path algorithm to determine the minimum query response delay. Subsequently, while adhering to user-specified delay constraints, ESTA transforms the in-network aggregation process into a series of set cover problems, which are solved recursively to build a Spatial-Temporal Aggregation Tree (STAT). This tree enables the identification of an energy-efficient routing path for aggregating and delivering query results. Extensive simulations demonstrate that ESTA reduces energy consumption by more than 50% compared to the baseline algorithm, all while satisfying the required query delay.**

Received 14 May 2025; revised 21 September 2025; accepted 6 October 2025. Date of publication 8 October 2025; date of current version 11 December 2025. This work was supported by the Open Fund of Key Laboratory of Civil Aviation Smart Airport Theory and System, Civil Aviation University of China, under Grant SATS202206. Recommended for acceptance by Prof. Geng Sun. *(Liang Liu and Wenbin Zhai contributed equally to this work.) (Corresponding author: Ran Wang.)*

Liang Liu, Xin Li, and Wanying Lu are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: liangliu@nuaa.edu.cn; nuaalix@nuaa.edu.cn; wanyinglu@nuaa.edu.cn).

Wenbin Zhai is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: wenbin.zhai@connect.polyu.hk).

Youwei Ding is with the School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing 210046, China (e-mail: ywding@njucm.edu.cn).

Ran Wang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China, and also with the College of Computer and Data Science, Fuzhou University, Fuzhou 350122, China (e-mail: wangran@nuaa.edu.cn).

Digital Object Identifier 10.1109/TNSE.2025.3619221

***Index Terms*—Autonomous aerial vehicle (AAV) networks, spatial-temporal range queries, in-network data aggregation, energy-efficient algorithms.**

I. Introduction

OVER the past few decades, advancements in sensor technology, navigation systems, and wireless communications have significantly improved the capabilities of Autonomous Aerial Vehicles (AAVs). These enhancements, coupled with their low cost, flexible deployment, and operational simplicity, have driven their widespread use across diverse military and civilian applications. Examples include military reconnaissance, border surveillance, disaster response, and agricultural monitoring [1], [2], [3]. As AAVs continue to evolve, their role in data-intensive missions has expanded, necessitating efficient methods for data collection and processing in networked environments.

In most AAV applications, multiple AAVs collaborate within a multi-hop network to gather data and execute missions effectively. This network can be conceptualized as a distributed database, with each AAV storing sensory data annotated with spatial and temporal metadata [4], [5]. When users seek information from a specific geographic region over a defined time interval, they issue query requests containing spatial-temporal constraints. These requests are directed to the AAVs holding the relevant data, which then search their local storage and return the qualifying records to a ground station. This process, while straightforward, poses significant challenges due to the resource limitations inherent in AAV networks.

AAVs are typically energy-constrained, and the bandwidth available in AAV networks is often limited [6]. Transmitting all raw query results to the ground station is thus resource-intensive, consuming excessive energy and bandwidth. Furthermore, in many practical scenarios, users require only statistical summaries—such as averages, maxima, or minima—rather than the entirety of the raw data [7]. For example, in forest fire monitoring, a user might need only the maximum temperature within a region r_1 between times t_1 and t_2. Sending the complete dataset in such cases is inefficient and wasteful, underscoring the need for optimized data handling strategies [8], [9].

Data aggregation offers a promising solution to this problem [10]. This technique involves in-network processing, where intermediate nodes within the network aggregate raw data using methods such as computing means, maxima, minima, or

summaries [11]. By aggregating data en route to the ground station, only the processed results are transmitted, significantly reducing data volume. This approach eliminates redundancy, extracts essential information, and lowers the number and size of transmissions, thereby conserving energy and bandwidth resources critical to AAV operations.

Although data aggregation has gained considerable attention for its efficiency [12], [13], existing research predominantly focuses on static Wireless Sensor Networks (WSNs) [14], [15], [16], with little exploration of its application in dynamic AAV networks. To the best of our knowledge, no prior studies have specifically addressed spatial-temporal range aggregation queries in AAV networks. Unlike static WSNs, AAV networks exhibit distinct characteristics, including high mobility, sparse node distribution, intermittent connectivity, variable link quality, and reliance on store-carry-forward (SCF) mechanisms. These properties introduce unique challenges for spatial-temporal range aggregation queries: First, the rapidly changing network topology and intermittent communication links hinder the establishment of efficient in-network aggregation routes. Second, users prioritize minimizing the time from issuing a query to receiving the result (query response time) over internal network dynamics [17], making it critical to balance aggregation efficiency with timely delivery.

Addressing these challenges requires a novel approach tailored to the AAV context. In this paper, we propose ESTA, an Efficient Spatial-Temporal range Aggregation query processing algorithm designed specifically for AAV networks. ESTA aggregates query results as early as possible during transmission to reduce energy and bandwidth consumption while ensuring that query response time remains within acceptable limits. The algorithm operates as follows: First, ESTA leverages pre-planned AAV trajectories to pinpoint target AAVs storing the required data. Meanwhile, a Topology Change Graph (TCG) is constructed to capture time-varying communication opportunities between AAVs. Then, using the TCG, an efficient shortest path algorithm calculates the minimum query response time. The aggregation process is modeled as a recursive set cover problem, resulting in a Spatial-Temporal Aggregation Tree (STAT). In this tree, the ground station serves as the root, and intermediate nodes aggregate data from their children before forwarding it to their parents, enabling optimized in-network processing. The main contributions of this paper are as follows:

- We propose an Efficient Spatial-Temporal range Aggregation query processing (ESTA) algorithm for AAV networks, which can find an efficient in-network aggregation path for target AAVs without the sacrifice of the user query delay. To the best of our knowledge, we are the first to study the spatial-temporal range aggregation query in AAV networks.
- We conduct extensive simulations on the Opportunistic Network Environment (ONE) simulator [18]. The experimental results show the superior performance of ESTA compared with the baseline spatial-temporal range aggregation query processing algorithm in terms of the query delay and energy consumption.

The remainder of this paper is structured as follows: Section II reviews the state-of-the-art in spatial-temporal range aggregation and AAV network routing protocols. Section III outlines the system model. Section IV provides a detailed description of the ESTA algorithm. Section V presents the simulation-based performance evaluation, and Section VII concludes the paper with a summary and future directions.

II. Related Work

This section synthesizes advancements in two critical domains: we first analyse foundational data aggregation techniques, which mainly focus on static network infrastructures, then survey routing protocol innovations in AAV networks that support our proposed framework. The intersectional gap between these domains motivates our research contribution. An overview of prior research and this work is shown in Table I.

A. Data Aggregation Techniques

Data aggregation mechanisms have become essential for enhancing energy efficiency and optimizing bandwidth utilization in resource-constrained networks [19]. Current methodologies predominantly focus on static network infrastructures and can be classified into three main paradigms: tree-based, cluster-based, and machine learning-based data aggregation [10].

1) Tree-Based Data Aggregation: In tree-based data aggregation schemes, nodes are hierarchically organized into a tree topology. Each non-leaf node serves as an aggregator, collecting sensory data from its child nodes, performing aggregation, and forwarding the results to its parent node. This recursive process continues until the aggregated data reaches the root node [20], [21].

In [22], the authors propose an approach that integrates genetic algorithms with density correlation metrics to construct adaptive aggregation trees, effectively reducing communication costs in Industrial Internet of Things (IoT) networks. Similarly, [23] introduces two innovative data aggregation models that employ mixed-integer programming to jointly optimize energy consumption for data aggregation and dissemination processes. Furthermore, [24] utilizes integer linear programming to address the trade-off between energy consumption and latency. Additionally, the same study applies heuristic methods to develop an aggregation tree rooted at the destination, aiming to minimize both energy usage and latency.

In [25], the authors design a tree topology for underwater sensor networks, taking into account inter-node distances and signal-to-noise ratios. They subsequently employ an ant search algorithm, which incorporates node energy levels, to identify efficient data transmission paths. In [26], the network topology is structured as a balanced binary tree to ensure uniform energy consumption across nodes. Moreover, cryptographic keys and error correction mechanisms are implemented to enhance the reliability of aggregation operations.

In [27], the authors apply a minimum spanning tree algorithm to establish a tree-based path, incorporating constraints on energy load and maximum depth to mitigate the hot spot problem

TABLE I
OVERVIEW OF PRIOR RESEARCH AND THIS WORK

Studies	Year	Network Model	Aggregation Algorithm	Routing Protocol	Limitations
Heidari et al. [22]	2024	Static IIoT	Tree-based	None	Poor Topological Robustness Limited Dynamic Adaptability Increased Topology Maintenance Overhead
Fitzgerald et al. [23]	2018	Static IoT			
Zhao et al. [24]	2020	Static SDN			
Zhang et al. [25]	2023	Static UWSNs			
Hasheminejad et al. [26]	2021	Static WSNs			
Wang et al. [27]	2022	Static WSNs			
Macriga et al. [28]	2023	Static WSNs			
Dao et al. [29]	2023	Static WSNs			
Bomnale et al. [30]	2024	Static WSNs			
Vo et al. [31]	2024	Static WSNs			
Sindhuja et al. [15]	2023	Static WSNs	Cluster-based	None	Frequent Reclustering Energy Consumption Imbalance Cluster Head Bottleneck Effect High Sensitivity to Parameter Selection
Mohseni et al. [13]	2023	Static IoT			
Amiri et al. [16]	2020	Static IIoT			
Saeedi et al. [33]	2022	Static IoT			
Qurabat et al. [34]	2020	Static IoT			
Idrees et al. [35]	2021	Static WSNs			
Alshehri et al. [36]	2024	Static IoT			
Zhang et al. [37]	2022	Static IoT			
Huang et al. [38]	2021	Static Smart Grid			
Yarinezhad et al. [39]	2021	Static IoT			
Ko et al. [40]	2019	Static IoT	Q-learning-based	None	Strong Reliance on Data Quality Weak Environmental Adaptability High Computational Resource Demand Significant Energy Consumption
Maivizhi et al. [41]	2021	Static WSNs			
Yun et al. [42]	2021	Static WSNs			
Ravi et al. [43]	2023	Static IoT	Deep-learning-based		
Zhang et al. [44]	2025	Static WSNs			
Hemalatha et al. [45]	2024	Static WSNs			
Zhang et al. [46]	2021	Static WSNs			
Wang et al. [47]	2023	Static WSNs	DRL-based		
Yarinezhad et al. [48]	2024	Static WSNs			
Mahmood et al. [49]	2024	Static WSNs			
Zhai et al. [51]	2022	AAV Networks	None	Topology-based	Significant Topology Maintenance Overhead Restricted Convergence Rate Insufficient Scalability
Ren et al. [52]	2024	AAV Networks			
Zhou et al. [53]	2024	AAV Networks			
Fu et al. [54]	2018	Mobile Networks			
Navarro et al. [55]	2022	AAV Networks			
Cui et al. [56]	2022	AAV Networks			
Arafat et al. [57]	2021	AAV Networks			
Yang et al. [58]	2022	AAV Networks			
Zheng et al. [59]	2024	AAV Networks	None	Geographic-based	Routing Hole Problem Uneven Energy Dissipation Degraded Complex Environments Adaptability
Yarinezhad et al. [61]	2021	Mobile Networks			
Huang et al. [63]	2017	Mobile Networks			
Huang et al. [62]	2017	Mobile Networks			
Cui et al. [64]	2023	AAV Networks			
Singh et al. [60]	2025	AAV Networks			
Usman et al. [65]	2021	AAV Networks			
Kumar et al. [66]	2023	AAV Networks			
Syed et al. [72]	2024	Vehicle Networks	None	SCF-based	High Latency Characteristics Storage Resource Constraints
Asadpour et al. [70]	2016	AAV Networks			
Bine et al. [71]	2023	Vehicle Networks			
Peng et al. [68]	2020	AAV Networks			
Peng et al. [69]	2019	AAV Networks			
Mahajan et al. [73]	2024	AAV Networks			
Tang et al. [74]	2025	AAV Networks			
Azzoug et al. [75]	2022	AAV Networks			
This work	-	AAV Networks	**Trajectory-based + Tree-based**	**Trajectory-based + SCF-based**	-

Note: IIoT, industrial internet of things; UWSNs, underwater wireless sensor networks; SDN, software-defined networking; AAV, Autonomous aerial vehicle; DRL, deep reinforcement learning; SCF, store-carry-forward.

in WSNs. In [28], an energy-efficient greedy tree algorithm is proposed for network modeling, considering global routing costs to minimize transmission energy consumption. Likewise, [29] employs a heuristic algorithm to construct an aggregation tree that reduces redundant packets, thereby decreasing the total energy cost associated with packet transmission and reception across the network.

In [30], an enhanced shortest path algorithm is presented, integrating node bandwidth utilization and hop count to optimize data aggregation operations. Finally, in [31], the authors explore a multi-channel duty cycle WSN scenario and dynamically select aggregation nodes by constructing a minimum sleep delay tree to reduce delivery delay.

However, existing tree-based data aggregation methods face challenges of poor topological robustness and insufficient dynamic adaptability when applied to AAV networks. The rapid movement of AAV nodes induces frequent topological changes, which existing tree-based data aggregation schemes designed for static networks cannot effectively accommodate. This limitation inevitably leads to the frequent reconstruction of routing trees, resulting in increased communication overhead and potential routing failures. The inherent mismatch between static

topology-dependent aggregation mechanisms and highly dynamic AAV network environments fundamentally undermines the efficiency and reliability of data transmission.

2) Cluster-Based Data Aggregation: Cluster-based data aggregation is a widely utilized technique in WSNs, wherein the network is partitioned into multiple clusters. Each cluster is supervised by a designated cluster head (CH) node, which is responsible for aggregating sensor data from cluster members and forwarding the aggregated data to the base station [32].

In [15], a distributed data aggregation algorithm is proposed that effectively integrates clustering and routing considerations. Cluster heads are selected based on several criteria, including connectivity, distance to the base station, energy levels, and delay. To improve efficiency, a self-attention mechanism is employed to optimize routing paths from CHs to the base station. Similarly, [13] employs fuzzy logic for CH selection and introduces the Capuchin search algorithm to enhance routing optimization. In [16], a multi-objective, multi-attribute data aggregation framework is presented, leveraging fuzzy logic for clustering and a hybrid metaheuristic algorithm for routing.

The method described in [33] partitions nodes into disjoint clusters according to their geographical locations. During data aggregation, energy consumption in network transmission is reduced by identifying and eliminating extreme data points. In [34], CHs are dynamically selected based on the similarity of transmitted data packets, with adjustments to sampling frequency and selective transmission triggered by variations in sensor data. Meanwhile, the authors in [35] propose an efficient data encoding scheme tailored to specific application scenarios, followed by data aggregation at CHs utilizing fog computing.

In [36], an optimized Firefly Algorithm (FA) is applied to clustering, taking into account the energy consumption and distance of IoT nodes to enhance network resource utilization. The approach in [37] incorporates node energy and duty cycle into CH selection. Additionally, it exploits the correlation of multi-dimensional, multi-attribute data to reconstruct and aggregate information, thereby reducing network energy consumption.

For IoT networks within smart grid environments, the authors in [38] propose a hierarchical tree structure comprising smart meters, data concentrator units, and meter data management systems. This architecture is complemented by data compression algorithms to enable efficient data aggregation. In [39], a novel clustering approach based on the Fixed Parameter Tractable (FPT) algorithm is introduced to balance traffic load and energy consumption among CHs in WSNs.

These advancements in cluster-based data aggregation underscore a diverse array of strategies—ranging from metaheuristic optimization and fuzzy logic to dynamic CH selection and hierarchical structuring—aimed at enhancing energy efficiency, scalability, and resource utilization in WSNs and IoT networks. However, when applied to highly dynamic AAV networks, cluster-based data aggregation methods inevitably encounter challenges related to frequent re-clustering and the reelection of cluster heads. This arises from the inherent mobility of AAV nodes, which disrupts cluster stability and requires topological reorganization as network configurations evolve. Furthermore, these approaches face significant limitations, including bottleneck effects at CH nodes and uneven energy consumption across the network. The performance of such methods is also highly dependent on parameter selection, with conventional configurations frequently proving inadequate in adapting to rapid topological changes. As a result, the reliability and efficiency of cluster head generation mechanisms remain uncertain in dynamic network environments, particularly under conditions of continuous node mobility and unpredictable link variations.

3) Machine-Learning-Based Data Aggregation: Machine Learning (ML) has emerged as a critical tool for enhancing data aggregation techniques in WSNs. For instance, the authors in [40] propose a novel data aggregation approach that incorporates duty cycles. This method leverages a Markov decision process to simultaneously optimize duty cycles and data aggregation, striking a balance among energy consumption, throughput, and data consistency. Similarly, the authors in [41] employ Q-learning, a reinforcement learning technique, to dynamically identify optimal routing paths for data aggregation. By constructing an optimal routing aggregation tree based on residual energy, inter-node distances, and link quality, this study also addresses the routing hole problem. Additionally, the authors in [42] introduce a Q-learning-based scheme aimed at improving energy efficiency and extending network lifetime.

Beyond reinforcement learning, deep learning techniques have been widely explored for WSN data aggregation. In [43], Deep Neural Networks (DNNs) are utilized to optimize routing strategies. By integrating DNNs with clustering algorithms, this approach reduces data latency and traffic congestion, thereby prolonging the network's operational lifespan. Furthermore, the authors in [44] present a semi-stationary clustering algorithm that combines Q-learning with an ant colony genetic algorithm to dynamically select cluster nodes. Another study, [45], harnesses Generative Adversarial Networks (GANs) for clustering sensor nodes and employs Long Short-Term Memory (LSTM) networks for data aggregation. A significant contribution from [46] integrates compressed sensing, sparse data reconstruction, and DNNs to develop an end-to-end learning framework for IoT networks, effectively aggregating data while minimizing transmission volume.

Other innovative strategies further enhance WSN performance. For example, the authors in [47] pair reinforcement learning with sleep scheduling schemes that consider node residual energy and sampling uniformity, resulting in aggregated and reconstructed data that extends network lifespan. Similarly, the authors in [48] use LSTM networks to improve data accuracy and variational autoencoders for data aggregation, enabling dynamic adaptation to vary data distributions for reliable aggregation and transmission. Lastly, the authors in [49] propose a scheme integrating Recurrent Neural Networks (RNNs) and LSTM networks to strengthen network resilience, supporting dynamic routing and load balancing for efficient data aggregation.

While machine learning, particularly deep learning and reinforcement learning, has been extensively adopted for data aggregation in static WSNs, its application to AAV networks poses significant research challenges. ML-based approaches typically

rely on high-quality training data and environmental stability, both of which are often lacking in AAV network scenarios due to their inherent variability. Moreover, pre-trained models struggle to adapt to the dynamic conditions of AAV networks, which are characterized by rapid node mobility and continuously evolving topologies. This adaptability issue necessitates frequent retraining, placing substantial computational and energy demands on the system. Such requirements are incompatible with the limited onboard energy reserves and computational capacities of AAVs, constraints that are particularly pronounced in aerial network operations. Consequently, addressing these challenges remains a critical area for future investigation to enable effective ML deployment in AAV networks.

In summary, compared with static networks, AAV networks have many unique characteristics, such as high mobility, sparse distribution, intermittent connectivity, unstable link quality and SCF mechanism. These characteristics make the above-mentioned approaches that rely on stable topology and constant connectivity inapplicable in AAV networks. As far as we know, there is no research on spatial-temporal range aggregation query processing in AAV networks.

B. Routing Protocols in AAV Networks

The efficiency of spatial-temporal range aggregation query processing in AAV networks hinges significantly on the design of routing protocols. To this end, we survey recent advancements in energy optimization within routing protocols tailored for AAV networks. These protocols can be broadly classified into three categories: topology-based, geographic, and store-carry-forward-based routing protocols.

1) Topology-Based Routing Protocols: Topology-based routing protocols utilize network topology information to enhance the efficiency of message forwarding and delivery [50]. In recent years, researchers have proposed various topology-aware strategies to optimize routing performance in AAV networks. These strategies, which leverage detailed knowledge of network structure, are summarized in the following sections.

Several studies have introduced innovative approaches to improve energy efficiency and adaptability in AAV routing protocols: In [51], a power-aware encounter tree is introduced, constructed based on pre-planned AAV trajectories. This approach aims to identify routing paths that are energy-efficient while adhering to specified delay constraints. In [52], the authors employ K-means online learning to dynamically reconstruct the network topology. This method facilitates load balancing and enhances data transmission efficiency through adaptive topological adjustments. In [53], a trajectory-based topology modeling approach is proposed, integrated with a modified version of Dijkstra's algorithm. This technique computes optimal routing paths tailored to dynamic environments.

Additional research has focused on multicast optimization and the use of topological redundancy: In [54], the multicast energy minimization problem in delay-constrained AAV networks is addressed by reformulating it as a directed Steiner tree optimization problem. This method accounts for both transmission and reception energy consumption within the communication model. In [55], an energy-efficient routing protocol is presented that leverages topological redundancy to optimize routing decisions. The protocol employs a dual-objective function to balance energy efficiency and load distribution, and incorporates a novel diagnostic and recovery framework that exploits topological diversity to enhance system reliability.

Advancements in machine learning and topology structuring have also been explored: In [56], an adaptive Q-learning strategy is developed to enable distributed and autonomous routing decisions. This strategy relies on real-time perception of the network topology and adapts effectively to dynamic topological changes. Building on Q-learning techniques, the authors in [57] introduce a topology-aware routing protocol that extends the topological visibility of nodes from one-hop to two-hop neighborhoods, thereby improving the quality of routing decisions. In [58], a network structuring methodology is proposed that transforms irregular topologies into organized architectures. This approach supports the implementation of a greedy forwarding mechanism, enhancing the efficiency of data dissemination.

Despite their application in AAV networks, topology-based routing protocols face significant challenges in highly dynamic operational environments. These protocols require continuous maintenance of topological information, including link status, neighbor relationships, and node connectivity. Such maintenance becomes particularly challenging when the network topology undergoes frequent changes.

In scenarios involving abrupt topological shifts, the process of recalculating routing paths can incur extended convergence times, potentially resulting in substantial packet delays or losses. Furthermore, as the density of AAV nodes increases, the size of routing tables grows significantly, leading to heightened computational resource demands and memory overhead. This scalability issue poses critical challenges for the deployment of large-scale AAV networks, particularly given the resource-constrained nature of typical AAV platforms.

2) Geographic Routing Protocols: Geographic routing protocols differ fundamentally from topology-based approaches by leveraging localized position information rather than global network topology data to inform routing decisions [50]. This method forwards packets to the neighbor node that is geographically closest to the destination, offering an efficient alternative to conventional routing strategies. In the context of AAV networks, recent advancements have significantly enhanced the application of geographic routing, enabling it to address the distinctive challenges posed by dynamic and resource-constrained environments.

A key challenge in AAV networks is adapting to rapidly changing network topologies. To address this, the authors in [59] have proposed adaptive mechanisms, such as integrating beaconing with greedy forwarding, which facilitates the dynamic selection of optimal next-hop nodes. This approach ensures robust performance despite frequent topological changes. Similarly, connectivity-aware algorithms, as presented in [60], improve routing reliability by utilizing link lifetime prediction and quality assessment to sustain stable communication paths.

Energy efficiency is a critical concern in AAV networks due to the limited battery capacity of these devices. To tackle this issue, advanced protocols have been developed that employ multi-criteria decision-making, integrating factors such as positional data, energy consumption, and latency to construct energy-aware routing trees [61]. These trees optimize resource utilization while maintaining connectivity along the shortest paths. Additionally, energy-efficient multicast trees, as explored in [62], leverage both positional and energy profile data to enhance routing performance for group communications.

Topological voids, often arising from sparse node deployment or equipment failures, pose significant barriers to effective routing in AAV networks. Topology-aware protocols, such as those described in [63], mitigate this challenge by assessing residual energy levels, spatial positioning, and consumption patterns to support route recovery in difficult scenarios. Furthermore, next-hop selection frameworks that consider wireless channel characteristics and inter-AAV interference incorporate recovery models to avoid local optima, thereby improving path reliability [64].

Innovative decision-making strategies further strengthen the capabilities of geographic routing. For instance, a protocol outlined in [65] integrates relative node velocity and angular alignment to reduce redundant transmissions and improve link quality. Another approach, detailed in [66], utilizes a utility function that combines multiple parameters—including energy reserves, inter-node distance, mobility speed, directional alignment, and link quality—to probabilistically select the most suitable next-hop.

Despite these advancements, pure geographic routing protocols face inherent limitations in AAV networks. Sparse node deployment and intermittent link availability disrupt the sustained connectivity assumed by greedy forwarding, necessitating hybrid approaches that integrate geographic awareness with predictive mechanisms or redundancy-based techniques to ensure robust operation. Additionally, the greedy forwarding strategy is prone to local optima, resulting in inefficient routing paths and increased latency. Compounding these challenges, nodes near destination areas often become communication hotspots, leading to accelerated energy depletion and a reduced network lifespan. These issues underscore the need for ongoing innovation to fully harness the potential of geographic routing in AAV applications.

3) Store-Carry-Forward-Based Routing Protocols: To address the limitations of traditional networking approaches in AAV systems, the SCF paradigm [67] has emerged as a robust solution, particularly in scenarios characterized by sparse connectivity. This approach enables nodes to store messages and physically transport them until a suitable forwarding opportunity arises, making it well-suited for environments where continuous end-to-end connectivity is impractical. Recent research has leveraged this paradigm, introducing diverse optimization strategies to enhance its applicability and performance in AAV networks.

A significant body of work has focused on trajectory-based methods to refine SCF protocols. For example, the authors in [68] propose a multicast priority encounter graph constructed from pre-planned AAV trajectories, enabling efficient forwarding sequences for group communications. Building upon this foundation, the authors in [69] integrate kinematic node information—such as location vectors and motion dynamics—to predict future trajectories, thereby enhancing path prediction accuracy and forwarding efficiency. Similarly, the authors in [70] advance packet forwarding by combining trajectory prediction models with AAV load balancing considerations, optimizing routing decisions through spatio-temporal analysis of node positions. In a related effort, the authors in [71] conduct a systematic evaluation of SCF protocols in sparse network scenarios, employing trajectory prediction algorithms to bolster message delivery reliability under intermittent connectivity conditions. Collectively, these studies underscore the pivotal role of trajectory-based techniques in improving SCF performance.

Reinforcement learning and probabilistic models have also been harnessed to optimize routing within SCF-based AAV networks. In [72], a Q-learning-based multi-objective routing protocol dynamically adjusts reinforcement learning parameters to adapt to AAV mobility patterns. This approach simultaneously minimizes energy expenditure and transmission delay by leveraging learned network states to identify optimal forwarding paths. Extending the scope of optimization, the authors in [73] introduce a multi-objective Markov chain routing mechanism that jointly optimizes node residual energy and expected end-to-end delay through probabilistic state transitions. These methods highlight the potential of machine learning and statistical techniques to address complex routing challenges in dynamic AAV environments.

Adaptive and bio-inspired strategies offer additional avenues for enhancing SCF protocols. For instance, the authors in [74] develop an adaptive routing protocol that dynamically selects next-hop nodes in sparse settings by integrating three key metrics: movement direction alignment, relative velocity differentials, and energy reserves. This method ensures efficient forwarding decisions tailored to real-time network conditions. In contrast, the authors in [75] propose a novel bio-inspired social spider optimization algorithm, which regulates packet replication rates and relay selection by utilizing historical forwarding statistics and topological features. These innovative approaches demonstrate the diversity of techniques being explored to refine SCF mechanisms.

The advancements outlined above illustrate the versatility of the SCF paradigm in tackling the inherent challenges of AAV networks. Nevertheless, significant hurdles persist, including scalability in large-scale deployments and the computational overhead associated with complex optimization models. These issues warrant further investigation to ensure the practical deployment of SCF-based solutions. Moreover, in highly dynamic network environments, achieving an optimal balance between energy efficiency, latency constraints, and high data delivery rates remains a critical priority. This multidimensional optimization challenge calls for innovative strategies capable of reconciling competing performance metrics under time-varying conditions, paving the way for future research to enhance the robustness and efficiency of AAV networks.

C. Research Gaps and Our Contributions

The preceding review in this section has detailed advances in two key domains pertinent to our work: data aggregation techniques and routing protocols for AAV networks. However, a significant research gap emerges at the intersection of these fields.

On one hand, while data aggregation is paramount for enhancing energy and bandwidth efficiency in wireless networks, current research on these mechanisms is predominantly focused on static networks, which are characterized by stationary nodes and fixed topologies. The high mobility of nodes and the perpetually dynamic topology of AAV networks render these static approaches unsuitable and ineffective.

On the other hand, existing energy-efficient routing protocols designed for AAV networks often overlook the potential of in-network data aggregation during query processing. This omission leads to significant inefficiencies when handling spatio-temporal range aggregation queries. Furthermore, these protocols fail to fully leverage the rich trajectory and topological information inherent in AAV operations for routing optimization.

This distinct gap motivates our research contribution. To the best of our knowledge, no prior work has systematically addressed the joint challenge of routing optimization and processing for spatio-temporal range aggregation queries specifically within AAV networks. Table I summarizes the key distinctions between our approach and previous studies. Specifically, our work introduces a holistic solution. First, we leverage pre-planned AAV trajectory information to dynamically model the network's topological evolution. Then, we transform the aggregation query into a series of set-covering problems. Finally, the query processing is directly integrated into the routing process through a SCF paradigm. This approach ensures that data is opportunistically aggregated during the forwarding phase, thereby minimizing energy consumption while adhering to message delivery latency constraints.

III. System Model

A. Network Model

We consider the AAV network which consists of multiple AAVs and a ground station. Without loss of generality, we abstract the AAV network from the three-dimensional space into a Euclidean space, ignoring the vertical space [51]. Furthermore, in most application scenarios like military missions, the flight trajectories of AAVs are pre-planned and can be obtained in advance through mission planning and path planning [76], [77], [78]. Even if AAVs re-plan the trajectories during the mission, their trajectories can also be obtained by the ground station in advance through the out-of-band channel [67], [68], [79].

Each AAV flies along its respective pre-planned trajectory, collects spatial-temporal sensory data and then stores it locally. The ground station may send an aggregation query request with a specific spatial-temporal constraint to target AAVs as needed, denoted as $Q = Request(T, R, D, A)$, where T is the time period of the query, R is the target query region, D is the data type of the query and A is the aggregation operation. For example, if the ground station queries the maximum temperature in the region r_1 between t_1 and t_2, the query request can be expressed as $Q_1 = Request([t_1, t_2], r_1, TEMP, MAX)$. After receiving the query request, each target AAV will search its local sensory data, perform the aggregation operation and return the qualified data as its query result to the ground station. For instance, each target AAV first searches its local sensory data on temperature in r_1 between t_1 and t_2. Then, it performs "MAX" aggregation operation and uses the maximum temperature as its aggregation query result.

Meanwhile, the computing resources of AAVs are relatively abundant, however, the energy and bandwidth resources are relatively scarce [51]. Moreover, in AAV networks, the cost of the communication is multiple times that of the computation [81]. Therefore, in the spatial-temporal range aggregation query processing, each AAV in the network can aggregate multiple received query results, and then forward the aggregated result to the ground station. For instance, after a AAV receives query results that contain the maximum temperature from multiple AAVs, it will aggregate these query results and select the highest maximum temperature as the aggregated query result. For convenience, in this paper, we assume that the packet size of the query result of each target AAV and the aggregated result is the same.

B. Topology Change Graph

After receiving the query requests, target AAVs will search their respective local sensory data and then send the aggregation query results back to the ground station. Meanwhile, these query results should be further aggregated within the network during transmission to reduce the energy and bandwidth consumption. However, due to the high dynamic of topology and intermittent connectivity of communications in AAV networks, it is difficult to find an efficient in-network aggregation routing for query results. To solve the above problem, in this paper, we build the topology change graph (TCG) based on the pre-planned trajectories of AAVs in order to accurately reflect communication windows between AAVs and topology changes of the network.

The changes of the AAV network topology can be abstracted into a topology change graph, which can be formalized as $TCG =< V, E >$, where V is the set of AAVs in the network and E represents communication links between AAVs. Based on the pre-planned trajectories of AAVs, we can calculate the communication windows between AAVs. It is worth noting that due to the high dynamic of topology, there may be multiple intermittent connections and communications between two AAVs. Without loss of generality, in this paper, we abstract them as a continuous time period [4]. In other words, for each $e_{ij} \in E$, it can be formalized as a quadruple $e_{ij} =< u_i, u_j, t^{ij}_{begin}, t^{ij}_{end} >$, which represents that u_i and u_j can communicate with each other between t^{ij}_{begin} and t^{ij}_{end}. For example, as Fig. 1 shows, there are ten AAVs $u_1 \sim u_{10}$ and a ground station g_0. The edges represent communication windows between AAVs, such as AAV u_1 and u_4 can communicate with each other between 0 s and 2 s, namely $e_{14} =< u_1, u_4, 0\,\text{s}, 2\,\text{s} >$.

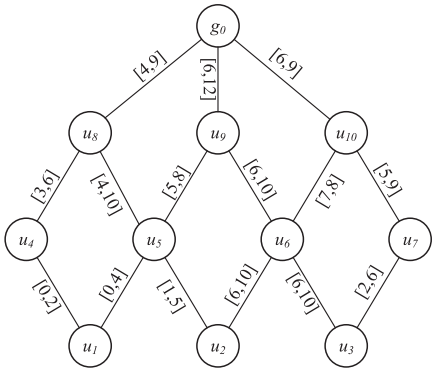

Fig. 1. The topology change graph.

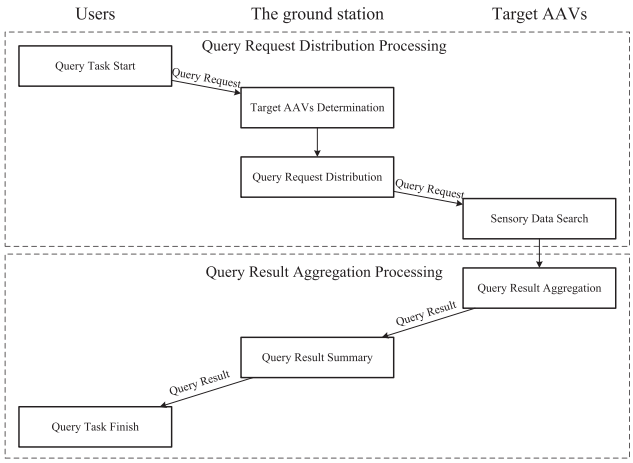

Fig. 2. The spatial-temporal range aggregation query processing architecture.

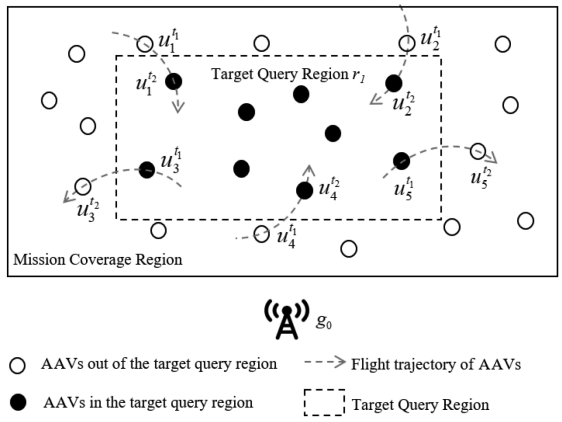

Fig. 3. The AAV network status.

IV. ESTA

A. Architecture

Fig. 2 shows the architecture of ESTA. It consists of three phases: target AAVs determination, query requests distribution and query results aggregation.

1) Target AAVs Determination (Section IV-B): When users are interested in the sensory data in a certain region during a certain time period, the ground station first uses the pre-planned trajectories of AAVs to determine target AAVs which store and carry the qualified data.

2) Query Requests Distribution (Section IV-C): After obtaining target AAVs, the ground station distributes aggregation query requests to target AAVs based on a certain query distribution routing protocol.

3) Query Results Aggregation (Section IV-D): When target AAVs receive the query request, they search their respective locally stored data, perform the aggregation operation on sensory data satisfying the spatial-temporal constraint and then return the aggregation query results to the ground station. Meanwhile, in order to reduce energy and bandwidth consumption without sacrificing the user query delay, an efficient spatial-temporal aggregation tree is constructed, based on which the query results from different target AAVs can be further aggregated at the intermediate nodes during transmission.

B. Target AAVs Determination

Most of the existing researches [10] on spatial-temporal range aggregation query processing focus on static WSNs. In static WSNs, since the positions of nodes are fixed and usually do not change, it is very easy to determine target nodes that store and carry the qualified data, that is, nodes in the target query region.

However, in AAV networks, due to the high mobility of nodes and high dynamic of topology, when the ground station sends out the spatial-temporal range aggregation query request, the qualified target AAVs may have left the target query region. For instance, as shown in Fig. 3, AAVs fly in the mission coverage region and collect sensory data, $u_i^{t_j}$ represents the position of u_i at t_j. At t_2, the ground station g_0 requests to query the maximum temperature in the target region r_1 between t_1 and t_2, denoted as $Q_1 = Request([t_1, t_2], r_1, TEMP, MAX)$. We can find that at t_2, AAV u_3 and u_5 have left the target query region r_1 due to the flight movement. In this case, if we collect and aggregate data in the same way as static WSNs, the stored data in u_3 and u_5 will be missed.

Another feasible method is to use flooding to distribute the spatial-temporal range aggregation query request to all nodes in the network without distinguishing whether they store the query data or not [82]. This approach ensures the accuracy and correctness of the query, however, it consumes a huge amount of energy and bandwidth resources, which are typically scarce in AAV networks.

Therefore, in order to make up for these shortcomings, in this paper, we utilize the pre-planned trajectory information of AAVs to assist in determining target AAVs. First, the time period of the query T is abstracted into n discrete time points, denoted as $T =< t_1, \ldots, t_j, t_{j+1}, \ldots, t_n >$. Then, since the trajectory of each AAV is pre-planned, for each AAV u_i, we can calculate its position at t_j, denoted as $u_i^{t_j} = (x_{ij}, y_{ij}), 1 \leq j \leq n$. Based on the obtained n position points, the trajectory of AAV u_i in the time period T can be approximately divided into $n-1$ line segments, denoted as $FT_i =< a_1, \ldots, a_j, \ldots, a_{n-1} >$, where

$a_j = (u_i^{t_j}, u_i^{t_{j+1}})$. For AAV u_i, if there is a line segment a_j that interacts with the target query region R, u_i is considered as a target AAV. Finally, we will obtain the set of target AAVs, denoted as U_{target}.

C. Query Requests Distribution

After obtaining target AAVs, the ground station needs to distribute query requests to target AAVs. However, due to the unique characteristics of AAV networks, e.g., high mobility and intermittent connectivity, traditional routing protocols for well-connected networks, such as Ad hoc On-Demand Distance Vector (AODV) based [83] or Optimized Link State Routing (OLSR) based [84] routing protocols, are not suitable for dynamic AAV networks. Therefore, in order to successfully distribute query requests to each target AAV, in this paper, multi-hop routing protocols for AAV networks [51] are utilized. Especially, multicast routing protocols, whose target is to deliver messages from the source to a group of destinations, are well suited for query requests distribution, and there is a considerable research effort for the development of multicast routing protocols for AAV networks [4], [85].

Based on the above method, the ground station can efficiently distribute the query requests to target AAVs. In addition, it is worth noting that no matter what routing protocol is used, there is no guarantee that the query requests reach target AAVs and target AAVs send out the query results at the same time, which makes the assumption of synchronous transmission in most existing methods [11], [86] untenable and unrealistic. However, our query result aggregation scheme (see in Section IV-D) does not rely on this assumption and can cope well with the asynchrony of the transmissions.

D. Query Results Aggregation

1) Basic Idea: In the spatial-temporal range aggregation query processing, the ground station needs to receive query results from multiple target AAVs, and the query task becomes successful only when all query results are delivered to the ground station. Therefore, the user query delay is the time when all query results arrive at the ground station. Meanwhile, for the spatial-temporal range aggregation query task, query results from different target AAVs can be aggregated at the intermediate nodes of the network during the transmission process. For example, in Fig. 1, u_5 can aggregate the query results from u_1 and u_2, then deliver the aggregated result to g_0.

Based on the above observations and analyses, the main idea of this paper is to aggregate query results as soon as possible on the basis of ensuring the user query delay, thereby reducing data communication and energy consumption in the network. As depicted in Fig. 1, u_1, u_2 and u_3 are target AAVs and need to send query results back to the ground station. For convenience, in this paper, we assume that the duration to transmit a query result from one node to its neighbor node is 0.1 s, denoted as $t_{trans} = 0.1$ s, and the corresponding energy consumption is E_m due to the same packet size. Their transmission paths with the earliest delivery time to the ground station are

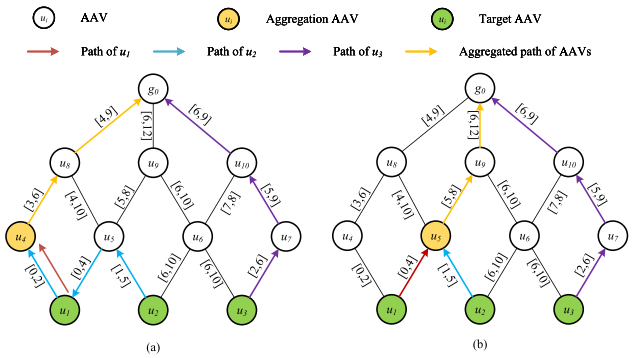

Fig. 4. The basic idea of ESTA. (a) Each query result is delivered along its shortest path. (b) Our proposed ESTA.

1) $pa_1 : u_1 \xrightarrow{[0,2]} u_4 \xrightarrow{[3,6]} u_8 \xrightarrow{[4,9]} g_0$. The earliest delivery time of the query result of u_1 is 4.1 s, and the corresponding energy consumption is $3E_m$.
2) $pa_2 : u_2 \xrightarrow{[1,5]} u_5 \xrightarrow{[0,4]} u_1 \xrightarrow{[0,2]} u_4 \xrightarrow{[3,6]} u_8 \xrightarrow{[4,9]} g_0$. The earliest delivery time of the query result of u_2 is 4.1 s, and the corresponding energy consumption is $5E_m$.
3) $pa_3 : u_3 \xrightarrow{[2,6]} u_7 \xrightarrow{[5,9]} u_{10} \xrightarrow{[6,9]} g_0$. The earliest delivery time of the query result of u_3 is 6.1 s, and the corresponding energy consumption is $3E_m$.

If each query result is delivered along its shortest path, the total energy consumption is $3E_m + 5E_m + 3E_m = 11E_m$. Meanwhile, if we take in-network aggregation into consideration, the query results of u_1 and u_2 can be aggregated at AAV u_4. As shown in Fig. 4(a), AAV u_4 will receive query results from u_1 and u_2 at 0.1 s and 1.3 s, u_4 can aggregate them and then forward the aggregated result to u_8 at 3 s. In this case, the total energy consumption becomes $3E_m + 1E_m + 2E_m + 3E_m = 9E_m$.

However, the above schemes are not efficient enough. Since the earliest delivery time of the last arrival packet (i.e., the query result of u_3) is 6.1 s, we take 6.1 s as the delay constraint of all query results in order to ensure the user query delay. Then, we aggregate query results of all target AAVs in the network as soon as possible, so we can obtain a better transmission scheme, as shown in Fig. 4(b). The query results of u_1 and u_2 will be aggregated at u_5 and then the aggregated result will be forwarded to g_0. The total energy consumption of this scheme is $1E_m + 1E_m + 2E_m + 3E_m = 7E_m$. In this case, the user query delay is the same as the above two schemes (i.e., 6.1 s), meanwhile, the energy consumption is further reduced.

2) Determine the User Query Delay: The main idea of this paper is to aggregate query results as soon as possible on the basis of ensuring the user query delay, thereby reducing data communication and energy consumption in the network. Therefore, we need to study and determine the user query delay first.

In the aggregation query processing, the query task becomes successful only when all query results return to the ground station, that is, the time when the last query result delivered to the ground station is the user query delay. We assume that there are a total of k target AAVs, denoted as $U_{target} =$

$\{u_1, u_2, \ldots, u_i, \ldots, u_k\}$. Without loss of generality, we take the minimum delay required for a query task as the user query delay. In other words, for the query result of each target AAV u_i, its earliest delivery time sent back to the ground station is t_i, so the user query delay is

$$t_d = \max\{t_1, t_2, \ldots, t_i, \ldots, t_k\} \tag{1}$$

Therefore, in order to obtain the user query delay, we need to calculate the earliest delivery time for each query result. In this paper, based on the constructed TCG, the problem of the earliest delivery time can be transformed into the shortest path problem. However, it is worth noting that, different from edges in traditional topology graph in static networks, edges in TCG have lifetime, which makes the shortest path problem unique and complex in AAV networks. Specifically, for $e_{ij} =< u_i, u_j, t^{ij}_{begin}, t^{ij}_{end} >$ and $e_{jk} =< u_j, u_k, t^{jk}_{begin}, t^{jk}_{end} >$, when u_j receives m from u_i at t^{ij}_{rx}, and $t^{jk}_{begin} + t_{trans} \leq t^{jk}_{end}$, where t_{trans} is the duration to transmit a query result m from u_j to u_k, there will be three cases:

- If $t^{ij}_{rx} < t^{jk}_{begin}$, the communication e_{jk} between u_j and u_k has not started yet, u_j has to store and carry m until t^{jk}_{begin}, and then u_j can forward m to u_k through e_{jk}.
- If $t^{jk}_{begin} \leq t^{ij}_{rx} \leq t^{jk}_{end} - t_{trans}$, u_j can immediately forward m to u_k through e_{jk}.
- If $t^{ij}_{rx} > t^{jk}_{end} - t_{trans}$, the communication e_{jk} between u_j and u_k is not enough to transmit m (i.e., $t^{jk}_{end} - t_{trans} < t^{ij}_{rx} \leq t^{jk}_{end}$) or even has finished (i.e., $t^{ij}_{rx} > t^{jk}_{end}$), u_j can not forward m to u_k through e_{jk}.

In brief, the query result m can be forwarded from u_j to u_k through e_{jk} if and only if

$$\max\{t^{ij}_{rx}, t^{jk}_{begin}\} + t_{trans} \leq t^{jk}_{end} \tag{2}$$

Based on the above observations and analyses, in this section, we propose the Shortest Path Algorithm (SPA) for TCG. For each target AAV u_s in U_{target}, we use A to denote the set of nodes that have found the shortest path which have the earliest delivery time from the source node (i.e., u_s) to themselves, and B to represent the set of nodes that have not yet found the shortest path, namely $B = V - A$. Meanwhile, t^s_{gen} is the time when the query result of u_s is generated, and $T(u_s, u_i)$ and $H(u_s, u_i)$ are the earliest delivery time and required hop count along the shortest path from u_s to u_i, where $u_i \in V$. To start with, set $A = \emptyset$ and $B = V$. Besides, $T(u_s, u_s) = t^s_{gen}, H(u_s, u_s) = 0$ and $T(u_s, u) = \infty, H(u_s, u) = \infty, u \in V - \{u_s\}$. The algorithm is expressed as follows:

1) The node that has the earliest delivery time in set B, denoted as u_i, is removed from this set and added into set A.
2) If u_i is the ground station g_0, return to Step 1; otherwise, consider each node which is adjacent to u_i and still in set B, denoted as $u_j \in U^i_{neighbor} - A$. If u_i can forward the query result of u_s to u_j through the edge e_{ij} (i.e., $\max\{T(u_s, u_i), t^{ij}_{begin}\} + t_{trans} \leq t^{ij}_{end}$), we judge whether $\max\{T(u_s, u_i), t^{ij}_{begin}\} + t_{trans}$ is smaller than $T(u_s, u_j)$, and if so, we update $T(u_s, u_j)$ as $\max\{T(u_s, u_i), t^{ij}_{begin}\} + t_{trans}$. Meanwhile, $H(u_s, u_j)$ is updated to $H(u_s, u_i) + 1$. This means through u_i, the query result of u_s can be delivered from u_s to u_j earlier.
3) If $B = \emptyset$, the process is done, which means the shortest paths, corresponding earliest delivery times and required hop counts from the target AAV u_s to each node in the network are all found; otherwise, return to Step 1.

Algorithm 1: SPA: The Shortest Path Algorithm for TCG.

Input:
U_{target}, TCG
Output:
$T(u_s, u_i), prev(u_s, u_i), H(u_s, u_i), u_s \in U_{target}, u_i \in V$

for each $u_s \in U_{target}$ **do**
  $A = \emptyset, B = V, T(u_s, u_s) = t^s_{gen}, H(u_s, u_s) = 0$
  for each $u \in V - u_s$ **do**
    $T(u_s, u) = \infty$
    $prev(u_s, u) = null$
    $H(u_s, u) = \infty$
  end for
  while $B \neq \emptyset$ **do**
    $u_i \Leftarrow \arg\min_{u \in B}\ T(u_s, u)$
    $A = A \cup \{u_i\}$
    $B = B - \{u_i\}$
    for each $u_j \in U^i_{neighbor} - A$ **do**
      $T_i(u_s, u_j) = \max\{T(u_s, u_i), t^{ij}_{begin}\} + t_{trans}$
      if $T_i(u_s, u_j) \leq t^{ij}_{end}$ && $T_i(u_s, u_j) < T(u_s, u_j)$ **then**
        $T(u_s, u_j) = T_i(u_s, u_j)$
        $prev(u_s, u_j) = u_i$
        $H(u_s, u_j) = H(u_s, u_i) + 1$
      end if
    end for
  end while
end for

After describing the main stages of the shortest path algorithm for TCG, we summarize it in Algorithm 1. If there is a path that can successfully deliver the query result from the target AAV to the ground station, SPA can always find the shortest one which has the earliest delivery time. Through SPA, we can obtain the earliest delivery time for the query result of each target AAV to the ground station, namely $t_1, t_2, \ldots, t_i, \ldots, t_k$. Then, as mentioned above, we take the maximum value as the user query delay, namely $t_d = \max\{t_1, t_2, \ldots, t_i, \ldots, t_k\}$.

Taking Fig. 1 as an example, target AAVs are u_1, u_2, u_3. For convenience, in this paper, we assume that $t_{trans} = 0.1\,\text{s}$ and $t^s_{gen} = 0\,\text{s}, u_s \in U_{target}$. According to Algorithm 1, we can calculate the earliest delivery time and required hop count from each target AAV to all nodes in the network, as depicted in Tables II and III. The earliest delivery times from u_1, u_2, u_3 to g_0 are $4.1\,\text{s}, 4.1\,\text{s}, 6.1\,\text{s}$ respectively, that is, $t_1 = 4.1\,\text{s}, t_2 = 4.1\,\text{s}, t_3 = 6.1\,\text{s}$. Therefore, the user query delay is $t_d = \max\{4.1\,\text{s}, 4.1\,\text{s}, 6.1\,\text{s}\} = 6.1\,\text{s}$.

3) Minimum Forwarding Set and Set Cover Problem: After obtaining the user query delay (i.e., t_d), we use it as the delay

TABLE II
THE EARLIEST DELIVERY TIME

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	g_0
u_1	0	1.1	6.2	0.1	0.1	6.1	7.2	3.1	5.1	7.1	4.1
u_2	1.2	0	6.2	1.3	1.1	6.1	7.2	3.1	5.1	7.1	4.1
u_3	∞	6.2	0	∞	6.3	6.1	2.1	6.4	6.2	5.1	6.1

TABLE III
THE HOP COUNT OF THE SHORTEST PATH

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	g_0
u_1	0	2	4	1	1	3	5	2	2	4	3
u_2	2	0	2	3	1	1	3	4	2	2	5
u_3	∞	2	0	∞	3	1	1	4	2	2	3

constraint of all query results, because the query task becomes successful only when the last query result reaches the ground station, and the early arrival of other query results is meaningless. Moreover, on the basis of ensuring the user query delay (i.e., ensuring all query result packets can be delivered to the ground station before t_d), aggregating query results in the network can reduce data communication and energy consumption without the sacrifice of the query delay.

Therefore, in order to enable the query results of different target AAVs to be aggregated in the network as soon as possible, in this section, we first propose the concept of minimum forwarding set for TCG.

Definition 4.1 (Minimum Forwarding Set): u_i is a node in TCG, and needs to receive the query results from U^i_{target} before t^i_d, where U^i_{target} is the subset of U_{target} and t^i_d is the delay constraint of query results of U^i_{target} to u_i. Moreover, $U^i_{neighbor}$ is the set of neighbor nodes of u_i, if there exists a set $N_i \subseteq U^i_{neighbor}$, such that

1) The query results of all target AAVs in U^i_{target} can be delivered to u_i through the nodes in N_i before t^i_d;
2) If any element in N_i is removed, the query result of at least one target AAV in U^i_{target} can not be delivered to u_i through N_i before t^i_d,

then N_i is called the minimum forwarding set of u_i.

Thus, we formulate the aggregation processing of query results as the minimum forwarding set problem, namely recursively finding the minimum forwarding set from the ground station to all target AAVs. In order to solve the above problem, for any u_j which is a neighbor node of u_i, we define the deliverable target AAVs of u_j as the subset of U^i_{target} whose elements are the target AAVs that can be delivered to u_i through u_j before t^i_d, denoted as $U^j_{target} \subseteq U^i_{target}$. Then, based on the shortest path algorithm for TCG, for each neighbor node u_j of u_i, we can further calculate the deliverable target AAVs of u_j.

The process can be expressed as follows: First, for any $u_j \in U^i_{neighbor}$, we can obtain the earliest delivery time from $u_s \in U^i_{target}$ to u_j (i.e., $T(u_s, u_j), u_s \in U^i_{target}$) according to the shortest path algorithm. Then, since u_j is the neighbor node of u_i, the edge between u_i and u_j is denoted as $e_{ij} =< u_i, u_j, t^{ij}_{begin}, t^{ij}_{end} >$. Therefore, for each $u_s \in U^i_{target}$, if $\max\{T(u_s, u_j), t^{ij}_{begin}\} + t_{trans} \leq \min\{t^{ij}_{end}, t^i_d\}$, the query result of u_s can be delivered to u_i through u_j before t^i_d and we add u_s into U^j_{target}. Meanwhile, in order to ensure that all query results of U^j_{target} can be delivered to u_i through e_{ij} before t^i_d, all query results of U^j_{target} should be delivered to u_j before t^j_d. In other words, t^j_d is the delay constraint of all query results from U^j_{target} to u_j, and $t^j_d = \min\{t^i_d, t^{ij}_{end}\} - t_{trans}$. Algorithm 2 summarizes this process in detail.

Algorithm 2: DTU: The Deliverable Target AAVs Algorithm.

Input:
$u_i, U^i_{target}, t^i_d, U^i_{neighbor}, TCG$
Output:
$U^j_{target}, t^j_d, u_j \in U^i_{neighbor}$
for each $u_j \in U^i_{neighbor}$ **do**
  $U^j_{target} = \emptyset$
  for each $u_s \in U^i_{target}$ **do**
    if $\max\{T(u_s, u_j), t^{ij}_{begin}\} + t_{trans} \leq \min\{t^{ij}_{end}, t^i_d\}$ **then**
      $U^j_{target} = U^j_{target} \cup \{u_s\}$
    end if
  end for
  $t^j_d = \min\{t^i_d, t^{ij}_{end}\} - t_{trans}$
end for

For example, in Fig. 1, AAV u_1, u_2, u_3 are target AAVs of g_0 (i.e., $U^0_{target} = \{u_1, u_2, u_3\}$) and the delay constraint of g_0 is 6.1 s (i.e., $t^0_d = 6.1$ s), which means the query results of all target AAVs need to be delivered to g_0 before 6.1 s. AAV u_8, u_9, u_{10} are the neighbor nodes of g_0. For AAV u_8, the adjacent edge is $e_{80} =< u_8, g_0, 4\,\text{s}, 9\,\text{s} >$, and according to Table II, $T(u_1, u_8) = 3.1$ s, $T(u_2, u_8) = 3.1$ s, $T(u_3, u_8) = 6.4$ s. Then for target AAV u_1, since $\max\{3.1\,\text{s}, 4\,\text{s}\} + 0.1\,\text{s} < \min\{9\,\text{s}, 6.1\,\text{s}\}$, the query result of u_1 can be delivered to g_0 through u_8 before 6.1 s. Similarly, the query result of u_2 can be delivered to g_0 through u_8 before 6.1 s. However, for target AAV u_3, since $\max\{6.4\,\text{s}, 4\,\text{s}\} + 0.1\,\text{s} > \min\{9\,\text{s}, 6.1\,\text{s}\}$, the query result of u_3 can not be delivered to g_0 through u_8 before 6.1 s. Therefore, the deliverable target AAVs of u_8 are u_1, u_2, denoted as $U^8_{target} = \{u_1, u_2\}$. Similarly, $U^9_{target} = \{u_1, u_2\}$ and $U^{10}_{target} = \{u_3\}$. Meanwhile, $t^8_d = \min\{t^0_d, t^{80}_{end}\} - t_{trans} = \min\{6.1\,\text{s}, 9\,\text{s}\} - 0.1\,\text{s} = 6\,\text{s}$. Similarly, $t^9_d = 6$ s and $t^{10}_d = 6$ s.

After obtaining the deliverable target AAVs of each neighbor nodes of u_i (i.e., $U^j_{target}, u_j \in U^i_{neighbor}$), the above minimum forwarding set problem can be transformed into a set cover problem.

Definition 4.2 (Set Cover Problem): u_i is a node in TCG and its deliverable target AAVs are U^i_{target}. The neighbor nodes of u_i are $U^i_{neighbor} = \{u_1, u_2, \ldots, u_j\}$, and their corresponding deliverable target AAVs are $U = \{U^1_{target}, U^2_{target}, \ldots, U^j_{target}\}$. Moreover, $\bigcup_{k=1}^{j} U^k_{target} = U^i_{target}$. The set cover problem is to identify the smallest subset of U whose union equals U^i_{target}.

It is well known that the optimization version of set cover is NP-hard [87] and there are considerable efforts to solve this

Algorithm 3: MSC: Minimum Set Cover Algorithm.

Input:
$u_i, U^i_{target}, U^i_{neighbor}, DTU$
Output:
N_i

$U = U^i_{target}$
$N_i = \emptyset$
while $U \neq \emptyset$ **do**
  for each $u_j \in U^i_{neighbor}$ **do**
    $M \Leftarrow \arg\max_{u_j \in U^i_{neighbor}} |U^j_{target}|$
    $u_k \Leftarrow \arg\min_{u_l \in M} \max\{H(u_s, u_l), u_s \in U^l_{target}\}$
    $U = U - U^k_{target}$
    $N_i = N_i \cup \{U^k_{target}\}$
    for each $u_r \in U^i_{neighbor} - \{u_k\}$ **do**
      $U^r_{target} = U^r_{target} - U^k_{target}$
    end for
  end for
end while

problem [88]. For convenience, in this paper, we use the greedy algorithm to solve the above set cover problem. The main idea is to choose the set that contains the largest number of uncovered elements at each stage. When multiple largest subsets exist, the subset exhibiting the minimum maximum hop count from its deliverable target AAV is selected. This criterion aims to generate shorter aggregate paths and reduce relay node load. Moreover, in each round, we remove the covered elements from the unselected subsets to ensure that each element appears in only one of the smallest subsets in the end. This is because the query result of each target AAV only need to be aggregated at most once. The repeated appearance of elements will cause the query results of target AAVs to be aggregated and delivered multiple times, which will lead to unnecessary consumption of energy and bandwidth resources. Algorithm 3 summarizes this process in detail.

For instance, $U^0_{target} = \{u_1, u_2, u_3\} = U$ and AAV u_8, u_9, u_{10} are neighbor nodes of g_0 where $U^8_{target} = \{u_1, u_2\}$, $U^9_{target} = \{u_1, u_2\}$, $U^{10}_{target} = \{u_3\}$. First, $N_0 = \emptyset$, since $|U^8_{target}| = |U^9_{target}| > |U^{10}_{target}|$, and according to Table III, $\max\{H(u_1, u_8), H(u_2, u_8)\} = \max\{2, 4\} = 4$, $\max\{H(u_1, u_9), H(u_2, u_9)\} = \max\{2, 2\} = 2 < 4$, we choose U^9_{target} and add it into the smallest subset, $N_0 = \{U^9_{target} = \{u_1, u_2\}\}$. Then, we delete the corresponding covered elements from U, U^8_{target} and U^{10}_{target}, namely $U = U - U^9_{target} = \{u_3\}$, $U^8_{target} = U^8_{target} - U^9_{target} = \emptyset$ and $U^{10}_{target} = U^{10}_{target} - U^9_{target} = \{u_3\}$. Next, since $|U^{10}_{target}| > |U^8_{target}|$, we choose U^{10}_{target} and add it into the smallest subset, $N_0 = \{U^9_{target} = \{u_1, u_2\}, U^{10}_{target} = \{u_3\}\}$. Meanwhile, $U = U - U^{10}_{target} = \emptyset$ and $U^8_{target} = U^8_{target} - U^{10}_{target} = \emptyset$. Since $U = \emptyset$, the process is done and the smallest subset of U^0_{target} is $\{U^9_{target} = \{u_1, u_2\}, U^{10}_{target} = \{u_3\}\}$.

4) Build the Spatial-Temporal Aggregation Tree: To ensure query results to be aggregated within the network as soon as possible, we propose the concept of minimum forwarding set and transform it into a set cover problem. In addition, the aggregation processing of query results can be transformed into recursively finding the minimum forwarding set from the ground station g_0 to all target AAVs U_{target}, thus constructing a Spatial-Temporal Aggregation Tree (STAT). In this section, we describe the construction procedure of the spatial-temporal aggregation tree. We use a queue Q to assist the insertion of nodes in STAT. Meanwhile, we use a set F to keep track of the nodes that have been added into STAT. The algorithm is expressed as follows:

1) Add the ground station g_0 into Q and insert g_0 into STAT as the root node. Meanwhile, we initialize set $F = \{g_0\}$.
2) Take the first element in the queue Q, denoted as u_i, and obtain its neighbor nodes $U^i_{neighbor}$. Note that $U^i_{neighbor}$ does not contain elements in set F, that is, nodes that have been added into STAT will not be considered as neighbor nodes of any AAV.
3) For each neighbor node of u_i, denoted as u_j, if $u_j \in U^i_{target}$, add u_j into STAT as the child node of u_i and add it into set F. Meanwhile, we delete u_j from U^i_{target} and $U^i_{neighbor}$. This means the aggregation path from the target AAV u_j to the ground station g_0 is found.
4) Calculate the deliverable target AAVs of each $u_j \in U^i_{neighbor}$, and then find the minimum forwarding set of u_i, that is, the minimum subset of U^i_{target}, denoted as N_i.
5) For each u_k whose $U^k_{target} \in N_i$, add u_k into the queue Q and insert it into STAT as a child node of u_i. Meanwhile, add u_k into set F, which means u_k has been added into STAT.
6) If the queue Q is empty or all target AAVs have been added into STAT (i.e., $U_{target} \subseteq F$), the construction process of STAT is done, which means the aggregation paths of all target AAVs have been found; otherwise, go to Step 2.

After describing the main stages of the construction of the spatial-temporal aggregation tree, we summarize it in Algorithm 4.

In Fig. 5, we build a spatial-temporal aggregation tree for the example in Fig. 1 to illustrate the construction process. First, g_0 is added into Q and inserted into STAT as the root node.

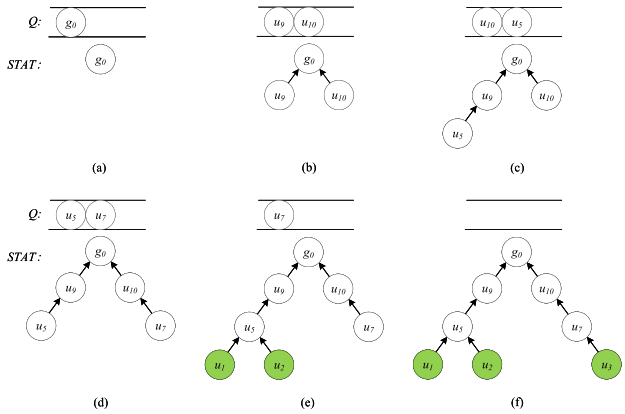

Fig. 5. The construction procedure of the spatial-temporal aggregation tree.

Algorithm 4: STA-Tree.

Input:
$g_0, U_{target}, t_d, TCG$
Output:
$STAT$

Q.push(g_0)
STA.add($g_0, null$)
$F = \{g_0\}$
while !Q.isEmpty() & $U_{target} \not\subset F$ **do**
  $u_i \Leftarrow Q$.poll()
  $U^i_{neighbor} = U^i_{neighbor} - F$
  for each $u_j \in U^i_{neighbor}$ **do**
    if $u_j \in U^i_{target}$ **then**
      STA.add(u_j, u_i)
      $F = F \cup \{u_j\}$
      $U^i_{target} = U^i_{target} - \{u_j\}$
      $U^i_{neighbor} = U^i_{neighbor} - \{u_j\}$
    end if
  end for
  $U_{delt} \Leftarrow DTU(u_i, U^i_{target}, t^i_d, U^i_{neighbor}, TCG)$
  $N_i \Leftarrow MSC(u_i, U^i_{target}, U^i_{neighbor}, U_{delt})$
  for each $u_k, U^k_{target} \in N_i$ **do**
    Q.push(u_k)
    STA.add(u_k, u_i)
    $F = F \cup \{u_k\}$
  end for
end while

$F = \{g_0\}$. We take out g_0 from Q, as mentioned above, since $U_{target} = \{u_1, u_2, u_3\}$ and the minimum forwarding set of g_0 is $N_0 = \{U^9_{target} = \{u_1, u_2\}, U^{10}_{target} = \{u_3\}\}$. We add u_9 and u_{10} into Q and insert them into STAT as the child nodes of g_0. Meanwhile, $F = \{g_0, u_9, u_{10}\}$.

Then, as shown in Fig. 5(c), we take out u_9 from Q. Since $g_0 \in F$, the neighbor nodes of u_9 are u_5 and u_6. For AAV u_5, $\max\{T(u_1, u_5), t^{59}_{begin}\} + t_{trans} = \max\{0.1\,\text{s}, 5\,\text{s}\} + 0.1\,\text{s} = 5.1\,\text{s}$, $\max\{T(u_2, u_5), t^{59}_{begin}\} + t_{trans} = \max\{1.1\,\text{s}, 5\,\text{s}\} + 0.1\,\text{s} = 5.1\,\text{s}$ and $\min\{t^{59}_{end}, t^9_d\} = \min\{8\,\text{s}, 6\,\text{s}\} = 6\,\text{s}$. Since $5.1\,\text{s} < 6\,\text{s}$, $U^5_{target} = \{u_1, u_2\}$. Similarly, for AAV u_6, $\max\{T(u_1, u_6), t^{69}_{begin}\} + t_{trans} = \max\{6.1\,\text{s}, 6\,\text{s}\} + 0.1\,\text{s} = 6.2\,\text{s}$, $\max\{T(u_2, u_6), t^{69}_{begin}\} + t_{trans} = \max\{6.1\,\text{s}, 6\,\text{s}\} + 0.1\,\text{s} = 6.2\,\text{s}$ and $\min\{t^{69}_{end}, t^9_d\} = \min\{10\,\text{s}, 6\,\text{s}\} = 6\,\text{s}$. Since $6.2\,\text{s} > 6\,\text{s}$, $U^6_{target} = \emptyset$. Therefore, the minimum forwarding set of u_9 is $N_9 = \{U^5_{target} = \{u_1, u_2\}\}$. We add u_5 into STAT as the child node of u_9 and add it into Q. Meanwhile, $F = \{g_0, u_9, u_{10}, u_5\}$.

Next, as Fig. 5(d) shows, we take out u_{10} from Q. The neighbor nodes of u_{10} are u_6 and u_7. For AAV u_6, $\max\{T(u_3, u_6), t^{610}_{begin}\} + t_{trans} = \max\{6.1\,\text{s}, 7\,\text{s}\} + 0.1\,\text{s} = 7.1\,\text{s}$ and $\min\{t^{610}_{end}, t^{10}_d\} = \min\{8\,\text{s}, 6\,\text{s}\} = 6\,\text{s}$. Since $7.1\,\text{s} > 6\,\text{s}$, $U^6_{target} = \emptyset$. For AAV u_7, $\max\{T(u_3, u_7), t^{710}_{begin}\} + t_{trans} = \max\{2.1\,\text{s}, 5\,\text{s}\} + 0.1\,\text{s} = 5.1\,\text{s}$ and $\min\{t^{710}_{end}, t^{10}_d\} = \min\{9\,\text{s}, 6\,\text{s}\} = 6\,\text{s}$. Since $5.1\,\text{s} < 6\,\text{s}$, $U^7_{target} = \{u_3\}$. Therefore, the minimum forwarding set of u_{10} is $N_{10} = \{U^7_{target} = \{u_3\}\}$. We add u_7 into STAT as the child node of u_{10} and add it into Q. Meanwhile, $F = \{g_0, u_9, u_{10}, u_5, u_7\}$.

Then, as depicted in Fig. 5(e), we take out u_5 from Q. The neighbor nodes of u_5 are u_1, u_2 and u_8. Since u_1 and u_2 are both target AAVs, we directly insert them into STAT as the child nodes of u_5. Meanwhile, we update $U^5_{target} = U^5_{target} - \{u_1, u_2\} = \emptyset$ and $F = \{g_0, u_9, u_{10}, u_5, u_7, u_1, u_2\}$. Finally, we take out u_7 from Q. Since u_3 is the neighbor nodes of u_7, we insert it as the child node of u_7. Meanwhile, we update $U^7_{target} = U^7_{target} - \{u_3\} = \emptyset$ and $F = \{g_0, u_9, u_{10}, u_5, u_7, u_1, u_2, u_3\}$. Since $U_{target} \subseteq F$, the construction procedure of STAT is done.

Based on the constructed spatial-temporal aggregation tree, we can obtain the aggregated routing paths of all target AAVs. In STAT, the root node is the ground station, and except for the leaf nodes, each node will aggregate query results of all its child nodes, and then forward the aggregated result to its parent node. For example, as shown in Fig. 5(f), u_1 and u_2 send their respective query results to u_5, then u_5 aggregates these query results and forwards the aggregated result to u_9, and finally u_9 delivers it to g_0.

5) Generalized Scenario Expansion: In the previous section, we take the minimum delay required to complete the query task (i.e., the earliest delivery time of the last arrival packet) as the user query delay. As shown in the above example, $t_d = \max\{t_1, t_2, t_3\} = \max\{4.1\,\text{s}, 4.1\,\text{s}, 6.1\,\text{s}\} = 6.1\,\text{s}$. However, in many practical application scenarios, the requirement of the user query delay is not so tight, that is, the query delay constraint given by the user is typically greater than the minimum delay required for a query task. Therefore, in order to extend to more general scenarios, in this section, we introduce the slack variable $\zeta \geq 0$ to relax the user query delay, namely

$$t_{nd} = t_d + \zeta \tag{3}$$

The parameter ζ is a user-definable trade-off factor. It balances query latency against energy consumption and can be adjusted based on user preferences or specific application requirements. For example, let $\zeta = 1\,\text{s}$, then $t_{nd} = 6.1\,\text{s} + 1\,\text{s} = 7.1\,\text{s}$. According to Table II, at this time, $U^8_{target} = \{u_1, u_2, u_3\}$, $U^9_{target} = \{u_1, u_2, u_3\}$ and $U^{10}_{target} = \{u_3\}$. Since $|U^8_{target}| = |U^9_{target}| = 3$, and $\max\{H(u_1, u_8), H(u_2, u_8), H(u_3, u_8)\} = \max\{2, 4, 4\} = 4$, $\max\{H(u_1, u_9), H(u_2, u_9), H(u_3, u_9)\} = \max\{2, 2, 2\} = 2$, the minimum forwarding set of g_0 is $N_0 = \{U^9_{target} = \{u_1, u_2, u_3\}\}$. Similarly, according to Algorithm 4, we can obtain the final spatial-temporal aggregation tree, as shown in Fig. 6(a), and the corresponding transmission scheme is shown in Fig. 6(b). The query results of u_1 and u_2 will be aggregated at u_5 first, and then further aggregated at u_9 with the query result of u_3. Finally, the aggregated query result will be delivered to g_0. The total energy consumption is $6E_m$, which is further reduced. By introducing the slack variable, we demonstrate that ESTA can work well with generalized scenarios.

E. Computational Complexity Analysis

In this section, we provide a formal analysis of the computational complexity of the ESTA algorithm to evaluate its

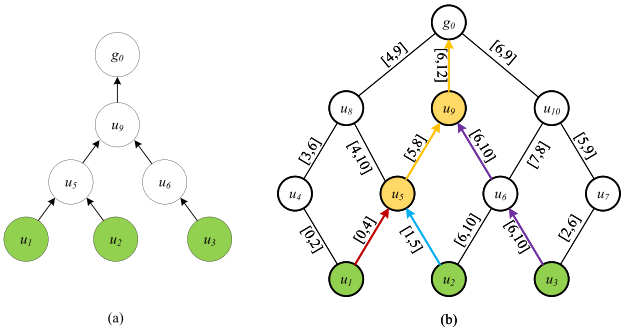

Fig. 6. The generalized scenario expansion when $t_{nd} = 7.1$ s. (a) The spatial-temporal aggregation tree. (b) The corresponding transmission scheme.

computational overhead. The algorithm's complexity is primarily determined by two core components: the execution of the Shortest Path Algorithm (SPA) to establish the user query delay, and the recursive construction of the Spatial-Temporal Aggregation Tree (STAT).

Shortest Path Algorithm (SPA): The first phase of ESTA involves determining the minimum query response delay by finding the earliest delivery time from each of the k target AAVs to the ground station. This is achieved by executing our Dijkstra-like SPA on the Topology Change Graph (TCG). Let V be the number of nodes (AAVs and the ground station) and E be the number of time-dependent edges (communication opportunities) in the TCG. For a single source, the complexity of our SPA, using a standard priority queue implementation (e.g., a binary heap), is $O(E + V \log V)$. Since this process is repeated for each of the k target AAVs, the total complexity for this phase is $O(k \cdot (E + V \log V))$.

STAT Construction: The second phase involves building the Spatial-Temporal Aggregation Tree (STAT) by recursively solving a series of set cover problems. The core of this process is a greedy heuristic that, at each node u_i in the tree's construction, selects a next-hop neighbor that can provide a path for the largest number of yet-uncovered target AAVs. Let d_{max} be the maximum degree of any node in the TCG (i.e., the maximum number of neighbors) and k be the total number of target AAVs. For a given node u_i, the greedy selection process involves iterating through its neighbors and checking their coverage, resulting in a computational cost of approximately $O(d_{max} \cdot k)$ for that step. As this procedure is applied recursively to build the tree, and the tree can have at most V nodes, the overall complexity for constructing the entire STAT is bounded by $O(V \cdot d_{max} \cdot k)$.

Therefore, the overall computational complexity of ESTA is manageable and primarily polynomial in terms of the network size and the number of target AAVs, ensuring its feasibility for practical deployment.

V. Performance Evaluation

In this section, we provide a comprehensive experimental design and performance analysis of ESTA on the Opportunistic Network Environment (ONE) simulator [18].

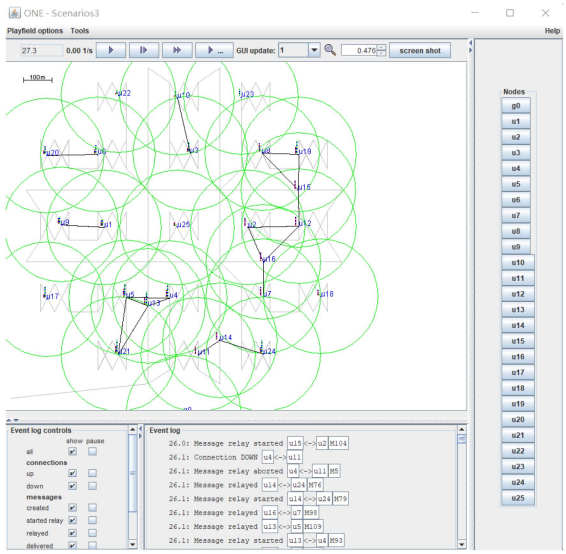

Fig. 7. An experimental screenshot of the ONE.

TABLE IV
Simulation Settings

Parameter	Default value
Simulation Area (m^2)	1200 × 1300
Simulation Time (s)	1000
Number of AAVs	25
Mobility Model	MapRouteMovement
AAV Speed (m/s)	10
Communication Range (m)	200
Query Result Packet Size (KB)	10
Message Time-to-Live (s)	200
Slack Variable (s)	0
Link Throughput (KB/s)	125
Query Time Period (s)	30
Query Region Ratio	10%
Query Number	1000

A. Simulation Setup and Scenarios

Referring to the simulation scenarios in [51], [76], [79], we design a simulation scenario inspired by the forest monitoring missions. One stationary ground station is placed together with 21 search AAVs and 4 ferry AAVs. Each search AAV is responsible for a 200 m × 200 m region and uses a typical zigzag movement pattern to efficiently cover the region, and each ferry AAV flies forth and back along the specified trajectory to assist search AAVs with message delivery. Note that the trajectories of all AAVs are pre-planned. That is, all nodes utilize the "MapRouteMovement" model, causing them to move along predefined map paths. The experimental screenshot is shown in Fig. 7 and Table IV summarizes the detailed experimental parameters.

Moreover, we use the energy model proposed in [51], [89]. This model is a probabilistic wireless link quality model based on the empirically-validated log-normal shadowing model, which captures both distance-dependent path loss and random signal fluctuations from multi-path effects. This realistic portrayal of

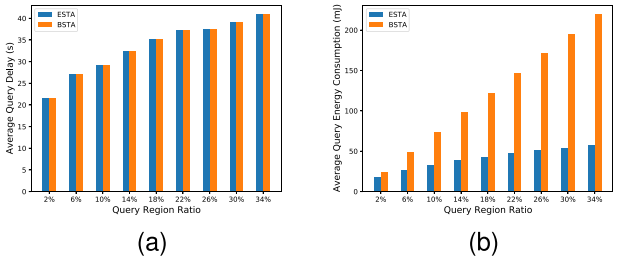

Fig. 8. The impact of the query region ratio. (a) Average query delay. (b) Average query energy consumption.

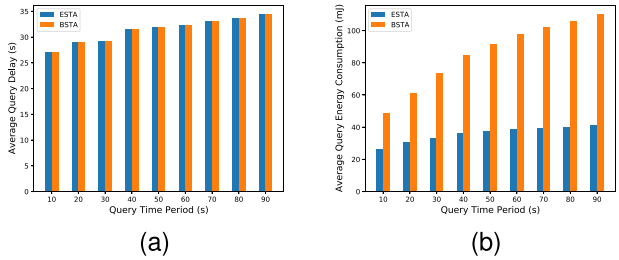

Fig. 9. The impact of the query time period. (a) Average query delay. (b) Average query energy consumption.

link behavior is essential for analyzing and designing robust network protocols.

In this paper, we use the following metrics to evaluate the performance of the spatial-temporal range aggregation query algorithms.

- *Query delay:* The query delay represents the time it takes to complete a query task, that is, the time for all query results to be successfully delivered to the ground station.
- *Query Energy consumption:* The query energy consumption is the energy consumed by all query result packets to be delivered to the ground station in a query task.

In addition, as far as we know, we are the first to research spatial-temporal range aggregation query processing in AAV networks. Therefore, we propose a Baseline Spatial-Temporal range Aggregation query processing (BSTA) algorithm and then compare it with ESTA. The BSTA consists of two main stages:

1) According to the constructed TCG and proposed SPA, we can obtain the shortest paths for query results of all target AAVs. Then, each target AAV delivers its query result along the shortest path to the ground station.
2) In the process of query result delivery, if a AAV stores and carries the query results of multiple target AAVs at the same time, it will aggregate these query results. Then the aggregated result will be forwarded to the ground station along the remaining shortest path.

For example, as shown in Fig. 1, target AAVs are u_1, u_2, u_3, and as mentioned above, the corresponding shortest paths are pa_1, pa_2, pa_3. According to the shortest paths, as shown in Fig. 4(a), AAV u_4 will receive the query results of u_1 and u_2 at 0.1 s and 1.3 s. Then, u_4 will store and carry them until 3 s, and forward them to u_8 at 3 s. Therefore, u_4 aggregates the query results of u_1 and u_2, and then forwards the aggregated result to u_8 at 3 s. Through the way of piggybacking in-network aggregation, BSTA reduces energy and bandwidth consumption without sacrificing the query delay.

B. The Impact of Different Variables

1) Impact of the Query Region Ratio: As Fig. 8(a) shows, as the query region ratio increases, the query delay increases. This is because the query delay is the time it takes for all query results to be delivered to the ground station, which is essentially determined by the arrival time of the last query result reaching the ground station. When the query region ratio becomes larger, the expectation of the delivery delay of each query result remains unchanged, but the variance becomes larger, resulting in a larger query delay. On the other hand, it is obvious that the query delays of ESTA and BSTA are the same. In BSTA, the query result of each target AAV is delivered along its own shortest path, while in ESTA, the earliest delivery time of the last arrival query result is the delay constraint, so the delivery delays of the two algorithms are the same.

Furthermore, with the increase of the query region, the query energy consumption of both BSTA and ESTA increases, however, the increase rate of ESTA is much smaller than that of BSTA, which means that ESTA can save more energy than BSTA with the increase of the query region. As depicted in Fig. 8(b), when the query region ratio becomes 34%, the query energy consumption of ESTA is 26.2% of that of BSTA. This is because by constructing a spatio-temporal aggregation tree, ESTA makes an overall plan for the query results of all target AAVs and performs in-network aggregation as early as possible, which can effectively reduce the query energy consumption.

2) Impact of the Query Time Period: As shown in Fig. 9(a), with the increase of the query time period, the average query delay of both ESTA and BSTA shows an overall upward trend. The reason is that the larger the query time period, the more target AAVs which store and carry the qualified data. Therefore, the increase of the query time period leads to the increase of the query delay, which is similar to the impact of the query region. However, the trajectories of AAVs are pre-planned, and AAVs cover the mission coverage region in an efficient pattern, rather than flying in a disorder way. Therefore, with the increase of the query time period, the number of target AAVs that satisfy the spatial-temporal constraint tends to be stable, and the query delay also tends to be stable.

Moreover, the query energy consumptions of both ESTA and BSTA increase with the query time period due to more query results which need to be delivered to the ground station. ESTA can efficiently reduce the energy consumption on the basis of ensuring query delay through the in-network aggregation, especially when the query time period is long. When the query time period is 90 s, the query energy consumption of ESTA is only 37.2% of that of BSTA.

3) Impact of the Query Number: In this experiment, different numbers of query tasks are sent out at a constant speed within 500 seconds. As depicted in Fig. 10(a), the query delays of both ESTA and BSTA do not fluctuate significantly with the query

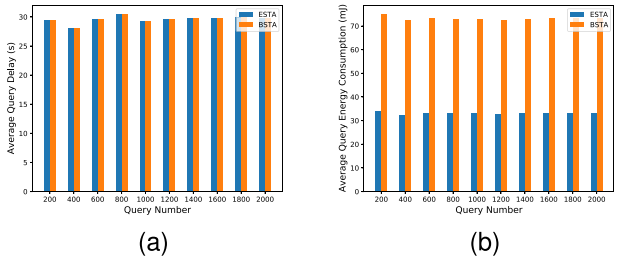

Fig. 10. The impact of the query number. (a) Average query delay. (b) Average query energy consumption.

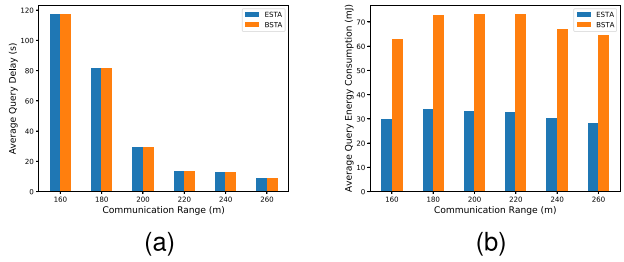

Fig. 12. The impact of the communication range. (a) Average query delay. (b) Average query energy consumption.

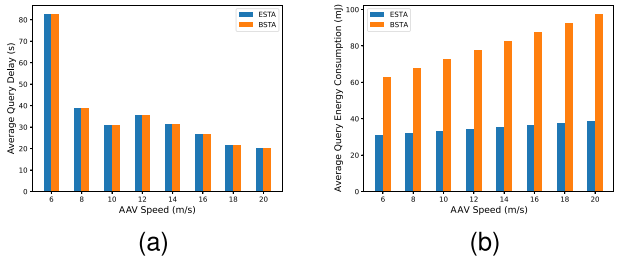

Fig. 11. The impact of the AAV speed. (a) Average query delay. (b) Average query energy consumption.

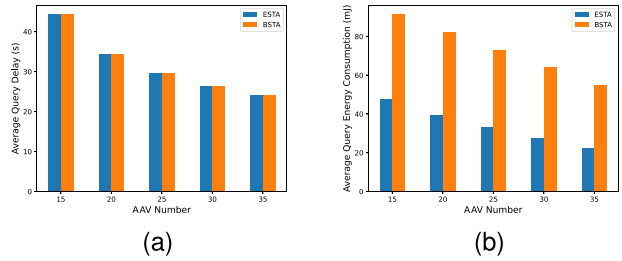

Fig. 13. The impact of the AAV density. (a) Average query delay. (b) Average query energy consumption.

number. This is because both ESTA and BSTA use pre-planned trajectory information to calculate routing paths in advance, so they can adapt to various AAV network environments.

At the same time, we can find that ESTA can effectively reduce the query energy consumption through in-network aggregation. As Fig. 10(b) shows, the query energy consumption of ESTA is stable at 45.2% of that of BSTA, demonstrating the efficiency and superiority of ESTA.

4) Impact of the AAV Speed: As shown in Fig. 11(a), the average query delay of both ESTA and BSTA generally decreases as the AAV speed increases. The reason is that as the AAV speed increases, there are more communication opportunities between AAVs, and less storage-and-carry time required by AAVs, so that AAVs can deliver query results to the ground station with less delivery delay.

As shown in Fig. 11(b), the average query energy consumptions of both ESTA and BSTA increase slowly as the AAV speed increases. This is because under the same spatial-temporal constraint, due to the increase of the AAV speed, the number of target AAVs increases, resulting in an increase of the query energy consumption.

Meanwhile, it is worth noting that the average query energy consumption of ESTA is only 39.8% of that of BSTA. This is because ESTA further performs in-network aggregation on the basis of BSTA, thereby further reducing the query energy consumption.

5) Impact of the Communication Range: In this experiment, we investigate the effect of the communication range, which determines the sparsity and connectivity of the AAV network. As shown in Fig. 12(a), as the communication range increases, the query delay decreases rapidly. The reason is that with the increase of the communication range, both ESTA and BSTA can find the transmission paths which have earlier delivery time for query results, thereby reducing the query delay. Meanwhile, the query energy consumption of ESTA is about 45% of that of BSTA. Furthermore, the better the connectivity of the AAV network, the better the effect of the in-network aggregation.

6) Impact of the AAV Density: This experiment evaluates the algorithms' performance under varying network densities by adjusting the number of AAVs from 15 (sparse) to 35 (dense) within a fixed area. As shown in Fig. 13(a), the average query delay for both ESTA and BSTA decreases significantly as the AAV density increases. In sparse networks with few AAVs, the lack of consistent connectivity forces a heavy reliance on the time-consuming store-carry-forward mechanism, resulting in high delays. As the network becomes denser, communication opportunities increase, enabling efficient multi-hop forwarding and drastically reducing the query delay. The rate of improvement diminishes as the network approaches a well-connected state, demonstrating a pattern of diminishing returns.

Fig. 13(b) illustrates the impact of AAV density on energy consumption. Interestingly, the energy consumption for both algorithms decreases as the network becomes denser. This is because higher connectivity allows for the discovery of more efficient, shorter-hop routing paths, reducing the overall transmission cost compared to the less efficient paths in a sparse network. However, the results clearly show that ESTA capitalizes on this increased connectivity far more effectively than BSTA. With more nodes available, ESTA can construct a more optimal Spatial-Temporal Aggregation Tree by selecting better-positioned aggregation points. This superior optimization strategy leads to a widening efficiency gap as density increases; for example, the energy consumption of ESTA is 52.0% of BSTA's at a density of 15 AAVs, but this ratio improves to just 41.0% at

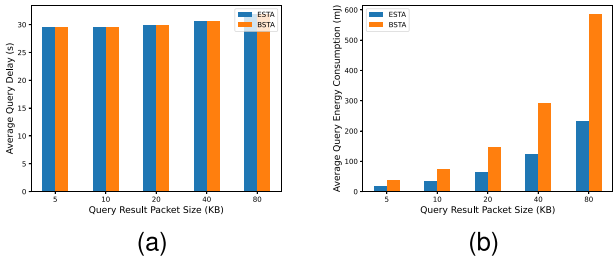

Fig. 14. The impact of the data load. (a) Average query delay. (b) Average query energy consumption.

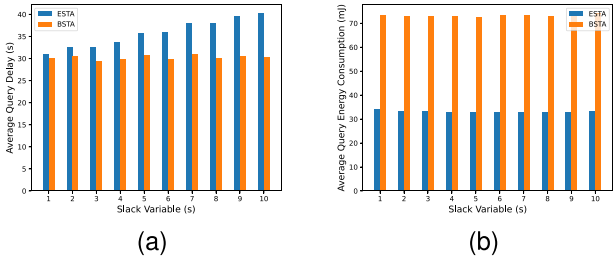

Fig. 15. The impact of the slack variable. (a) Average query delay. (b) Average query energy consumption.

35 AAVs. This demonstrates ESTA's robustness and its advanced capability to leverage network resources to save energy.

7) Impact of the Data Load: In this experiment, we investigate the performance of the algorithms under varying data loads by adjusting the Query Result Packet Size from 5 KB to 80 KB. As shown in Fig. 14(a), the average query delay for both ESTA and BSTA exhibits a slight and steady increase as the packet size grows. This is because a larger packet requires more time for transmission at each hop. However, since the total delay in AAV networks is often dominated by store-and-carry waiting times due to intermittent connectivity, the increase in transmission time has a limited, linear impact on the overall query delay. As ESTA's optimization is designed to reduce energy consumption without sacrificing latency, the query delays for both algorithms remain identical.

Fig. 14(b) illustrates the impact of data load on energy consumption. The query energy consumption for both algorithms increases significantly with the packet size, which is expected as transmitting larger packets requires more energy. However, the key observation is that the energy efficiency gap between ESTA and BSTA widens dramatically as the data load increases. While BSTA's energy consumption scales linearly with the packet size, ESTA's in-network aggregation strategy becomes far more effective at mitigating this increase. For instance, at a data load of 80 KB, the energy consumption of ESTA is only about 40.0% of BSTA's. This demonstrates that ESTA is not only efficient but also highly robust, showcasing superior energy-saving advantages, especially in data-intensive application scenarios.

8) Impact of the Slack Variable: As shown in Fig. 15(a), with the increase of the slack variable, the query delay of ESTA also increase gradually, because the introduction of the slack variable

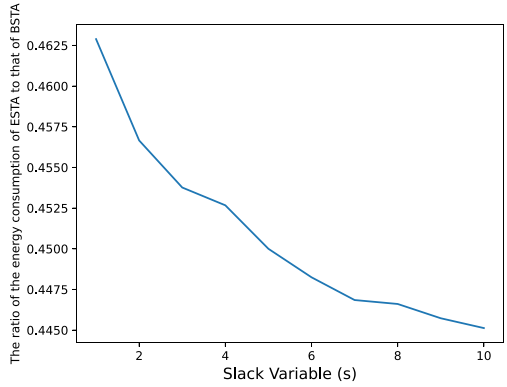

Fig. 16. The impact of the slack variable on the ratio of the energy consumption of ESTA to that of BSTA.

makes ESTA relax the requirement of the query delay constraint in order to further aggregate query results within the network. Furthermore, in order to more intuitively reflect the impact of the slack variable on the query energy consumption, Fig. 16 shows the ratio of the energy consumption of ESTA to that of BSTA. It can be seen that with the increase of the slack variable, the ratio decreases, which indicates that the introduction of the slack variable can better optimize the spatial-temporal aggregation tree, so that ESTA can further reduce the energy consumption.

C. Comparison to Current State-of-The-Art

In this section, we present a comprehensive performance evaluation of our proposed ESTA and BSTA methods against the current state-of-the-art. The comparison is conducted against six representative methods from two highly relevant research domains: Data Aggregation Technologies and Routing Protocols for AAV Networks. As detailed in Section II, the selected methods encompass diverse paradigms: tree-based (Vo et al. [31]), cluster-based (Mohseni et al. [13]), and machine-learning-based (Zhang et al. [44]) approaches for data aggregation; and topology-based (Zhai et al. [51]), geographic (Singh et al. [60]), and store-carry-forward-based (Guo et al. [67]) protocols for AAV routing.

To ensure a fair and rigorous comparison, we adhered to the original parameter settings of each method where possible, with a few necessary adaptations for the AAV network context. For the data aggregation methods, the message delay constraint was set to the message Time-To-Live (TTL) of 200 s to prioritize the query success rate. For the Q-learning algorithm in Zhang et al. [44], we trained its routing model using data generated from random initializations within our simulated AAV network environment. For the AAV routing protocols, the delay constraint for Zhai et al. [51] was set to 35 s to facilitate a direct comparison of energy consumption under similar latency conditions. For Singh et al. [60], the link quality was set based on our energy model. For Guo et al. [67], its "factor" parameter was set to 1 to mandate the delivery of all query messages. To provide a holistic performance assessment, we introduced the Query

TABLE V
COMPARISON TO CURRENT STATE-OF-THE-ART

Type	Method	Query Success Rate	Query Delay (s)	Energy Consumption (mJ)
Date Aggregation for Static Networks	Vo et al. [31]	28.45%	159.65	223.72
	Mohseni et al. [13]	30.21%	151.47	202.18
	Zhang et al. [44]	35.15%	125.19	187.53
Routing Protocol for AAV Networks	Zhai et al. [51]	**100%**	33.57	87.88
	Singh et al. [60]	94.36%	65.39	131.51
	Guo et al. [67]	**100%**	30.24	197.82
This work	BSTA	**100%**	**29.64**	73.23
	ESTA	**100%**	**29.64**	**33.19**

Success Rate, defined as the ratio of successfully completed aggregation queries to the total number of initiated queries. An aggregation query is deemed successful only if all corresponding query results are received by the sink node.

The experimental results, as summarized in Table V, demonstrate the unequivocal superiority of our proposed methods. Both ESTA and BSTA achieve a 100% Query Success Rate while simultaneously realizing the lowest query delay and energy consumption among all compared approaches. This outstanding performance is primarily attributed to ESTA's novel architecture, which constructs a Topology Change Graph (TCG) to accurately model the network's dynamic nature and subsequently builds a Spatio-Temporal Aggregation Tree (STAT) to guide efficient, aggregation-aware message transmission.

A detailed analysis reveals the limitations of existing technologies in dynamic AAV environments. The data aggregation methods, originally designed for static wireless networks, perform poorly. Despite relaxing the delay constraint, their query success rates remained below 40%, with query delays exceeding four times that of ESTA and consuming substantially more energy. The tree-based approach of Vo et al. [31] frequently experiences routing failures as the rigid tree structures cannot withstand the constant link breakages in a mobile AAV network. In contrast, ESTA's TCG proactively adapts to such topological changes. The cluster-based method of Mohseni et al. [13] suffers from severe instability. The high mobility of AAVs leads to frequent re-clustering and changes in cluster leadership, incurring excessive overhead that results in extremely low success rates and high latency and energy costs. The reinforcement learning approach of Zhang et al. [44] shows marginally better performance, as it was trained on AAV network data. However, it still fails to match the efficiency and reliability of our deterministic, topology-aware approach.

The compared AAV routing protocols achieve higher query success rates but are suboptimal in terms of efficiency due to their lack of an in-network aggregation mechanism. Zhai et al. [51] achieves a respectable performance within its category by dynamically adjusting transmission power. Nevertheless, its energy consumption is nearly three times that of ESTA and also higher than our baseline BSTA. Singh et al. [60] exhibits query failures because its geographic forwarding strategy is prone to getting trapped in local optima. Its reliance on geographic progress for relay selection also leads to inefficient paths and higher energy consumption. Guo et al. [67] leverages an optimized flooding-based multicast approach to match the low query delay of ESTA. Nevertheless, its reliance on this broadcast-like foundation results in substantial transmission redundancy and energy consumption that is roughly six times greater than that of ESTA.

VI. LIMITATIONS AND FUTURE WORKS

While this paper introduces ESTA as a novel and efficient algorithm for spatial-temporal range aggregation queries in AAV networks, demonstrating significant performance gains, we acknowledge several limitations that also open promising avenues for future research.

Dependence on Pre-planned Trajectories and Network Dynamics: A core assumption of ESTA is the availability of pre-planned AAV trajectories, which are used to construct the Topology Change Graph (TCG). While applicable in many mission-based scenarios, this assumption limits the algorithm's adaptability in highly unpredictable environments where AAVs must frequently deviate from their paths due to unforeseen obstacles, adverse weather, or dynamic mission retasking.

Future research could focus on developing a more dynamic version of ESTA that can handle real-time trajectory adjustments. This could involve integrating probabilistic trajectory prediction models to anticipate near-future connectivity changes or designing a hybrid framework that uses pre-planned routes as a baseline but employs a reactive mechanism to locally update the TCG and repair the Spatial-Temporal Aggregation Tree (STAT) in response to unexpected deviations.

Computational Complexity and Scalability: The ESTA algorithm relies on centralized computations, including the execution of a shortest path algorithm for the entire network and the recursive solving of the set cover problem to build the STAT. While the simulations with 25 AAVs show excellent performance, the computational overhead may become a bottleneck in large-scale AAV swarms comprising hundreds or thousands of nodes. The complexity of generating a global TCG and solving optimization problems could challenge the resource constraints of the ground station.

To enhance scalability, a decentralized or hierarchical version of ESTA could be investigated. The network could be partitioned into dynamic clusters, with each cluster constructing a local STAT. An inter-cluster aggregation strategy could then be overlaid, creating a hierarchical aggregation structure that distributes the computational load across the network, thereby improving scalability and resilience.

Simplifications in System and Communication Models: ESTA assumes that query and aggregated result packets are of a uniform size and focuses on distributive aggregation functions like MAX. The model also abstracts multiple intermittent connections into a single continuous communication window. These simplifications, while pragmatic, do not capture the full complexity of real-world scenarios.

Future iterations of the algorithm could be enhanced to support heterogeneous data types and variable packet sizes, which would require a more nuanced energy consumption model. Research should also explore strategies for handling more complex, non-distributive aggregation functions (e.g., median, variance), which may necessitate modifications to the aggregation logic within the STAT. Enhancing the communication model to account for MAC-layer contention and interference would also provide a more realistic performance evaluation, especially in dense network deployments.

VII. Conclusion

In this paper, we study the spatial-temporal range aggregation query in AAV networks, and then propose an Efficient Spatial-Temporal range Aggregation query processing (ESTA) algorithm. First, ESTA utilizes the pre-planned trajectory information to construct a topology change graph (TCG) in order to reflect the communication windows between AAVs. Then, an efficient shortest path algorithm is proposed based on which the user query delay can be obtained. Next, ESTA transforms the aggregation processing of query results into recursively solving the set cover problem, thus constructing a spatial-temporal aggregation tree (STAT). With the constructed STAT, an efficient in-network aggregation routing path for query results without the sacrifice of the user query delay can be found. Extensive experimental results show that ESTA outperforms the baseline algorithm in terms of time delay and energy consumption.

References

[1] P. Cao et al., "Computational intelligence algorithms for UAV swarm networking and collaboration: A comprehensive survey and future directions," *IEEE Commun. Surv. Tuts.*, vol. 26, no. 4, pp. 2684–2728, Fourthquarter 2024.

[2] G. Sun et al., "Aerial reliable collaborative communications for terrestrial mobile users via evolutionary multi-objective deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 24, no. 7, pp. 5731–5748, Jul. 2025.

[3] J. Li, G. Sun, L. Duan, and Q. Wu, "Multi-objective optimization for UAV swarm-assisted IoT with virtual antenna arrays," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4890–4907, May 2024.

[4] X. Li, L. Liu, L. Wang, J. Xi, J. Peng, and J. Meng, "Trajectory-aware spatio-temporal range query processing for unmanned aerial vehicle networks," *Comput. Commun.*, vol. 178, pp. 271–285, 2021.

[5] Y. Wang et al., "Toward realization of low-altitude economy networks: Core architecture, integrated technologies, and future directions," 2025, *arXiv:2504.21583*.

[6] S. Javed et al., "State-of-the-art and future research challenges in UAV swarms," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19023–19045, Jun. 2024.

[7] N. Hrovatin, A. Tošić, M. Mrissa, and J. Vičič, "A general purpose data and query privacy preserving protocol for wireless sensor networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 4883–4898, 2023.

[8] G. Sun et al., "Online collaborative resource allocation and task offloading for multi-access edge computing," *IEEE Trans. Mobile Comput.*, vol. 24, no. 11, pp. 11430–11448, Nov. 2025.

[9] G. Sun et al., "Joint task offloading and resource allocation in aerial-terrestrial UAV networks with edge and fog computing for post-disaster rescue," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8582–8600, Sep. 2024.

[10] B. A. Begum and S. V. Nandury, "Data aggregation protocols for WSN and IoT applications–A comprehensive survey," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 2, pp. 651–681, 2023.

[11] M. Kang and S.-W. Jeon, "Energy-efficient data aggregation and collection for multi-UAV-enabled IoT networks," *IEEE Wireless Commun. Lett.*, vol. 13, no. 4, pp. 1004–1008, Apr. 2024.

[12] E. S. Ali, R. A. Saeed, I. K. Eltahir, and O. O. Khalifa, "A systematic review on energy efficiency in the Internet of Underwater Things (IoUT): Recent approaches and research gaps," *J. Netw. Comput. Appl.*, vol. 213, 2023, Art. no. 103594.

[13] M. Mohseni, F. Amirghafouri, and B. Pourghebleh, "Cedar: A cluster-based energy-aware data aggregation routing protocol in the Internet of Things using capuchin search algorithm and fuzzy logic," *Peer Netw. Appl.*, vol. 16, no. 1, pp. 189–209, 2023.

[14] N. Chandnani and C. N. Khairnar, "A reliable protocol for data aggregation and optimized routing in IoT WSNs based on machine learning," *Wireless Pers. Commun.*, vol. 130, no. 4, pp. 2589–2622, 2023.

[15] M. Sindhuja et al., "Multi-objective cluster head using self-attention based progressive generative adversarial network for secured data aggregation," *Ad Hoc Netw.*, vol. 140, 2023, Art. no. 103037.

[16] I. S. Amiri et al., "Dabpr: A large-scale Internet of Things-based data aggregation back pressure routing for disaster management," *Wireless Netw.*, vol. 26, no. 4, pp. 2353–2374, 2020.

[17] A. Rovira-Sugranes, A. Razi, F. Afghah, and J. Chakareski, "A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook," *Ad Hoc Netw.*, vol. 130, 2022, Art. no. 102790.

[18] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for DTN protocol evaluation," in *Proc. 2nd Int. Conf. Simul. Tools Techn.*, 2009, pp. 1–10.

[19] S. Yousefi, H. Karimipour, and F. Derakhshan, "Data aggregation mechanisms on the Internet of Things: A systematic literature review," *Internet Things*, vol. 15, 2021, Art. no. 100427.

[20] T.-D. Nguyen, D.-T. Le, V.-V. Vo, M. Kim, and H. Choo, "Fast sensory data aggregation in IoT networks: Collision-resistant dynamic approach," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 766–777, Jan. 2021.

[21] P. A. Kale and M. J. Nene, "A survey on network lifetime maximization using data aggregation trees," *Int. J. Commun. Syst.*, vol. 38, no. 2, 2025, Art. no. e5962.

[22] A. Heidari, H. Shishehlou, M. Darbandi, N. J. Navimipour, and S. Yalcin, "A reliable method for data aggregation on the industrial Internet of Things using a hybrid optimization algorithm and density correlation degree," *Cluster Comput.*, vol. 27, no. 6, pp. 7521–7539, 2024.

[23] E. Fitzgerald, M. Pióro, and A. Tomaszwski, "Energy-optimal data aggregation and dissemination for the Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 955–969, Apr. 2018.

[24] Z. Zhao, W. Yang, and B. Wu, "Flow aggregation through dynamic routing overlaps in software defined networks," *Comput. Netw.*, vol. 176, 2020, Art. no. 107293.

[25] L. Zhang, J. Qi, and H. Wu, "A novel data aggregation method for underwater wireless sensor networks using ant colony optimization algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 4, 2023, doi: 10.14569/IJACSA.2023.0140411.

[26] E. Hasheminejad and H. Barati, "A reliable tree-based data aggregation method in wireless sensor networks," *Peer Netw. Appl.*, vol. 14, no. 2, pp. 873–887, 2021.

[27] N.-C. Wang, C.-Y. Lee, Y.-L. Chen, C.-M. Chen, and Z.-Z. Chen, "An energy efficient load balancing tree-based data aggregation scheme for grid-based wireless sensor networks," *Sensors*, vol. 22, no. 23, 2022, Art. no. 9303.

[28] G. A. Macriga et al., "Energy efficient greedy tree based algorithm for data aggregation in wireless sensor network," *Meas., Sensors*, vol. 30, 2023, Art. no. 100910.

[29] T. C. Dao, N. T. Tam, N. Q. Quy, and H. T. T. Binh, "An energy-efficient scheme for maximizing data aggregation tree lifetime in wireless sensor network," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 9, pp. 12329–12344, 2023.

[30] A. Bomnale and A. More, "Node utilization index-based data routing and aggregation protocol for energy-efficient wireless sensor networks," *J. Supercomputing*, vol. 80, no. 7, pp. 9220–9252, 2024.

[31] V.-V. Vo, D.-T. Le, S. M. Raza, M. Kim, and H. Choo, "Active neighbor exploitation for fast data aggregation in IoT sensor networks," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 13199–13216, Apr. 2024.

[32] P. Karpurasundharapondian and M. Selvi, "A comprehensive survey on optimization techniques for efficient cluster based routing in WSN," *Peer Netw. Appl.*, vol. 17, no. 5, pp. 3080–3093, 2024.

[33] I. D. I. Saeedi and A. K. M. Al-Qurabat, "An energy-saving data aggregation method for wireless sensor networks based on the extraction of extrema points," in *Proc. AIP Conf.*, vol. 2398, no. 1, 2022, Art. no. 050004.

[34] A. K. M. Al-Qurabat and A. Kadhum Idrees, "Data gathering and aggregation with selective transmission technique to optimize the lifetime of Internet of Things networks," *Int. J. Commun. Syst.*, vol. 33, no. 11, 2020, Art. no. e4408.

[35] A. K. Idrees and A. K. M. Al-Qurabat, "Energy-efficient data transmission and aggregation protocol in periodic sensor networks based fog computing," *J. Netw. Syst. Manage.*, vol. 29, no. 1, 2021, Art. no. 4.

[36] H. S. Alshehri and F. Bajaber, "A cluster-based data aggregation in IoT sensor networks using the firefly optimization algorithm," *J. Comput. Netw. Commun.*, vol. 2024, no. 1, 2024, Art. no. 8349653.

[37] X. Zhang, J. He, Y. Zhou, and Y. Chi, "A subspace approach to sparse-sampling-based multi-attribute data aggregation in IoT," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 18054–18063, Sep. 2022.

[38] J.-F. Huang, G.-H. Zhang, and S.-Y. Hsieh, "Real-time energy data compression strategy for reducing data traffic based on smart grid AMI networks," *J. Supercomputing*, vol. 77, no. 9, pp. 10097–10116, 2021.

[39] R. Yarinezhad and M. Sabaei, "An optimal cluster-based routing algorithm for lifetime maximization of Internet of Things," *J. Parallel Distrib. Comput.*, vol. 156, pp. 7–24, 2021.

[40] H. Ko, J. Lee, and S. Pack, "CG-E2S2: Consistency-guaranteed and energy-efficient sleep scheduling algorithm with data aggregation for IoT," *Future Gener. Comput. Syst.*, vol. 92, pp. 1093–1102, 2019.

[41] R. Maivizhi and P. Yogesh, "Q-learning based routing for in-network aggregation in wireless sensor networks," *Wireless Netw.*, vol. 27, no. 3, pp. 2231–2250, 2021.

[42] W.-K. Yun and S.-J. Yoo, "Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks," *IEEE Access*, vol. 9, pp. 10737–10750, 2021.

[43] G. Ravi, M. S. Das, and K. Karmakonda, "Reliable cluster based data aggregation scheme for IoT network using hybrid deep learning techniques," *Meas., Sensors*, vol. 27, 2023, Art. no. 100744.

[44] Z. Zhaohui, Z. Jiaqi, and L. Jing, "Q-learning-based semi-fixed clustering routing algorithm in WSNs," *Ad Hoc Netw.*, vol. 174, 2025, Art. no. 103837.

[45] K. Hemalatha and M. Amanullah, "Effective hybrid deep learning model of GAN and LSTM for clustering and data aggregation in wireless sensor networks," *Int. J. Sensors Wireless Commun. Control*, vol. 14, no. 2, pp. 122–133, 2024.

[46] M. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Learning-based sparse data reconstruction for compressed data aggregation in IoT networks," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11732–11742, Jul. 2021.

[47] X. Wang, H. Chen, and S. Li, "A reinforcement learning-based sleep scheduling algorithm for compressive data gathering in wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2023, no. 1, 2023, Art. no. 28.

[48] R. Tharmalingam, N. Nachimuthu, and G. Prakash, "An efficient energy supply policy and optimized self-adaptive data aggregation with deep learning in heterogeneous wireless sensor network," *Peer Netw. Appl.*, vol. 17, no. 6, pp. 3991–4012, 2024.

[49] T. Mahmood, J. Li, T. Saba, A. Rehman, and S. Ali, "Energy optimized data fusion approach for scalable wireless sensor network using deep learning-based scheme," *J. Netw. Comput. Appl.*, vol. 224, 2024, Art. no. 103841.

[50] A. H. Wheeb, R. Nordin, A. Samah, M. H. Alsharif, and M. A. Khan, "Topology-based routing protocols and mobility models for flying ad hoc networks: A contemporary review and future research directions," *Drones*, vol. 6, no. 1, 2021, Art. no. 9.

[51] W. Zhai, L. Liu, J. Peng, Y. Ding, and W. Lu, "Par: A power-aware routing algorithm for UAV networks," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.*, 2022, pp. 333–344.

[52] Z. Ren et al., "K-means online-learning routing protocol (K-MORP) for unmanned aerial vehicles (UAV) adhoc networks," *Ad Hoc Netw.*, vol. 154, 2024, Art. no. 103354.

[53] Z. Zhou, J. Tang, W. Feng, N. Zhao, Z. Yang, and K.-K. Wong, "Optimized routing protocol through exploitation of trajectory knowledge for UAV swarms," *IEEE Trans. Veh. Technol.*, vol. 73, no. 10, pp. 15499–15512, Oct. 2024.

[54] L. Fu et al., "Joint optimization of multicast energy in delay-constrained mobile wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 633–646, Feb. 2018.

[55] M. Navarro, Y. Liang, and X. Zhong, "Energy-efficient and balanced routing in low-power wireless sensor networks for data collection," *Ad Hoc Netw.*, vol. 127, 2022, Art. no. 102766.

[56] Y. Cui, Q. Zhang, Z. Feng, Z. Wei, C. Shi, and H. Yang, "Topology-aware resilient routing protocol for fanets: An adaptive Q-learning approach," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18632–18649, Oct. 2022.

[57] M. Y. Arafat and S. Moh, "A Q-learning-based topology-aware routing protocol for flying ad hoc networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1985–2000, Feb. 2022.

[58] J. Yang, K. Sun, H. He, X. Jiang, and S. Chen, "Dynamic virtual topology aided networking and routing for aeronautical ad-hoc networks," *IEEE Trans. Commun.*, vol. 70, no. 7, pp. 4702–4716, Jul. 2022.

[59] B. Zheng, K. Zhuo, H. Zhang, and H.-X. Wu, "A novel airborne greedy geographic routing protocol for flying ad hoc networks," *Wireless Netw.*, vol. 30, no. 5, pp. 4413–4427, 2024.

[60] V. Singh, K. P. Sharma, and H. K. Verma, "CLGR: Connectivity and link quality aware geographical routing using AHP-TOPSIS for FANETs," *Peer Netw. Appl.*, vol. 18, no. 1, pp. 1–20, 2025.

[61] R. Yarinezhad and S. Azizi, "An energy-efficient routing protocol for the Internet of Things networks based on geographical location and link quality," *Comput. Netw.*, vol. 193, 2021, Art. no. 108116.

[62] H. Huang, J. Zhang, X. Zhang, B. Yi, Q. Fan, and F. Li, "EMGR: Energy-efficient multicast geographic routing in wireless sensor networks," *Comput. Netw.*, vol. 129, pp. 51–63, 2017.

[63] H. Huang, H. Yin, G. Min, J. Zhang, Y. Wu, and X. Zhang, "Energy-aware dual-path geographic routing to bypass routing holes in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1339–1352, Jun. 2018.

[64] Y. Cui, H. Tian, C. Chen, W. Ni, H. Wu, and G. Nie, "New geographical routing protocol for three-dimensional flying ad-hoc network based on new effective transmission range," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16135–16147, Dec. 2023.

[65] Q. Usman, O. Chughtai, N. Nawaz, Z. Kaleem, K. A. Khaliq, and L. D. Nguyen, "A reliable link-adaptive position-based routing protocol for flying ad hoc network," *Mobile Netw. Appl.*, vol. 26, no. 4, pp. 1801–1820, 2021.

[66] S. Kumar, R. S. Raw, A. Bansal, and P. Singh, "UF-GPSR: Modified geographical routing protocol for flying ad-hoc networks," *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 8, 2023, Art. no. e4813.

[67] K. Guo, L. Liu, W. Zhai, and Y. Ding, "Ekr: An efficient k-anycast routing in UAV networks," in *Proc. 9th Int. Conf. Comput. Commun.*, 2023, pp. 39–45.

[68] J. Peng, H. Gao, L. Liu, N. Li, and X. Xu, "Tbm: An efficient trajectory-based multicast routing protocol for sparse UAV networks," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst.*, 2020, pp. 867–872.

[69] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun, "Fanet: Communication, mobility models and security issues," *Comput. Netw.*, vol. 163, 2019, Art. no. 106877.

[70] M. Asadpour, K. A. Hummel, D. Giustiniano, and S. Draskovic, "Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 843–856, Mar. 2017.

[71] L. M. Bine, A. Boukerche, L. B. Ruiz, and A. A. Loureiro, "IoDMix: A novel routing protocol for delay-tolerant Internet of Drones integration in intelligent transportation system," *Ad Hoc Netw.*, vol. 148, 2023, Art. no. 103204.

[72] L. Syed et al., "Deep learning-based route reconfigurability for intelligent vehicle networks to improve power-constrained using energy-efficient geographic routing protocol," *Wireless Netw.*, vol. 30, no. 2, pp. 939–960, 2024.

[73] P. Mahajan, B. Palanisamy, A. Kumar, G. Chalapathi, V. Chamola, and M. Khabbaz, "Multi-objective MDP-based routing in UAV networks for search-based operations," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13777–13789, Sep. 2024.

[74] H. Tang, D. Li, Y. Zhang, X. Chen, and A. Rauf, "An adaptive bendable virtual tunnel routing protocol for flying ad-hoc networks," *Veh. Commun.*, vol. 54, 2025, Art. no. 100922.

[75] Y. Azzoug and A. Boukra, "Enhanced UAV-aided vehicular delay tolerant network (VDTN) routing for urban environment using a bio-inspired approach," *Ad Hoc Netw.*, vol. 133, 2022, Art. no. 102902.

[76] W. Zhai, L. Liu, Y. Ding, S. Sun, and Y. Gu, "ETD: An efficient time delay attack detection framework for UAV networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 2913–2928, 2023.
[77] J. Song, L. Liu, Y. Liu, J. Xi, and W. Zhai, "Path planning for multi-vehicle-assisted multi-UAVs in mobile crowdsensing," *Wireless Commun. Mobile Comput.*, vol. 2022, no. 1, 2022, Art. no. 9778188.
[78] K. Zhang, L. Liu, W. Zhai, Y. Ding, and J. Hu, "OSIS: Obstacle-sensitive and initial-solution-first path planning," in *Proc. IEEE 29th Int. Conf. Parallel Distrib. Syst.*, 2023, pp. 2795–2797.
[79] W. Zhai, S. Sun, L. Liu, Y. Ding, and W. Lu, "Hotd: A holistic cross-layer time-delay attack detection framework for unmanned aerial vehicle networks," *J. Parallel Distrib. Comput.*, vol. 177, pp. 117–130, 2023.
[80] K. Meng et al., "UAV-enabled integrated sensing and communication: Opportunities and challenges," *IEEE Wireless Commun.*, vol. 31, no. 2, pp. 97–104, Apr. 2024.
[81] A. Aldweesh et al., "MLORA-CBF: Efficient cluster-based routing protocol against resource allocation using modified location routing algorithm with cluster-based flooding," *Wireless Netw.*, vol. 30, no. 2, pp. 671–693, 2024.
[82] S. Tamizharasu and P. Kalpana, "An intelligent AODV routing with energy efficient weight based clustering algorithm (EEWCA) in wireless ad hoc network (WANET)," *Wireless Netw.*, vol. 29, no. 6, pp. 2703–2716, 2023.
[83] S. Gangopadhyay and V. K. Jain, "A position-based modified OLSR routing protocol for flying ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 12087–12098, Sep. 2023.
[84] Y. Cheriguene et al., "Cocoma: A resource-optimized cooperative UAVs communication protocol for surveillance and monitoring applications," *Wireless Netw.*, vol. 30, no. 5, pp. 4429–4445, 2024.
[85] Y. Nabil, H. ElSawy, S. Al-Dharrab, H. Mostafa, and H. Attia, "Data aggregation in regular large-scale IoT networks: Granularity, reliability, and delay tradeoffs," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17767–17784, Sep. 2022.
[86] M. Kulichenko et al., "Uncertainty-driven dynamics for active learning of interatomic potentials," *Nature Comput. Sci.*, vol. 3, no. 3, pp. 230–239, 2023.
[87] S. Bhattacharya, M. Henzinger, and D. Nanongkai, "A new deterministic algorithm for dynamic set cover," in *Proc. IEEE 60th Annu. Symp. Foundations Comput. Sci.*, 2019, pp. 406–423.
[88] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proc. 1st Annu. IEEE Commun. Soc. Conf. Sensor Ad Hoc Commun. Netw.*, 2004, pp. 517–526.

Liang Liu received the B.S. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2005, and the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2012. He is currently an Associate Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include distributed system, Big Data, and system security.

Wenbin Zhai received the B.S. degree from the Nanjing University of Chinese Medicine, Nanjing, China, in 2020, and the M.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, in 2023. He is currently working toward the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include AI security, cybersecurity, and wireless sensor networks.

Xin Li received the B.S. and M.S. degrees from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2019 and 2022 respectively. His research interests include UAV routing and spatio-temporal range query for UAV networks.

Youwei Ding received the B.S. and M.S. degrees in computer science from Yangzhou University, Yangzhou, China, in 2007 and 2010, respectively, and the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China in 2016. He is currently an Associate Professor with the School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing. His research interests include energy efficient data management, Big Data analysis, and data security.

Wanying Lu received the B.S. degree from Henan Polytechnic University, Jiaozuo, China, in 2021, and the M.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2024. Her research interests include time series Big Data storage and data mining.

Ran Wang (Senior Member, IEEE) received the B.E. degree in electronic and information engineering from Honors School, Harbin Institute of Technology, Harbin, China, in 2011, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2016. He is currently a Professor and Doctoral Supervisor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. He is also a ChangKong Scholar with NUAA. He has authored or coauthored more than 100 papers in top-tier journals and conferences. His current research interests include telecommunication networking and cloud computing. He was the recipient of Nanyang Engineering Doctoral Scholarship (NEDS) Award in Singapore and Innovative and Entrepreneurial Ph.D. Award of Jiangsu Province, China, in 2011 and 2017, respectively, Second Prize of the Jiangsu Provincial Science and Technology Award in 2023, Second Prize of the China Communications Society Science and Technology Award in 2022, and Best Paper Award of ACM MobiArch'23.