Hindawi
Mobile Information Systems
Volume 2022, Article ID 7433556, 16 pages
https://doi.org/10.1155/2022/7433556

Research Article

All-Domain Fusion-Based Time Synchronization Protocol in SD-ATSN

Meng-Yuan Zhu,[1] **Ke-Fan Chen**,[2] **Zhuo Chen**,[3] **and Na Lv**[1]

[1]*Information and Navigation College, Air Force Engineering University, Xi'an 710077, China*
[2]*People's Liberation Army of China, Nan Jing 210000, China*
[3]*People's Liberation Army of China, Lu An 237000, China*

Correspondence should be addressed to Na Lv; lvnn2007@163.com

Received 20 October 2021; Revised 17 January 2022; Accepted 1 February 2022; Published 11 March 2022

Academic Editor: Robin Singh Bhadoria

As combat scales up and weapons become more intelligent, airborne networks (AN) must facilitate high-precision communications. This means that the airborne network should have communication capability with minimum delay, low jitter, and high reliability, and this new type of AN is called an airborne time-sensitive network (ATSN). A prerequisite to guarantee the above communication capability is to have a high precision time synchronization protocol. To plug this gap, we have designed a software-defined airborne time-sensitive network (SD-ATSN) architecture based on the benefits of software-defined centralization and flexibility. It supports our proposed all-domain fusion-based time synchronization protocol (AF-TSP) to support precise time synchronization between ATSN platforms. In AF-TSP, we innovatively propose an all-domain (land, sea, air, and space) master clock hot standby mechanism to cope with the existing instability and poor robustness in AN. In the key problem of the master clock election, we first completed a rough election utilizing improved K-Means++. Subsequently, on top of the rough election, we inserted a mixed mutation operator to improve the convergence of the multiobjective optimization algorithm No Dominant Sorting Genetic Algorithm-II (NSGA-II). The control delay, clock accuracy, and path reliability coefficient are the optimization objectives for selecting the appropriate master clock. Simulation results demonstrate that our protocol has advantages in terms of synchronization precision, communication delay, and network robustness.

1. Introduction

1.1. Motivation. As the number of airborne platforms (drones, manned aircraft, early warning aircraft, etc.) continues to increase, weapon systems (radio systems, ammunition, and missiles, etc.) are all over the skies [1–4]. In this context, the airborne network (AN), which connects the whole sky elements, has become a core component of military aviation. At present, the airborne networks are mainly composed of independent data link systems (Link 11, Link 16, MADL, etc.) [5]. Generally speaking, current ANs have little ability to communicate with each platform precisely. Accordingly, it is tough to support high-speed target attacks, signal-level information fusion, precise weapon control, and other situations requiring time-sensitive network services. In addition, without precise electromagnetic spectrum management, the problem of the low utilization rate of the AN communications resources are becoming increasingly prominent [6–8]. Faced with these problems, the DARPA has determined a large number of programs, such as ASTARTE, ICASS, and SoSITE, to provide certain bandwidth, low delay, and minimum jitters for AN [9, 10]. Therefore, building an airborne time-sensitive network (ATSN) has become an important development direction in the present and future.

Similarly, there is an inevitable and urgent need for time-sensitive network services in many industries, such as medical care, augmented reality and virtual reality (AR/VR), and self-driving vehicles [11–15]. Among all solutions, time synchronization is the key prerequisite, and so is the ATSN

[16, 17]. Time synchronization facilitates network nodes to transmit information while maintaining accurate time. Enhancing the performance of time synchronization would make communication across different platforms faster, more precise, and more flexible [18]. In view of the above advantages, there is an urgent need to design a time synchronization protocol adapted to ATSN applications.

1.2. Related Work. Global Navigation Satellite System Timing and Round-Trip Timing (RTT) are the most widely employed synchronization methods for airborne networks. These methods rely on an external clock source or a single master clock specified statically [19]. The airborne network has the characteristics of the heterogeneous platform, strong mobility of nodes, and dynamic change of network topology. As the change of battlefield situation, an external or static clock source is easy to lose its function, and it is tough to retain the best synchronization with the fluctuation of link quality and computing power. Particularly, attacks on the clock source or a surge in synchronization load will lead to the rapid decline of the network synchronization accuracy and even the collapse of the synchronization network [20, 21].

For the Internet, standard time synchronization protocols, such as Network Time Protocol (NTP), Precise Time Protocol (PTP), Reference Broadcast Synchronization (RBS), and Hierarchical Reference Time Synchronization Protocol (HRTS), rarely pay attention to the importance of designating the master clock [22, 23]. In addition to statically specifying the master clock, the most commonly used master clock selection algorithm for time synchronization protocols is the Best Master Clock Algorithm (BMC). At any time, BMC has only one master clock, which needs to send Announce_ messages continuously, resulting in huge communication costs. If the master clock fails, it needs to be re-elected, and it takes a long time to conduct the election, which will readily cause the synchronous network to collapse [22]. IEEE 802.1 AS-rev provides a dual master clock redundancy mechanism in the general Precise Time Protocol (gPTP) based on the BMC algorithm. When the master clock fails, the controller can promptly switch to the standby master clock. However, this protocol requires static configuration and is only suitable for small wired local area networks [24]. Tinner has substantially widened the network range of 802.1 AS-rev synchronization, but it cannot exceed 7 hops at most [25].

1.3. Contributions. From this point of view, neither the AN nor the Internet has a direct time synchronization mechanism, which can meet the precise communication requirements of ATSN. Subsequently, in order to achieve high-precision time synchronization of ATSN applications, we propose a software-defined based ATSN architecture and also design a new time synchronization protocol. The innova- tion of this paper can be summarized as follows.

(i) We propose a software-defined airborne time-sensitive network (SD-ATSN) architecture. SD-ATSN integrates the advantages of centralized control and agile management of software-defined networking (SDN) and is used to satisfy the low latency and high-reliability communication requirements of airborne networks to adapt to future highly dynamic and highly adversarial battlefields. It is designed with a three-layer architecture, whose central control layer is responsible for real-time configuration and precise management of the network; the data forwarding layer implements functions such as measuring network latency and jitter, forwarding and transmitting data. This architecture provides the foundation support for our proposed time synchronization protocol.

(ii) We design the all-domain fusion-based time synchronization protocol (AF-TSP), which innovatively combines a hot standby mechanism and a full-domain deployment strategy. Based on the SD-ATSN architecture, we can activate the master clocks located in different operational domains such as land, sea, air, and space according to different requirements, while the master clocks of all domains are directly hot-standby to each other. This effectively improves the survivability of a single master clock, improves network robustness, and ensures high-precision time synchronization of the ATSN.

(iii) We focus on the AF-TSP protocol master clock election problem. Because the merit of the master clock directly affects time synchronization precision. We propose a step-by-step master clock optimization selection algorithm to determine the optimal by two steps: virtual centroids (central clock coordinates) and multiobjective optimization. In the first step, we divide multiple sub-domains by improving the K-means++ algorithm and selecting the virtual centroid at the same time. The rough election of the master clock is completed. In the second step, we define the control node failure probability, link failure probability, clock accuracy, propagation delay, and processing delay as the optimization objectives and insert the mixed mutation operator to strengthen the performance of the NSGA-II multiobjective optimization algorithm. Subsequently, the precise optimal master clock is selected from the rough selection result in the first step.

The rest of the paper is organized as follows: Section 2 introduces the SD-ATSN network architecture and the basic operation flow of AF-TSP. Section 3 describes the master clock election algorithm of AF-TSP. In Section 4, the simulation verified the performance characteristics of AF-TSP. Section 5 is the conclusion.

2. All-Domain Fusion-Based Time Synchronization Protocol (AF-TSP)

To better satisfy the communication requirements of low latency, high precision, and high reliability for airborne time-sensitive network applications, we propose SD-TSAN,

a network architecture, in Section 2.1. Section 2.2 exhibits AF-TSP, a time synchronization protocol. Lastly, we will present a case to further illustrate the protocol.

2.1. Network Architecture of SD-ATSN. The airborne network is exposed to wireless and hostile environments, including heterogeneous nodes and dynamic topologies. Due to the complex control in management, it is tricky to synchronize time in the ATSN. Currently, the high precision time synchronization mechanism is essentially optimized offline for specific scenarios and is preset before use in a mission [26]. If the actual task or environment is different from the assumptions of originally configuring the network, the settings cannot be modified promptly. When intelligent opponents can unpredictably influence the network topology and operation in a short time, this problem will become more significant.

SDN strictly separates the network control logic from the processing and forwarding behavior of the underlying data stream, which greatly simplifies the management and configuration of the network, and is conducive to the rapid and on-demand deployment of network services [27]. In [28], the use of SDN effectively solves the problems of weak scalability and static configuration of Ethernet time-sensitive networks. reference [29] utilizes SDN to recognize the flexible structure, dynamic monitoring, real-time management, and adjustment on demand. And, it presents a method to exchange timing information of the synchronous network on the central controller. Simulation results demonstrate that the reduction of communications interaction would decrease the response time and throughput. Although this research is all conducted on local area networks, researchers have verified the advantages of employing SDN.

Therefore, we use SDN to simplify real-time management and control of ATSN. As shown in Figure 1, we designed the SD-ATSN architecture, which provides basic support for precise time synchronization in ATSN.

The SD-ATSN consists of two types of nodes: master control nodes and service nodes. There are several kinds of master control nodes in the SD-ATSN, such as airborne early warning aircraft, command post, aircraft carrier, and geostationary orbit satellite. These nodes are invulnerable, have sufficient computing power and powerful payload capacity. Therefore, these nodes can be the master clock to meet the resource requirement and robustness of the time synchronization. Other operational platforms are categorized as service nodes.

Each SD-TSAN node has a service layer, a subcontrol layer, and a data forwarding layer. The master node furthermore contains the network application layer and the central control layer. The network application layer in the master control node implements the application layer function of SDN. This layer provides various applications to control and arrange the behavior and functional characteristics of SD-ATSN. The central control layer maps to the control plane of the SDN and is responsible for collecting and processing the network measurement information. At the same time, this layer constructed a global network view to deal with centralized data forwarding.

The service layer of SD-ATSN nodes is responsible for generating and processing various tactical information. On the contrary, as the supplement and substitute for SD-ATSN centralized network control, the subcontrol layer is responsible for the distributed network control in the service nodes [30]. The data forwarding layer corresponds to the data layer of SDN. These functions, such as measuring network delay and jitter, forwarding and transmitting data, are implemented in the data forwarding layer. Under this three-tier structure, it exhibits better robustness and invulnerability to the accurate control of ATSN communication network. In addition, this architecture is superior to traditional architecture in satisfying the dynamic characteristics of airborne networks.

2.2. All-Domain Fusion Based Time Synchronization Protocol (AF-TSP). The performance of time synchronization depends on the precise clock source provided by the master clock [14–18]. That is, the merit of the master clock determines the performance of time synchronization. Thus, what type of clock is selected as the master clock? How to select this master clock? It is a tricky problem that must be settled. Nonetheless, orthodox and common time synchronization protocols are used to continuously transmit such synchronization messages as *Announce_* and *Sync_*, so that the clock that receives the message continually compares the size of its own data with the received data, so as to compare the best master clock. Called Best Master Clock Algorithm (BMCA), it has been mostly used in time synchronization protocols such as gPTP, IEEE 1588, and 1588V2 [17, 31].

Nonetheless, this algorithm has a massive communication overhead, a fixed synchronization link, and relies on a single master clock. Unfortunately, the airborne network has the characteristics of platform node heterogeneity, network dynamics, and vulnerability to attacks [5, 7, 9], and BMCA can hardly be directly applied to SD-ATSN. Consequently, we need to adopt a new master clock election algorithm in the designed AF-TSP time synchronization protocol.

We believe that the master clock will be assumed by a specific operational platform node, which will also become the master control node in the SD-ATSN. In subsequent sections, we collectively refer to it as the master clock node or master clock. Subsequently, in order to maximize the reliability, sustainability, and time synchronization precision of the SD-ATSN, a suitable master clock node needs to be selected. This node needs to have the advantages of survivability, communication coverage, and system capability. This is because strong platform survivability assured not only the impeccable master clock but also the robustness of the synchronization network. The wider the master clock coverage, the lower the communication overhead will be. The system performance of the nodes themselves is pivotal to the accuracy of synchronization. Nevertheless, in a variety of battlefield environments, it is unrealistic to expect a single node to meet the full capability.

As shown in Table 1, when we have to extend the range of operations in an emergency, if only the AEW in the air

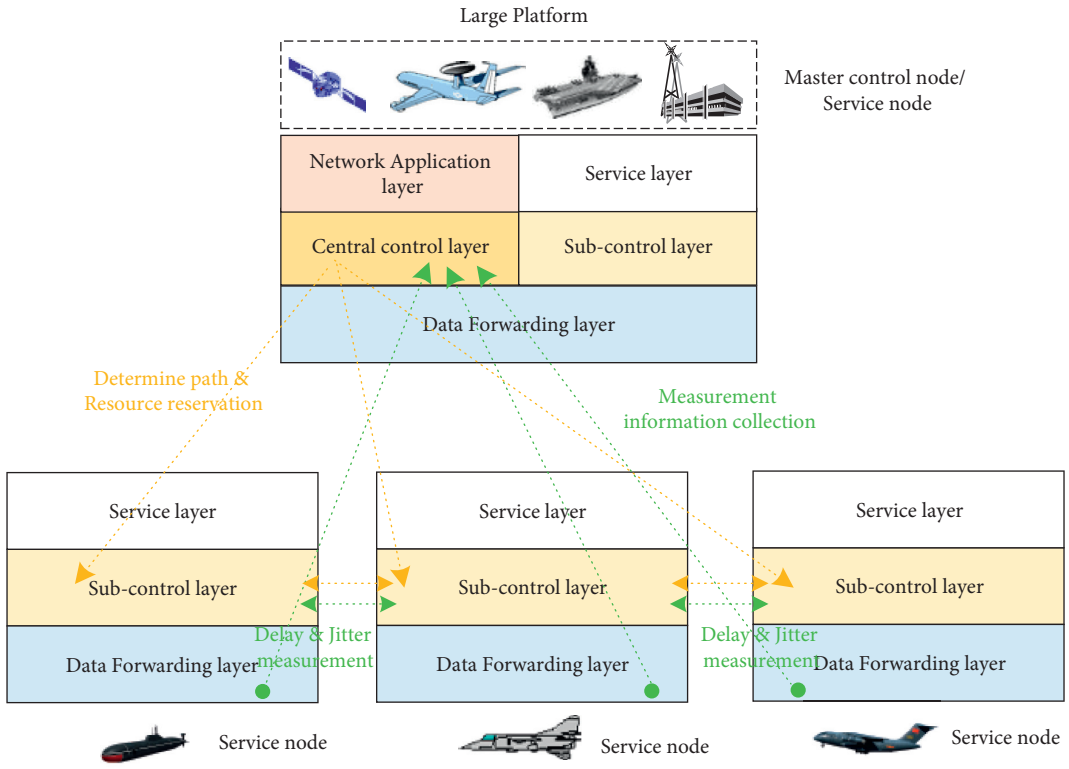

Figure 1: SD-ATSN architecture.

domain is selected as the master clock, it increases the uncertainty of synchronization precision because of the AEW's limited communication range. In addition, the multihop transmission of synchronization messages of early warning aircraft will inevitably increase the communication overhead. However, in this scenario, if we switched the master clock to the satellite in the space domain, we can utilize its communication covering capacity to avoid the above problems. Therefore, establishing the hot standby mechanism of the master clock in the all-domain can significantly enhance the time synchronization capability of SD-TSAN.

Hot standby is a core idea of AF-TSP and an innovation in our master clock election algorithm. This mechanism forms a master clock set by selecting multiple master clocks across the entire domain. These master clocks are backed up to each other through the SD-ATSN's network controllers. They are centrally governed by the controller and activated on demand. This redundancy mechanism ensures that only one master clock is responsible for synchronization at a given point in time. When this master clock is assaulted or damaged, the controller switches to the hot standby master clock in time. After a brief adjustment, the original time synchronization network will be synchronized utilizing the time of the hot standby master clock. Relying on SD-ATSN, we effectively establish a hot standby mechanism that can settle the unreliability of the ATSN communication link and the vulnerability of the synchronization network caused by a single master clock. The network will not collapse because the master clock is damaged, which improves the reliability of the synchronization network and accordingly ensures synchronization precision.

For ease of understanding, we refer to the master clocks of different domains as domain master clocks in this subsection, and the domain master clocks of the whole domain mutually constitute the master clock. Master clock election algorithm section (Section 3) will not be differentiated, collectively referred to as the master clock.

As shown in Figure 2, AF-TSP is logically divided into three functional layers: the domain master clock election layer, the domain master clock layer and the subordinate time layer. Based on the global network view, the SD-ATSN network controller selects the appropriate domain master clock from the entire domain at the domain master clock election layer based on customized evaluation metrics and related requirements. Simultaneously, these domain master clocks form the master clock set. In Section 3, we describe the election algorithm in detail. The domain master clock layer is responsible for maintaining the single valid master clock for the time-synchronous network; in other words, only one domain master clock will be activated at a given time. This activated domain master clock will assume the function of synchronizing the network. During network operation, the network controller dynamically activates the domain master clock based on the network conditions observed in the global network view and the status of each master clock. If the current domain master clock becomes corrupted or the communication link fails, it will swiftly switch to another domain master clock. In addition, the network controller maintains the transmission of

Table 1: Comparison of advantages and disadvantages of master clock platform.

Node	Combat domain	Advantage	Disadvantage
AEW	Air	Better tactical position, stronger flexibility, stronger carrying capacity	The communication coverage ability is general, the concealment is poor, and the action time is limited
Ground command post	Land	Better tactical position, the concealment is good, the bearing capacity is strong, and the action time is long	Poor communication coverage and flexibility
Aircraft carrier	Sea	The tactical position is good, the flexibility is good, the bearing capacity is strong, and the action time is long	The ability of communication coverage is poor and the concealment is poor
GEO	Space	The communication coverage capacity is strong, the bearing capacity is strong, and the action time is long	Poor tactical position, poor flexibility and poor concealment

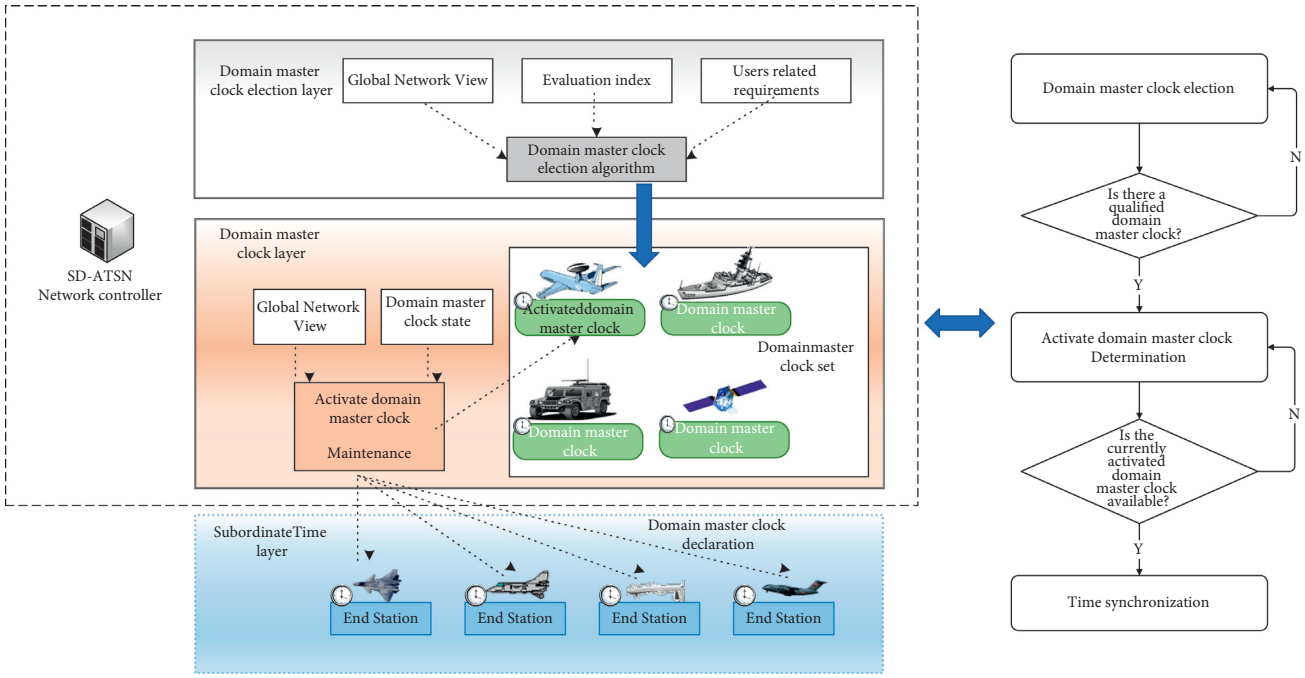

Figure 2: AF-TSP time synchronization protocol diagram.

synchronization messages from the domain master clock to the remaining nodes within the slave time layer that need to be synchronized. This logical approach ensures that the domain master clock provides time synchronization services precisely, continuously, and reliably. According to this time synchronization protocol, the relevant nodes interact with the active domain master clock to complete time synchronization.

2.3. Application Case. Through an application case, the time synchronization protocol we proposed can be more readily understood. Firstly, we outline the details of the case and then explain how AF-TSP implements the network services desired by this application scenario.

Case: identification and strike task under the environment of all-domain fusion.

All-domain fusion operation breaks the complex link between perceptual resources and weapon resources of the original combat platform. By adopting SD-ATSN, the fixed platform does not need to rely on its sensor data to continuously guide and track the target and does not need to launch its weapons to complete the attack on the target. SD-TSAN moreover assured the survivability of its platform. The scenario is illustrated in Figure 3.

Platform B and Platform C bear the SD-ATSN network controller and became the domain master clock which is mutually backed up after the election. The enemy target detected by platform A will move to the right along the direction of the dashed arrow. At this time, platform B is the active domain master clock. B comprehensively calculates the resource and task requirements of all nodes and determines that platform A is the first fighting node. At the same time, platform B synchronizes its control domain members. With the movement of the target position, platform G will be in the best strike position, and platform B will transmit the target tracking information to platform G to guide the missile launched by platform A to eventually complete the strike mission.

In the process of completing the task, the domain master clock will call any suitable network application through the central control layer in the network controller according to the task requirements at different moments. Consequently, the domain master clock will design a reasonable strategy for

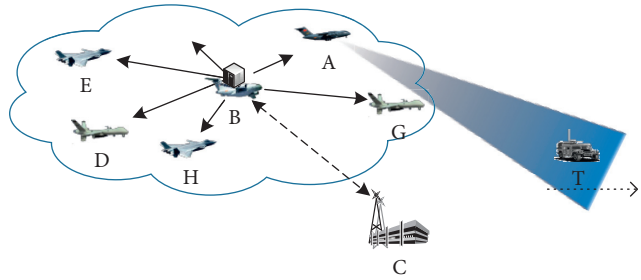

Figure 3: Mission diagram.

the network configuration. Meanwhile, the data forwarding layer continuously collects measurement information such as delay and jitter and establishes information transmission links for the high-precision communication network.

3. AF-TSP Master Clock Election Algorithm

As mentioned in Section 2, in order to meet the precise communication requirements of time-sensitive airborne networks. We put forward AF-TSP, in which the selection of the master clock is the strategic point. The evaluation index and election algorithm of the master clock will be described in detail.

3.1. Evaluation Index of Master Clock Election. We define three evaluation indexes: path reliability coefficient, control delay, and clock accuracy. Each index is targeted for different master clock demands: platform survivability, communication covering capability, and system capability.

3.1.1. Control Delay. Network delay principally consists of transmission delay, propagation delay, processing delay, and queuing delay. In the application scenario with large-scale distributed SD-ATSN, the transmission delay and queuing delay are comparatively smaller than the processing delay and propagation delay [32]. Therefore, the communication overhead of the SD-ATSN is chiefly determined by the propagation delay on the round-trip data path. In order to ensure that the network has stronger system capability, we tend to select the node with the strongest processing capability. We define the control delay T including the propagation delay t and the processing delay m.

Previously, we refer to [33] using a matrix to represent the communication between the nodes and the master clock in the domain, as in

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,N} \\ \cdots & \cdots & \cdots \\ x_{N,1} & \cdots & x_{N,N} \end{bmatrix}, \tag{1}$$

where $x_{i,j}$ indicates the connection of nodes v_j and the master clock c_i, $x_{i,j}$ has only two values, 0 or 1. 0 means v_j is not connected to c_i, and 1 means connected. N represents the set of all nodes in the subdomain controlled by c_i.

Definition 1. Propagation delay t.

The data propagation delay is predominantly affected by the deployment position of the master clock. For example, GEO in the spatial domain and the AEW in the airborne domain can detect, locate and track the target in a large range. Nevertheless, these propagation delays which interact with the domain edge nodes will be too high [32]. In this section, t is defined using only the time on the data round-trip path. For each node v_j,

$$t_{i,j} = 2 \cdot f_j \cdot \frac{d_i j}{c}, \tag{2}$$

where f_j represents v_j propagation data size; $d_{i,j}$ is the length of the shortest path between v_j and c_i; and the constant c is the speed of light. $t_{i,j}$ denotes the propagation delay of the connection between v_j and c_i.

Definition 2. Processing delay m.

The processing delay m is only related to the size of the transmitted data and the processing capability of the controller. For each node v_j,

$$m_{i,j} = \frac{f_j}{p_i}, \tag{3}$$

where p_i represents the processing capacity of the master clock node. $m_{i,j}$ denotes the processing delay of master node c_i to process the data from v_j.

Definition 3. Control delay T.

Control delay T is the sum of propagation delay t and processing delay m.

$$T_{i,j} = m_{i,j} + t_{i,j}, \tag{4}$$

$T_{i,j}$ represents the control delay between v_j and c_i.

Then the control delay for the master clock node to the nodes in its domain can be expressed as

$$T = \sum_{j=1}^{N} x_{i,j} \cdot T_{i,j}. \tag{5}$$

3.1.2. Clock Accuracy. The clock accuracy P is one of the important parameters to describe clock performance. It is stored in the data set, and the node carries an asynchronous data packet. The research showed that the higher the

numerical value, the better the clock performance [31]. p_j denotes the clock accuracy of node v_j.

3.1.3. Path Reliability Coefficient. The SD-ATSN supports complex topologies and has multiple routes between its nodes with different path durations. We considered nodes' survivability and the path's survivability to ensure synchronous network invulnerability and robustness in this intensely dynamic network.

We use link failure probability r_a and node failure probability Ψ to define the path reliable coefficient R.

Definition 4. Link failure probability r_a.

The link failure probability r_a is determined by the link length, and the longer link is, the higher the failure probability is [33]

$$r_a = (1-\delta)^{d_a}, a \in A_{i,j}, \tag{6}$$

where $A_{i,j}$ is the set of direct links through which the shortest path between v_j and c_i passes; δ is failure probability per unit length; d_a is the length of a direct link.

Due to the heterogeneity of SD-ATSN combat units, different domains and different types of links have different failure probabilities per unit length.

Definition 5. Node failure probability Ψ.

In theory, all nodes in the domain can participate in the election of the master clock, but with the SD-TSAN architecture, the master clock node must carry the SDN controller, consequently for each node v_j:

$$a_j = \begin{cases} 1, v_j \text{iscarriedSDNcontroller}, \\ 0, \text{else}. \end{cases} \tag{7}$$

At the same time, to ensure the reliability of the network, the master clock should have a low probability of node failure.

$$\Psi = \frac{\sum_{j=1}^{B_{i,j}} \left(r_j \cdot a_j\right)}{n}, \tag{8}$$

where r_j represents the probability that v_j in the network cannot function normally; $B_{i,j}$ is the set of the shortest path between v_j and c_i passing through nodes; n is the total number of SDN controllers in this domain.

Definition 6. Path reliable coefficient R.

Node failure probability Ψ and link failure probability r_a determined the path reliability coefficient R on the master node control link. For each node v_j, $R_{i,j}$ represents the control path reliability coefficient of v_j and c_i [34].

$$R_{i,j} = \prod_{a \in A_{i,j}} (1 - r_a) \cdot \prod_{b \in B_{i,j}} \left(1 - r_j\right). \tag{9}$$

Combined with equation (1), the path reliability coefficient of c_i is the average value of the reliability coefficient in its domain all control paths.

$$R = \frac{\sum_{j=1}^{N} R_{i,j} \cdot x_{i,j}}{\sum_{j=1}^{N} x_{i,j}}. \tag{10}$$

3.2. Objective Function. According to the above definition, the smaller the control delay, the better the effect of time synchronization. The more precise the clock is, the more capable it is of acting as the master clock. The larger the path reliability coefficient, the higher the robustness of the synchronization network. Hence, in order to comprehensively optimize the evaluation index of the domain master clock, we summarize the election objective function of the AF-TSP master clock as follows:

$$Z = \min F(X) = \begin{bmatrix} \min f_T\left(x_{i,j}\right) = \sum_{j=1}^{N} x_{i,j} \cdot T_{i,j} \\ \min f_P\left(x_{i,j}\right) = \dfrac{1}{p_j} \\ \min f_R\left(x_{i,j}\right) = \dfrac{\sum_{j=1}^{N} x_{i,j}}{\sum_{j=1}^{N} R_{i,j} \cdot x_{i,j}} \end{bmatrix}. \tag{11}$$

$$\text{s.t.} \begin{cases} \forall j \in N, \sum_{i=1}^{N} x_{i,j} = 1 \\ \forall i \in N, q_i < f_i \\ \Psi < \dfrac{\sum_{j=1}^{N} \left(r_j\right)}{N} \end{cases}$$

The first point of the constraint indicates that each node in the domain is controlled by only one master clock; the second constraint indicates that the load of the node cannot exceed the processing capacity, where $q_i = \sum_{j=1}^{N} (x_{i,j} \cdot f_i)$; the last point of constraint indicates that the failure probability of the master node should be smaller than the average failure probability of all nodes in its domain.

3.3. Master Clock Election Algorithm Based on K-Means++ Virtual Centroid and Mixed Mutation Operator NSGA-II

3.3.1. Rough Election of Virtual Centroid of K-Means++. ATSN is a large-scale, node-rich, and heterogeneous network, while it differs from the Internet in that it does not utilize protocols such as IPV6, so it is challenging to precisely identify and classify nodes in different domains. The clustering algorithm is favorable to solving this difficulty. It can multicenter cluster the nodes in the same network to differentiate different domains.

K-Means++ is an unsupervised fast iterative clustering method with near-linear time complexity. It is highly efficient in large data processing and generates clustering centers as far away from each other as possible [35]. In

Step 1: rasterize the global combat network and randomly select a location coordinate as the initial clustering centroid c_i
Step 2: calculate the shortest distance between each node and the existing cluster centroid $D(v) = \operatorname{argmin}_{i=1}^{k_{\text{selected}}} \|v_j - c_i\|^2$
Step 3: calculate the probability that each node is selected as the next cluster centroid $D(v_j)^2 / \sum_{j \in v} D(v_j)^2$
Step 4: using the roulette method to select the next cluster centroid c_i
Step 5: repeat step 2 until a total centroid is selected
Step 6: determine the initial centroid $C = \{c_1, c_2, \ldots, c_k\}$
Step 7: calculate the distance from each node to a centroid and divide it into the subdomain corresponding to the minimum centroid
Step 8: recalculate the centroid of clustering $c_i = 1/|c_i| \sum_{v \in c_i} v$
Step 9: repeat steps 5 and 6 until the position of the center of mass no longer changes

Algorithm 1: Rough election of virtual centroid of K-means++.

Algorithm 1, we abstract the clustering centers as virtual centroids, which are represented in the form of coordinates.

We divide the SD-ATSN into different domains, while the virtual centroid c_i of the algorithm output is located at the center of the domain as much as possible. This approach prevents long-distance communication with any edge node and improves the synchronization accuracy of the edge nodes.

Because the virtual centroid is a coordinate, it may have one real node, no node, or have multiple nodes, so we call this process of Algorithm 1 a rough election of the master clock. After forming the set C of virtual centroids, in Algorithm 2, we further execute a multiobjective optimization of the actual nodes within a specific range of the virtual centroid to gain the exact selection of the master clock.

3.3.2. Mixed Mutation Operator-Based NSGA-II Multiobjective Optimization Precision Election. No Dominated Sorting Genetic Algorithm II (NSGA-II) is a biological heuristic algorithm for solving complex optimization problems. The algorithm is characterized by fast merging speed and high precision and improves the graphics near the real Pareto optimal frontier [36]. Nonetheless, the performance of the simulated binary crossover operator and polynomial mutation operator is relatively weak, which limits the searching ability of the algorithm. When the algorithm solves more than or equal to three optimization objectives, it will cause uneven distribution of convergence and easily fall into the local optimum [37].

Therefore, we introduce a mixed mutation operator into the NSGA-II to improve the population diversity and prevent the algorithm from falling into local optimization. Gaussian mutation operators have strong local stability. On the other hand, the uniformly distributed mutation operator has a solid ability to guide individuals to jump out of the local optimum, which is conducive to global convergence [38]. So, we try to mix the two mutation operators, which are expressed as follows:

$$X_i^! = \begin{cases} X_i + C(0, \delta), t \le 1/2T, \\ X_i + N(0, \sigma), 1/2T < t \le T. \end{cases} \tag{12}$$

$C(0, \delta)$ denotes the one-dimensional Cauchy distribution random variable with 0 mean and the scale parameter is δ. $N(0, \sigma)$ represents the one-dimensional normal distribution random number with mean value 0 and standard deviation is σ; t represents the current evolutionary algebra and T represents the maximum evolutionary algebra. The pseudo code is shown in Algorithm 2.

In steps 1 and 2, the population is generated and initialized. In steps 3–6, the population is sorted according to the non-advantage-based sorting. The best half elite population is selected as parents in steps 7–8. In step 9, crossover and mutation operations are accomplished to produce a better solution from the selected parent population. In steps 10–14, select the better half of the elite population from the combined population (generated by the elderly and child). Repeat these steps until the predetermined maximum number of generations is reached.

Figure 4 portrays the AF-TSP master clock selection algorithm, with Algorithm 1 in the green section on the left and Algorithm 2 in the blue section on the right. The output of Algorithm 1 is used as the input to Algorithm 2.

The left part is Algorithm 1, which enters the modified K-Means++ algorithm for looping after randomly generating the virtual centroids, and outputs C when $i = K$ and C no longer change position. Hence completing the rough election, and the output of Algorithm 1 is the input of Algorithm 2 (right part of Figure 4). With the nodes within the virtual centroids as the initial population, Algorithm 2 is used to iterate in a loop, and the optimal result is output after attaining the maximum number of iterations to complete the precise election of the master clock.

4. Simulation Verification

4.1. Simulation Conditions. To verify the performance of the master clock time synchronization protocol and master clock election algorithm, we conducted a joint simulation based on EXata5.1 and MatlabR2019b on the desktop equipped with Corei9-9820x and GTX1080Ti, and the detailed simulation parameters are exhibited in Table 2.

AF-TSP is an overlay protocol that is an enhancement after the airborne time-sensitive network has established a relatively stable and efficient connection using the underlying routing and MAC protocols. The main enhancement goal of AF-TSP is to improve the time synchronization accuracy of the airborne time-sensitive network. At the same time, the optimal of the enhanced protocol is to make the dominant protocol more advantageous, so we should still

Notations: DS_p^t: dominance set of t solution; S_p^t: t solution of the population; n_p: this is the number of solutions that dominate p; F_q^t: t front; Cr^{dist}: crowding distance; S_{child^P}: size of child population; S_{old^P}: size of the old population considered for execution; P_{rank}^t: rank of t solution; t: number of generations used for looping; T: max number of generations; old^P: old population; child^P: child population

Input: t; T; s_{old^P}

(1) Near the virtual centroid c_i. Generate initial population of size S_{old^P} by random distribution of decision variable in given range [,]. Save one copy of population as old^P

(2) **For each** $S_p^t \in \text{old}^P$

(a) Calculate Multiobjective using equation (11)

(b) Meet the relevant constraints using equation (12)

End for

$t = 1$

(3) **While** $(t \le T)$

Nondominated_sorting(old^P)

(4) **For each** $S_p^t \in \text{old}^P$

Calculate DS_p^t

End for

$q = 1$,

(5) **For each** $S_p^t \in \text{old}^P$

If $(DS_p^t = \varnothing)$

$F_q^t = F_q^t \cup S_p^t$

$P_{\text{rank}}^t = 1$

End If

End For

$q = 2$

(6) **For each** $S_p^t \in \text{old}^P$

If $(DS_p^t = \varnothing)$

$F_q^t = F_q^t \cup S_p^t$

$P_{rank}^t = q$

$q = q + 1$

End If

End for

Crowing_distance (old^P)

Assume Cr^{dist} from boundary point to ∞ for any solution

(7) **For each** $S_p^t \in \text{old}^P$

calculate Cr^{dist} from all point excluding boundary points

End For

(8) **Select** the best half elite population as parent^P using Binary tournament selection approach.

$\text{child}^P = \varnothing$

$S_{child^P} = 0$

(9) **While** $(S_{\text{child}^P} \le S_{\text{old}^P})$

(a) **Randomly** select two chromosomes from the parent^P

(b) **Perform crossover** to produce two child chromosomes

(c) **Mutate chromosome** to produce a child chromosome using Mixed mutation operator (13)

Update child^P and

End While

(10) **Generate** new population of size $(2 \times S_{\text{old}^P})$ by $\text{child}^P \cup \text{old}^P$

(11) **Non-dominated_sorting**($\text{child}^P \cup \text{old}^P$)

(12) **Crowing_distance** ($\text{child}^P \cup \text{old}^P$)

(13) **Select again** the best half elite population as old^P using rank and Cr^{dist}

End While

(14) **Select** the minimum sum of Multiobjective

(15) **Exit**

Output: optimized chromosomes

Algorithm 2: Mixed mutation operator NSGA-II.

consider the relevant requirements of the airborne time-sensitive network when selecting the underlying protocol.

We use two underlying protocols, the Optimized Link State Routing (OLSR) protocol and the Statistical Priority Multiple Access (SPMA) MAC protocol, to build links in our simulation experiments. OLSR modifies the response to network topology changes by changing the maximum transmission interval, and it works in a distributed manner

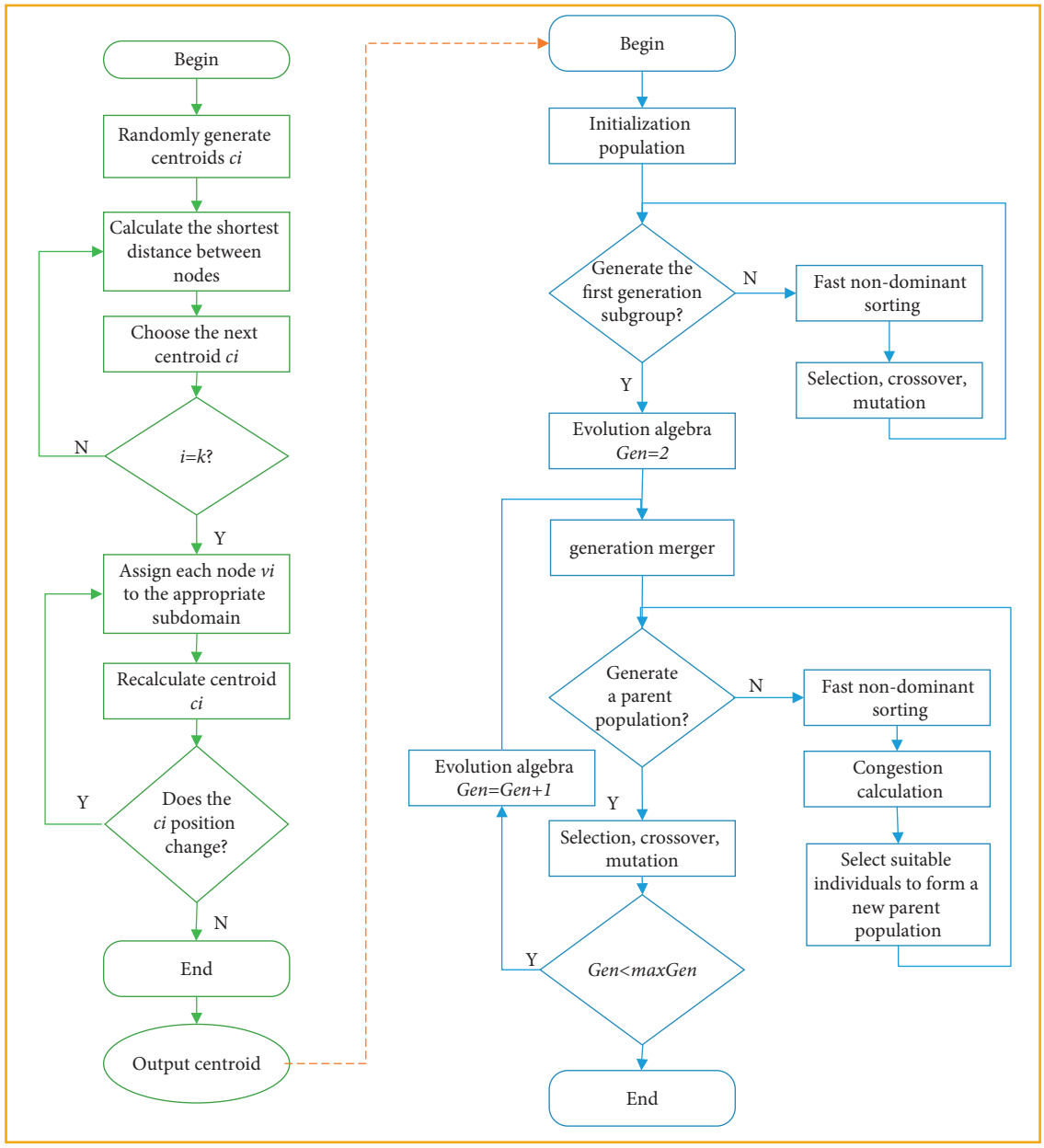

Figure 4: Flow chart of AF-TSP master clock election algorithm.

Table 2: Network simulation parameters.

Parameter name	Parameter value
Network coverage	1000 km × 1000 km
Scenario height	500 km
Number of nodes	100
Node mobility model	Custom path planning
Moving speed	300 m/s~1000 m/s
Channel bandwidth	13.5 KHz
Routing protocol	OLSR
MAC protocol	SPMA
Transmission rate	20 Mbps
Sync message send interval	0.125 s
Phase correction period	0.125 s
Timestamp granularity	4 ns

with low network transmission overhead, so it is the best choice for airborne time-sensitive network scenarios with large scale and high node density. The SPMA protocol uses an access control method based on channel load statistics, defines multiple channel states, allows multiple packets to be sent simultaneously on the channel, and provides a quality of service assurance mechanism. At the same time, the size of the network load has less impact on the probability of successful transmission and latency of high-priority services (time-sensitive tasks), effectively ensuring the success rate and low latency of high-priority data transmission. It shows great advantages in the latest Tactical Targeting Network Technology (TTNT) tactical data chain at this stage, and it especially meets the mission requirements of ATSN in a combat environment.

4.2. Analysis of Master Clock Election Algorithm. In this part, we designed four experiments to evaluate the performance of the proposed algorithm, and test the platform's survivability, communication coverage ability and master clock system capability in the SD-ATSN environment.

4.2.1. Rough Election of Virtual Centroids with K-Means++. In this experiment, the number of virtual centroids is 4, and the number of iterations is 15 times. We intercepted results 5, 10, and 15, as shown in Figure 5. It can be proved that through the clustering of Algorithm 1, the whole combat domain is predominantly divided into four different subdomains. And as the number of iterations increases, the location of the centroids of each domain is continuously adjusted. In Figure 5(c), it can be illustrated that the center of clustering, i.e., the virtual center of mass, is approximately close to the center of the geographic coordinates of each domain.

In Figure 5, we can see that the virtual centroids keep adjusting their positions to be as close to the edge nodes as possible through the continuous iterations of Algorithm 1. This is because if the edge node is too far from the synchronization clock, it will accumulate propagation delay due to the increase of propagation distance and relay nodes, which will lead to too low synchronization accuracy of the edge node. Consequently, we should try to select the node located at the center of the domain as the master clock to balance the distance between the center and edge nodes and minimize the accumulation of the abovementioned offset. Nevertheless, we should acknowledge that the virtual centroid selected by Algorithm 1 is a theoretical location and is a coordinate. This location may have no nodes, only one node, or multiple nodes. Accordingly, based on the virtual centroid, we will perform the next round of fine election by improving the NSGA-II algorithm.

Because the exact selection of Algorithm 2 is run after the results of Algorithm 1 are generated, the results of Algorithm 2 are not directly related to the initial nodes number of

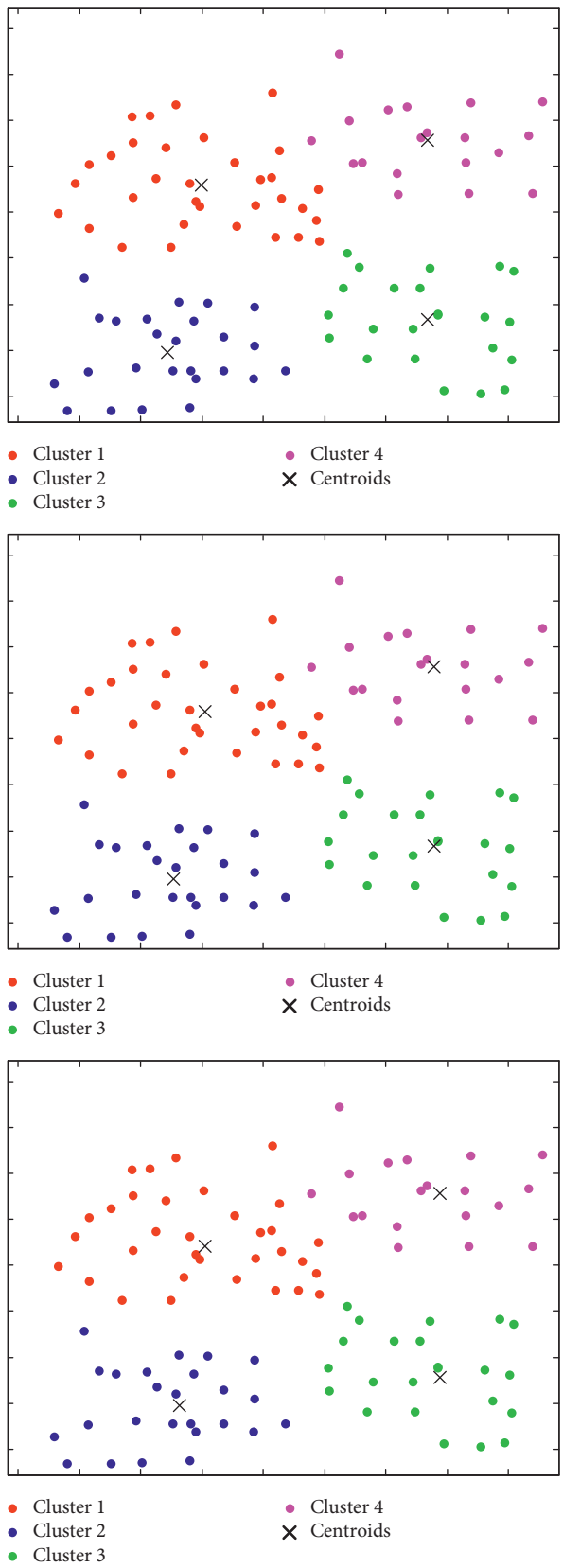

Figure 5: Result of virtual centroid rough election.

simulations, and it is not affected by the increase in the number. Therefore, we further evaluated the impact of the increase in the number of nodes on Algorithm 1, as shown in Table 3.

The space complexity of the K-Means++ algorithm is $O(n)$, and the time complexity is $O(pkn)$, which is approximately linear. Where n is the number of sample points, k is the number of centroids, and p is the number of iterations. Meanwhile, we insert the Silhouette coefficient (SC) to evaluate the effect of the different numbers of nodes on the clustering results. SC combines the cohesion and separation of clusters. It is expressed as

$$SC = \frac{(b-a)}{\max(a,b)}, SC \in [-1, 1], \tag{13}$$

where, a is the average distance of sample nodes within a cluster v_j from other samples in the same cluster, called cohesiveness. b is the average distance between v_j and all samples in the nearest cluster, called separation. The value range of SC is [−1, 1], the closer the distance of samples within the cluster, the farther the distance of samples between the clusters, the larger the silhouette coefficient, the better the clustering effect.

We can see that as the number of nodes increases, the number of iterations increases, which is caused by the increase in computation. Nevertheless, the silhouette coefficient value is stable and close to 1. This means that the increase in the number of nodes does not have much impact on the clustering results, and has little effect on the selection results of the virtual centroids, and accordingly does not have a large impact on the network performance. While confirming that the algorithm clusters nicely, it also indicates that our algorithm has generalization and reliability, which further proves the effectiveness of our algorithm.

Table 3: Comparison of clustering effect with different no. of nodes.

No of nodes	No of iterations	Silhouette coefficient (SC)
50	10	0.876
100	15	0.954
150	17	0.949
200	20	0.937
250	26	0.944
300	28	0.950

4.2.2. Synchronization Precision Test. In the simulation, we set the synchronization model as a first-order model, i.e., we ignore the rate of change of the clock frequency offset with time [39]. The clock model in the simulation is given by the following equation:

$$t_{\text{local}} = t_{\text{lastModefiedLocalTime}} + \left(t_{\text{simulatorTime}} - t_{\text{lastModefiedSimulatorTime}}\right) \times f_{\text{local}}, \tag{14}$$

where t_{local} is the local clock time; $t_{\text{lastModefiedLocalTime}}$ is the local clock time at the time of the last correction. $t_{\text{simulatorTime}}$ is the simulation time; $t_{\text{lastModefiedSimulatorTime}}$ is the simulation time at the time of the last correction; f_{local} is the local clock frequency.

The case presented in Section 2.3 performs the construction of the network topology. There are 8 nodes in the network, node B is the master clock node in the network, the rest of the nodes as master clock slave nodes, directly or indirectly synchronized with the master clock node, we use the gPTP clock synchronization method [14, 39]. The offset of the rest clocks from the master clock is calculated using the following equation. For any node i,

$$\text{offset} = t_{r,i} - GM\left(t_{r,i}\right). \tag{15}$$

$t_{r,i}$ is the time when the synchronization information is received from node i. $GM(t_{r,i})$ is the master clock time calculated from node i at the time $t_{r,i}$.

The slave node will utilize the calculated phase deviation for phase correction.

$$t_{\text{local}} = t_{\text{local}} - \text{offset}. \tag{16}$$

The function $GM(t)$ is the master clock time when the local time of the time synchronization system is t. In the simulation, we overlook the frequency drift of the clock crystal due to environmental factors. Thus, the expression of the function in the ideal case can be obtained as

$$GM\left(t_{r,i}\right) = t_{GM} = t_B. \tag{17}$$

t_{GM} is the current time of the master clock, which is the current time t_B of the master clock node B. The results of the time synchronization offset simulation for the 8 nodes using AF-TSP are as follows.

In Figure 6, we first observe that from node H to node F, their synchronization offset is progressively increasing. This result is caused by the increase of the synchronization distance between the node and the master clock B. The further away from the synchronized master node, the larger the synchronization offset is [25]. This phenomenon has also been mathematically proven theoretically in the literature [39]. This is further evidence that the location of the master clock is crucial and reflects the importance of the virtual centroids we generate through Algorithm 1.

Fortunately, we can also see that AF-TSP retains the synchronization precision within 500 ns despite the inevitable accumulation of synchronization offsets of the nodes. This precision is markedly smaller than the offset produced by existing on-board network synchronization mechanisms, such as Link-16,22 [9]. All these results emphasize the superiority of our protocol and the urgency of electing a master clock.

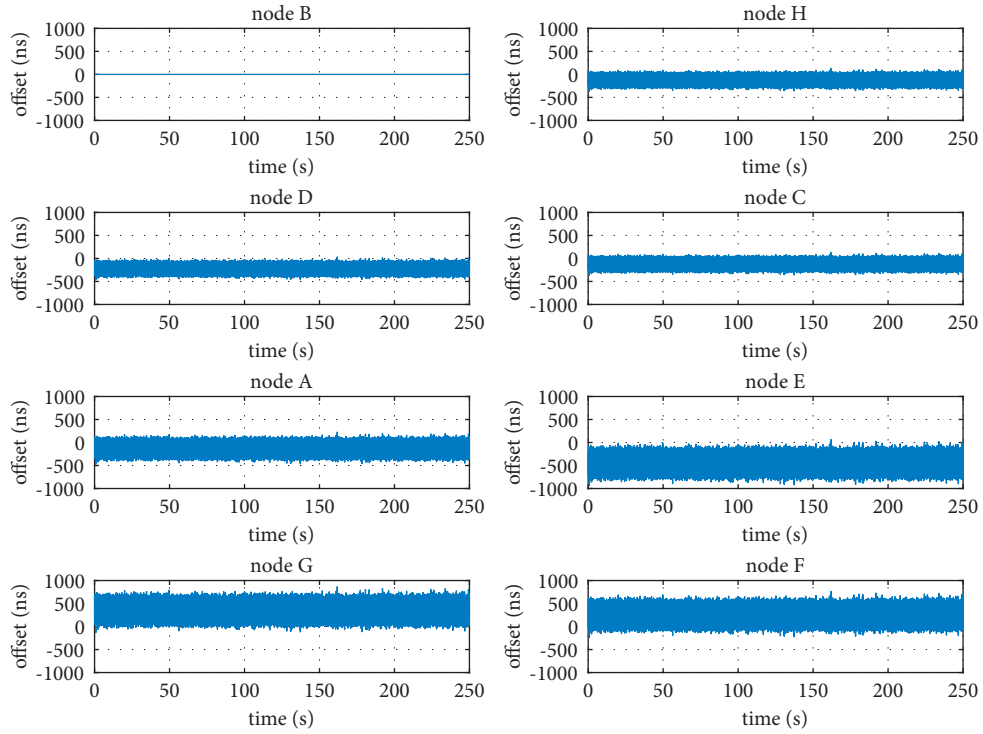

Figure 6: Simulation of synchronization precision of case nodes.

4.2.3. Influence of Once Transmission Data Size on Control Delay. In order to test the control delay, we compared our master clock election algorithm with the best master clock algorithm (BMCA), NSGA-II, and MOEA/D. BMCA is a common algorithm for Ethernet clock synchronization, which selects the best performing clock in the network as the master clock through a large number of interactions and comparisons between nodes, and establishes a synchronous network topology by doing so. MOEA/D are two classical multiobjective optimization algorithms. By increasing the size of the transmitted packets and thus comparing the advantages and disadvantages of the control delay (Equation (11)), the results are exhibited in Figure 7.

As the increase of data, the control delays of the four algorithms disclose an obvious upward trend. This trend is due to the increasing amount of data transmission in the network, and the processing time of the central controller is bound to increase. In addition, the propagation delay of data in the channel will likewise be prolonged. Nevertheless, in Figure 7, the growth rates of the four results are substantially different. Compared with other algorithms, the control delay of BMCA is longer. When the data increases, the delay of BMCA grows speedy. The algorithm requires all nodes to participate in comparing their own performance. It caused a great deal of interactive information and increased communication redundancy. In comparison, NSGA-II, MOEA/D and the algorithms we proposed had slower growth rates. Practice has verified that intelligent algorithms have obvious advantages in dealing with big data. These algorithms are very suitable for large-scale airborne network combat

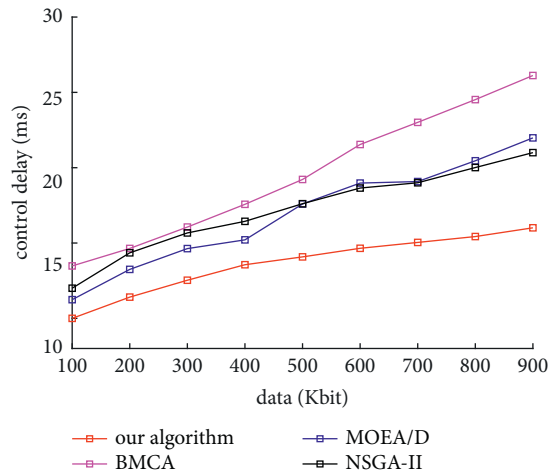

Figure 7: Comparisons of control delay.

scenarios. However, NSGA-II and MOEA/D both ignore the influence of the position of the master clock, which leads to slow convergence and easily fall into local optimum. Thus, the control delay is higher than that of our algorithm.

Our algorithm added the master clock position to the synchronization mechanism. It avoided the offset accumulation caused by the excessive distance between edge nodes and reduced the communication redundancy of repeated message transmission. Hence, our algorithm has a fast convergence rate and the slowest delay growth rate. The control delay is decidedly less than other algorithms. Even if

the data reached 900 Kbit, our proposed algorithm could retain a low delay of about 20 ms, which is far superior to other comparison algorithms.

4.2.4. Influence of Node Failure Probability on Path Reliability. In this experiment, we investigated the impact of node failure probability on path reliability coefficient ((11) after adopting AF-TSP. The experiment compares two cases of overload (data greater than 105 m) and normal load (data less than or equal to 105 m). The results are exhibited in Figure 8.

It is well known that the more reliable the nodes in a synchronous network, the more robust the network will be. In the intensely hostile network environment of the ATSN, robustness is critical and determines whether tasks can be completed in a timely and efficient manner. As we can see in Figure 8, in both network cases, the reliability of the paths inevitably decreases as the probability of node failure increases. Fortunately, even when the node failure probability increases to 14% in the overloaded network, which is the worst case, the synchronous path reliability coefficient of our protocol is still above 0.5. This can satisfy the basic operational requirements [9]. This result reflects the superiority of our protocol and also the urgency of establishing a hot standby mechanism. Enhance the destructive resistance of a single node, thus preventing the degradation of network performance due to node failure. Hence, in the follow-up experiments, we experimentally validated the robustness simulation of the synchronous network. Meanwhile, we expect to verify the superiority of the hot standby mechanism in AF-TSP.

4.2.5. Path Reliability Test. To test the path reliability, we compared our master clock election algorithm with the Ethernet clock synchro-nization algorithm BMCA, the classical multiobjective optimization algorithm NSGA-II, and MOEA/D. As the simulation time increases, the path reliability coefficient (equation (11)) all decreases, and the results are shown in Figure 9.

It can be seen that the algorithms behave differently at the beginning of the simulation, but all guarantee path reliability of 0.9 or more. However, the reliability of BMCA decreases steeply with time, which is due to the fact that this synchronization method relying on long- term interaction between nodes is not suitable for the highly dynamic and strongly adversarial environment of ATSN, and the whole network will collapse only if there is a node failure. Two methods, NSGA-II and MOEA/D, are better compared to BMCA, but the reliability is not as good as our algorithm, and the phenomenon of reliability decreasing with time still occurs, not as stable as our algorithms. Thus, it can be found that our algorithm has high path reliability and is relatively stable and robust.

4.2.6. Robustness Verification of Synchronous Network. We build on the experiments in Section 4.2.2 to further verify the robustness of AF-TSP. This experiment still

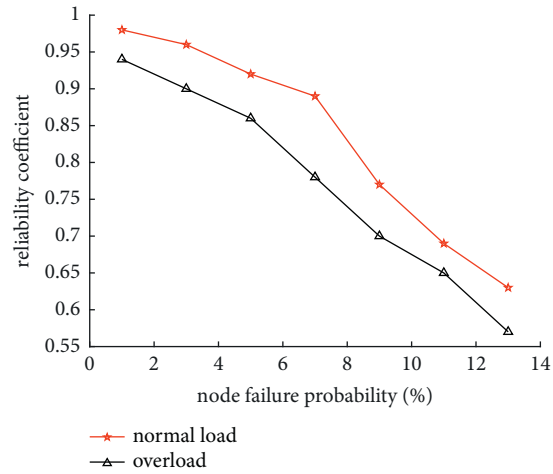

Figure 8: Comparison of path reliability in two cases.

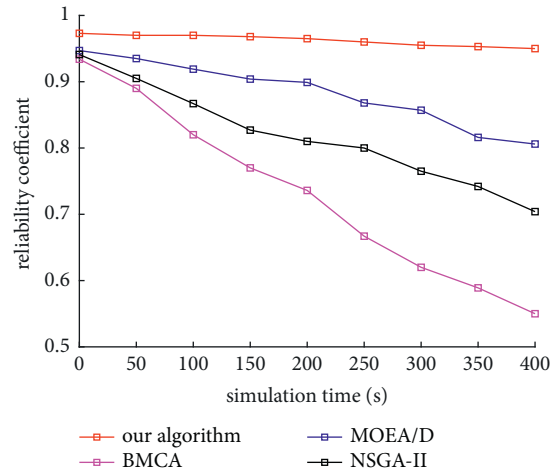

Figure 9: Comparisons of path reliability.

utilizes the topology from the previous 8-node case. The simulation principally evaluates the hot standby mechanism of AF-TSP after the master clock B is destroyed. The simulation results are depicted in Figure 10.

Clock B was damaged at 125 s, and the synchronization time could not be rendered. Consequently, the AF-TSP started the hot standby mechanism. After a short time, each node switched to the communication link maintained by the master clock C in the hot domain. After 125 seconds, clock C commands and synchronizes these remaining nodes until the end of the simulation experiment. Notwithstanding, the offset of some nodes has been modified. In Figure 10, the offset of clock A is increased from 400 ns to 500 ns, but clock E is reduced from about 500 ns to 300 ns. Switched the activated domain master clock, the synchronization path of clock A and E has been altered. So, it can be proved that the synchronization distance remarkably affected the synchronization precision. Fortunately, we can find that the synchronization offset of all nodes is still within 500 ns, which is far beyond the existing

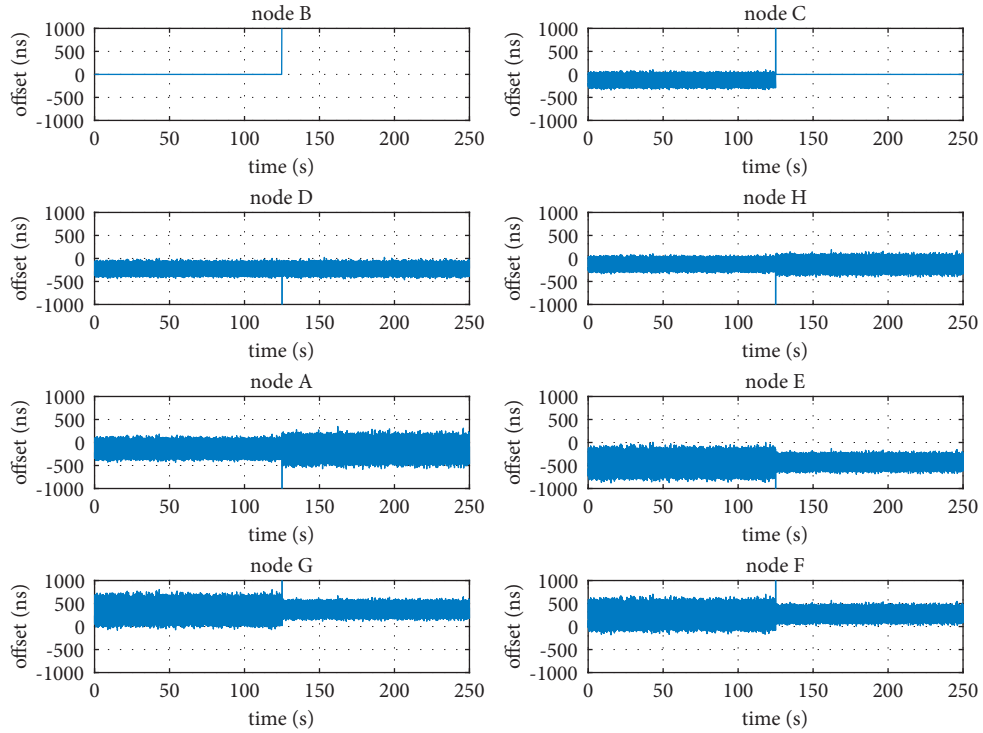

Figure 10: Simulation results of hot standby mechanism.

AN time synchronization protocol. The simulation effectively verified that our AF-TSP mechanism has strong robustness, and can still support the high-precision operation of the synchronous system even after the master clock is damaged.

5. Conclusion

To ensure high precision time synchronization in airborne time-sensitive networks, we designed the AF-TSP time synchronization protocol. It contains a new network architecture SD-ATSN and a master clock selection algorithm. The architecture integrates the SDN paradigm on the airborne network to meet the requirements of accurate communication with low latency, bounded jitter, and high reliability in the future airborne environment. At the same time, we innovatively propose a hot standby mechanism. Multiple master clocks located in different operational domains are backed up to each other and activated on demand. Subsequently, a domain-wide hot standby is initiated, which strengthens the resistance to destruction of a single master clock and assured the robustness of the synchronization network. We propose a new master clock election algorithm, which first utilizes improved K-means++ for rough elections. Based on the rough election, we insert a hybrid mutation operator to enhance the convergence of the multiobjective optimization algorithm NSGA-II with control delay, clock accuracy, and path reliability coefficient as the optimization objectives. Simulation results demonstrate that our AF-TSP protocol has advantages in synchronization precision, network communication delay, reliability, and robustness.

Data Availability

Data are available at the following link https://www.unb.ca/cic/datasets/vpn.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Liu, Y. Shi, and Z. M. Lullah, "Space-air-ground integrated network: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, 2018.

[2] J. Zhang, "Aeronautical ad-hoc networking for the internet-above-the-clouds," *Proceedings of the IEEE*, vol. 107, no. 5, 2019.

[3] Ieee Future Networks, *IEEE 5G and beyond Technology Roadmap-WHITE PAPER*, ieee, New York, NY, USA, 2017.

[4] J. Rexford, "The compositional architecture of the Internet," *Communications of the ACM*, vol. 62, no. 3, pp. 78–87, 2019.

[5] M. Cheon, H. Baek, and J. Lim, "Dynamic relay node selection scheme for multi-hop time synchronization in Link-16," in *Proceedings of the Military Communications Conference*, pp. 1329–1334, Tampa, FL, USA, October, 2015.

[6] K. H. K. H. L. Kim, "Technology trends analysis of airborne network," *International Journal of Future Generation Communication and Networking*, vol. 10, no. 9, pp. 299–310, 2016.

[7] B. N. Bow-Nan Cheng, F. J. Block, B. R. Hamilton et al., "Design considerations for next-generation airborne tactical networks," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 138–145, 2014.

[8] B. Zheng, L. I. Yong, and W. Cheng, "Design and analysis of a multi-channel S-ALOHA protocol for airborne tactical

networks," *Systems Engineering and Electronics*, vol. 41, no. 12, pp. 2864–2871, 2019.

[9] M. Darpa, "Technology programs list," 2010, https://DARPA (newatlas.com).

[10] H. Zhao, H. Wang, and W. Wu, "Deployment algorithms for UAV airborne networks toward on-demand coverage," *IEEE Journal on Selected Areas in Communications*, vol. 36, 2018.

[11] X. Jiang, S. G. Hossein, and F. Gabor, "Low-latency networking: where latency lurks and how to tame it," *Proceedings of the IEEE*, vol. 107, pp. 1–27, 2018.

[12] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Mobile edge computing and networking for green and low-latency internet of things," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 39–45, 2018.

[13] X. Zuo, Y. Cui, M. Wang, T. Xiao, and X. Wang, "Low-latency networking: architecture, techniques, and opportunities," *IEEE Internet Computing*, vol. 22, no. 5, pp. 56–63, 2018.

[14] T. Adame and M. Carrascosa, "Bellalta B time-sensitive networking in IEEE 802.11be: on the way to low-latency WiFi 7," *Sensors*, vol. 21, 2019.

[15] E. Khorov, I. Levitsky, and I. F. Akyildiz, "Current status and directions of IEEE 802.11be, the future wi-fi 7," *IEEE Access*, vol. 8, Article ID 88664, 2020.

[16] J. Prados-Garzon, T. Taleb, and M. Bagaa, "Optimization of flow allocation in asynchronous deterministic 5G transport networks by leveraging data analytics," *IEEE Transactions on Mobile Computing*, vol. 99, p. 1, 2021.

[17] N. Sertba, D. Ergen, and M. Fischer, "SDN-based Self-Configuration for Time-Sensitive IoT Networks," in *Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN)*, Edmonton, AB, Canada, October, 2021.

[18] Z. Li, "Networks with CoMP for URLLC and time sensitive network architecture," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 947–959, 2019.

[19] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, 2004.

[20] Z. Li, H. Wan, and Y. Deng, "Time-triggered switch-memory-switch architecture for time-sensitive networking switches," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, p. 1, 2018.

[21] X. Huan, K. S. Kim, and S. Lee, "Improving multi-hop time synchronization performance in wireless sensor networks based on packet-relaying gateways with per-hop delay compensation," *IEEE Transactions on Communications*, vol. 69, no. 99, p. 1, 2021.

[22] S. Kehrer, O. Kleineberg, and D. Heffernan, "A comparison of fault-tolerance concepts for IEEE 802.1 Time Sensitive Networks (TSN)," in *Proceedings of the Emerging Technology & Factory Automation*, September, 2014.

[23] J. T. Tsai, "Transmission rate scheduling and stopping time for time-sensitive multicast stream traffic in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 31, 2014.

[24] A. Nasrallah, A. S. Thyagaturu, and Z. Alharbi, "Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS)," *IEEE Access*, vol. 7, no. 99, p. 1, 2019.

[25] C. Park, J. Lee, and T. Tan, "Simulation of scheduled traffic for the IEEE 802.1 time sensitive networking," *Information Science and Applications (ICISA)*, Springer, Singapore, 2016.

[26] X. Huan and K. S. Kim, "On the practical implementation of propagation delay and clock skew compensated high-precision time synchronization schemes with resource-constrained sensor nodes in multi-hop wireless sensor networks," *Computer Networks*, vol. 166, no. Jan.15, pp. 106959.1–106959.8, 2020.

[27] M. Gundall, "Integration of 5G with TSN as prerequisite for a highly flexible future industrial automation time synchronization based on IEEE 802.1AS," in *Proceedings of the 2020 IEEE 46th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, IEEE, Singapore, October, 2020.

[28] S. Song, H. Park, B.-Y. Choi, T. Choi, and H. Zhu, "Control path management framework for enhancing software-defined network (SDN) reliability," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 302–316, 2017.

[29] A. Mahmood, R. Exel, and T. Sauter, "Delay and jitter characterization for software-based clock synchronization over WLAN using PTP," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1198–1206, 2014.

[30] S. T. Q. H. B. Said, "SDN-based configuration solution for IEEE 802.1-time sensitive networking (TSN)," *ACM SIGBED Review*, vol. 1, no. 16, pp. 27–32, 2019.

[31] I. C. Society, "IEEE standard for local and metropolitan area networks—timing and synchronization for time," *IEEE Std 802.1AS-2011*, pp. 1–421, 2020.

[32] N. Lyu, C. Liu, and K. Chen, "A method for centralized control network deployment of aeronautic swarm[J]," *Aeronautical Journal*, vol. 39, no. 7, 2018.

[33] N. Lv, X. Zou, and K. Chen, "An update strategy for a software-defined airborne network of the aviation swarm[J]," *Systems Engineering and Electronics Technology*, vol. 42, no. 1, p. 11, 2020.

[34] R. Salazar, T. Godfrey, and N. Finn, *White paper - utility applications of time sensitive networking white paper*, IEEE, New York, NY, USA, 2019.

[35] S. K. Mydhili, S. Periyanayagi, and S. Baskar, "Machine learning based multi scale parallel K-means++ clustering for cloud assisted internet of things," *Peer-to-peer Networking and Applications*, vol. 13, 2019.

[36] V. Kanwar and A. Kumar, "Multi-objective optimization-based DV-hop localization using NSGA-II algorithm for wireless sensor networks," *International Journal of Communication Systems*, vol. 33, no. 11, p. 4431, 2020.

[37] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, Part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[38] M. P. Behera, A. Sarangi, and D. Mishra, "Analysis of Gaussian and Cauchy mutations in K-means particle swarm optimization algorithm for data clustering," *Technical Advancements of Machine Learning in Healthcare*, vol. 936, 2021.

[39] Y. Wang, *Research on Key Technologies for High-Precision Clock Synchronization in Time-Sensitive Networks*, Xi'an University of Electronic Science and Technology, Xi'an, China, 2019.