Sensing and Communication Integrated Fast Neighbor Discovery for UAV Networks

WEI Zhiqing, ZHANG Yongji, JI Danna, LI Chenfei
(Beijing University of Posts and Telecommunications, Beijing 100876, China)

DOI: 10.12142/ZTECOM.202403009
https://kns.cnki.net/kcms/detail/34.1294.TN.20240921.1450.002.html, published online September 23, 2024
Manuscript received: 2024-08-06

Abstract: In unmanned aerial vehicle (UAV) networks, the high mobility of nodes leads to frequent changes in network topology, which brings challenges to the neighbor discovery (ND) for UAV networks. Integrated sensing and communication (ISAC), as an emerging technology in 6G mobile networks, has shown great potential in improving communication performance with the assistance of sensing information. ISAC obtains the prior information about node distribution, reducing the ND time. However, the prior information obtained through ISAC may be imperfect. Hence, an ND algorithm based on reinforcement learning is proposed. The learning automaton (LA) is applied to interact with the environment and continuously adjust the probability of selecting beams to accelerate the convergence speed of ND algorithms. Besides, an efficient ND algorithm in the neighbor maintenance phase is designed, which applies the Kalman filter to predict node movement. Simulation results show that the LA-based ND algorithm reduces the ND time by up to 32% compared with the Scan-Based Algorithm (SBA), which proves the efficiency of the proposed ND algorithms.

Keywords: unmanned aerial vehicle networks; neighbor discovery; integrated sensing and communication; reinforcement learning; Kalman filter

Citation (Format 1): WEI Z Q, ZHANG Y J, JI D N, et al. Sensing and communication integrated fast neighbor discovery for UAV networks [J]. ZTE Communications, 2024, 22(3): 69 - 82. DOI: 10.12142/ZTECOM.202403009
Citation (Format 2): Z. Q. Wei, Y. J. Zhang, D. N. Ji, et al., "Sensing and communication integrated fast neighbor discovery for UAV networks," *ZTE Communications*, vol. 22, no. 3, pp. 69 - 82, Sept. 2024. doi: 10.12142/ZTECOM.202403009.

1 Introduction

Recently, integrated sensing and communication (ISAC), which improves hardware and spectrum efficiency, has attracted wide attention in both academia and industry. For unmanned aerial vehicle (UAV) networks, ISAC can save limited space and power, minimize the payload, and increase the endurance of UAVs. Therefore, ISAC-driven UAV networking becomes vital to achieve better performance of UAV networking, especially for neighbor discovery (ND). Obtaining the prior knowledge through ISAC can speed up neighbor discovery. The estimated number and directions of neighbors can be obtained through ISAC to avoid invalid transmission in advance. Based on the prior knowledge, the scan-based algorithm (SBA) is used for neighbor discover[1-2]. In Ref. [3], LIU et al. used the sensing information obtained by a double-sided phased array radar to assist neighbor discovery and proposed a new algorithm that applies two beams to transmit and receive independently or simultaneously. In Ref. [4], based on the accuracy of the prior knowledge and the response mechanism, WEI et al. proposed four radar-assisted ND algorithms, including Reply and Non-Stop (RnS), Non-Reply and Non-Stop (nRnS), Reply and Stop (RS), and Non-Reply and Stop (nRS). In Ref. [5], a 79-GHz millimeter-wave radar was used to detect the location and mobility of neighbors and the identity information of neighbors via 5.9-GHz braodcasting was obtained. The UAV node identifies its neighbors by combining the information obtained from the two frequency bands.

Besides ISAC-based ND algorithms, there are other methods to acquire prior knowledge. In Ref. [6], KHAMLICHI et al. applied energy detectors to distinguish whether the receiving node was in a collision state or an idle state. If the receiving node is in the collision state, it will switch to the collision-resolving listening mode and the transmitting node will switch to the collision-resolving re-transmitting mode until the receiving node successfully receives the data packets transmitted by at least two transmitting nodes. GAO et al. proposed an anti-collision ND protocol that avoids beacon collisions through carrier sensing[7]. The node can decide whether to send a bea-

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant No. 2024ZCJH01, in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62271081, and in part by the National Key Research and Development Program of China under Grant No. 2020YFA0711302.

con according to the prior knowledge of the channel state obtained by carrier sensing, which can effectively avoid collisions. SUN et al. proposed an ND protocol based on pre-handshake[8]. The pre-handshake is carried out by adding small sub-slots before the ordinary time slots, which allows nodes to learn the activities of their neighbors in advance to reduce collisions. In Ref. [9], a dual-band system was applied, where one frequency band was used to obtain prior knowledge about neighbors, and the other frequency band was used for neighbor discovery. The paper also discussed how to maximize the efficiency of neighbor discovery in the presence of prior knowledge. Except for applying prior knowledge to improve the performance of algorithms, some researchers consider introducing reinforcement learning algorithms to find an optimal ND strategy by interacting with the environment[10 - 11]. Optimizing the classical ND algorithms by dynamically adjusting the parameters are also considered[12 - 13]. However, these algorithms are all direct ND algorithms. As the number of nodes increases, some researchers have focused on ND algorithms based on gossip, in which nodes can indirectly discover neighbors by exchanging neighbor lists with their neighbors[14].

The above studies still have several limitations: 1) When they apply prior knowledge to improve neighbor discovery, they ideally assume that the sense information is perfect. However, ISAC may make the radar detection range smaller than the communication range and lead to the problem of incomplete prior knowledge obtained by nodes; 2) only the initialization phase of the neighbor discovery is studied in the above papers. Some temporary events in the neighbor maintenance phase can easily destroy the node connection, such as node failure, energy exhaustion, and power increase. As neighbor discovery is a continuous process, the maintenance phase should be studied with the initialization phase together. In this paper, reinforcement learning is introduced to solve the problem of incomplete prior knowledge. A Kalman filter is applied to predict the movement of nodes in the neighbor maintenance phase, which can speed up neighbor discovery in the maintenance phase. The main contributions of this paper are summarized as follows.

1) We introduce a reinforcement learning algorithm to address the incompleteness of prior information obtained by ISAC, which occurs in the ND problem. The problem is mapped as a multi-agent learning model, and an ND strategy based on learning automata (LA) is proposed. The algorithm estimates whether the neighbor nodes have been completely discovered according to the distribution of nodes and the partial information obtained by detection. We design a linear reward and punishment mechanism for LA. The simulation results show that when the ratio of radar detection range to communication range is set to 0.6, the time efficiency of the algorithm based on LA can be increased by 32%.

2) Due to the high mobility of UAVs, the neighbor list needs to be continuously updated to maintain the constructed network topology. Unlike the existing ND algorithms that update the entire network, we separate the neighbor maintenance phase from the traditional initialization phase. In the neighbor maintenance phase, an efficient ND algorithm is designed, where the Kalman filter is introduced based on the prior knowledge obtained in the initialization phase. The switching mechanism between the initialization phase and the maintenance phase is also designed. When an error occurs in node prediction or the duration of the neighbor maintenance phase is greater than a threshold, the maintenance phase will be switched to the initialization phase to ensure the accuracy of the network topology. The simulation results prove that the networking efficiency with this algorithm is much higher than that of traditional neighbor discovery.

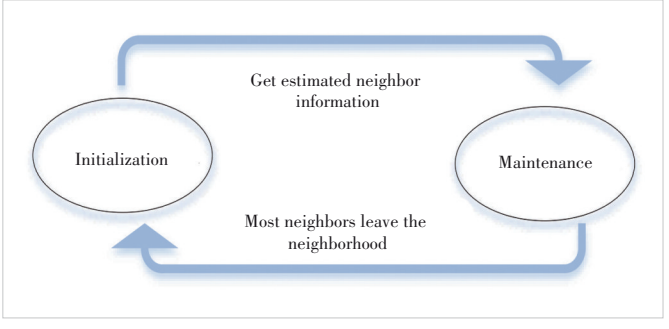
The rest of this paper is organized as follows. In Section 2, the system model and assumptions are described. In Section 3, considering the gap between the radar detection range and communication range, we introduce the reinforcement learning algorithm and design the learning strategy. The upper and lower bounds of time slots required to discover all neighbors are derived in the algorithm. Moreover, the impact of parameter settings on the performance is simulated and analyzed in two-dimensional and three-dimensional UAV networks. In Section 4, the Kalman filter is introduced, which greatly improves the efficiency of neighbor discovery through predictions. In Section 5, the algorithms proposed in Sections 3 and 4 are simulated and analyzed. Section 6 concludes this paper.

2 System Model and Assumptions

2.1 System Model

In UAV networks, neighbor discovery needs to be carried out periodically due to the mobility of nodes and its possible failure. Neighbor discovery can be divided into two procedures: the initialization and maintenance phases[15]. Previous works performed these two phases with the same scheme[6, 16 - 20]. Considering the mobility of nodes and the highly dynamic topology of UAV networks, we use different strategies for the two phases and design their switching mechanism. Fig. 1 shows the transformation of the two phases.

Initialization phase is a traditional stage for neighbor dis-



▲Figure 1. Two-stage conversion of neighbor discovery

covery. In this stage, we introduce LA for the neighbor discovery. Nodes are regarded as agents with learning ability, and automatic learning machines are applied to change policies to speed up the discovery process. More detailed information is introduced in Section 3.

Maintenance phase refers to the stage in which the topology needs to be updated due to possible failures, which adapts to the highly dynamic topology due to the mobility of UAVs. At this stage, nodes need a more efficient scheme to cope with the rapid changes in topology with limited energy. In Section 4, a neighbor maintenance method based on the Kalman filter is proposed to accelerate the construction of network topology by using the speed and location of the neighbors.

2.2 Neighbor Discovery Assumptions

Undirected graphs $G = (V, E)$ are here used to describe the neighbors of nodes in the network, where $V = (V_1, V_2, \cdots, V_N)$ is the set of nodes distributed uniformly and randomly and E is the set of link edges. A link edge $(i, j) \in E$ represents that two nodes (i, j) are one-hop neighbors of each other. Each node is equipped with a directional antenna. For a two-dimensional model, the neighborhood of a node is a circle with radius R_s. The antenna width is α and the neighborhood can be divided into $k = 2\pi/\alpha$ non-overlapping beams. For a three-dimensional model, its neighborhood is a sphere with a radius R_s. The vertical and horizontal widths of a beam of the antenna are both α, and the neighborhood can be divided into $k = 2\pi^2/\alpha^2$ non-overlapping beams. Moreover, the following definitions and assumptions are made.

1) Identification: All nodes are distinguished by unique identification (ID), which can be a media access control (MAC) address.

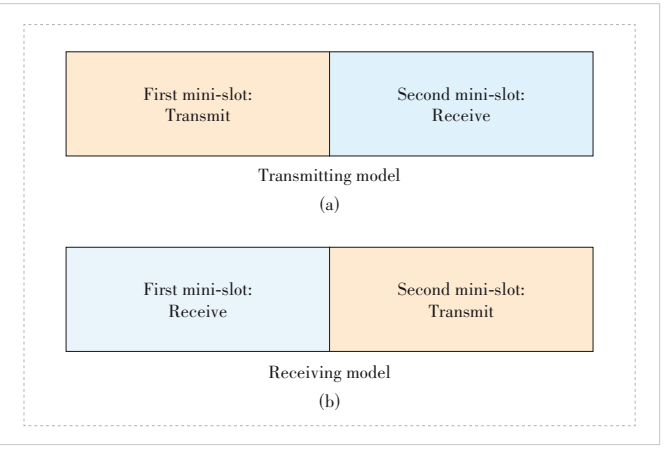
2) Resource: All nodes will transmit messages on the same frequency band with the same power.

3) Time slot division: The network adopts the synchronous time slot division model[21], which divides time into time slots of the same length. The algorithm proposed in this paper is based on a two-way handshake mechanism. Therefore, each time slot is divided into two mini-slots, as shown in Fig. 2. In the second mini-slot, after receiving the Hello packet, the receiving node replies with an acknowledgment packet (ACKP) with probability 1.

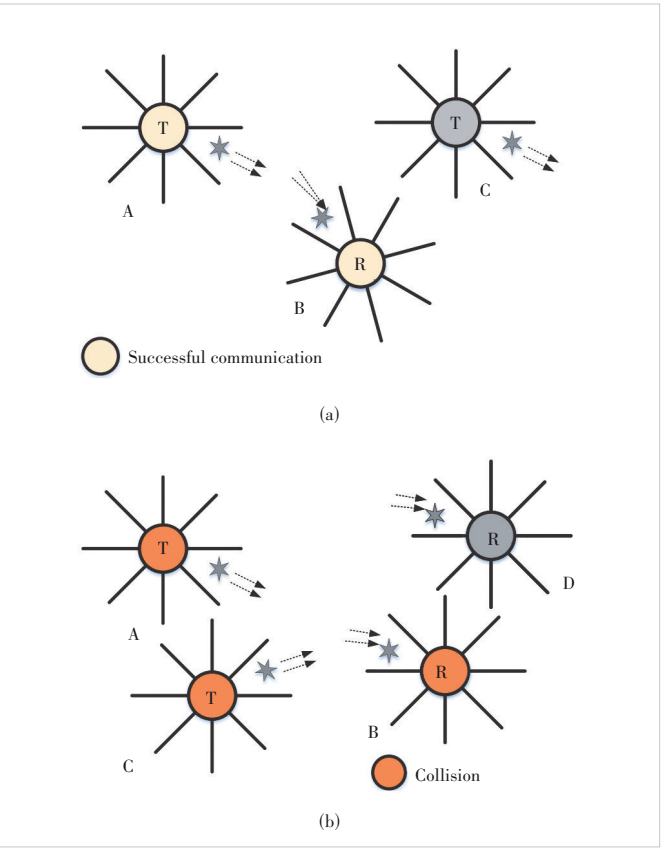
4) Half-duplex: The nodes work in the half-duplex mode[22]. A node will be in a transmitting or receiving state at any time slot. In this paper, a node has three modes: transmit (T), receive (R), and idle (I)[23]. Unlike being in the active states (T/R), nodes in the idle state save power consumption[24].

5) Successful communication: Nodes A and B are neighbors to each other only when they are within the one-hop communication range of each other. As shown in Fig. 3(a), successful communication between a pair of nodes requires that antennas are in a complementary state (one node transmits and the other receives) and point to each other at the same time. It is supposed that the transmitting direction of node A is θ' and the receiving direction of node B is θ''. For the directional transmission and directional reception mode, the following conditions must be met for successful communication: a) The antenna patterns of the two nodes are complementary; b) the direction meets $\theta' = (\theta'' + \pi) \bmod 2\pi$; c) no collision occurs during the interaction.

6) Collision: When a node receives two or more Hello packets, a data packet conflict has occurred. As shown in Fig. 3(b),



▲Figure 2. Time slot division



▲Figure 3. Successful communication and data packet collision

a collision will occur when two nodes send signals to one node at the same time.

7) ISAC: Each node is equipped with a set of transceivers to send and receive ISAC signals. A node transmits the ISAC signal that has detection capabilities and carries the Hello packet. The receiver can process the echo signals and the communication data packets.

8) Neighbor information: Each node maintains an antenna beam number list (ABNL) and a neighbor information list (NIL). Maintained by node i, $\forall i \in N$, the ABNL is denoted as $i, \forall i \in N$, $\xi_i^k \in \{0,1\}$, where ξ_i^k denotes the existence of node i in the k-th beam direction according to the result of radar detection. If the radar detects that there are neighbors in the k-th beam, ξ_i^k is marked as 1. Otherwise, it is marked as 0. Maintained by node j, $\forall j \in N$, the NIL is expressed as $I_j = \{I_1, I_2, \cdots, I_N\}$, which maintains the information of one-hop neighbors.

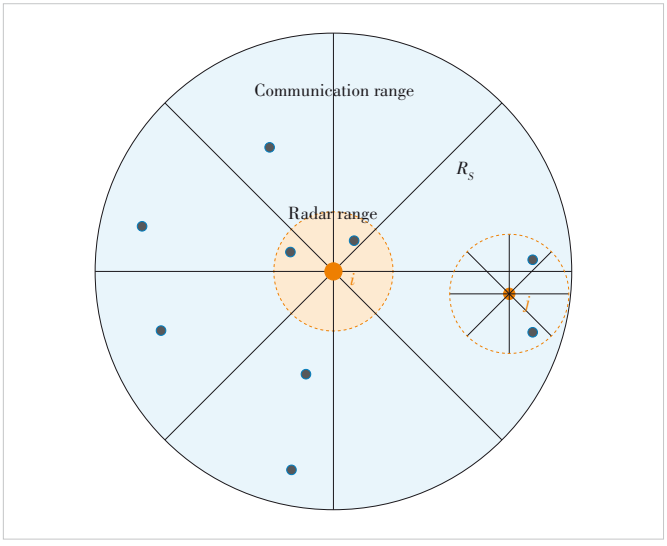
With an ISAC signal, the radar detection range is smaller than the communication range; $\eta = R_r/R_c$ is the ratio of radar detection range to communication range. As shown in Fig. 4, nodes can only detect part of neighborhood information[25].

The ND algorithm uses LA to solve this problem. The ND process is modeled as autonomous learning. Nodes are regarded as agents, and the neighborhood is regarded as the environment to be learned. Based on the prior knowledge obtained by each time slot, the node optimizes the behavior of selecting the beam in the ND algorithm[25].

3 Neighbor Discovery Algorithm Based on Learning Automata

3.1 Learning Automata Node

In a non-stationary environment, the reward distribution is related to time. So the learning goal of LA is to adjust behavior based on each time slot[26]. This paper is based on a non-stationary environment and applies a finite state automaton (FSA) to model neighbor discovery.



▲Figure 4. Communication and radar detection range

In each time slot, LA selects an action and receives an enhanced signal β_i from the environment. β_i describes whether the selected action is beneficial or unfavorable. LA observes the input signal and updates the action probability distribution vector $\boldsymbol{P}_i$ to enable a higher probability of successful neighbor discovery in the next time slot. Moreover, in the LA model, there are the following definitions.

Antenna mode $M_i = \{T, R\}$: In each time slot, LA_i independently selects the transmission mode (T) with probability p_t and selects the reception mode (R) with probability $p_l = 1 - p_t$.

Action space $A_i = \{a_1, a_2, \cdots, a_k\}$: Action $a_k = 1$ means that LA_i transmits or receives in beam k.

Action probability distribution vector $\boldsymbol{P}_i = \{p_i^1, p_i^2, \cdots, p_i^k\}$: p_i^m is the probability that LA_i selects beam m. In each time slot, LA_i selects a beam according to $\boldsymbol{P}_i$.

Reinforcement signal $\beta_i(t)$: The reinforcement signal $\beta_i(t) \in \{0,1\}$ is obtained by LA_i in time slot t.

Probability update: The action probability distribution vector is updated to $\boldsymbol{P}_i(t+1)$ according to $\beta_i(t)$ in time slot t.

3.2 Probability Update Strategy

According to the range of radar detection and communication, neighbors are divided into two categories, including the nodes both in the radar detection range and communication range, and the nodes in the communication range and outside the radar detection range. Two lists are defined to describe these two types of neighbors: the radar neighbor list (RNL) and communication neighbor list (CNL). The RNL of node i is a matrix $\boldsymbol{R}_{N\times K}^i$ with dimension $N \times K$, which records the neighbors in the RNL of node i. For example, if node f exists in the RNL of node i, the f-th row of matrix $\boldsymbol{R}_{N\times K}^i$, namely vector $\boldsymbol{R}_{f\times K}^i$, represents the discovery of node f in the beams of node i. When node f is in beam g with $g \in \{1, 2, \cdots, K\}$, $\boldsymbol{R}_{f\times K}^i$ is

$$\boldsymbol{R}_{f\times K} = \begin{bmatrix} 0 & 0 & \underset{g}{1} & \cdots & 0 \end{bmatrix}_{1\times K}. \tag{1}$$

CNL of node i is denoted by $C_{N\times K}^i$ and $\boldsymbol{R}_{N\times K}^i \in C_{N\times K}^i$. The vector $\boldsymbol{R}_{1\times K}^i = [\, r_{11} \quad \cdots \quad r_{1K} \,]$ is the number of neighbors detected by node i in the second mini-slot. The radar can detect the number and the position of nodes in the current beam[4].

LA_i optimizes the action selection by updating the action probability distribution vector $\boldsymbol{P}_i$[27]. The update strategy is as follows.

Environmental interaction: At time slot t, LA_i selects an action a_m, calculates the environmental feedback information, gives an enhanced signal $\beta_i(t)$, and updates it to $\boldsymbol{P}_i(t+1)$ according to Eqs. (3) or (4). When the radar detects the same beam multiple times, the inaccurate detection in the previous

time slot can be corrected by updating matrix $\boldsymbol{R}^i_{1\times K}$.

Reward and punishment signal β_i^k: When LA_i selects beam k and the transmission mode, the ISAC signal is transmitted in the first mini-slot. In the second mini-slot, LA_i updates $C^i_{N\times K}$ and $\boldsymbol{R}^i_{N\times K}$ according to radar detection and the communication result. If there are potential neighbors in beam k, assign $\beta_i^k = 0$. For example, if a neighbor is in beam k in RNL and the ACK packet is not received in the second mini-slot, there is a potential neighbor in beam k and should be rewarded. If there are no potential neighbors in beam k, assign $\beta_i^k = 1$.

In UAV networks, the number of neighbors in each beam is mostly 0 and 1[4]. In a beam, when a node appears in the radar detection range, the probability of other nodes appearing in the communication range outside the radar detection range is small. $\boldsymbol{P}^i_{\mathrm{pte}}$ is the probability that there are still neighbors for node i in beam m when all the nodes in the radar detection range are discovered, with expression as follows.

$$\boldsymbol{P}^i_{\mathrm{pte}} = \boldsymbol{P}^i_{\mathrm{pte}}\left(N_1^R\right) + \boldsymbol{P}^i_{\mathrm{pte}}\left(N_2^R\right) + \cdots = \frac{P_{B_2}+\cdots+P_{B_m}}{P_{B_1}+P_{B_2}+\cdots+P_{B_m}} + \frac{P_{B_3}+\cdots P_{B_m}}{P_{B_2}+\cdots+P_{B_m}} + \cdots, \tag{2}$$

where $\boldsymbol{P}^i_{pte}(N_1^R)$ represents the probability that the sum of the m-th column of $C^i_{N\times K} - \boldsymbol{R}^i_{N\times K}$ is greater than 1 when $r_{1m} = 1$, and $P^i_{\mathrm{pte}}(N_2^R)$ represents the probability that the sum of the m-th column of $C^i_{N\times K} - R^i_{N\times K}$ is greater than 1 when $r_{1m} = 2$.

According to the reinforcement signal, the update of the action probability distribution vector is divided into the following two cases.

When $\beta_i^k = 0$,

$$\boldsymbol{P}_i{}^k(t+1) = \begin{cases} \boldsymbol{P}_i{}^k(t) + \phi(\cdot)\left(1 - \boldsymbol{P}_i{}^k(t)\right), a_i(t) = a_i \\ \left(1-\phi(\cdot)\right)\boldsymbol{P}_i{}^k(t), a_i(t) \neq a_i \end{cases}, \tag{3}$$

where $\phi(\cdot)$ is a function of the number of potential neighbors.

$$\phi(\cdot) = \begin{cases} \gamma_1 n_i(t), & n_i = 1 \\ \gamma_2 n_i(t), & n_i > 1 \end{cases}, \tag{4}$$

where γ_1 represents the reward coefficient of a single potential neighbor, γ_2 represents the reward coefficient of multiple potential neighbors, and $n_i(t)$ represents the number of potential neighbors.

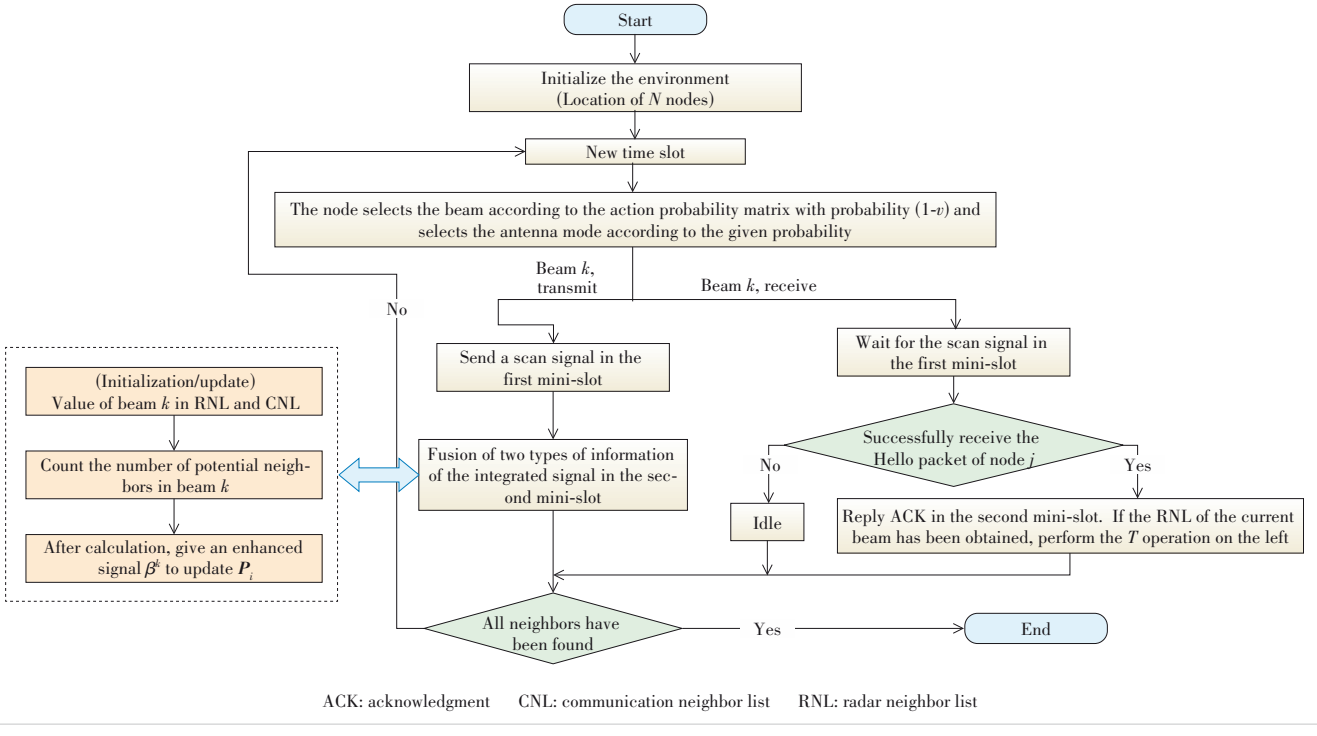
When $\beta_i^k = 1$,

$$\boldsymbol{P}_i{}^k(t+1) = \begin{cases} (1-\mu)\boldsymbol{P}_i{}^k(t), a_i(t) = a_i \\ \dfrac{\mu}{K-1} + (1-\mu)\boldsymbol{P}_i{}^k(t), a_i(t) \neq a_i \end{cases}, \tag{5}$$

where μ is the penalty coefficient.

3.3 ND Process

The ND algorithm based on LA is depicted as follows, and its flow chart is shown in Fig. 5.



▲Figure 5. Flow chart of the ND algorithm

When neighbor discovery starts, LA_i initializes its action probability distribution vector $\boldsymbol{P}_i(t=0)$, $\boldsymbol{P}_i^k = 1/K, \quad \forall i \in N$, and assigns matrices $\boldsymbol{R}^i_{1\times K}$, $\boldsymbol{R}^i_{N\times K}$, and $\boldsymbol{C}^i_{N\times K}$ to zero.

Repeat the following steps for each time slot.

LA_i randomly selects a_k with probability υ or selects a_k according to the action probability distribution vector $\boldsymbol{P}_i$ with probability $1-\upsilon$.

In the second mini-slot, when $M_i = T$, update $\boldsymbol{R}^i_{N\times K}$ according to radar detection, update $C^i_{N\times K}$ according to the communication result, and update $\boldsymbol{P}_i$ according to the reinforcement signal. When $M_i = R$, update $C^i_{N\times K}$ according to the communication result in the first mini-slot. If the Hello packet is received, it will reply with an ACK packet, otherwise it will be in the idle state.

3.4 Time Convergence Upper and Lower Bound Analysis

In the learning-based ND scheme, the action probability distribution vector changes with time. This section gives the probability update strategy when $\eta = 1$. When the current beam k does not receive an echo, let $\mu = 1$ in Eq. (5), otherwise no update is made.

3.4.1 Discovery Probability Analysis

The ND process can be described as a Bernoulli experiment in all beams[12].

It takes J frames for a node u to find its neighbor v with probability P_J^v. K is the number of beams. In the two-dimensional model, the node density is $\sigma = N/\pi R_s^2$. In the three-dimensional model, the node density is $\sigma = 3N/4\pi R_s^3$. The average number of beams is $m = N/K$. The probability that any node is transmitting in any time slot is p_t.

After time slot t, the number of neighbors discovered by node u is $\chi(t)$; $(i+1)_{\text{th}}$ means the direction in which the beam is likely to point from $\chi(t) = i$ to $\chi(t) = i+1$. Suppose there are a total of m_j neighbors of node u in beam j. $X = p_t/K$ represents the probability that m_j (including v) neighbors in beam j each send to node u. $X_j = p_r/K$ represents the probability of node u receiving in beam j.

Lemma 1: The probability of finding a specific neighbor when there is a collision is

$$P_{ro} = XX_j(1-X)^{m_j-1}. \tag{6}$$

Based on Lemma 1, in the two-way handshake, if node u discovers its neighbor v in time slot t, the antenna beams of u and v need to point to each other with probability $1/K^2$. The antenna patterns of (u,v) are complementary to each other, which is (T,R) or (R,T) with probability $2\times p_t\times(1-p_t)$. In addition, the other $N-1$ neighbors must not interfere with the ND process. When other neighbors select the transmitting mode, the probability of aligning with the receiving node is $1/K^2$ and the non-interference probability is $(1-1/K^2)$. When other neighbors select the receiving mode, it does not send an ACK packet in the second mini-slot[25]. Therefore, the probability of finding any neighbor is

$$P_{\text{suc}}^{u-v} = 2\frac{1}{K^2}p_t(1-p_t)\cdot\left(\left(1-\frac{1}{K^2}\right)p_t+\left(1-\frac{1}{K^2}p_t\right)(1-p_t)\right)^{N-1}, \tag{7}$$

where $K=1$ means the antenna is omni-directional. $P_{\text{suc}}^{u-v}\approx 2p_t(1-p_t)^{2N-1}$ and the best transmission probability is $p_t^{\text{opt}} = 1/(2N)$[25]. When the beam width is small, the optimal value can be set as $p_t^{\text{opt}} = 0.5$ through simulation.

3.4.2 Analysis of Upper and Lower Bounds of Time Slot Mean

When K is a constant, the probability of node u finding any neighbor v in a time slot is a function of N and p_t, which can be expressed as $F_p^{u-v}(N,p_t)$. In ISAC, the central node can update the antenna beam list with the information obtained by radar detection. Because of that, K also affects the probability of success and function $F_p^{u-v}(N,p_t)$ is rewritten as $F_p^{u-v}(N,p_t,K)$. P_{sud}^{u-v} in $E_u^{all}(N-1)$[28] is changing. The expectation number of slots required for node u to find a new neighbor E_1[28] is also changing. As the neighbor discovery progresses, K gradually becomes smaller and the discovery probability becomes larger.

$$E_i^{\text{all}}(N-1) = \sum_{i=1}^{N-1}\frac{1}{(N-i)P_{\text{sud}}^{u-v}}, \tag{8}$$

$$E_1 = \frac{1}{(N-1)P_{\text{sud}}^{u-v}}. \tag{9}$$

The range of the beams that can be selected for any node i is $x\in[K-N_0^E, K]$. The average number of empty beams is $N_0^E = N(K,N,0)$, where N is the number of neighbors and K is the number of beams.

The second type of Stirling number $S_2(n,m)$ represents the number of schemes in which n different elements are divided into m sets. According to the principle of tolerance and exclusion, we can get

$$S_2(n,m) = \frac{1}{m!}\sum_{k=0}^{m}(-1)^2\binom{m}{k}(m-k)^n. \tag{10}$$

The second type of Stirling recurrence formula is

$$S_2(n,m) = m\cdot S_2(n-1,m)+S_2(n-1,m-1), \quad 1\leqslant m\leqslant n-1. \tag{11}$$

The boundary conditions are

$$\begin{cases} S_2(n,n) = 1, & n\geqslant 0 \\ S_2(n,0) = 0, & n\geqslant 1. \end{cases} \tag{12}$$

We bring $S_2(n,n)=1$ into Eq. (11) to get

$$\sum_{k=0}^{m}(-1)^2\binom{m}{k}(m-k)^n=m!. \tag{13}$$

When $N \leqslant K$, N_0^E is expressed as

$$N(K,N,0)=\sum_{h=0}^{K-1}h\times C_K^h(K-h)!\times S_2(N,K-h)/K^N$$
$$N\geqslant K, \tag{14}$$

where C_K^h represents the selection of $h\in[K-N,K-1]$ from K beams as empty beams, the fraction means putting the remaining N neighbors into $K-h$ beams and ensuring that no beam is empty, and K^N means the number of schemes in which N neighbors are put into K beams.

When $N>K$, the range of the empty beam is $[1,K-1]$ and N_0^E is

$$N(K,N,0)=\sum_{h=K-N}^{K-1}h\times\binom{K}{h}E_{K-h}(N)/K^N, N<K, \tag{15}$$

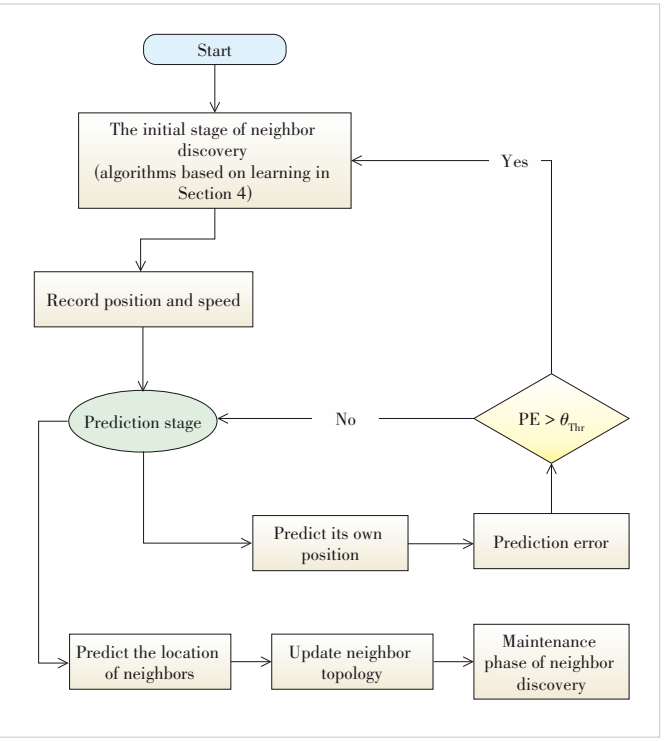
where

$$E_m(j)=\begin{cases}m^j-\sum_{l=1}^{m-1}\binom{m}{l}E_{m-l}(j), j\geqslant m\\ 0, j<m.\end{cases} \tag{16}$$

According to the algorithm, once the node finds that the beam is empty, it does not send Hello messages to the beam afterwards. Therefore, K in Eq. (7) is changing. At the beginning of neighbor discovery, when the central node randomly selects in each slot, the probability of finding an empty beam is the largest, which is $P_{\text{dis},0}=N_0^E/K$. When an empty beam is found, the probability of finding a new empty beam is $P_{\text{dis},1}=(N_0^E-1)/(K-1)$. By analogy, the probability of finding the last empty beam is $P_{\text{dis},N_0^E}=1/(K-N_0^E+1)$.

We construct function $f(t,P_{\text{dis}})=(1-P_{\text{dis}})^{t-1}P_{\text{dis}}$ to represent the probability of beam quality change in t time slots, where P_{dis} represents the probability of selecting an empty beam in the current time slot. When the beam is fixed at $x=K-N_0^E$, the velocity reaches its peak. When $x=K$, the time is the longest. As shown in Fig. 6, we can give more precise upper and lower bounds according to the characteristics of the discovery process.

It can be seen intuitively from Eq. (17) that $E_i^{\text{all}}(N)$ and P_{sud}^{u-v} are inversely proportional, so the lower bound of $E_i^{\text{all}}(N)$ is the upper bound of P_{sud}^{u-v}. The average value of the empty beam is rounded to $N_{\text{ceil}}=\lceil N_0^E\rceil$. Then in this time slot, time T_x^{dis} for each beam x is set to be the same and the average time $E_i^{\text{all}}(N)$ is obtained. P_{sud} is revised as



▲Figure 6. Flow chart of neighbor discovery based on Kalman filter

$$P_{\text{upper}}^{u-v}=\frac{1}{N_0^E+1}\sum_{x=K-N_0^E}^{K}\frac{1}{2x^2}\left(\left(1-\frac{1}{x^2}\right)\times\frac{1}{2}+\left(1-\frac{1}{2x^2}\right)\times\frac{1}{2}\right)^{N-1}. \tag{17}$$

The mathematical upper bound of the neighbor discover time $E_i^{\text{all}}(N)$ is the lower bound of P_{sud}^{u-v}. The number of possible empty beams in the entire ND process of a node is N_0^E. When K in P_{sud}^{u-v} is unchanged, the time is the longest, which is the upper bound of the time. In this section, to get an accurate upper bound, in any time slot, set time T_x^{dis} of these (N_0^E-2) beams appearing to be the same. It is equivalent to dividing the time of the larger discovery probability to the smaller discovery probability, so as to obtain the upper bound of the time of neighbor discovery. P_{sud} can be expressed as

$$P_{\text{lower}}^{u-v}=\frac{1}{N_0^E-2}\sum_{x=K-N_0^E+1}^{K}\frac{1}{2x^2}\left(\left(1-\frac{1}{x^2}\right)\times\frac{1}{2}+\left(1-\frac{1}{2x^2}\right)\times\frac{1}{2}\right)^{N-1}. \tag{18}$$

4 Neighbor Maintenance Method Based on Kalman Filter

4.1 Flight Model of UAV

The initial speed and direction of a UAV are initialized

with a random pattern[29]. In a two-dimensional scene, the physical movement of the UAV is represented as vector $\boldsymbol{X}_k = [\,x \quad V_x \quad y \quad V_y\,]$. In a three-dimensional scene, it can be represented as $\boldsymbol{X}_k = [\,x \quad V_x \quad y \quad V_y \quad z \quad V_z\,]$. Fig. 7 shows the movement of two UAVs. The two-dimensional scene is a flat circle, and the three-dimensional scene is a cross-cut circle. Both of them have a radius of R_s.

The predefined time interval is T_{th} (PTI)[30], which represents the time when B leaves the one-hop communication range of A.

$$T_{\text{th}} = \frac{2R_s}{E[V']}, \tag{19}$$

where $E[V']$ represents the average value of relative speed.

In Fig. 7, V_1 and V_2 obey the uniform distribution on $[L_1, L_2]$ and θ_1 and θ_2 obey the uniform distribution on $[0, \pi]$. They satisfy $V' = V_1\cos\theta_1 + V_2\cos\theta_2$. The expression of the probability density function of V_1 is

$$f(V_1) = \begin{cases} \dfrac{1}{(L_2 - L_1)} & L_1 \leqslant V_1 \leqslant L_2 \\ 0 & \text{others} \end{cases}. \tag{20}$$

The function $f(D)$ is the probability density of $D = \cos\theta_1$, and its expression is

$$f(D) = \begin{cases} \dfrac{2}{\pi\sqrt{1 - D^2}} & 0 \leqslant D \leqslant 1 \\ 0 & \text{others} \end{cases}. \tag{21}$$

V_1 and D are distributed independently, so the joint probability density function is the product of the two, denoted as

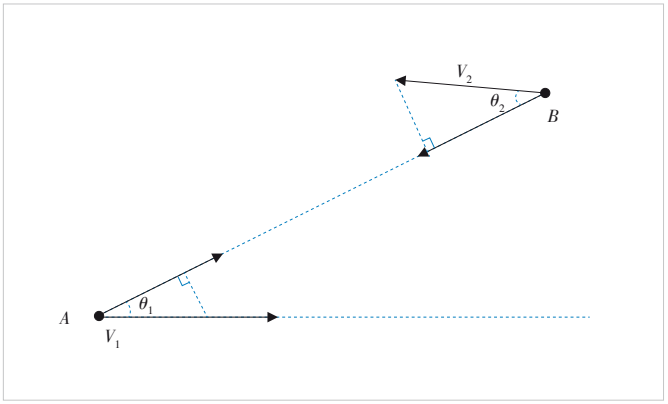
$$f(V_1, D) = \begin{cases} \dfrac{2}{\pi(L_2 - L_1)\sqrt{1 - D^2}} & 0 \leqslant D \leqslant 1, L_1 < V_1 < L_2 \\ 0 & \text{others} \end{cases}. \tag{22}$$

Given that $E[V'] = 2E[V]$, V_1 and θ_1 are taken as examples to find $E[V]$, and the process is

$$E[V] = \int_{L_1}^{L_2}\int_{0}^{1} V_1 D\, f(V_1, D)\,\mathrm{d}V_1\,\mathrm{d}D = \int_{L_1}^{L_2}\int_{0}^{1} V_1 D \frac{2}{\pi(L_2 - L_1)\sqrt{1 - D^2}}\,\mathrm{d}V_1\,\mathrm{d}D = \frac{(L_2 + L_1)}{\pi}. \tag{23}$$

We put the result of Eq. (23) into Eq. (19) to get

$$T_{\text{th}} = \frac{\pi R_s}{(L_2 + L_1)}. \tag{24}$$



▲Figure 7. Relative movement of two nodes

4.2 Kalman Filter Model

The Kalman filter estimates the value of variables based on observations, noise and errors, which is more accurate than the estimation based on observations[29].

We assume that each UAV saves $N + 1$ quadruples $\langle \text{ID} \quad V \quad P \quad T_k \rangle$, where V and P represent the speed and position of the UAV respectively, and T_k is the timestamp. The state is represented as $X_k = \begin{bmatrix} V_{T_k} & P_{T_k} \end{bmatrix}$.

$$X_{k+1} = A_k X_k + w_k, \tag{25}$$

$$\boldsymbol{Z}_k = \boldsymbol{H}_k X_k + v_k. \tag{26}$$

In Eq. (25), X_k is the real state at time k, X_{k+1} is the predicted state, and A_k is the 4×4 state transition matrix. That is $X_k \xrightarrow{A_k} X_{k+1}$, where Δt can be set according to actual conditions. The process noise w_k obeys Gaussian distribution of $N(0, Q)$. In Eq. (26), $\boldsymbol{Z}_k$ represents the estimated position vector, $\boldsymbol{H}_k$ is the observation matrix, and observation noise v_k obeys the Gaussian distribution of $N(0, R)$.

Covariance Q and R can be obtained through empirical analysis, and the settings in this section are as

$$Q = 0.01 \times E_{4\times 4}, \tag{27}$$

$$R = 10 \times E_{2\times 2}, \tag{28}$$

$$A_k = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{29}$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{30}$$

At the prediction stage[29], the state value $\hat{X}'_k$ at the next moment is estimated according to

$$\hat{X}'_k = A_{k-1}\hat{X}_{k-1}. \tag{31}$$

1) Prediction and estimation

The covariance matrix $\boldsymbol{P}'_k$ is predicted and estimated according to

$$\boldsymbol{P}'_k = A_k \boldsymbol{P}_{k-1} A_k^T + Q_{k-1}. \tag{32}$$

2) Stage update[29]

The optimal Kalman gain is calculated by

$$K_k = \boldsymbol{P}'_k H^T_k \left(H_k \boldsymbol{P}'_k H^T_k + R\right)^{-1}. \tag{33}$$

The estimated state $\hat{X}'_k$ is updated by

$$\hat{X}_k = \hat{X}'_k + K_k(\boldsymbol{Z}_k - \boldsymbol{H}_k \hat{X}'_k). \tag{34}$$

The estimated covariance matrix $\boldsymbol{P}_k$ is updated by

$$\boldsymbol{P}_k = \left(I - K_k \boldsymbol{H}_k\right)\boldsymbol{P}'_k. \tag{35}$$

$\hat{X}_k$ is then used to predict the next moment, so that the prediction is more accurate[30].

3) Error forecast

In order to measure the effect of Kalman filtering, position error (PE) is

$$\mathrm{PE} = \sqrt{\left[\left(y'_i - y_i\right)^2 + \left(x'_i - x_i\right)^2\right]}, \tag{36}$$

where (x_i, y_i) is the true value and (x'_i, y'_i) is the predicted value.

The right-angled side R_l conforms to the uniform distribution $R_l \sim U[0, R_s\cos(\alpha/2)]$, and the mathematical expectation is

$$E_{R_b} = R_s\cos(\alpha/2)/2, \tag{37}$$

where the value of the bottom R_b is the error threshold $\theta_{\mathrm{Thr}} = 2E_{R_b} \times \tan(\alpha/2)$.

4.3 Switching Mechanism Between Initialization and Maintenance Phases

In the maintenance phase, after the node gets a highly accurate topology, a stop mechanism is joined[4]. The difference between the maintenance and stop mechanisms[4] is that the beam selection probability changes with the number of undiscovered neighbors.

The size of the Hello packet can be changed according to the needs of the protocol. For example, node ID (4 B), node location (usually two integers, 8 B), speed (1 B) and direction (1 B) can be added and deleted as needed[30]. In the initialization phase, the Hello packet contains ID, speed, and location information. In the maintenance phase, to reduce the data overhead, neighbor information can be obtained through prediction, and the Hello packet only contains the ID.

Next, the outgoing and incoming of UAVs are explored during the maintenance period of neighbor discovery[31].

UAV i calculates the distance R_{i-j} from neighbor j. When $R_{i-j} > R_s$, the UAV determines that the neighbor has left and no longer sends Hello packets, which can reduce bandwidth consumption[31].

In each time slot, UAV predicts its own position. When PE > θ_{Thr}, it will switch to the initial stage of neighbor discovery. When the neighbor maintenance phase continues to be greater than the predefined time interval (PTI), even if PE < θ_{Thr}, it will also switch to the initial phase. In addition, to prevent the emergence of extreme situations, a random scan factor P_{arbit} is added to optimize the algorithm. All directions are arbitrarily selected with a small probability, which trades discovery time for accuracy. The parameters θ_{Thr}, PTI and P_{arbit} all have an impact on the protocol performance. If the random scan factor P_{arbit} is too small, the neighbors that accidentally enter will not be discovered. If P_{arbit} is too large, the discovery time of the neighbor maintenance phase is prolonged. These parameters need to be set according to the actual scene[30]. The distance prediction error threshold θ_{Thr} and the time interval threshold T_{th} will have an impact on the performance of the two-phase handover. First of all, if θ_{Thr} is too large, the predicted topology in the maintenance phase is not accurate enough, which prolongs the discovery time of the maintenance phase. If θ_{Thr} is too small, it will enter the initial stage too frequently, which will increase data overhead. Secondly, if θ_{Thr} is too large, the UAV may still not enter the initialization phase, but the actual topology has changed, which will also increase the average discovery time. If T_{th} is too small, it will frequently enter the initial phase and increase data overhead.

5 Simulation Results and Analysis

5.1 Neighbor Discovery Algorithm Based on Learning Automata

Discovery time $E_i^{\mathrm{all}}(t)$ and speed $V_i^{\mathrm{all}}(t)$ are used as measurement standards. Some studies use neighbor convergence of 70%[11] as indicators to measure the algorithm performance. However, the algorithm based on purely directional Completely Random Algorithms (CRA) has a long tail problem[11]. In this section, to fully see the effect of the algorithm, 100% convergence of the nodes in the network is used as the criterion.

Each parameter influences the performance of the algorithm. Function $E_i^{\mathrm{all}}(t) = f(N, K, \eta, \varpi, \nu)$ represents the discovery time of the algorithm based on the LA, where N is the

network size, K is the number of beams, η is the radar communication ratio, and $\boldsymbol{\varpi} = \{\gamma_1, \gamma_2, \mu\}$ and ν are the reward and punishment factors. In order to eliminate the influence of ν, we set $\nu = 0.01$. Next, the controlled variable methods are used to perform multiple simulations and the average value is taken. In order to better analyze the influence of the parameters, Table 1 shows the parameter values obtained through multiple simulation observations.

Fig. 8 illustrates the relationship between the radar communication ratio η and the convergence time $E_i^{\text{all}}(t)$. When K = 36, for the same topology size N, the larger the value of η , the smaller the time. The reason is that the larger η is, the more accurate the information obtained by the node in each time slot is. The action probability distribution obtained by iteration A can maximize the success rate of neighbor discovery in each slot. For N = 30, compared with classic CRA, when $\eta = \{\eta_1, \eta_2, \eta_3, \eta_5\}$, the efficiency increases by 13.5%, 30.9%, 57.9%, and 65.2%, respectively.

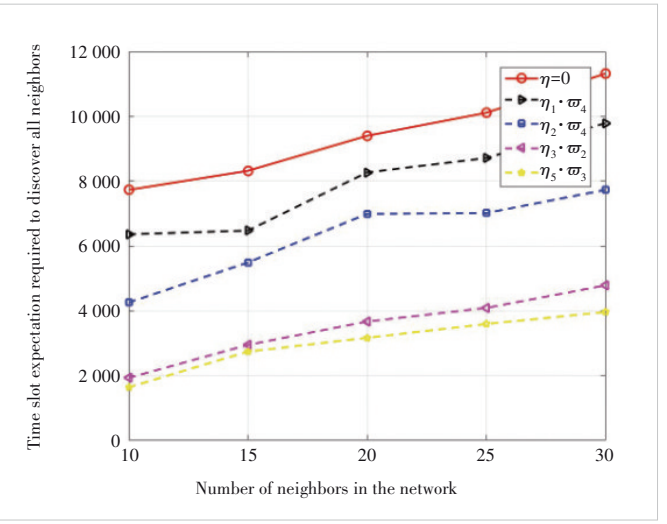
Fig. 9 shows the effect of the reward and punishment factor $\boldsymbol{\varpi}$ on $E_i^{\text{all}}(t)$ and the velocity when N and K = 24 are fixed. Given $\gamma_1, \gamma_2, \mu \in [0, 1]$, the simulation parameters cannot be enumerated due to the complex and diverse combinations of parameters. When $\mu/\gamma_1 = 2$, let the set $\boldsymbol{\varpi}_2 = \{\boldsymbol{\varpi}_2^6 \quad \boldsymbol{\varpi}_2^5 \quad \boldsymbol{\varpi}_2^4 \quad \boldsymbol{\varpi}_2^3\}$, $\gamma_1 = 0.1$, $\mu = 0.2$, $\gamma_2 = 0.6$ in $\boldsymbol{\varpi}_2^6$, etc. Comparing the curves longitudinally, it can be seen that the optimal values of $\boldsymbol{\varpi}_{\text{opt}}$ are different in different scenarios. The simulation results also show that the algorithm can maintain a better performance effect than the classic ND algorithm in actual scenarios.

Fig. 10 shows the influence of the beam change on time $E_i^{\text{all}}(t)$ when N = 10. For the ISAC signal that shares the waveform, its power is both the communication power and the radar power. Under normal circumstances, the radar signal has two path losses, while the communication has only one path loss. Without tuning parameters, the detection range of radar is generally about half of that of communication[32]. It is more practical to set η to 0.5 and 0.6. Set $\boldsymbol{\varpi}_{\text{opt}} = \boldsymbol{\varpi}_2^4$. Next, the first three sets of data in the histogram are analyzed. When K = 36, the data are (8 166, 6 099, 5 558). When K = 10, the data are (692.4, 591.2, 560.4). It can be calculated that compared with CRA, the narrower the beam, the more obvious the efficiency improvement. When the number of neighbors is fixed, the narrow beam means that probability P_{pten}^i is small, and the probability of learning correct is high.

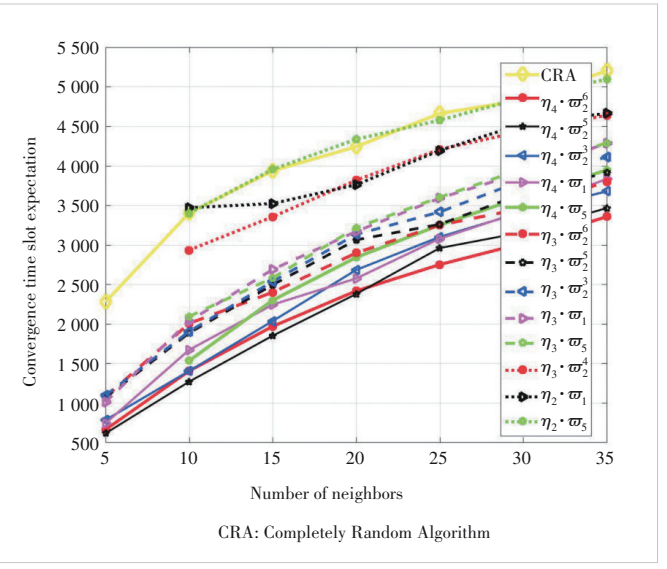
Convergence (discovery) rate $V_i^{\text{all}}(t)$ is quantitatively analyzed from a computational perspective here. When the number of beams is K = 36 and the number of neighbors is N = 25, a different η is set, and the reward and punishment factor is set to $\boldsymbol{\varpi}_{\text{opt}} = \boldsymbol{\varpi}_2^6$. Then the success rate of neighbor discovery is observed under different time slot consumption (the ratio of successfully discovered neighbors to all neighbors). Reciprocal k' of the slope of the curve is used to represent $V_i^{\text{all}}(t)$. The highlighted part in Fig. 11 is the time period when the central node discovers 62.5% of neighbors. The rate of the learning-based algorithm is k'_{Learn} = 78.9 (slot/a new neighbor). The rate of the traditional CRA is k'_{class} = 192.3 (slot/a new neighbor). The discovery rate is in-

▼Table 1. Parameter value setting

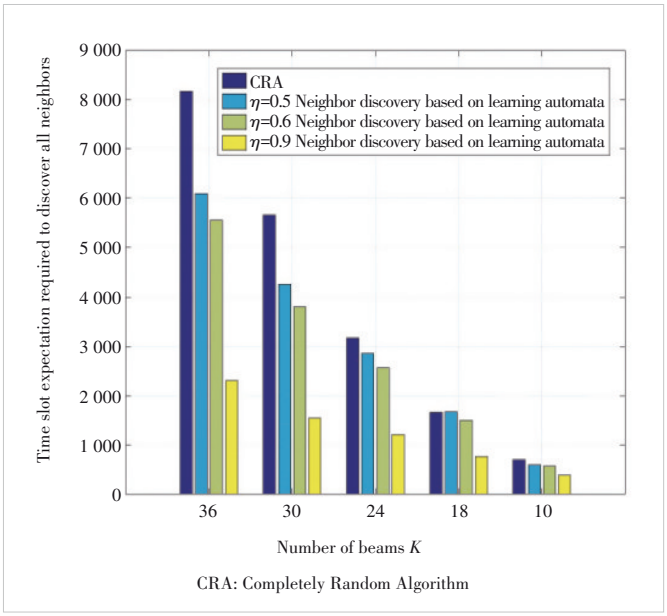
η	$\boldsymbol{\varpi}$
$\eta_1 = 0.3$	$\boldsymbol{\varpi}_1 = (0.2, 0.3, 0.2)$
$\eta_2 = 0.5$	$\boldsymbol{\varpi}_2 = (0.1, 0.4, 0.1)$
$\eta_3 = 0.7$	$\boldsymbol{\varpi}_3 = (0.1, 0.2, 0.05)$
$\eta_4 = 0.8$	$\boldsymbol{\varpi}_4 = (0.1, 0.2, 0.1)$
$\eta_5 = 0.9$	$\boldsymbol{\varpi}_5 = (0.1, 0.3, 0.2)$



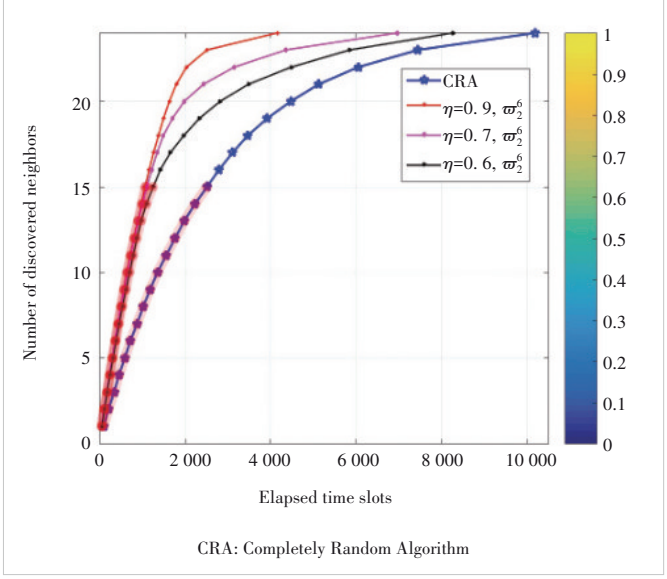
▲Figure 8. The relationship between network size and time slot expectations



▲Figure 9. Relationship between the covergence time slot expectation and the number of beams



▲ **Figure 10. Trends of neighbors discovered in different algorithms over time**



▲ **Figure 11. Comparison of discovery efficiency in the same node density network**

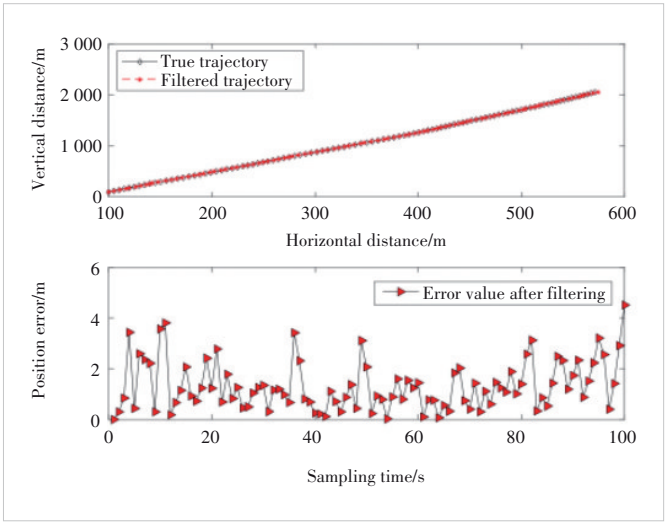
creased by about 60%. In addition, the discovery of the last neighbor takes time slots respectively. That is to say, the algorithm in Section 3 improves the convergence speed, but there is still a long tail problem.

5.2 Neighbor Maintenance Methods Based on Kalman Filter

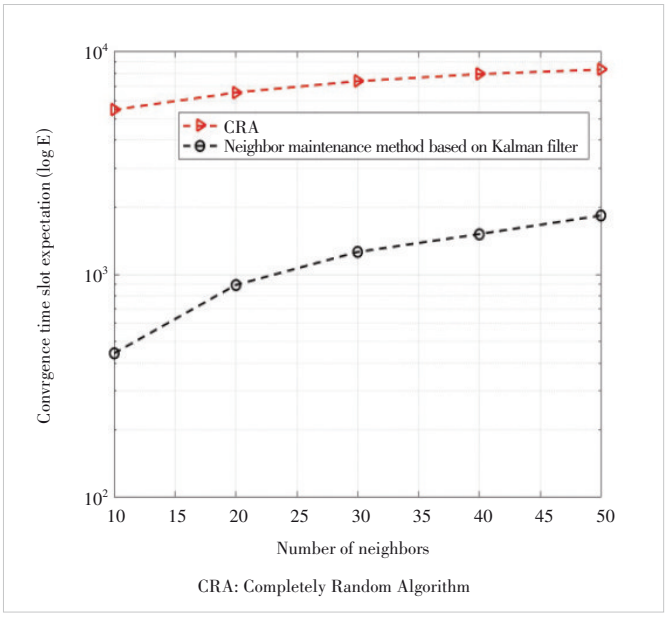
1) Analysis of forecast errors: Fig. 12 shows the PE at different prediction moments. In the simulation, the horizontal velocity is 5 m/s, the vertical velocity is 20 m/s, and the prediction period is 1 s[33]. From a numerical point of view, the error is about 5 m, which shows that the prediction result based on the Kalman filter has a higher accuracy rate. The specific reason is that the filtering model predicts by the estimated $\hat{X}_k$ at the next moment. If we consider the influence of actual noise, the prediction will be more accurate.

2) Analysis of discovery efficiency: In Fig. 13, $N \in [10, 50]$ nodes are randomly scattered in an area of 2 km × 2 km. The number of beams is 30. Comparing the two curves, the algorithm in Section 4 can greatly reduce the network delay.

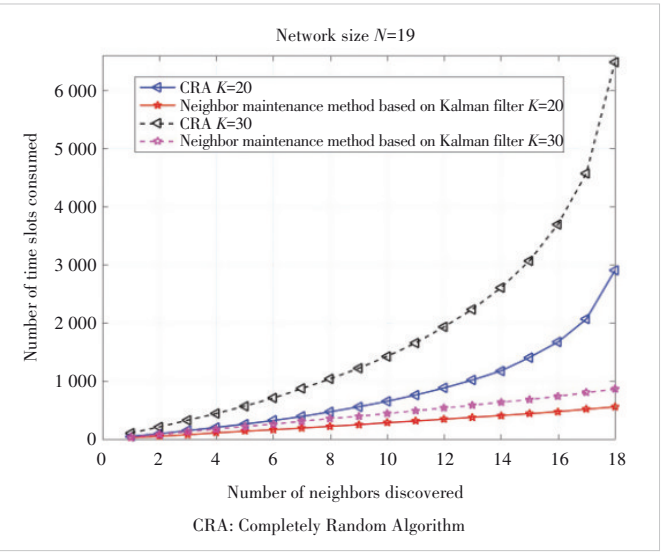
The following part studies the ND convergence process of any node i. In Fig. 14, the topology size is fixed at 19, and K is fixed at 30 and 20. The meaning of the ordinate is time interval $E_i^1(t)$ between the discovery of two new neighbors. Ana-



▲ **Figure 12. Position error after Kalman filtering**



▲ **Figure 13. Comparison of convergence time of different algorithms in differentnetwork sizes**

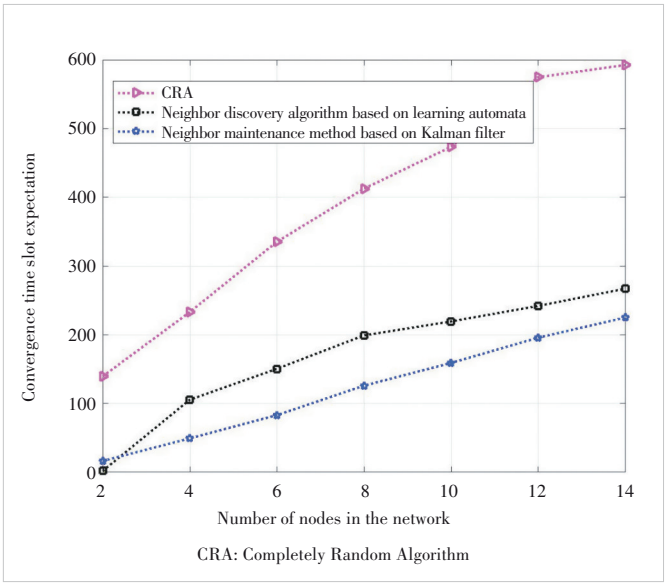


▲Figure 14. Comparison of discovery efficiency in the same node density network

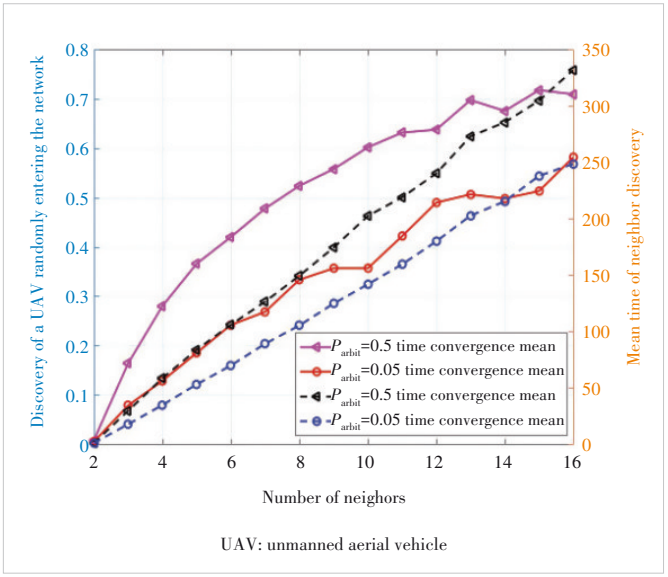
lyze the "★" and "△" curves longitudinally from the slope of the curve. The filtering-based algorithm represented by the "★" curve maintains a high ND rate throughout the process. The curve can be fitted to a straight line, which means that the long tail problem of neighbor discovery is overcome. In K = 30, when the filtering-based algorithm converges to 100%, the CRA algorithm only converges about 37%. Figs. 13 and 14 show that adding the filtering model not only increases the discovery speed but also shortens the time. The reason is that before the ND phase starts, the node has predicted a topology through the Kalman filter model as prior knowledge. Based on this prior knowledge, nodes can scan the targeted to shorten the discovery time.

3) Impact of mobility: In a three-dimensional scene, the UAV applies a phased array antenna to scan the entire sphere. To simplify the performance research of the proposed algorithm, a scene where a single UAV is in the center and other UAVs are uniformly and randomly distributed is observed. The following part studies probability P_{dis}^{x} of any UAV being discovered after entering the network topology in the neighbor maintenance phase.

Fig. 15 is the comparison of the three neighbor discovery strategies. For example, when N = 8, compared with CRA, the efficiency of filtering-based algorithms is increased by 69.6%, and compared with learning-based algorithms by 36.9%. In addition, from the perspective of slope, $k_{\text{KF}} < k_{\text{LA}} < k_{\text{CRA}}$, the node sensitivity is overcome to a certain extent. Compared with the classic CRA, the algorithm proposed in Section 4 avoids the invalid scanning of blank areas and reduces the time. Compared with the LA scheme based on time slots to obtain prior knowledge, the algorithm in Section 3 obtains all prior knowledge before the neighbor discovery starts, which can improve the timeliness. In addition, because the nodes (beams) that have been discovered no longer occupy communication resources, the sensitivity of the nodes is overcome to a certain extent. In Fig. 16, the left and right ordinates are P_{dis}^{x} and $E_i^{\text{all}}(t)$ respectively. It can be seen that under the same UAV density, P_{dis}^{x} and $E_i^{\text{all}}(t)$ increase with the increase of P_{arbit}. The reason is that the increase of the random factor P_{arbit} (in a mobile scene, an optimization factor that increases the probability that a random UAV enters the network topology is found) increases the scanning probability of other beams, which increases the probability of random UAVs discovery chances. At the same time, it also increases the prob-



▲Figure 15. Comparison of convergence time of different neighbor discovery strategies



▲Figure 16. Probability of discovery and the average time slot for different numbers of neighbors

ability of invalid scanning, which in turn increases the time of the entire network topology construction.

6 Conclusions

This paper proposes the sensing and communication integrated intelligent ND algorithms for UAV networks. Reinforcement learning is introduced to solve the problem that the radar detection range is not equal to the communication range in the integration. Simulation results show that the algorithm based on LA can increase the time efficiency by up to 32% compared with the classic scan-based algorithm, when the radar communication ratio $\eta = 0.6$. This paper also considers the high mobility of UAVs and designs an efficient neighbor maintenance algorithm that predicts the node motion through the Kalman filter. In the current work, we only introduce a relatively simple LA algorithm to solve the problem of neighbor discovery in the UAV network. In the future, more learning algorithms can be introduced in combination with specific scenarios to improve the efficiency of neighbor discovery.

References

[1] LI J M, PENG L X, YE Y L, et al. A neighbor discovery algorithm in network of radar and communication integrated system [C]//Proc. IEEE 17th International Conference on Computational Science and Engineering. IEEE, 2014: 1142 – 1149. DOI: 10.1109/CSE.2014.224
[2] JI D N, WEI Z Q, CHEN X, et al. Radar-communication integrated neighbor discovery for wireless ad hoc networks [C]//Proc. 11th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2019: 1 – 5. DOI: 10.1109/WCSP.2019.8927896
[3] LIU N X, PENG L X, XU R H, et al. Neighbor discovery in wireless network with double-face phased array radar [C]//Proc. 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). IEEE, 2016: 434 – 439. DOI: 10.1109/MSN.2016.080
[4] WEI Z Q, HAN C Y, QIU C, et al. Radar assisted fast neighbor discovery for wireless ad hoc networks [J]. IEEE access, 2019, 7: 176514 – 176524. DOI: 10.1109/ACCESS.2019.2950277
[5] HAN Y, EKICI E, KREMO H, et al. Automotive radar and communications sharing of the 79-GHz band [C]//Proc. First ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services. ACM, 2016: 6 – 13. DOI: 10.1145/2980100.2980106
[6] EL KHAMLICHI B, NGUYEN D H N, EL ABBADI J, et al. Collision-aware neighbor discovery with directional antennas [C]//Proc. International Conference on Computing, Networking and Communications (ICNC). IEEE, 2018: 220 – 225. DOI: 10.1109/ICCNC.2018.8390323
[7] GAO M J, SHEN R J, MU L H, et al. An anti-collision neighbor discovery protocol for multi-node discovery [C]//Proc. 11th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2019: 1 – 5. DOI: 10.1109/WCSP.2019.8927864
[8] SUN G B, WU F, GAO X F, et al. Time-efficient protocols for neighbor discovery in wireless ad hoc networks [J]. IEEE transactions on vehicular technology, 2013, 62(6): 2780 – 2791. DOI: 10.1109/TVT.2013.2246204
[9] BURGHAL D, TEHRANI A S, MOLISCH A F. On expected neighbor discovery time with prior information: modeling, bounds and optimization [J]. IEEE transactions on wireless communications, 2018, 17(1): 339 – 351. DOI: 10.1109/TWC.2017.2766219
[10] WANG Y H, PENG L X, XU R H, et al. A fast neighbor discovery algorithm based on Q-learning in wireless ad hoc networks with directional antennas [C]//Proc. IEEE 6th International Conference on Computer and Communications (ICCC). IEEE, 2020: 467 – 472. DOI: 10.1109/ICCC51575.2020.9345296
[11] EL KHAMLICHI B, EL ABBADI J, ROWE N W, et al. Adaptive directional neighbor discovery schemes in wireless networks [C]//Proceedings of International Conference on Computing, Networking and Communications (ICNC). IEEE, 2020: 332 – 337. DOI: 10.1109/ICNC47757.2020.9049819
[12] BAI W, XU Y H, WANG J L, et al. Cognitive neighbor discovery with directional antennas in self-organizing IoT networks [J]. IEEE Internet of Things journal, 2021, 8(8): 6865 – 6877. DOI: 10.1109/JIOT.2020.3037067
[13] ZHU L N, GU W Y, YI J J, et al. On mobility-aware and channel-randomness-adaptive optimal neighbor discovery for vehicular networks [J]. IEEE Internet of Things journal, 2021, 8(8): 6828 – 6839. DOI: 10.1109/JIOT.2020.3036251
[14] KAROWSKI N, WILLIG A, WOLISZ A. Cooperation in neighbor discovery [C]//Proc. Wireless Days. IEEE, 2017: 99 – 106. DOI: 10.1109/WD.2017.7918123
[15] LUO Y, DANG J, SONG Z X, et al. Compressed sensing algorithm for neighbour discovery in mobile ad hoc networks [J]. IET communications, 2019, 13(12): 1781 – 1786. DOI: 10.1049/iet-com.2018.5127
[16] ASTUDILLO G, KADOCH M. Neighbor discovery and routing schemes for mobile ad-hoc networks with beamwidth adaptive smart antennas [J]. Telecommunication systems, 2017, 66(1): 17 – 27. DOI: 10.1007/s11235-016-0268-x
[17] CHEN L, LI Y, VASILAKOS A V. On oblivious neighbor discovery in distributed wireless networks with directional antennas: theoretical foundation and algorithm design [J]. IEEE/ACM transactions on networking, 2017, 25(4): 1982 – 1993. DOI: 10.1109/TNET.2017.2673862
[18] LIAO Y L, PENG L X, XU R H, et al. Neighbor discovery algorithm with collision avoidance in ad hoc network using directional antenna [C]//Proc. IEEE 6th International Conference on Computer and Communications (ICCC). IEEE, 2020: 458 – 462. DOI: 10.1109/ICCC51575.2020.9344952
[19] ZHANG Z S. Performance of neighbor discovery algorithms in mobile ad hoc self-configuring networks with directional antennas [C]//Proc IEEE Military Communications Conference (MILCOM 2005). IEEE, 2005: 3162 – 3168. DOI: 10.1109/MILCOM.2005.1606143
[20] GAMMARANO N, SCHANDY J, STEINFELD L. Q-SAND: a quick neighbor discovery protocol for wireless networks with sectored antennas [C]//Proc. Ninth Argentine Symposium and Conference on Embedded Systems (CASE). IEEE, 2018: 19 – 24
[21] ZHANG Z S, RYU B, NALLAMOTHU G, et al. Performance of all-directional transmission and reception algorithms in wireless ad hoc networks with directional antennas [C]//Proc. MILCOM 2005 - 2005 IEEE Military Communications Conference. IEEE, 2005: 225 – 230. DOI: 10.1109/MILCOM.2005.1605690
[22] RUSSELL A, VASUDEVAN S, WANG B, et al. Neighbor discovery in wireless networks with multipacket reception [J]. IEEE transactions on parallel and distributed systems, 2015, 26(7): 1984 – 1998. DOI: 10.1109/TPDS.2014.2321157
[23] SHEN Z, JIANG H, DONG Q K, et al. Energy-efficient neighbor discovery for the Internet of Things [J]. IEEE Internet of Things journal, 2020, 7 (1): 684 – 698. DOI: 10.1109/JIOT.2019.2949922
[24] SARAEREH O A, KHAN I, LEE B M. An efficient neighbor discovery scheme for mobile WSN [J]. IEEE access, 2019, 7: 4843 – 4855. DOI: 10.1109/ACCESS.2018.2886779
[25] ZHANG Z S, LI B. Neighbor discovery in mobile ad hoc self-configuring networks with directional antennas: algorithms and comparisons [J]. IEEE transactions on wireless communications, 2008, 7(5): 1540 – 1549. DOI: 10.1109/TWC.2008.05908
[26] WUNDER M, LITTMAN M L, BABES-VROMAN M. Classes of multia-

gent Q-learning dynamics with e-greedy exploration [C]//Proc. 27th International Conference on Machine Learning (ICML-10). ICML, 2010: 1167 – 1174
[27] EL KHAMLICHI B, NGUYEN D H N, EL ABBADI J, et al. Learning automaton-based neighbor discovery for wireless networks using directional antennas [J]. IEEE wireless communications letters, 2019, 8(1): 69 – 72. DOI: 10.1109/lwc.2018.2855120
[28] LIU L J, PENG L X, XU R H, et al. A neighbor discovery algorithm for flying ad hoc network using directional antennas [C]//Proc. 28th Wireless and Optical Communications Conference (WOCC). IEEE, 2019: 1 – 5. DOI: 10.1109/WOCC.2019.8770698
[29] XU H, CHO J, LEE S. A predictive and synchronized neighborhood tracking scheme for mobile ad hoc networks [C]//Proc. 7th IEEE International Conference on Computer and Information Technology. IEEE, 2007: 429 – 434. DOI: 10.1109/CIT.2007.33
[30] LIU C F, ZHANG G, GUO W S, et al. Kalman prediction-based neighbor discovery and its effect on routing protocol in vehicular ad hoc networks [J]. IEEE transactions on intelligent transportation systems, 2020, 21(1): 159 – 169. DOI: 10.1109/TITS.2018.2889923
[31] ZHOU L, WANG H H, GUIZANI M. How mobility impacts video streaming over multi-hop wireless networks? [J]. IEEE transactions on communications, 2012, 60(7): 2017 – 2028. DOI: 10.1109/TCOMM.2012.051712.110165
[32] MA H, WEI Z Q, CHEN X, et al. Performance analysis of joint radar and communication enabled vehicular ad hoc network [C]//Proc. IEEE/CIC International Conference on Communications in China (ICCC). IEEE, 2019: 887 – 892. DOI: 10.1109/ICCChina.2019.8855937
[33] OOMMEN B J, THATHACHAR M A L. Multiaction learning automata possessing ergodicity of the mean [J]. Information sciences, 1985, 35(3): 183 – 198. DOI: 10.1016/0020-0255(85)90049-0

Biographies

WEI Zhiqing (weizhiqing@bupt.edu.cn) received his BE and PhD degrees from Beijing University of Posts and Telecommunications (BUPT), China in 2010 and 2015, respectively. He is an associate professor with BUPT. He has authored one book, three book chapters, and more than 50 papers. His research interest is the performance analysis and optimization of intelligent machine networks. He was granted the Exemplary Reviewer of *IEEE Wireless Communications Letters* in 2017, the Best Paper Award of WCSP 2018. He was the Registration Co-Chair of IEEE/CIC ICCC 2018 and the Publication Co-Chairs of IEEE/CIC ICCC 2019 and IEEE/CIC ICCC 2020.

ZHANG Yongji is an undergraduate student studying at the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), China. His research interests include integrated perception, communication, computing, machine learning, etc.

JI Danna received her BE degree from University of Science and Technology Beijing (USTB), China and ME degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), China. Her research interests include wireless communication networking, joint radar and communication, etc.

LI Chenfei received her BE degree from Nankai University (NKU), China in 2021 and master's degree from Beijing University of Posts and Telecommunications (BUPT), China in 2024. Her research interests include integrated sensing and communication neighbor discovery and positioning.