FMAC: A Self-Adaptive MAC Protocol for Flocking of Flying *Ad Hoc* Network

Xinquan Huang, Aijun Liu, *Member, IEEE*, Haibo Zhou, *Senior Member, IEEE*, Kai Yu, *Graduate Student Member, IEEE*, Wei Wang, *Student Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Considering the high-density and high-dynamic feature of cooperative unmanned aerial vehicles (UAVs) swarm, also referred to as flocking of flying *ad hoc* networks (FANETs), reliable medium access control (MAC) protocol design for network connectivity maintaining and network information sharing is a challenging issue. In this article, we propose a self-adaptive carrier sense multiple access with collision avoidance (CSMA/CA)-based MAC protocol for flocking of FANET, namely, FMAC, to provide reliable broadcast information service under density-varying flocking scenarios. To represent the varying trend of UAV density during flocking, we define the collective neighboring potential (CNP) in the FMAC protocol. Specifically, at the beginning of each period, each UAV computes the current CNP based on available neighbors' motion states. Then, the value of CNP at the start of the next period regarding the same neighbors is predicted using UAV's kinetic equation. After that, each UAV can update the contention window (CW) size by comparing the current CNP and the predicted CNP, and CW will be decreased (increased) if the current CNP is larger (smaller) than the predicted one for enough period. The simulation results show that the proposed FMAC protocol can ensure high successful transmission probability under density-varying flocking scenarios and outperforms the typical MAC solutions.**

***Index Terms*—Collective neighboring potential (CNP), contention window (CW), flocking, medium access control (MAC), unmanned aerial vehicles (UAVs) swarm.**

I. Introduction

With the rapid popularity of unmanned aerial vehicles (UAVs) and their great potentials in various applications, such as industrial and emergency services, UAVs swarm networking techniques have attracted intensive research interests from both industry and academia [1]–[4]. Swarm of small UAVs, also referred to as flying *ad hoc* network (FANET), can conduct common missions cooperatively, thus higher efficiency and lower cost can be realized than using one single large UAV [5], [6]. For most swarm missions, the distributed coordination control capability called flocking is required [7]–[10], which is used to achieve the flocking motion of the swarm. The flocking motion aims to maneuver the swarm as a whole from source *A* to destination *B* while realizing collision avoidance (between UAVs or with obstacles) and cohesion among the swarm. To achieve flocking motion, it is necessary for each UAV to periodically broadcast their flight status to its one-hop neighbors [11]–[13]. The broadcast service requires both reliability and latency, however, the reliability is more important. This is because one packet loss of the UAV will make its neighbors unaware of its status for at least two periods. Although medium access control (MAC) protocols aiming at efficient broadcast[1] service have been widely investigated in mobile *ad hoc* networks (MANETs) [14]–[19], the protocols for that service associated with UAV flocking have not been well studied. In this article, we explore the available MAC protocol prioritizing high reliability for the broadcast service under flocking scenarios.

The carrier sense multiple access with collision avoidance (CSMA/CA) protocol defined in 802.11 distributed coordination function (DCF) can provide broadcast service for UAV flocking in a cost-effective manner and can be conveniently implemented without precise time synchronization requirement compared with the time-division multiple access (TDMA) protocol. However, the unchanged contention window (CW) imposes challenges on typical CSMA/CA (with unchanged CW size) to accommodate channel deterioration by a sudden increase of UAV density when the swarm is moving through the narrow space [20]. To address such a problem, it is anticipated to adaptively adjust the size of CW as node density changes in the MAC protocol. However,

[1]The term "broadcast" and periodic one-hop broadcast is used interchangeably throughout this article.

Manuscript received February 5, 2020; revised June 4, 2020; accepted June 30, 2020. Date of publication July 7, 2020; date of current version December 21, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61871211; in part by the Natural Science Foundation of Jiangsu Province Youth Project under Grant BK20180329; in part by the China Postdoctoral Science Foundation under Grant 2019M651648; in part by the National Natural Science Foundation of China under Grant 61901516; and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). *(Corresponding author: Aijun Liu.)*

Xinquan Huang and Aijun Liu are with the College of Communication Engineering, Army Engineering University, Nanjing 210007, China (e-mail: huangxinquan1993@sina.com; liuaj.cn@163.com).

Haibo Zhou and Kai Yu are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: haibozhou@nju.edu.cn; kaiyu@smail.nju.edu.cn).

Wei Wang is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (e-mail: wei_wang@nuaa.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2020.3007071

most existing researches on adaptive CW mechanisms focus on the unicast service, based on collision detection with the help of Acknowledge (ACK) and clear to send (CTS) packets [21]–[24]. In the case of broadcast service, since no ACK/CTS is transmitted for the successful reception, the collisions cannot be recognized. In fact, only several CW adjustment mechanisms adapting to varying density have been proposed for CSMA/CA to provide efficient broadcast service in *ad hoc* networks [25]–[29]. These mechanisms are designed dedicated to vehicular *ad hoc* networks (VANETs) [30], and they adaptively adjust the CW size based on varied states that can be observed locally, including expiration, node density, and reception ratio [31]. The expiration means the occurrence of the expired packet which is not transmitted during its lifetime. The expiration-based mechanisms [25], [26] usually update the CW size periodically according to the number of expirations observed locally within each interval. The local node density typically denotes the number of neighbors perceived by each node within each interval. To adjust the CW size, the node density-based mechanisms need to evaluate the local node density directly or indirectly [27], [28]. One efficient method to evaluate the node density is to count the different sources from which at least one broadcast packet has been received within each interval [28]. The local reception ratio within one interval defines the percentage of broadcast packets that are successfully received by one node sent to it from its neighbors [29]. The reception ratio is obtained by utilizing the reception continuity of sequence numbers of broadcast packets sent by one specific sender, such that any unsuccessfully received packet can be found when there is one gap between two received sequence numbers. Generally, in the reception ratio-based mechanisms, the large CW will be adjusted by one node if it estimates the low reception ratio.

Although, the implementation convenience and performance efficiency, the above-mentioned triggering states still have some drawbacks which make them not fulfill great broadcast performance in density-varying flocking scenarios. First, as for packet expiration used in [25] and [26], it occurs only when the UAV density becomes very large. In the case, certain UAV may wait for too many other UAVs' transmission in the shared channel, and hence, it can be of a high risk that the lifetime of its packet is exhausted before putting on the channel. However due to limited communication range, even for the worst case of the swarm (when the swarm topology densifies), the node density is not large enough such that few packet expirations will occur. Furthermore, as for the node density-based method, when the packet collision occurs frequently as node density increases, many packets sent by neighboring sources may not be successfully received, causing an incorrect perception of node density. Since the local perception on node density is incorrectly evaluated when the packet collision probability increases, the CW size will not be adjusted in an available manner. Third, the length of each interval should be long enough to make the local reception ratio converge to the real transmission performance during the interval. This makes the CW adjustment based on the local reception ratio insensitive to the network state change.

Moreover, the FANET requires a higher quality of broadcast service. In VANET scenarios, the increasing of node density within one road segment, typically means the road segment is witnessing the congestion. In the congestion case, the average velocity of vehicles is generally in low range. As a result, the high-density VANET scenarios can tolerate the failure of beacons as most vehicles are stationary or only moderately moving. However, in the considered FANET scenario, each fixed-wing UAV must keep the high forwarding speed to keep itself in the sky. This introduces a higher requirement on periodic broadcast service for the FANET than general VANETs.

To further improve the broadcast performance, it is of great importance to introduce one new state to trigger the CW update process of CSMA/CA. The new triggering state should be able to handle the above-mentioned drawbacks so as to achieve better broadcast performance in density-varying flocking scenarios. To find the appropriate triggering state, we need to know how the UAV density changes under the flocking control. The flocking capability has been extensively investigated in the field of control and aviation, which analyzes how the whole UAVs in the swarm collectively move from one source to one destination in the sky. One term called collective neighboring potential (CNP) is used by the flocking capability, to enforce appropriate intervehicle spacings, so as to prevent the swarm from collisions (between UAVs or with obstacles) and achieve cohesion. Since the inter-UAV distance directly relates to UAV density, motivating us to treat the potential term as one triggering state to adaptively adjust the CW size. This article proposes an FMAC protocol, which is based on the CSMA/CA protocol and can adjust the CW size of each UAV adapting to the change of UAV density. In the FMAC protocol, before broadcasting packet during each period, each UAV first calculates the current CNP based on neighbors' motion states that are announced in their received packets during the past period. After that, the CNP at the next period start regarding the same neighbors is predicted based on UAV's kinetic equation. Finally, each UAV could adjust the CW size by comparing the two CNP values. Particularly, if the current CNP is smaller (larger) than the predicted one for enough periods, the CW will be increased (decreased) by one constant offset value. We summarize our contributions in threefold.

1) The FMAC protocol achieves high reliability in the density-varying flocking scenario at a low cost. The main feature of FMAC is its adaptability to the level of contention in the network so that under low contention, it uses normal CW size, and under high contention, uses large CW adaptively.
2) We propose one analytical model of CSMA/CA in the flocking scenario when the UAV swarm densifies, and analyze the MAC performance of CSMA/CA by integrating the typical Markov chain model and flocking traffic model. The Markov chain model is used to analyze the broadcasting performance (packet collision probability and successful transmission probability) of CSMA/CA in the one-hop cluster of the FANET, while the traffic flow model is used to approximately

describe the spatial distribution of the swarm when moving through the narrow space.

3) Different from expiration, node density, and reception ratio, the CNP introduced in FMAC can correctly and sensitively reflect the change of UAV density since it directly relates to the dynamics of inter-UAV spacing. Moreover, the CNP value can be obtained locally by using only the motion states of neighbors. Additionally, as for FMAC, even with limited received packets, it can appropriately adjust the CW size of CSMA/CA since it neatly utilizes the information contained in these packets.

The remainder of this article is organized as follows. In Section III, we present the communication model and the flocking functions. Then, the main problem is formulated in Section IV, where we theoretically point out the challenge of the typical CSMA/CA protocol when used in flocking-based UAV scenarios. The FMAC protocol is elaborated in Section V and the evaluation of the proposed mechanism is provided in Section VI. Finally, the conclusion is given in Section VII.

II. Related Works

Adaptive CW mechanisms for CSMA/CA have been investigated for decades. However, most studies such as [21]–[24] address the adaptive CW problem in unicast service and are mainly based on collision detection with the help of ACK and CTS packets. Since no ACK/CTS mechanism exists, the adaptive CW mechanisms proposed in these studies will be unavailable in the case of broadcast service.

As far as we concerned, only several adaptive CW mechanisms of CSMA/CA have been proposed for one-hop broadcast services [25]–[29]. These mechanisms are generally designed for VANETs, utilizing various triggering states that are observed locally. The states mainly include expiration, node density, and reception ratio. These mechanisms are simple enough to be implemented and can provide efficient broadcast service.

The adaptive CW mechanisms involved in [25] and [26] are based on packet expiration, which means one packet is not transmitted during its lifetime. According to the reverse binary exponential back-off (RBEB) mechanism in [25], each node treats one fairly large value as initial CW size, which is halved every time one packet is found expired. Once no expiration happens when there is one packet being successfully put on the channel, the CW resets to its initial value. The collision-expiration backoff (CEB) mechanism in [26] utilizes the local observed ratio of packet expiration. Within each observation interval (OI) which includes several beacon periods, the number of expired packets is counted. Then, at the end of the OI, the ratio of the expired packet number counted in the current OI to one preset threshold is computed. If the ratio is smaller than 1, the CW will be divided by the ratio, otherwise, the CW will be multiplied by the ratio. However, packet expiration occurs only when the node density becomes pretty large. In the case with aggressive network node density, one specific node may wait for enormous other nodes' transmissions in the shared channel, and hence it can be of the high risk that its packet is not transmitted during its lifetime. As the matter of fact, due to limited communication range, even for the worst case of the swarm (when the swarm topology densifies), the node density will not be dramatically high such that few packet expirations will be witnessed.

The adaptive CW mechanisms in [27] and [28] are based on local node density, which is estimated within each interval by counting the different sources from which at least one beacon has been received. Chrysostomou *et al.* [27] designed the fuzzy logic-based broadcast MAC controller (FLB) which is responsible for the adjustment of CW size. The node density measured at the current and previous sampling intervals is two inputs of the FLB controller, while the output being the CW size. The FLB controller is conducted by the nonlinear control law called linguistic IF-THEN rule, to dynamically adjust the output CW according to input values. The idea behind the control law is that when the current node density is higher than one threshold, the CW size is increased no matter how small the previous node density is. In [28], at each interval, the CW is computed by directly multiplying the node density with one predefined parameter.

However, when the packet collision occurs frequently as node density increases, many packets sent by neighboring sources may not be successfully received, causing incorrect estimation of node density. This will make the CW size under node density-based mechanisms cannot be adjusted in an available manner. It is worth mentioning that the node density-based mechanism is similar to our CNP-based mechanism, which is all related to the number of neighboring sources from which at least one packet is received within the last interval/period. However, there are primary differences and strengths in our CNP-based mechanism in comparison with node density-based mechanisms, which will be briefly analyzed in Section V.

Moreover, in [28], other efficient methods, which are idle time based, stop time based, and speed based, have been proposed to evaluate the local node density. The idea involved in the idle time-based method is to compare the time the channel is idle due to the back-off mechanism and the time the channel is occupied by collisions within each interval. Intuitively, the higher the ratio of idle time to the collision time, the lower the node density. However, since the collisions cannot be detected in broadcast environments, one approach to compute the collision time is taken by multiplying the number of lost beacons with the time needed for transmitting one beacon. As for the stop time based and speed-based methods, local node density is evaluated based on the time the car is stopped within each interval or the speed of each node, respectively. The reasonability lies in the fact that the vehicle might be stopped or running at low speed if the road traffic is jammed, which means the network is witnessing one increase of the node density. However, the methods are not available in FANET since each fixed-wing UAV requires high forwarding speed to keep itself in the sky.

In [29], the adaptive CW mechanism is proposed based on the local reception ratio, which is estimated by the percentage of packets that are successfully received. Since packets are sequenced and broadcasted accordingly by each generator, one node can find unsuccessful packets sent to it from one specific neighbor if there are gaps between the two latest received sequence numbers. Then, the weighted reception

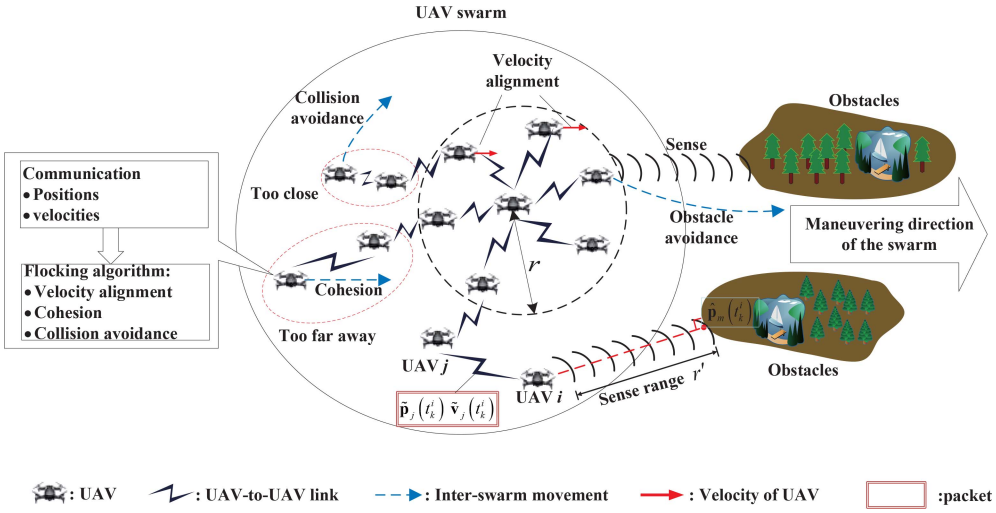

Fig. 1. Collective movement of UAV swarm under flocking control in the obstacle-based environment.

ratio regarding the neighbor can be computed iteratively by the node, where one gap is counted by 0 and one received sequence number is counted by 1. The channel is divided into intervals with constant lengths, and at the end of each interval, the local reception ratio is then computed by the node as the mean value of weighted reception ratios regarding all potential neighbors. If the local reception ratio is larger than that computed within the last interval for one certain threshold, the CW size will decrease, otherwise the opposite. However, for the reception ratio-based mechanism, the length of each interval should be long enough such that the local reception ratio could reflect the real transmission performance during the interval as much as possible.

III. System Model

A. Network Model

We consider N UAVs in the swarm, each of which is equipped with one communication device and the UAVs form the FANET among the swarm, as shown in Fig. 1. We assume that all UAVs in the swarm have the same carrier sensing range and communication range, and the two ranges for each UAV are identical, denoted by $r > 0$. For certain UAV i, its neighbors are the UAVs that are within the distance of r from UAV i. One channel denoted as C_0 is used by the FANET, and the state packets containing the position and velocity information of the sender UAVs are transmitted among neighbors on the channel and will be used for the swarm to develop various cooperation control capabilities. Without loss of generality, we focus on the flocking capability in this article.

We also assume that the packets are periodically broadcasted in one-hop mode with the predetermined period of T_p, and the transmission period for each node is identical but with varying starting time. Moreover, the period length T_p remains unchanged to guarantee the equal interval between arrivals of every two consecutive packets. Denote the packet generating time sequence of the ith UAV as $t_1^i, t_2^i, t_3^i, \ldots, t_k^i, \ldots$, where the superscript k denotes the kth period. When one packet is generated by the application layer of the ith UAV at t_k^i, the packet will be sent immediately to the MAC layer. In order to simplify the model, we neglect the transmission time between the application layer and the MAC layer, as well as the propagation delay of signals.

In fact, there will be heterogeneous periods for different transmission services. For example, UAVs send flight status messages to the ground station at the rate of 2 Hz and UAVs also send pose messages to update the ground station and other swarm vehicles as to their current states at the rate of 10 Hz. However, in this article, we only consider the transmission of periodic packets containing UAVs' states, which is needed to develop flocking capability. Therefore, we assume that the considered periodic packets are transmitted independently via one transceiver, while other types of packets are transmitted via another transceiver, as assumed in [14].

B. Broadcast Mode of CSMA/CA Protocol

Since the CSMA/CA can be conveniently implemented without precise time synchronization requirement which is needed in the TDMA protocol, in this article, we consider the UAVs share the channel C_0 with its neighbors based on the CSMA/CA protocol. Note that the CSMA/CA protocol in this article refers to the typical contention mechanism of the IEEE 802.11 DCF [32], and it will be described in detail in the following contexts.

In the broadcast mode of CSMA/CA, when one packet is put into the MAC queue, the MAC layer informs the radio interface to sense the channel. When the channel is sensed idle for a certain time interval equaling to DCF interframe spacing (DIFS), the MAC layer starts the backoff mechanism. During the backoff mechanism, the channel is divided into slots and the node activates one backoff counter, which holds one integer value randomly chosen from the range $(0, \mathrm{CW})$. If the channel is sensed as idle at one slot, the backoff counter will be decreased by 1 at the end of that slot, otherwise, it will be frozen until the channel becomes empty and ready for the next DIFS time. When the backoff counter decreases to 0, the node will immediately broadcast its message. At the broadcast

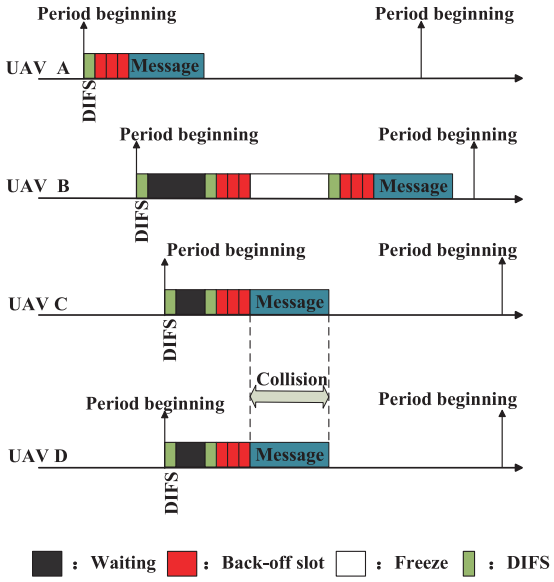

Fig. 2. CSMA/CA at broadcast mode.

mode, the size of CW is fixed and no ACK is sent when one packet is successfully received. Hence, no retransmission is required if the packet is not correctly decoded at the receiver. Fig. 2 illustrates the process, where UAVs *A*, *B*, *C*, and *D* are within the communication range of each other, and their packets are sent to the MAC queue at different periods with different backoff values, e.g., 3, 6, 3, and 3 for each UAV. When UAV *A* sends its packet, other UAVs keep sensing the channel until it is idle for DIFS. After the transmission of *A*'s packet, other UAVs start their backoff process. Fortunately, UAVs *C* and *D* could send their packets after three slots, while UAV *B* needs to freeze its backoff counter and waits for one total DIFS time after their transmissions.

Note that in CSMA/CA, access collision could occur on channel C_0, when two or more UAVs within two hop of each other broadcast their packets simultaneously. For example, in Fig. 2, UAVs *C* and *D* will send their packets simultaneously, resulting in the access collision at their potential receivers. Moreover, for each broadcast packet, it will be dropped if it is not sent on the channel before the arrival of its next packet. We denote such a phenomenon as packet expiration.

C. Collective Motion Model

We assume that all UAVs move in the $n = 2$ dimensional Euclidean space. Actually, in practice, the UAV always moves in the $n = 3$ dimensional Euclidean space. To simplify the description in this article and make some figures more intuitive, we assume all UAVs move in the 2-D Euclidean space. In fact, this 2-D assumption can be achieved by using the consensus algorithm [33] in a practical scenario, while treating the flight height of UAV as the status that needs to be a consensus. Moreover, the proposed adaptive protocol is independent of the dimension of UAV's motion.

1) Single UAV Motion Model: Each UAV in the swarm is equipped with the flight control module, which is used to steer the movement of the UAV. The position and velocity of the *i*th UAV at any time instant t are represented by $\mathbf{p}_i(t) \in R^n$ and $\mathbf{v}_i(t) \in R^n$, respectively. The kinetic dynamics of the UAV is described by

$$\begin{cases} \dot{\mathbf{p}}_i(t) = \mathbf{v}_i(t) \\ \dot{\mathbf{v}}_i(t) = \mathbf{u}_i(t) \end{cases} \tag{1}$$

where $\mathbf{u}_i(t) \in R^n$ represents the input of the control module at time instant t.

We consider the flight environment in the presence of obstacles, which could be mountains, buildings, or dangerous fields. For simplicity, all considered obstacles are restricted to be connected convex region in R^n. The connected convex region can refer to [20], which is the conception in the field of Geometry. Intuitively, the connected convex region means that the region is convex and there is no concave place. In most practical flight environments, some narrow spaces may be formed by pairwise obstacles as shown in Fig. 1.

2) Flocking Motion Model: In this article, we consider the swarm moves through the narrow space in a flocking manner. Note that the narrow space flight environment is motivated by the scenario mentioned in [20], and it is used to create a scenario where the UAV swarm will densify. As for flocking motion, it requires collision avoidance (between UAVs or with obstacles) and cohesion among the swarm during the collective motion. The collision avoidance aims to ensure the safety of the swarm, while cohesion is supposed to preserve the connectivity of the FANET, such that each UAV in the swarm can alert each other to environmental threats and attacks [34]. Such cooperative movement relies on the flocking capability, which is developed by flocking algorithms through the control module [9], [20]. For each UAV, the flocking algorithm updates its control input at the start of each period based on motion states announced in neighbors' packets, which are received during the last period on channel C_0. As a result, the control module will decide the upcoming interswarm motion (moving far away/close to neighboring UAVs) of the UAV, to achieve cooperative movement. Moreover, the control input of each UAV is also affected by the obstacle states sensed by onboard sensors.

In this article, the control module of each UAV uses the flocking algorithm proposed in [20]. Particularly, at the beginning of the kth period t_k^i, UAV i updates the control input $\mathbf{u}_i(t_k^i)$ according to

$$\mathbf{u}_i(t_k^i) = \mathbf{u}_i^{\text{nei}}(t_k^i) + \mathbf{u}_i^{\text{obs}}(t_k^i) + \mathbf{u}_i^{\text{leader}}(t_k^i). \tag{2}$$

The first term $\mathbf{u}_i^{\text{nei}}(t_k^i)$ aims to steer UAV i to move to the appropriate position regarding its neighbors to avoid collision and achieve cohesion. The second term $\mathbf{u}_i^{\text{obs}}(t_k^i)$ acts when any obstacle is sensed within the range of r' by UAV i and can be used to achieve obstacle avoidance for UAV swarm. The third term $\mathbf{u}_i^{\text{leader}}(t_k^i)$ is affected by the motion states of the virtual leader, making the swarm follow the virtual leader to the desired destination. As derived in [20], the explicit form of each term in (2) is given by

$$\mathbf{u}_i^{\text{nei}}(t_k^i) = -c_1^{\text{nei}} \sum_j \nabla_{\mathbf{p}_i} \psi_{\text{nei}}\left(\left\|\tilde{\mathbf{p}}_j(t_k^i) - \mathbf{p}_i(t_k^i)\right\|_\sigma\right) + c_2^{\text{nei}} \sum_j a_{ij}\left(\tilde{\mathbf{v}}_j(t_k^i) - \mathbf{v}_i(t_k^i)\right) \tag{3}$$

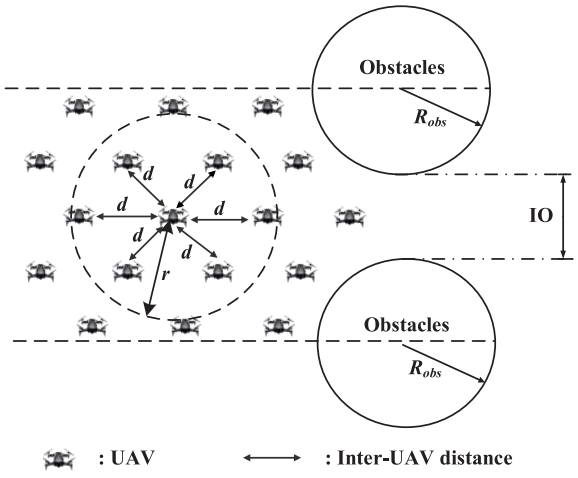

Fig. 3. Considered flight environment.

$$\mathbf{u}_i^{\text{obs}}\left(t_k^i\right) = -c_1^{\text{obs}} \sum_m \nabla_{\mathbf{p}_i} \psi_{\text{obs}}\left(\left\|\hat{\mathbf{p}}_m\left(t_k^i\right) - \mathbf{p}_i\left(t_k^i\right)\right\|_\sigma\right) + c_2^{\text{obs}} \sum_m b_{i,m}\left(\hat{\mathbf{v}}_m\left(t_k^i\right) - \mathbf{v}_i\left(t_k^i\right)\right) \tag{4}$$

and

$$\mathbf{u}_i^{\text{leader}}\left(t_k^i\right) = -c_1^{\text{leader}} \sigma_1\left(\mathbf{p}_i\left(t_k^i\right) - \mathbf{p}_r\left(t_k^i\right)\right) - c_2^{\text{leader}}\left(\mathbf{v}_i\left(t_k^i\right) - \mathbf{v}_r\left(t_k^i\right)\right) \tag{5}$$

respectively, where c_β^α are positive constants for all $\beta \in \{1, 2\}$ and $\alpha \in \{\text{nei},\ \text{obs},\ \text{leader}\}$, and the σ-norm of the vector z is defined as

$$\|z\|_\sigma = \frac{1}{\varepsilon}\left[\sqrt{1 + \varepsilon\|z\|^2} - 1\right] \tag{6}$$

with $\varepsilon > 0$. As shown in Fig. 1, $\tilde{\mathbf{p}}_j(t_k^i)$ and $\tilde{\mathbf{v}}_j(t_k^i)$ in (3), respectively, represent the position and velocity of UAV j, which are derived from the received packet UAV j from UAV i between t_{k-1}^i and t_k^i. Moreover, $\hat{\mathbf{p}}_m(t_k^i)$ and $\hat{\mathbf{v}}_m(t_k^i)$ in (4), respectively, denote the position and velocity of "on-vehicle neighbor" m of UAV i at time t_k^i, with m being the nearest point sensed by UAV i on the boundary of the obstacle. $\mathbf{p}_r(t_k^i)$ and $\mathbf{v}_r(t_k^i)$ represent the position and velocity of one virtual leader. $\psi_{\text{nei}}(\|\tilde{\mathbf{p}}_j(t_k^i) - \mathbf{p}_i(t_k^i)\|_\sigma)$ is called the neighboring potential function in i's perspective, depending on the relative position between i and its neighbor j, which is the map $R^n \to R^+$.

If UAV i flies too close to UAV j, this potential will act as the repulsion via $\mathbf{u}_i^{\text{nei}}$ to steer i move away from j to avoid collision. Otherwise, if UAV i is too far away from UAV j, this potential will act as attraction to steer i move toward j. Another potential function $\psi_{\text{obs}}(\|\hat{\mathbf{p}}_m(t_k^i) - \mathbf{p}_i(t_k^i)\|_\sigma)$ performs similarly as $\psi_{\text{nei}}(\|\tilde{\mathbf{p}}_j(t_k^i) - \mathbf{p}_i(t_k^i)\|_\sigma)$, except that it acts as the repulsion through $\mathbf{u}_i^{\text{obs}}$ to steer UAV i move far away from obstacles.

IV. Problem Formulation

As shown in Fig. 3, only two pairwise obstacles are considered in this section, which form the narrow space. We will analyze the MAC-layer performance of the CSMA/CA protocol in the scenario where the swarm has to flock through the narrow space. Even though many models have concentrated on the broadcasting performance of the CSMA/CA protocol in VANET [25], [35]–[39], they cannot be directly used in the flocking scenario. This is because the motion behavior of the swarm is under control, causing that the UAV traffic flows in the FANET are different from that in VANET. Specifically, the motion behavior of all UAVs within each period is automatically and locally decided by the control input via (1), which is updated at the beginning of the corresponding period. The control input is computed using the flocking algorithm (2) with the knowledge of the flight status of neighbors. Each UAV does not move casually or randomly such that making the UAV behave like it is being controlled. Based on the analytical model, we will provide insight on the limitation of the CSMA/CA protocol in the flock scenario. The notations in this section are summarized in Table I.

TABLE I
Summary of Important Symbols

Symbols	Notations
N	The whole number of UAVs.
$\mathbf{p}_i(t)$	The position vector of UAV i at time instant t.
$\mathbf{v}_i(t)$	The velocity vector of UAV i at time instant t.
$\mathbf{u}_i(t)$	The control input vector of UAV i at time instant t.
r	The communication range of each UAV.
C_0	The control channel.
T_p	The time length of each period.
t_k^i	The start time instant of the kth period of UAV i.
r'	The sensing range of the sensor onboard at each UAV.
$\tilde{\mathbf{p}}_j\left(t_k^i\right)$	The position of UAV j announced by j's packet received by i from time instant t_{k-1}^i to t_k^i.
$\tilde{\mathbf{v}}_j\left(t_k^i\right)$	The velocity of UAV j announced by j's packet received by i from time instant t_{k-1}^i to t_k^i.
$\mathbf{p}_r\left(t_k^i\right)$	The position vector of the virtual leader at time instant t_k^i.
$\mathbf{v}_r\left(t_k^i\right)$	The velocity vector of the virtual leader at time instant t_k^i.
CW	The size of the contention window.
P_{col}	The packet collision probability on channel C_0.
P_{succ}	The successful transmission probability on channel C_0.
p_g	The probability of each UAV for generating one packet.
p_{busy}	The probability that the channel C_0 is busy.
N_{one}	The number of UAVs within the one-hop cluster.
IO	The minimum inter-obstacle distance of the pairwise obstacles in Fig. 3.
R_{obs}	The radius of each obstacle in Fig. 3.
$N_{segl}(t)$	The number of UAVs from segtion l ($l = 1, 2, 3$) passed the reference line during the time interval $(0, t)$ in Fig. 3.
$N_{out}(t)$	The number of UAVs passed the reference line within the time interval $(0, t)$ in Fig. 3.
$\lvert\mathbf{v}_{ave}\rvert$	The absolute value of the average speed of all UAVs.
$L(x_{obs})$	The length of reference line located at x_{obs}.
d	The inter-UAV distance when the swarm is maintaining the stable topology.

A. Analytical Performance of the CSMA/CA Mechanism

Consider a set of UAVs, where any two UAVs are within the communication range of each other. Note that the overlay FANET among the swarm is a multihop communication network in this article. However, in this section, we would like

to provide one approximate and a theoretical glimpse of the CSMA/CA performance deterioration caused by the increase of UAV density, which is defined as the number of UAVs within a one-hop communication range. Since hidden terminals are ignored in the following theoretical model (however we have stated their effects on MAC performance in the last paragraph of this section), we therefore only need to consider the CSMA/CA performance within the one-hop cluster. In the following, we will analyze the broadcasting performance (i.e., the packet collision probability P_{col} and the successful transmission probability P_{succ}) of CSMA/CA in the one-hop cluster of the FANET.

For certain time instant t, the MAC states of each UAV mainly include: 1) idle, referring that the UAV does not have any packet to send and 2) backoff, referring the UAV's backoff state, which is represented by the remaining value of its backoff counter. This section uses -1 to represent the idle state and uses $0, 1, \ldots, \text{CW}-1$ to represent possible backoff states. For each UAV, as the time tends to be infinite, the state distribution at each time instant tends to be steady. Therefore, we use $b(k)$ where $k = -1, 0, 1, \ldots, \text{CW}-1$, to represent the stationary distribution of corresponding state of each UAV.

We simplify the chain model in [35] by only considering periodic one-hop broadcast service, as shown in Fig. 4. Note that the starts of all nodes are synchronized in [35], which is different from the model in this section. In our model, the period starts of all UAVs may not be synchronized. Assume each period can be divided into N_v "virtual time slot," and the UAV sends its packet at one virtual time slot. Then, we have

$$p_g = \frac{1}{N_v} \tag{7}$$

where p_g represents the probability of each UAV for generating one packet. p_{busy} in (8) denotes the probability that the channel C_0 is busy, then based on Fig. 4, the transition equation of MAC states in [35] is modified as

$$\begin{cases} p[k|-1] = \frac{1}{\text{CW}}, & \text{for } k = 0, 1, 2, \ldots, \text{CW}-1 \\ p[k-1|k] = 1 - p_{\text{busy}}, & \text{for } k = 0, 1, 2, \ldots, \text{CW}-1 \\ p[k|k] = p_{\text{busy}}, & \text{for } k = 1, 2, \ldots, \text{CW}-1. \end{cases} \tag{8}$$

According to (8), we have

$$\begin{cases} \frac{b(-1)}{\text{CW}} + b(1)\big(1-p_{\text{busy}}\big) = b(0)\big(1-p_{\text{busy}}\big) \\ \frac{b(-1)}{\text{CW}} + b(2)\big(1-p_{\text{busy}}\big) = b(1)\big(1-p_{\text{busy}}\big) \\ \qquad\qquad\qquad\qquad \vdots \\ \frac{b(-1)}{\text{CW}} + b(k)\big(1-p_{\text{busy}}\big) = b(k-1)\big(1-p_{\text{busy}}\big) \end{cases} \tag{9}$$

and

$$b(-1)p_g = b(0)\big(1-p_{\text{busy}}\big). \tag{10}$$

According to (9), we can further derive the relations between $b(-1)$, $b(0)$, and $b(k)$ $(k = 1, 2, \ldots, \text{CW}-1)$ as follows:

$$b(-1)\frac{k}{\text{CW}} + b(k)\big(1-p_{\text{busy}}\big) = b(0)\big(1-p_{\text{busy}}\big). \tag{11}$$

Additionally, by substituting (10) into (11), we have

$$b(k) = \frac{p_g\text{CW}-k}{p_g\text{CW}}b(0). \tag{12}$$

Finally, by using the normalized condition $\sum_{k=-1}^{\text{CW}-1} b(k) = 1$, we can obtain $b(0)$ as

$$b(0) = \left\{1 + \frac{1-p_{\text{busy}}}{p_g} + \frac{1}{p_g\text{CW}}\left[(\text{CW}-1)p_g\text{CW} - \frac{\text{CW}(\text{CW}-1)}{2}\right]\right\}^{-1}. \tag{13}$$

Note that the state distribution that the channel is busy at N_v we discussed here refers to a steady-state distribution. The considered virtual time slot N_v is assumed to be long enough to finish the transmission of any broadcast packet. Therefore, the steady distribution that the channel is busy is only decided by the distribution of $b(0)$ at the corresponding virtual time slot N_v. Based on the above discussion, p_{busy} can be represented by $b(0)$ as

$$p_{\text{busy}} = 1 - [1-b(0)]^{N_{\text{one}}} \tag{14}$$

where N_{one} denotes the number of UAVs in the one-hop cluster. The value of $b(0)$ can be easily calculated by substituting (7) and (14) to (13), and therefore, p_{succ} and p_{col} can be computed as follows:

$$P_{\text{succ}} = N_{\text{one}}b(0)\Big\{[1-b(0)]^{N_{\text{one}}-1}\Big\} \tag{15}$$

$$P_{\text{col}} = 1 - N_{\text{one}}b(0)\Big\{[1-b(0)]^{N_{\text{one}}-1}\Big\}. \tag{16}$$

B. Traffic Flow Model in the Flocking Scenario

As per (15) and (16), only the value of N_{one} needs to be solved. To calculate N_{one}, the distribution of UAVs within the swarm is required. During flocking in the considered scenario, all the three terms in (2) will affect the motion behaviors of all UAVs, and further their spatial distributions within the swarm. To simplify the analysis of the UAV distribution of the scenario in Fig. 3, several assumptions are made, which are only applied to the rest of this section. First, we assume that there are infinite UAVs in the swarm and the maximum width of the swarm along the vertical direction is $IO + 2R_{\text{obs}}$, as shown in Fig. 3. In fact, the traffic passes through the obstacles tends to be 0 since the number of UAVs in the swarm is limited. Therefore, we assume that there are infinite UAVs in the swarm such that as the time tends to be infinite, the UAV traffic passing through the reference line will finally reach steady. The following analyses are based on the steady traffic assumption. Furthermore, we assume the swarm has formed a stable structure before moving to the proximity of the two obstacles. In the stable structure of the UAV swarm, the distance between each pair of neighbors is maintained as d [20].

Based on the considered scenario, the traffic flow model is developed in Fig. 5, which approximately captures the spatial dynamics of the UAVs. In Fig. 5, the x axis lies in the middle of the two obstacles, and the y axis is tangent to the left edge of the obstacles. Moreover, the reference line located at $x_{\text{obs}} = (R_{\text{obs}}, 0)$ and parallel to the y axis is used to observe the UAVs passing though it.

In the considered scenario, the UAVs moving through the reference line mainly come from three segments, i.e., segment 1, segment 2, and segment 3, as shown in Fig. 5, resulting

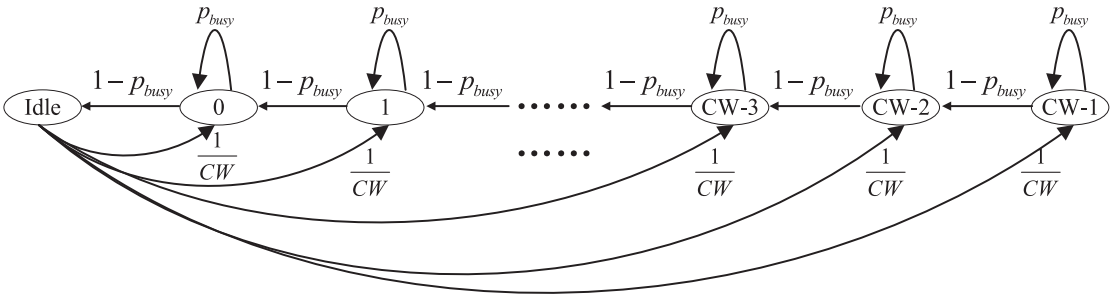

Fig. 4. 1-D chain model.

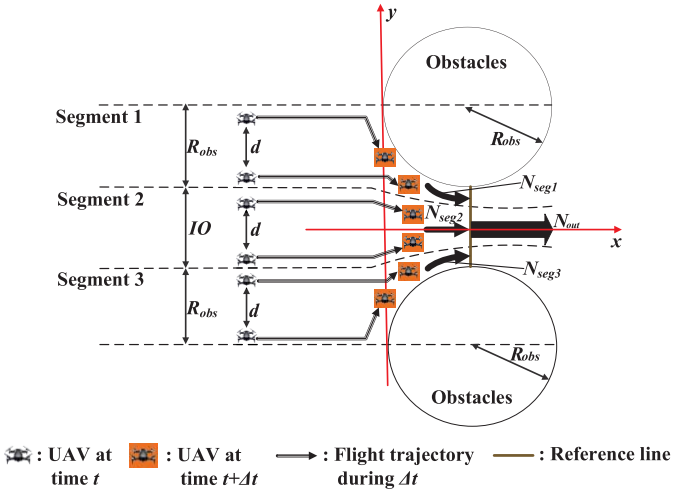

Fig. 5. Model of UAV traffic flow.

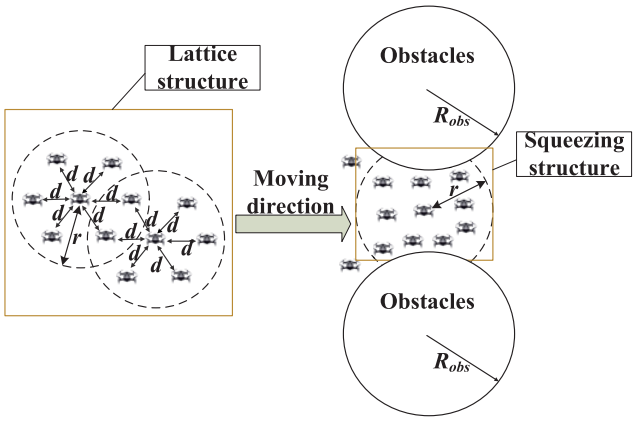

Fig. 6. Move through narrow space.

in following differential equations:

$$\frac{\partial N_{\text{out}}(t)}{\partial t} = \frac{\partial N_{seg1}(t)}{\partial t} + \frac{\partial N_{seg2}(t)}{\partial t} + \frac{\partial N_{\text{seg3}}(t)}{\partial t}. \quad (17)$$

Obviously, $\partial N_{seg1}(t)/\partial t$ equals $\partial N_{\text{seg3}}(t)/\partial t$. To calculate $\partial N_{seg1}(t)/\partial t$ or $\partial N_{\text{seg3}}(t)/\partial t$, the UAV flow in the frontier fraction of segment 1, can be treated as the one-way path tangent to the boundary of the obstacle, as illustrated in the macroscopic flow in Fig. 5. Additionally, according to the assumption that the swarm holds the stable structure before meeting the obstacles, we can obtain

$$\frac{\partial N_{\text{seg1}}(t)}{\partial t} = \frac{\partial N_{\text{seg3}}(t)}{\partial t} = \frac{\sqrt{2}d|\mathbf{v}_{\text{ave}}|}{\sqrt{2}R_{\text{obs}}} = \frac{d|\mathbf{v}_{\text{ave}}|}{R_{\text{obs}}} \quad (18)$$

where $|\mathbf{v}_{\text{ave}}|$ represents the modulus of the average speed of UAVs in the swarm. Moreover, according to the macroscopic flow in Fig. 5, we have

$$\frac{\partial N_{\text{seg3}}(t)}{\partial t} = \frac{IO}{d}|\mathbf{v}_{\text{ave}}|. \quad (19)$$

Therefore, the value of the swarm, N_{one}, over the narrow space can be obtained as

$$N_{\text{one}} = \int_0^r \frac{r}{L(x_{\text{obs}})}\frac{\partial N_{\text{out}}(t)}{\partial t}dx_{\text{obs}}. \quad (20)$$

By substituting (20) to (15) and (16), the broadcasting performance of CSMA/CA under the flocking scenario can be developed. Note that the proposed approach can be used to analyze other flight environments with more complicated obstacle distributions.

C. Limitation of CSMA/CA in the Flocking Scenario

Since the cooperation movement of UAV swarm is achieved by the flocking algorithm in (2), two major swarm structures will be formed as shown in Fig. 6.

1) *Lattice Structure:* When all UAVs fly as the flock in areas without obstacles nearby, they will maintain the lattice structure, and the nearest distance to the neighbors is maintained as d. The value of d is designed to maximize the mission coverage of the swarm. Since no obstacle is encountered, this structure is mainly achieved by $\psi_{\text{nei}}(\|\tilde{\mathbf{v}}_j(t_k^i) - \mathbf{v}_i(t_k^i)\|_\sigma)$.
2) *Squeezing Structure:* When the swarm moves within the proximity of some obstacles, the term $\psi_\beta(\|\hat{\mathbf{p}}_m(t_k^i) - \mathbf{p}_i(t_k^i)\|_\sigma)$ acts, making the fraction of swarm close to the obstacles acts like being squeezed to avoid collision with obstacles.

According to the macroscopic flow in Fig. 5, it can be observed that during the switching from lattice structure to squeezing structure, the UAV density of the swarm is increased. This is because the flocking enforces shorter inter-UAV distance for the UAVs close to the obstacles to avoid collisions. In contrast, the UAV density will be decreased during the switching from squeezing structure to lattice structure. Since N_{one} is directly proportional to the UAV density of

the swarm, the increase of UAV density will cause deterioration on the broadcasting performance of CSMA/CA according to (15) and (16). In fact, (15) and (16) do not consider hidden UAVs, but it can be foreseen that the hidden UAVs will impose extra deterioration. Therefore, the CSMA/CA protocol fails to maintain the reliable broadcast service during flocking in the considered scenario, owing to the fixed CW size and the varying UAV density.

V. Proposed Flocking MAC Protocol

To achieve the flocking capability in the density-varying scenario, the MAC protocol should be able to maintain high reliability for the broadcast service during the whole flocking motion. In order to achieve this goal, we propose the flocking MAC (FMAC) protocol, where each UAV could update the CW size at each period adapting to the density variation. Once the new CW is acquired, each UAV executes the same backoff procedure as CSMA/CA and broadcasts its packet on channel C_0 to its one-hop neighbors until the backoff counter reaches 0. In the proposed FMAC protocol, each broadcast packet must contain its own identifier (e.g., UAV ID), position, velocity, and control input. The position information is encoded as two 4-B floating point quantities, for x and y coordinate values. Similarly, the velocity and control input is, respectively, encoded as two 4-B floating point quantities, for components of velocity along x axis and y axis. We assume that the lifetime of the received packet is T_p, which is the same as the period length. Upon not receiving one beacon from one neighbor for longer than timeout interval T_p, the UAV assumes that the neighbor has failed or gone out-of-range, and deletes the neighbor information from its table. For certain UAV i, $N_i^{\text{nei}}(t_k^i)$ is the node set at the kth period containing the IDs of its one-hop neighbors. In this section, we will elaborate on the CW update mechanism within the FMAC protocol.

A. Current Collective Neighboring Potential

To decide the new CW in the FMAC protocol before transmitting one packet, each UAV should calculate the current CNP first, defined as $V(N_i^{\text{nei}}(t_k^i))$ and is calculated based on the positions of neighboring UAVs in $N_i^{\text{nei}}(t_k^i)$, i.e.,

$$V\left(N_i^{\text{nei}}\left(t_k^i\right)\right) = \sum_{j \in N_i^{\text{nei}}\left(t_k^i\right)} \psi_{\text{nei}}\left(\left\|\tilde{\mathbf{p}}_j\left(t_k^i\right) - \mathbf{p}_i\left(t_k^i\right)\right\|_\sigma\right) \quad (21)$$

where $\psi_{\text{nei}}(\|\cdot\|_\sigma)$ is given by [20]

$$\psi_{\text{nei}}(\|\cdot\|_\sigma) = \int_{d_\sigma}^{\|\cdot\|_\sigma} \phi_{\text{nei}}(s)ds. \quad (22)$$

Note that $\phi_{\text{nei}}(z) = \rho_h(z/r_\sigma)\phi(z - d_\sigma)$, with

$$\rho_h(z) = \begin{cases} 1, & \text{if } z \in [0, h) \\ \frac{1}{2}\left[1 + \cos\left(\pi \frac{(z-h)}{(1-h)}\right)\right], & \text{if } z \in [h, 1] \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

and

$$\phi(z) = \frac{1}{2}[(a+b)\sigma_1(z+c) + (a-b)] \quad (24)$$

where $\sigma_1(z) = z/\sqrt{1+z^2}$ and $0 < h < 1$. The three constants a, b, and c satisfy $0 < a \leq b$ and $c = |a-b|/\sqrt{4ab}$. Moreover, the terms d_σ and r_σ are obtained by substituting d and r into (6), respectively. Note that (23) and (24) were all designed in [20].

As for CNP $V(N_i^{\text{nei}}(t_k^i))$, it is the resultant potential in i's perspective. Choose any subset S_i^{nei} from $N_i^{\text{nei}}(t_k^i)$, and assume the elements in S_i^{nei} remain unchanged for enough periods T_s. Note that this assumption is only used for the following discussions in this section. Then, we describe some interesting observations on the CNP.

1) If the swarm maintains the lattice structure after t_k^i as in the left part of Fig. 6, the intervehicle distance between i and any neighbor in S_i^{nei} will remain d for upcoming T_s periods. Therefore, the change of CNP will maintain 0 or constant small value (since there may be some offset between their broadcasting moments).
2) If the swarm switches from the lattice structure to the squeezing structure after t_k^i, the neighborhood of i will become more and more cohesive, meanwhile, the intervehicle distance between i and any neighbor in S_i^{nei} will decrease for upcoming T_s periods. Therefore, the CNP will experience one sharp increase.
3) Oppositely, when inverse switching happens, the neighborhood of i becomes more and more separative and the intervehicle distance between i and any neighbor in S_i^{nei} will increase for upcoming T_s periods, causing the decrease of the CNP.

To verify our above-mentioned arguments, the CNP variations of all UAVs within 1 s over the scenario in Fig. 6 is measured. Specifically, the boundaries of the two obstacles are circles and are of the same radius of 25 m centered at (300, 30) and (300, 100), respectively. One swarm consisting of 150 UAVs is considered to conduct collective motion with the help of a flocking algorithm in (2). Fig. 7 shows the snapshots of the swarm at several time instants, where the two red hollow circles represent obstacles, and black hollow circles represent UAVs. The simulation runs 20 time intervals, with each equaling to 1 s, and we measure the average CNP variation within each interval. The average CNP variation is the average value of CNP variations of all UAVs at each interval. At the initial time $t = 0$, the swarm has formed the lattice structure as shown in Fig. 7(a), and moves toward the right with the speed of $(6, 0)$ m/s. At $t = 10$ s, the swarm reaches the neighborhood of the obstacles as shown in Fig. 7(b). Then, the swarm has to maneuver through the two obstacles, as shown in Fig. 7(c). Fig. 7(d) shows our measurement of average CNP variations within each time interval in the considered scenario. It can be observed that from 1st to 10th interval, the average CNP variations equal to 0 within these intervals since the swarm is maintaining the lattice structure. After 10 s, the swarm is moving through the obstacles, therefore, it can be observed that the average CNP variations are relatively large, indicating that the CNP in one's perspective is increasing. Moreover, the average CNP variation tends to increase along with the time since the number of UAVs that move into the narrow space within the two obstacles increases.

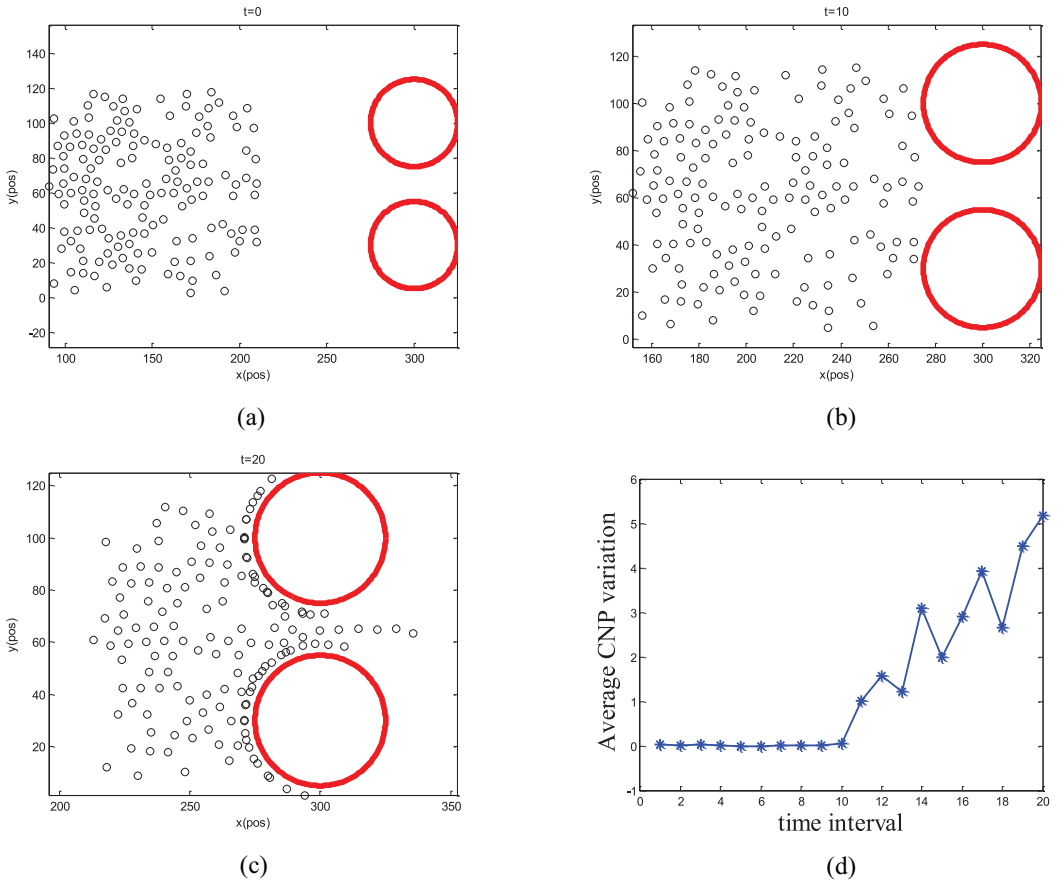

Fig. 7. Swarm snapshots and average CNP variation. (a) Snapshot of the swarm at $t = 0$ s. (b) Snapshot of the swarm at $t = 10$ s. (c) Snapshot of the swarm at $t = 20$ s. (d) Average CNP variations at different intervals.

Based on the aforementioned discussions, we can conclude that the evolution of CNP regarding the same neighbors in one's perspective can reflect the switching state that the swarm is performing, and thus can further show the varying trend of UAV density.

The simulation results in Fig. 7 are simply used to help readers to identify our observations on the CNP described above. Actually, T_s here is set as 1 s (including ten periods) rather than one period since we would like to show more intensive and intuitive results on the evolution of CNP, so as to make our statement above more clear. The parameters in Fig. 7 do not relate to our proposed FMAC protocol, instead, it is just utilized to make the readers be more clear of the motivation of the FMAC protocols.

B. Predicted Collective Neighboring Potential

To be informed the evolution of CNP regarding neighbors in $N_i^{\text{nei}}(t_k^i)$ after t_k^i, UAV i must acquire the new positions of neighbors in $N_i^{\text{nei}}(t_k^i)$. However, the new packet sent by any neighbor j in $N_i^{\text{nei}}(t_k^i)$ may not be received successfully by UAV i after t_k^i due to access collision. Hence, the evolution of CNP based on neighbors in $N_i^{\text{nei}}(t_k^i)$, cannot be observed directly by UAV i. Fortunately, for any UAV j in $N_i^{\text{nei}}(t_k^i)$, the control input $\mathbf{u}_j(t_k^j)$ updated at time instant t_k^j is constant during the upcoming period, which decides the motion of UAV j within the period. This is different from the vehicles in VANET, where the acceleration of each vehicle may vary occasionally by the willing of the driver.

We first illustrate how to predict the upcoming position and velocity of neighboring UAVs in detail. At UAV j's period start t_k^j, it updates its control input and broadcasts it to its surrounding neighbors. UAV i which is the neighbor of UAV j receives the packet and will be informed of the control input as well as the position and velocity of UAV j at t_k^j. UAV i defaults that the control input (announced by the received packet) of UAV j will decide the motion of the UAV j's upcoming one period length since the control input remains unchanged for upcoming one period. In this article, we do not consider the wind force or other intruders in the air environment. Consequently, UAV j's motion within its upcoming period can be predicted by UAV i with small error, while the error is introduced by the latency of broadcast packet send by j. Specifically, according to Newton's second law of motion, we have

$$\begin{cases} \hat{\mathbf{p}}_j = \tilde{\mathbf{p}}_j(t_k^i) + \tilde{\mathbf{p}}_j(t_k^i)T_p + \frac{1}{2}\tilde{\mathbf{u}}_j(t_k^i)T_p^2 \\ \hat{\mathbf{v}}_j = \tilde{\mathbf{v}}_j(t_k^i) + \tilde{\mathbf{u}}_j(t_k^i)T_p \end{cases} \tag{25}$$

where $\tilde{\mathbf{u}}_j(t_k^i)$ is the control input of UAV j announced by j's packet received by i between t_{k-1}^i and t_k^i. Moreover, $\hat{\mathbf{p}}_j$ and $\hat{\mathbf{v}}_j$ represent the predicted position and velocity of j at its next period start. Then, the CNP at i's subsequent period start regarding $N_i^{\text{nei}}(t_k^i)$ can be predicted by substituting (25)

Algorithm 1 CNP-Based Adaptive CW Mechanism in FMAC

Input: $V_{pre}\left(N_i^{nei}\left(t_k^i\right)\right)$ and $V\left(N_i^{nei}\left(t_k^i\right)\right)$ of UAV i.
Output: The size of CW of UAV i.

```
At UAV i's period start t_k^i.
if V_pre(N_i^nei(t_k^i)) < V(N_i^nei(t_k^i)) then
    count = count − 1
else if V_pre(N_i^nei(t_k^i)) > V(N_i^nei(t_k^i)) then
    count = count + 1
else
    count is unchanged
end if
if count > δ then
    CW = CW + Δ
    count=initial
    Anti-overflow
else if count < δ then
    CW = CW − Δ
    count=initial
    Anti-overflow
else
    CW is unchanged
end if
```

into (21), i.e.,

$$V_{\text{pre}}\left(N_i^{\text{nei}}\left(t_k^i\right)\right) = \sum_{j \in N_i^{\text{nei}}\left(t_k^i\right)} \psi_{\text{nei}}\left(\left\|\hat{\mathbf{q}}_j - \hat{\mathbf{q}}_i\left(t_{k+1}^i\right)\right\|_\sigma\right). \quad (26)$$

C. CW Adjustment Mechanism in the FMAC Protocol

By computing $V_{\text{pre}}(N_i^{\text{nei}}(t_k^i))$ with $V(N_i^{\text{nei}}(t_k^i))$, the change of CNP regarding the same neighbors $N_i^{\text{nei}}(t_k^i)$ could be perceived by UAV i. As discussed in the first part of this section, the evolution of CNP for each UAV regarding the same neighbors can reflect the varying trend of the UAV density. Therefore, the size of CW can be adjusted by comparing $V_{\text{pre}}(N_i^{\text{nei}}(t_k^i))$ and $V(N_i^{\text{nei}}(t_k^i))$, which will be described as follows.

Let δ and Δ be two positive constant values. In the adaptive CW mechanism of the FMAC protocol, for any UAV i at the kth period start t_k^i, it first computes the current CNP $V(N_i^{\text{nei}}(t_k^i))$ using (21). Then, the predicted CNP $V_{\text{pre}}(N_i^{\text{nei}}(t_k^i))$ based on motion states of $N_i^{\text{nei}}(t_k^i)$ can be acquired using (26). Subsequently, the two CNPs are compared directly. If $V_{\text{pre}}(N_i^{\text{nei}}(t_k^i))$ is larger (smaller) than $V(N_i^{\text{nei}}(t_k^i))$, another counter of i (different from backoff counter) will be increased (decreased) by 1. If the value of the counter is higher (smaller) than the threshold δ, the CW of i in this period will be increased (decreased) by Δ. Moreover, after each adjustment of the CW size, the counter will be reset to its initial value. Brief illustration of this mechanism is illustrated in Algorithm. 1.

The anti-overflow process in Algorithm. 1 is designed to avoid relative large CW size or relative small CW size. Consider two positive constant values $\text{CW}_{\max}$ and $\text{CW}_{\min}$, the size of any updated CW should be smaller than $\text{CW}_{\max}$ and larger than $\text{CW}_{\min}$. Otherwise, $\text{CW}_{\max}$ or $\text{CW}_{\min}$ will be treated as the new CW size.

D. Differences Between CNP Metric in Comparison With the Other Triggering States

First, the CNP metrics (the current value and predicted value) within each period can be obtained if at least one broadcast packet is received. This makes FMAC more stable in evaluating the network condition than the expiration-based method in the considered flocking scenario since packet expiration in the scenario is almost impossible to be observed. Second, under FMAC, one period is enough to efficiently evaluate the network condition trend by comparing the practical CNP value and predicted CNP value. This makes FMAC more sensitive to the change of network condition than the reception ratio-based mechanism since it will take too much time for the estimated reception ratio to converge to a practical one.

Third, the node density metric is similar to CNP metric, since their values all relate to the number of neighboring sources from which at least one packet is received within one interval. However, there is one primary difference between them. The node density metric utilizes only the number of different sources from which at least one broadcast packet has been received within one interval to evaluate the network condition. However, the CNP utilizes not only the number of sources but also the information involved in their packets, that is, the positions and velocities of the sources. The difference makes our CNP-based mechanism capture the practical network condition more sensitively and accurately than the node density-based method. This is because the serious loss of broadcast packets when the FANET densifies will make the number of received sources be much more different from the real neighbors in the proximity. However, for the CNP metric, it mainly utilizes the information announced by the received sources rather than the number of them.

For example, assume one node has nine neighbors indexed by 1–9, however it only receives packets of neighbors 1, 2, 3, 5, 6, 7, and 9 within one specific period, in which the packets of neighbors 4 and 8 are lost. Therefore the node will treat the node density around itself as 7. After several periods, the FANET densifies and the number of the node's neighbors increases to 13, with the neighbors being indexed by 1–13. Consider another period after the FANET densifies, the node only receives packets of neighbors 1, 3, 4, 6, 8, 11, and 12, where the packets of neighbors 2, 5, 7, 9, 10, and 13 are lost due to increasing contentions. The node will still treat the node density around itself as 7. Therefore, when only utilizing the number of received sources to evaluate the network condition, one will result in an incorrect conclusion. However, even only 7 out of 13 packets are received, the CNP could still notice the change of node density around itself by analyzing the motion information in the seven packets.

VI. Simulation Results and Discussions

This section presents simulation results to evaluate the performance of the proposed FMAC protocol, as compared with the typical CSMA/CA protocol, the CEB-based CSMA/CA protocol [26], the protocol proposed in [28], and the protocol proposed in [29]. The three protocols [26], [28], [29] are chosen as comparative protocols since they are typical

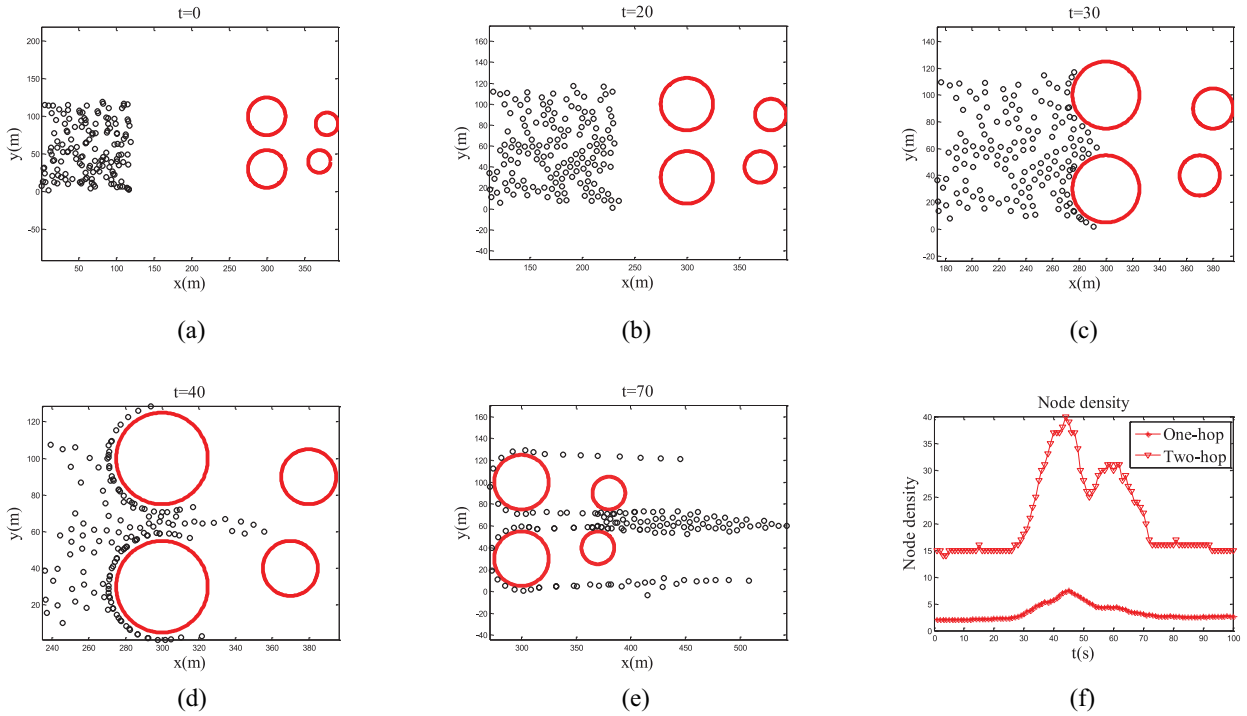

Fig. 8. Flocking motion in the presence of obstacles. (a) $t = 0$. (b) $t = 20$. (c) $t = 30$. (d) $t = 40$. (e) $t = 70$. (f) Node density over time.

adaptive mechanisms and have been proved to achieve efficient broadcast performances. Moreover, the three protocols, respectively, utilize packet expiration, node density, and reception ratio as triggering states, while our FMAC utilizes the CNP. Therefore, choosing the three protocols as comparative protocols will show the advantage of the triggering condition (CNP) proposed in FMAC. Some common parameters at the MAC layer for these protocols are shown in Table II. As for the proposed FMAC protocol in this simulation, some other parameters, such as $\mathrm{CW_{min}}$ and $\mathrm{CW_{max}}$ are set to 32 and 128, respectively.

TABLE II
SIMULATION PARAMETERS

Common parameters at the MAC layer of different MAC protocols	
Parameters	Values
T_p(ms)	100
Packet size (Byte)	64
$T_{slot}(\mu s)$	20
Simulation time (s)	100
Bit Rate (Mb/s)	5
Initial CW	32
DIFS(μs)	50
Parameters of flocking algorithm	
Parameters	Values
Weighting coefficient c_1^{nei}	0.1
Weighting coefficient c_2^{nei}	$2\sqrt{c_1^{nei}}$
Weighting coefficient c_1^{obs}	20
Weighting coefficient c_2^{obs}	$2\sqrt{c_1^{obs}}$
Weighting coefficient c_1^{leader}	0.2
Weighting coefficient c_2^{leader}	$2\sqrt{c_1^{leader}}$
Desired inter-UAV distance d (m)	7
Constant value b	5
Constant value ε	0.1
Constant value h	0.2
Constant value a	5

A. Simulation Scenario and Performance Metrics

In our simulation, the flight environment under consideration consists of four obstacles that are relatively close to each other. The simulation flight environment is motivated by the scenario in [20]. The simulation flight environment is used to create a scenario where the UAV density of the swarm will densify, so as to identify our proposed FMAC protocol. The boundaries of the four obstacles are circles, and two obstacles are of the same radius of 25 m centered at (300, 30) and (300, 100), respectively, while the other two obstacles have the same radius of 15 m centered at (370, 40) and (380, 90), respectively. There are $N = 150$ UAVs in the swarm and the number of UAVs remains constant during the simulation. Initially, all UAVs hover within the region $[0, 120]^2$, with their forwarding velocities being all 0 and positions drawn from the random distribution. The black hollow circles in Fig. 8(a) shows the snapshot of all UAVs at the initial time.

Consider that each UAV can receive information from one virtual leader moving with the speed (6, 0) m/s from the initial coordination (380, 60). With the flocking algorithm in (2), the swarm has to maneuver through the four obstacles tracking the virtual leader. Table II summarizes the simulation parameters of the flocking algorithm. The packet containing motion states is generated at the frequency of 10 Hz by each UAV, which is needed by the flocking algorithm. Moreover, the communication range r and sensor range r' are set to 1.2 and 0.72 d, respectively, as provided in [20]. The total simulation time is 100 s in each run. Fig. 8(a)–(e) shows the snapshots of the swarm at several time instants, where the four red hollow

circles represent obstacles, and black hollow circles represent UAVs. From Fig. 8(a)–(e), we can observe that the UAVs avoid collisions with all obstacles as moving forward.

In the evaluation, two metrics in the MAC layer, namely, successful transmission probability (P_{succ}) and packet collision probability (P_{col}), are considered. These two metrics are measured every 5 s by

$$P_{\text{succ}}(s) = \frac{\sum_{i=1}^{N}\sum_{m=1}^{M_i}\frac{N_i^r(m)}{N_i^{\text{nei}}(m)}}{\sum_{i=1}^{N} M_i} \tag{27}$$

and

$$P_{\text{col}}(s) = \frac{\sum_{i=1}^{N}\sum_{m=1}^{M_i}\frac{N_i^c(m)}{N_i^{\text{nei}}(m)}}{\sum_{i=1}^{N} M_i} \tag{28}$$

respectively, where s represents the sth interval which is of 5-s length. $N_i^{\text{nei}}(m)$ is the total number of vehicles within the communication range of certain UAV i when i is transmitting its m packet, while $N_i^r(m) < N_i^{\text{nei}}(m)$ denotes the number of neighboring vehicles that can receive the mth packet of i successfully. Moreover, $N_i^c < N_i^{\text{nei}}(m)$ denotes the number of vehicles that cannot receive the packet due to access collision. M_i means the total number of packets sent by i on channel C_0 within the sth interval. Note that the wireless channel is assumed ideal, i.e., the only source of packet errors is the access collision. It is worth mentioning that as the distance varies, being closer will achieve higher packet reception probability. This is because as the distance tends to be closer, the signal to noise ratio tends to be higher. Moreover, the capture effect of the MAC layer will also make a shorter distance channel obtain better reception probability. However, in this article, we only focus on the contention mechanism of CSMA/CA, and we will consider this facts in our future work.

B. Simulation Results

In Fig. 8(f), both one-hop density and two-hop density are evaluated. The one-hop density of the swarm at time t is evaluated by $(1/N)\sum_{i=1}^{N} N_i^{\text{nei}}(t)$, while the two-hop density is computed by the maximum number of two-hop neighbors at that time. As shown in Fig. 8(f), the two-hop density has the same trend as the one-hop density, however, it acts more sharply than one-hop density. The two-hop density plays a more important role in determining the broadcast performance of one node since the hidden terminal problem. We then use the two-hop density as an example to illustrate the change of the network condition.

It can be seen from Fig. 8(f) that the two-hop density remains one small value from 0 to 30 s since the swarm is maintaining the lattice structure. Then, the swarm moves to the proximity of the two left obstacles and switches to squeezing structure to avoid collisions from $t = 30$ s and $t = 45$ s, during which the two-hop density experiences one sharp increase. After that, the swarm continues to move toward the two right obstacles from 45 to 50 s, with its two-hop density undergoing one decline. This is because, during the movement between the two left obstacles and the two right obstacles, the swarm tends to return to the lattice structure. From 50 to 60 s, the two-hop density experiences another increase since the swarm is moving through the two right obstacles. After that, from 60 to 70 s, the two-hop density declines as the swarm passes through the two right obstacles. During 70 to 100 s, all UAVs of the swarm has totally passed the obstacles, and the swarm switches back to the lattice structure, resulting in the unchanged low two-hop density.

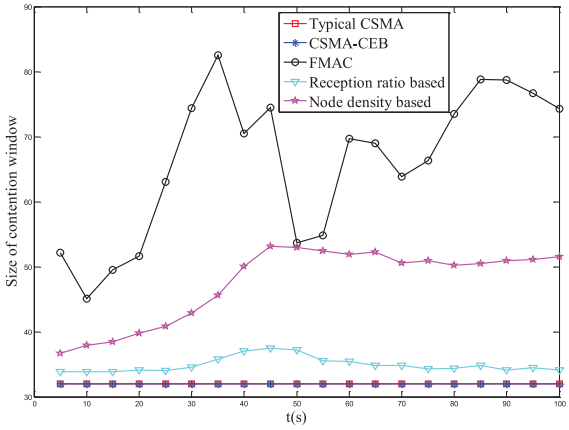

Fig. 9. Size of CW.

δ and Δ of FMAC are set as 1 and 4, which means the CW update interval of FMAC is 0.1 s (one period) and the step value of CW is 4. To make fair comparisons, in the following simulations, we set the update interval of the other four protocols to be 0.1 s, and set their CW step values to be 4.

1) Size of Contention Window: Fig. 9 shows the evolution of the average CW size for the five MAC protocols under consideration. As for the CEB-based CSMA/CA protocol, its average CW remains the same trend as that of the typical CSMA/CA protocol, which remains constant during the simulation time. This is because excessive cohesion of the swarm can be avoided by the flocking algorithm (2), and the number of UAVs sharing a common channel even in the worst case is not large enough to trigger the expiration.

Under the reception ratio-based protocol [29], the CW size always exceeds that of both the CEB-based CSMA/CA protocol and typical CSMA/CA protocol, which indicates that the reception ratio is able to capture the network condition change in the density-varying flocking scenario. The CW size of the reception ratio-based protocol witnesses one slight increase and one slight decline along with time from 30 to 50 s. This adjustment corresponds to the change of node density from 30 to 50 s as shown in Fig. 8(f), however, the change is slighter than the node density-based protocol [28] and the FMAC protocol. This is because the reception ratio-based method always takes long periods to make the local reception ratio converge to the real one, which makes the CW change more slowly and insensitively to the network condition change.

As shown in Fig. 9, the CW size of the node density-based protocol [28] changes more heavily than the typical CSMA/CA protocol, CEB-based CSMA/CA protocol, and reception ratio-based protocol [29]. The CW size of the node density-based protocol witnesses one increase from 30 to 45 s

and one slight decrease from 45 to 50 s, which corresponds to the change of node density from 30 to 50 s. However, the change is slighter than that of the FMAC protocol as shown in Fig. 9. Moreover, its CW adjustment cannot sensitively reflect the change of node density from 50 to 70 s. This is mainly because the loss of received sources will cause incorrect and insensitive understanding of node density change.

For the proposed FMAC, its CW size increases sharply as the node density arises excessively from 20 to 45 s, while undergoing one decline when the density decreases from 45 to 50 s. After that, another increase from 50 to 60 s and one decrease from 60 to 70 s are witnessed by the CW size of FMAC protocol, which corresponds to the node density change from 50 to 70 s as shown in Fig. 8(f). Since the CNP utilizes the motion character of the flocking scenario, it can easily reflect the density change in one's surrounding more than one-hop range. Moreover, we can find that the CW size of FMAC is always higher than the other four protocols. Since the update intervals and CW step values are all considered the same, the simulation results indicate that the proposed CNP-based CW adjustment mechanism in FMAC is much better than the other four protocols in the density-varying flocking scenario. After 70 s, the CW size witnesses one increase. This is because when the swarm moves through the obstacle in the simulation, it is divided into three fractions as shown in Fig. 8(e). As the swarm totally passes through the obstacles, the three fractions will aggregate together due to the virtual leader function involved in the flocking algorithm as shown in (2). In the aggregation process, the nodes in the upside and downside fractions will witness one increase in CNP values since the swarm has the cohesion trend.

2) Performances of Different Protocols: The successful transmission probability and packet collision probability for all protocols are shown in Figs. 10 and 11, respectively. It is notable that as the node density from 30 to 45 s is increasing, both p_{succ} and p_{col} of all five protocols witness deteriorations. This is because the increase of CW size cannot fulfill the performance deterioration caused by the increase of node density, and can be solved by increasing the step value of CW size.

We can observe that the typical CSMA/CA and CEB-based CSMA/CA can achieve almost the same performance on p_{succ} and p_{col}. This is mainly because the adaptive CW mechanism in the CEB protocol is not triggered, resulting in the unchanged CW as that of typical CSMA/CA according to Fig. 9. Moreover, the p_{succ} (p_{col}) of the five protocols during 70–100 s is lower (higher) than that during 0–30 s, even the node densities of the two intervals are the same. The reason can be illustrated as follows. As can be judged by Fig. 8(a)–(e), the nodes in the upside and downside fractions of the swarm during 70–100 s have less two-hop neighbors than during 0–30 s. Since the two-hop density within the two intervals remains almost the same according to Fig. 8(f), we can infer that the nodes in the middle fraction of the swarm during 70–100 s will have more two-hop neighbors than during 0–30 s. Therefore, the nodes in the middle fraction will perform more terrible broadcast service during 70–100 s than during 0–30 s.

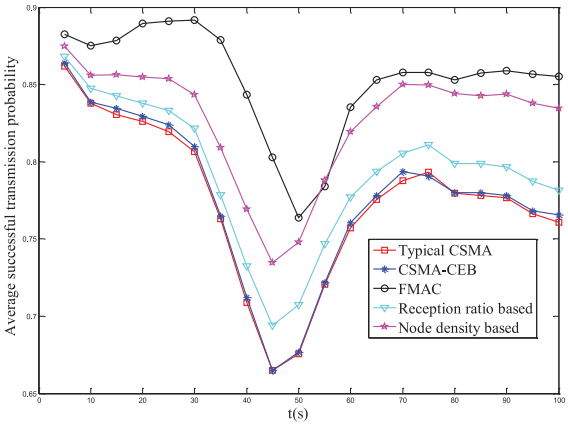

Fig. 10. Successful transmission probability of different protocols.

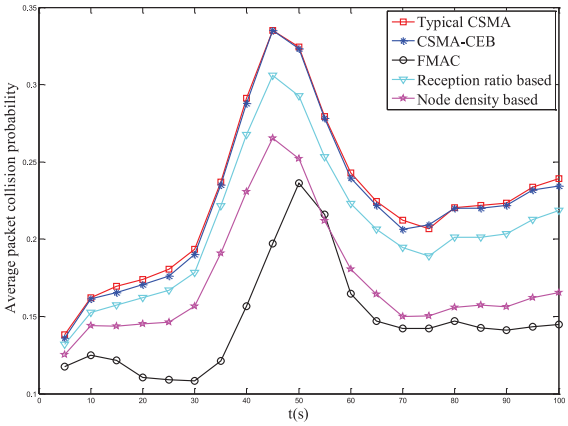

Fig. 11. Packet collision probability of different protocols.

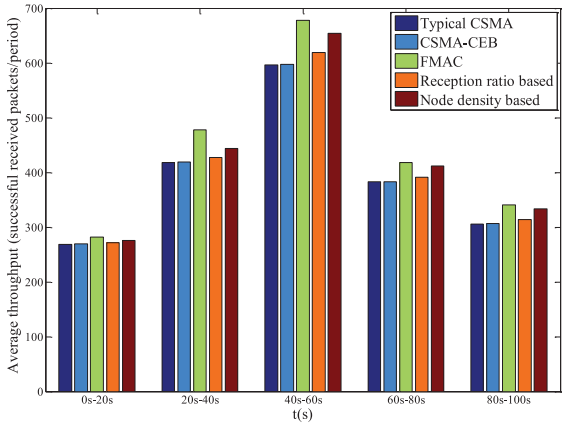

Fig. 12. Average throughput of different protocols.

The proposed FMAC protocol always achieves the highest p_{succ} and lowest p_{col} than the other four protocols. This promotion lies in the fact that the CNP metric can sensitively reflect the varying trend of UAV density in the considered scenario. In other words, the CNP metric makes the CW changes more sensitively as the node density changes than other metrics involved in other protocols.

The throughput TP_{Rx} of the FANET is defined as the average number of successfully received packets within each

period. The throughputs of the five protocols are presented in Fig. 12, where the throughput is measured every 20 s. It can be seen from Fig. 12 that the FMAC protocol achieves the highest throughput than the other four protocols because it provides a reliable broadcast service. Note that all protocols witness higher throughput from 40 to 60 s than that at other intervals, due to the fact that the potential receivers of each UAV increase as the UAV density increases.

VII. Conclusion

In this article, the FMAC protocol has been proposed for the FANET to enable the flocking capability of UAV swarm. To adaptively change the CW size, the triggering state CNP relating to the dynamics of inter-UAV spacing has been introduced in FMAC, which is more sensitive to the variation of UAV density during flocking motion than typical triggering states, that is, expiration, node density, and reception ratio. In addition, the FMAC protocol can maintain a more reliable one-hop broadcast service on the control channel in the considered dynamic density scenario. Compared with the typical adaptive CW mechanism-based CSMA/CA protocols, the FMAC protocol can provide higher successful transmission probability when UAV density bursts. For the future work, the effect of parameters δ and Δ (which decide the length of the interval and the change in CW size, respectively, for each CW update) on the performance of the FMAC protocol will be investigated analytically, which is useful to choose one optimum combination of (δ, Δ). Moreover, more complex flight environments will be considered for the transmission of periodic one-hop packets.

References

[1] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in UAV networks for emergency communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8958–8976, Oct. 2019.

[2] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3727–3742, Apr. 2019.

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

[4] W. Shi *et al.*, "Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks," *IEEE Access*, vol. 6, pp. 12518–12529, 2018.

[5] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang, and X. Shen, "Drone assisted vehicular networks: Architecture, challenges and opportunities," *IEEE Netw.*, vol. 32, no. 3, pp. 130–137, Jan. 2018.

[6] H. Wu, X. Tao, N. Zhang, and X. Shen, "Cooperative UAV cluster-assisted terrestrial cellular networks for ubiquitous coverage," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2045–2058, Sep. 2018.

[7] H. Fang, Y. Wei, J. Chen, and B. Xin, "Flocking of second-order multiagent systems with connectivity preservation based on algebraic connectivity estimation," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1067–1077, Apr. 2017.

[8] S. A. Quintero, G. E. Collins, and J. P. Hespanha, "Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 2025–2031.

[9] Y. Jia, Q. Li, and S. Qiu, "Distributed leader–follower flight control for large-scale clusters of small unmanned aerial vehicles," *IEEE Access*, vol. 6, pp. 32790–32799, 2018.

[10] H. Zhao, H. Wang, W. Wu, and J. Wei, "Deployment algorithms for UAV airborne networks toward on-demand coverage," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2015–2031, Sep. 2018.

[11] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, May 2013.

[12] E. Ferrante, A. E. Turgut, A. Stranieri, C. Pinciroli, M. Birattari, and M. Dorigo, "A self-adaptive communication strategy for flocking in stationary and non-stationary environments," *Nat. Comput.*, vol. 13, no. 2, pp. 225–245, Jun. 2014.

[13] C. Kownacki and D. Ołdziej, "Flocking algorithm for fixed-wing unmanned aerial vehicles," in *Advances in Aerospace Guidance. Navigation and Control*. Cham, Switzerland: Springer, 2015, pp. 415–431.

[14] H. A. Omar, W. Zhuang, and L. Li, "VeMAC: A TDMA-based MAC protocol for reliable broadcast in VANETs," *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1724–1736, Sep. 2013.

[15] F. Lyu *et al.*, "SS-MAC: A novel time slot-sharing MAC for safety messages broadcasting in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3586–3597, Apr. 2018.

[16] F. Lyu *et al.*, "MoMAC: Mobility-aware and collision-avoidance MAC for safety applications in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10590–10602, Nov. 2018.

[17] S. Bharati and W. Zhuang, "CRB: Cooperative relay broadcasting for safety applications in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9542–9553, Dec. 2016.

[18] Q. Ye and W. Zhuang, "Distributed and adaptive medium access control for Internet-of-Things-enabled mobile networks," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 446–460, Apr. 2016.

[19] D. Tian *et al.*, "A distributed position-based protocol for emergency messages broadcasting in vehicular ad hoc networks," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1218–1227, Apr. 2018.

[20] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[21] B.-J. Kwak, N.-O. Song, and L. E. Miller, "Performance analysis of exponential backoff," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 343–355, Apr. 2005.

[22] C. Zhang, P. Chen, J. Ren, X. Wang, and A. V. Vasilakos, "A back-off algorithm based on self-adaptive contention window update factor for IEEE 802.11 DCF," *Wireless Netw.*, vol. 23, no. 3, pp. 749–758, Apr. 2017.

[23] M. W. Lee and G. Hwang, "Adaptive contention window control scheme in wireless ad hoc networks," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 1062–1065, May 2018.

[24] R. Ali, N. Shahin, Y.-T. Kim, B.-S. Kim, and S. Kim, "Channel observation-based scaled backoff mechanism for high-efficiency WLANs," *Electron. Lett.*, vol. 54, no. 10, pp. 663–665, May 2018.

[25] R. Stanica, E. Chaput, and A.-L. Beylot, "Reverse back-off mechanism for safety vehicular ad hoc networks," *Ad Hoc Netw.*, vol. 16, pp. 210–224, May 2014.

[26] H. Zhao, A. Du, H. Zhu, D. Li, and N. Liu, "A self-adaptive back-off optimization scheme based on beacons probability prediction for vehicle ad-hoc networks," *China Commun.*, vol. 13, no. 12, pp. 132–138, Dec. 2016.

[27] C. Chrysostomou, C. Djouvas, and L. Lambrinos, "Contention window adaptation for broadcast beaconing in vehicular ad hoc networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2014, pp. 1039–1044.

[28] R. Stanica, E. Chaput, and A. Beylot, "Local density estimation for contention window adaptation in vehicular networks," in *Proc. IEEE 22nd Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2011, pp. 730–734.

[29] N. Balon and J. Guo, "Increasing broadcast reliability in vehicular ad hoc networks," in *Proc. 3rd Int. Workshop Veh. Ad Hoc Netw.*, 2006, pp. 104–105.

[30] H. Zhou, N. Cheng, J. Wang, J. Chen, Q. Yu, and X. Shen, "Toward dynamic link utilization for efficient vehicular edge content distribution," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8301–8313, Sep. 2019.

[31] S. A. A. Shah, E. Ahmed, F. Xia, A. Karim, M. Shiraz, and R. M. Noor, "Adaptive beaconing approaches for vehicular ad hoc networks: A survey," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1263–1277, Jun. 2018.

[32] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2012, 2012.

[33] M. T. Hale and M. Egerstedt, "Convergence rate estimates for consensus over random graphs," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 1024–1029.

[34] H. Zhao, H. Liu, Y.-W. Leung, and X. Chu, "Self-adaptive collective motion of swarm robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1533–1545, Oct. 2018.

[35] H. J. Qiu, I. W.-H. Ho, K. T. Chi, and Y. Xie, "A methodology for studying 802.11p VANET broadcasting performance with practical vehicle distribution," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4756–4769, Oct. 2015.
[36] K. A. Hafeez, L. Zhao, Z. Liao, and B. N.-W. Ma, "Performance analysis of broadcast messages in VANETs safety applications," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.
[37] M. Khabazian, S. Aïssa, and M. Mehmet-Ali, "Performance modeling of safety messages broadcast in vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 380–387, Mar. 2013.
[38] A. Vinel, V. Vishnevsky, and Y. Koucheryavy, "A simple analytical model for the periodic broadcasting in vehicular ad-hoc networks," in *Proc. IEEE GLOBECOM Workshops*, Dec. 2008, pp. 1–5.
[39] C. Campolo, A. Vinel, A. Molinaro, and Y. Koucheryavy, "Modeling broadcasting in IEEE 802.11p/WAVE vehicular networks," *IEEE Commun. Lett.*, vol. 15, no. 2, pp. 199–201, Feb. 2011.
[40] M. Van Eenennaam, A. Remke, and G. Heijenk, "An analytical model for beaconing in VANETs," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Nov. 2012, pp. 9–16.

Xinquan Huang received the B.S. degree in electrical and information engineering from the College of Information Science and Technology, Northwest University, Xi'an, China, in 2013, and the M.S. degree in communications and information system from the College of Communications Engineering, PLA University of Science and Technology, Nanjing, China, in 2016. He is currently pursuing the Ph.D. degree in communications and information system with Army Engineering University, Nanjing.

His research interests mainly focus on multiagent networks, UAV swarms, and medium access control protocols.

Aijun Liu (Member, IEEE) received the B.S. degree in microwave communications, and the M.S. and Ph.D. degrees in communications engineering and information systems from the College of Communications Engineering, PLA University of Science and Technology, Nanjing, China, in 1990, 1994, and 1997, respectively.

He is currently a Full Professor with the College of Communication Engineering, Army Engineering University. Since March 2015, he has been a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His current research interests mainly focus on satellite communication system theory, signal processing, space heterogeneous networks, channel coding, and information theory.

Haibo Zhou (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014.

From 2014 to 2017, he worked as a Postdoctoral Fellow with the Broadband Communications Research Group, Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China. His research interests include resource management and protocol design in cognitive radio networks and vehicular networks.

Kai Yu (Graduate Student Member, IEEE) received the B.S. degree in detection guidance and control techniques form the University of Electronic Science and Technology of China, Chengdu, China, in 2019. He is currently pursuing the Ph.D degree in communications and information system with Nanjing University, Nanjing, China.

His research interests include resource allocation, machine learning for wireless communications, and heterogeneous cellular networks.

Wei Wang (Student Member, IEEE) received the B.Eng. degree in information countermeasure technology and the M.Eng. degree in signal and information processing from Xidian University, Xi'an, China, in 2011 and 2014, respectively, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2018.

He is currently a Professor with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He was as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from September 2018 to August 2019. His research interests include wireless communications, space–air–ground-integrated networks, wireless security, and physical-layer security.

Dr. Wang received the IEEE Student Travel Grants for IEEE ICC'17 and the Chinese Government Award for Outstanding Self-Financed Students Abroad in 2019.

Xuemin (Sherman) Shen (Fellow, IEEE) received the B.Sc. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982, and the M.Sc. and Ph.D. degrees in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1987 and 1990, respectively.

He is currently a University Professor and an Associate Chair for Graduate Studies with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on resource management, wireless network security, social networks, smart grid, and vehicular *ad hoc* and sensor networks.

Dr. Shen was a recipient of the Excellent Graduate Supervision Award in 2006. He received the Premiers Research Excellence Award from the Province of Ontario, Canada, in 2003. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom16, Infocom14, IEEE VTC10 Fall, and Globecom07, the Symposia Chair for IEEE ICC10, the Tutorial Chair for IEEE VTC11 Spring and IEEE ICC08, the General Co-Chair for ACM Mobihoc15, Chinacom07, and QShine06, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, *Peer-to-Peer Networking and Application*, and *IET Communications*; a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Computer Networks*, and *ACM/Wireless Networks*; and the Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, *IEEE Communications Magazine*, and *ACM Mobile Networks and Applications*. He is a Registered Professional Engineer of ON, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He was an Elected Member of the IEEE ComSoc Board of Governor and the Chair of the Distinguished Lecturers Selection Committee.