A Link-Prediction Based Multi-CDSs Scheduling Mechanism for FANET Topology Maintenance

Xiaohan Qi[1], Xinyi Gu[1], Qinyu Zhang[1], and Zhihua Yang[1,2](✉)

[1] Communication Engineering Research Center, Harbin Institute of Technology,
Shenzhen 518055, Guangdong, China
`yangzhihua@hit.edu.cn`
[2] Pengcheng Laboratory, Shenzhen 518055, Guangdong, China

Abstract. In Flying Ad hoc Network (FANET), maintenance of topology is a quite difficult task due to rapid change of connectivity between flight nodes, i.e. unmanned aerial vehicles (UAVs). Aiming to this issue, in this article, we proposed a link prediction-based multiple Connected Dominating Sets (CDSs) scheduling mechanism for stable maintenance of FANET's topology. In particular, a group of candidate CDSs are periodically scheduled for developing a stable backbone subnet of the topology. The proposed mechanism could achieve an early detection of topological changes by employing a Markov chain predicting model on the node's mobility. The simulation results show that the proposed algorithm has a better success rate and less update overheads than the single CDS maintenance method, especially in a typical swarm pattern of UAV team.

Keywords: FANET · Connected dominated set · Link-prediction · Swarm

1 Introduction

Typically, flying ad hoc network (FANET) is a specific category of self-organizing networked systems with a group of swarming unmanned aerial vehicles (UAVs) [1,2], which is widely exploited for remoting sensing, surveillance and emergency communication scenes. With un-even node density, FANET develops a highly dynamic topology with unreliable links due to fast relative motions of UAV nodes, which has caused adverse effects on the maintenance of the network. Currently, virtual backbone network (VBN) is introduced for providing an effective way to maintain the time-related topological stability, i.e. in mobile sensor networks and general ad hoc networks.

At present, constructing methods of VBN are developed into two main categories, such as hierarchically clustering and connected dominating set (CDS). Comparatively, connected dominating set, with less dominating nodes, could provide a smaller scale of backbone network than the clustering method. Therefore,

M. Jia et al. (Eds.): WiSATS 2019, LNICST 280, pp. 587–601, 2019.
https://doi.org/10.1007/978-3-030-19153-5_59

CDS is considered as a preferable candidate solution for constructing a backbone network for FANET. Considering the constrained transmission power of UAV platform, a smaller connected dominating set, thus a slighter scale of VBN, is quite expected for greatly reducing the forwarding overheads of messages during routine communications [3,4]. More heavily, the topology of FANET will change with time due to highly relative motions of nodes, leading to a lower operating efficiency and possible failure of the backbone network. As a result, it is indispensable for constantly monitoring and maintaining the backbone network after the construction of VBN to resist the dynamic of topology. Currently, there exists a lot of works on the topology maintenance of mobile sensor networks [5,6], most of which focus on guaranteeing the connectivity of the backbone network and extend the network lifetime as long as possible. In [7], a k-hop CDS constructing algorithm is proposed, in which the cost and stability of algorithm are closely depending on the value of k without an optimal solution. Mainly, a part of existing maintenance algorithms are dedicated for improving the fault tolerance of the backbone network by a couple of approaches, i.e., adjusting the relationship of nodes and controlling node's transmission power. This type of topology control algorithms with fault-tolerant capability include active and passive modes. Among them, the active algorithm includes power control [8] and network hierarchical partitioning respectively, which mainly consider the fault tolerance of VBN construction. On the other side, the passive topology control algorithm attempt to find a k-connected graph of the topology. In [9], the problem of solving the least cost k-connected graph is proved to be NP-hard, in which the value of k is constrained by the maximum transmission power of node.

Although there are a variety of works discussing on the control and maintenance of topology in diversified scenarios, most of them are aiming to guarantee the connectivity of the backbone network in a relatively quasi-static condition, without considering the dynamic of topology and motion characteristic of in-situ platform. To address this issue, in this paper, we propose a link-prediction based multi-CDSs scheduling mechanism for effective maintenance of FANET topology. The contributions of this paper are as follows: (a) we developed a novel scheduling mechanism of multiple CDSs for constructing a stable VBN. The proposed mechanism could achieve a constantly virtual backbone for the time-changing topology by periodically recruiting a group of candidate CDSs; (b) we proposed a Markov chain-based link-prediction model of connectivity state between nodes by considering the Markovian property of UAV mobility, which could achieve an early detection of topological changes of FANET, especially with a swarming motion pattern. As a result, the proposed maintenance algorithm could obviously reduce the communication overheads and achieve a high success rate with a small time delay.

The remainder of the paper is organized as follows. Section 2 defines the topology model and formulates the problem, the proposed algorithm is detailed in Sect. 3. Section 4 presents the simulation results and Sect. 5 gives the conclusion.

2 Problem Formulation

2.1 Topology Model

For the convenience of analysis, we assume that each UAV node in FANET has the same communication range. Typically, we define an undirected graph $G\left(V^t, E^t\right)$ to represent the network topology at the t-th time slot. In the graph, V^t represents the set of nodes at time t, denoted by $V^t = \{v_1, v_2, v_3, \ldots, v_n\}$, and E^t is the edges with bi-directional connection at time t, denoted by $E^t = \{e_1, e_2, e_3 \ldots e_n\}$, respectively. For less communication overheads, we will adopt a minimum spanning tree based algorithm to obtain a minimum connected dominating set in order to obtain a small scale of VBN. Firstly, we give a group of preliminary definitions of CDS.

Definition 1. Dominating Set (DS). Given graph $G =$ (V, E), V is a DS of $G =$ (V, E), only if $\forall \left(u, v\right) \in E$, either $u \in V'$ or $v \in V'$ is true.

Definition 2. Connected Dominating Set (CDS). $C \subseteq V$ is a CDS of G if (1) C is a DS and (2) a graph induced by C is connected.

Definition 3. Degree of node. For node u in graph $G =$ (V, E), the number of neighbors of u is defined as the degree of u, denoted by $D\left(u\right)$.

The VBN constructing algorithm is based on the minimum spanning tree and local routing information. The backbone network can achieve a smaller scale and reduce routing overheads through the algorithm. The selection criterion of the nodes in CDS is the weight of nodes. When conflict occurs in weight of nodes, the edge weight is chosen as the new criteria. Therefore, we set the weight definitions of nodes and edges below.

Definition 4. Weigh of node. For node u in graph $G =$ (V, E), the node weight of u is defined as the weighting summation of $D\left(u\right)$ and energy of node $E\left(u\right)$, denoted by $W\left(u\right) = \alpha D\left(u\right) + \beta E\left(u\right)$, and $\alpha + \beta = 1$.

Definition 5. Weigh of edge. For edge $\langle u, v\rangle$ in graph $G =$ (V, E), the summaion of $D\left(u\right)$ and $D\left(v\right)$is defined as the weight of edge, denoted by $W\left(\langle u, v\rangle\right)$.

2.2 Problem Formulation

Without loss of generality, we assume that the initial topology graph is connected, which means each node has a reachable path to any other node in the network. The topology will change with intermittent links due to temporal motions of UAV nodes, i.e., join, sojourn and leave the network, leading to a fragile connectivity of backbone network. Typically, three main types of topological changes are frequently encountered during the swarm flight of a FANET, as shown in Fig. 1.

(1) The node of the backbone network is failed.
(2) The link of the backbone is disconnected.
(3) The dominatee exits the network.

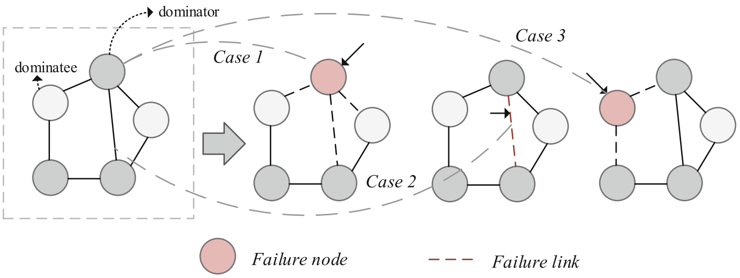


Fig. 1. Topological changes.

As a result, it is scarcely possible for maintaining a stable virtual backbone by only single CDS during a long time. To address this issue, we introduce a novel multiple CDSs based maintenance by reasonably scheduling multiple CDSs in a duration of network operation as long as possible. Firstly, as discussed in the previous section, a smaller connected dominating set is expected since it could easily maintain the connectivity with less communication overheads. In this paper, therefore, we will construct a minimum dominating set with the minimum spanning tree algorithm. The notation employed are illustrated in Table 1.

Moreover, triggering the proposed maintenance algorithm in time is quite indispensable for ensuring the stability of the VBN. The proposed mechanism includes two subsequent phases: the construction of connection dominating sets and the maintenance of VBN. If current VBN is failure, resulting in disconnections of relevant links, the duty backbone cannot guarantee the normal communication connectivity between nodes. Therefore, it is essential to invoke another candidate CDS on duty in time with a feasible schedule, in order to ensure continual effectiveness of the backbone network.

In particular, we define two well-tailored metrics to evaluate the proposed algorithm, i.e., update overhead and successful update rate, respectively.

Update Overhead. The update overheads of CDS is defined as the total overheads of broadcasting messages during the detection and updating phases in a complete working time.

$$T_{cost} = \sum_{i} P_{t_i}(C_{t_i} \rightarrow NC_{t_i}) \tag{1}$$

where $C_{t_i} \rightarrow NC_{t_i}$ represents the change of nodes and edges in current CDS on duty at time t_i, in which C_{t_i} represents the status of current CDS on duty, and NC_{t_i} represents the status of CDS to-be-on-duty if the backbone changes, and $P_{t_i}(C_{t_i} \rightarrow NC_{t_i})$ is the overheads incurred during the updating phase at i-th time slot, respectively.

Table 1. Notation description

Notation	Representation of the symbol or symbol
S_{cds}	The set of CDSs in the graph
$ESm^{t_c}(S_{cds})$	CDS with maximum energy at time t_c
$Em^{t_c}(cds_i)$	energy of CDS_i
V_{bn}	Nodes in CDS_{ct}
E_{bn}	Links of nodes in CDS_{ct}
$N_1(v)$	Node lists in $v's$ one-hop range
$N_2(v)$	Node lists in $v's$ two-hop range
$v.parent$	Dominator of v
$Nlist(v)$	Neighbor list of v
$Domlist(v)$	Dominator nodes in $N_1(v)$
CDS_{ct}	The CDS of graph at current time
CDS_{bc}	The backup CDS
T_{cost}	Update overheads of the network
P_{srt}	Total successful update rate
p_{Θ}	Connected probability threshold of the link
$p(u \to v)$	Predicted connection probability of u and v
K_{TS}	Maximum similarity among CDSs
$MAX\left\{W\left\{u\right\}\right\}$	The node with maximum weight

Successful Update Rate. It is exactly the percentage of the rounds of successful updates to the total rounds of updates.

$$P_{srt} = \frac{\sum_i S_{t_i}(C_{t_i} \to NC_{t_i})}{\sum_i R_{t_i}(C_{t_i} \to NC_{t_i})} \tag{2}$$

in which $S_{t_i}(C_{t_i} \to NC_{t_i})$ is the amounts of successfully updated nodes at time t_i, and $R_{t_i}(C_{t_i} \to NC_{t_i})$ is the amounts of failed nodes at time t_i.

In this paper, a Multi-CDSs scheduling algorithm is proposed by employing the minimum spanning tree based on weight of node and maximum similarity criterion, which could maintain the connectivity of topology with lower overheads. In particular, a link prediction algorithm is designed by employing the Markovian property of the mobility model, which could effectively predict node connectivity through calculating one-step transition probability of nodes. The block diagram of the proposed maintenance algorithm is shown in Fig. 2.

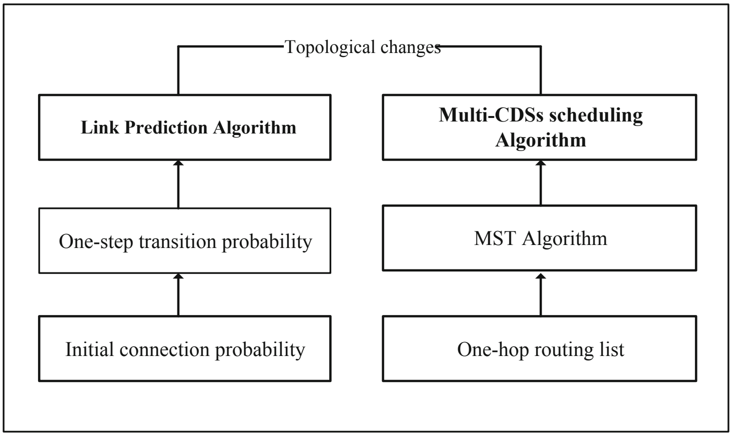


Fig. 2. The block diagram of the proposed mechanism.

3 Algorithms Description

3.1 Link Prediction

In this section, we will propose a Markov chain based link prediction algorithm of FANET topology, which could predict the connectivity state between a pair of nodes. For the convenience of analysis, we employ a SYN-boid model [10] to describe the swarming motions of nodes in FANET. In the model, the velocity of an individual node at the next moment is mainly determined by the previous step, which presents obvious Markovian property.

In specific, in the model, the speed variation is denoted by $v = V_{\max} e^{\beta[syn-1]}$, in which V_{max} is the maximum velocity of UAV nodes, syn is the synchronization coefficient and β is a constant regulating the velocity distribution, respectively. With a given probability density function of syn, expressed as $f_{syn}(syn)$. We can obtain the probability density distribution function of the speed $f_v(v)$ as follows:

$$\begin{aligned} f_v(v) &= f_{syn}(syn)\frac{\partial syn}{\partial v} \\ &= \frac{1}{\beta v} f_{syn}\left(\frac{\ln \frac{v}{v_{\max}}}{\beta} + 1\right) \end{aligned} \tag{3}$$

in which

$$syn = \frac{\ln \frac{v}{v_{\max}}}{\beta} + 1 \tag{4}$$

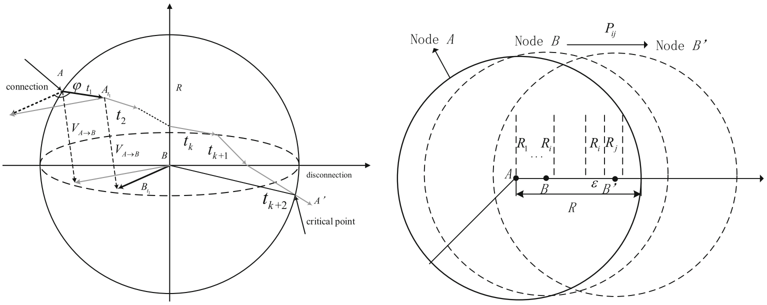
As shown in Fig. 3(a), for node A and node B, the speeds at the current time are denoted as $\overrightarrow{V_A}$ and $\overrightarrow{V_B}$ respectively, and the relative speed is calculated by

$V_{A\to B} = \overrightarrow{V_A} - \overrightarrow{V_B}$ accordingly. Define Φ as the angle between $\overrightarrow{V_A}$ and $\overrightarrow{V_B}$, then we have

$$|\overrightarrow{V_{A\to B}}| = \sqrt{{V_A}^2 + {V_B}^2 - 2V_A V_B \cos\varphi} \tag{5}$$

and

$$\varphi = \arccos \frac{|\overrightarrow{V_A}|^2 + |\overrightarrow{V_B}|^2 - |\overrightarrow{V_{A\to B}}|^2}{2|\overrightarrow{V_A}||\overrightarrow{V_B}|} \tag{6}$$



(a) The relative positions of two nodes.

(b) One-step state transition probability.

Fig. 3. Relative movements of two nodes.

In order to conveniently describe changes in connection states of two adjacent nodes, we discrete the relative motion of nodes into a series of individual states. Considering the Markov property of UAV's motion, we proposed a link prediction algorithm for forecasting the VBN changes in advance. Figure 3(b) presents relative positions distribution of node A and B during their motions. Without loss of generality, we fix node A exactly at the center of transmission area with a radius of R. For the convenience of analysis, we dividing the allowed communication range of node i into n segments with equal length ε. In particular, each small segment represents one state of the relative distance l between two nodes, where $l \in [k\varepsilon, (k+1)\varepsilon]$ indicates that the current status is R_k. Therefore, we propose a Markov chain with N states for modeling the variety of relative positions of two nodes. Given $\mathbf{P}$ as the one-step transition probability matrix, the one-step distance transition probability is

$$\mathbf{P} = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ & \cdots & \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} = [\mathbf{P_1} \ldots \mathbf{P_k} \ldots \mathbf{P_n}] \tag{7}$$

in which $P_{ij} = \Pr\{l_m \in R_i | l_m \in R_j\}$ is the probability that the relative distance l shifts from the i-th segment to the j-th segment. In particular, the maximum change of transition states should be $N = 2v_{\max}/\varepsilon$. It can be deserved that P_{ij} will be equal to 0 if $|i-j| > N$. Define the probability distribution of the related nodes at the initial time as ρ_{t_0}. In general, the connection probability of the initial state is known, expressed as

$$\rho_{t_0} = [\rho_1 \,...\, \rho_1 \,...\, \rho_n] \tag{8}$$

in which $\rho_{t_0}(k)$ represents the probability that the initial connected state of the node is at R_k. Furthermore, we call the probability that the node locates at certain small segment after m time slots as the m-step transition probability, denoted by ρ_{t_m}. If the initial connected state is at R_k, accordingly, the marginal probability of relative distance after m time slots is derived as

$$\begin{aligned} \rho_{t_m}(k) &= \rho_{t_0}(k)\mathbf{P}^m \\ &= [\rho_{t_0}(1)\mathbf{P}^m_{\mathbf{1}} ... \,\rho_{t_0}(k)\mathbf{P}^m_{\mathbf{k}} \,...\, \rho_{t_0}(n)\mathbf{P}^m_{\mathbf{n}}] \\ &= [0 ... \,\rho_{t_0}(k)\mathbf{P}^m_{\mathbf{k}} \,...\, 0] \end{aligned} \tag{9}$$

in which the one-step state transition probability density function is expressed as

$$P_{ij} = \Pr\{l_m \in R_i | l_m \in R_j\} = \int\limits_{(j-1)\varepsilon}^{j\varepsilon} \int\limits_{(i-1)\varepsilon}^{i\varepsilon} f_{l_m|l_{m-1},v_m}(l_m|l_{m-1}) dl_m dl_{m-1} \tag{10}$$

and

$$\begin{aligned} &f_{l_m|l_{m-1}}(l_m|l_{m-1}) \\ &= \int_0^{2v_{\max}} f_{l_m|l_{m-1},v_m}(l_m|l_{m-1}, v_m) f_v(v_m) dv_m \\ &= \int_0^{2v_{\max}} \frac{2}{\pi} . \frac{l_m f_v(v_m) dv_m}{\sqrt{4l_{m-1}^2 v_m^2 - (l_{m-1}^2 + v_m^2 - l_m^2)^2}} \\ &= \int_0^{2v_{\max}} \frac{2}{\pi} .T(v_m) f_v(v_m) \end{aligned} \tag{11}$$

In order to ensure that the value of probability distribution is greater than 0, we can further derive the efficient solution, where

$$\begin{aligned} &f_{l_m|l_{m-1}}(l_m|l_{m-1}) = \int_0^{2v_{\max}} f_{l_m|l_{m-1},v_m}(l_m|l_{m-1}, v_m) f_v(v_m) dv_m \\ &= \int_{\max\{0,|l_m - l_{m-1}|\}}^{\min\{2v_{\max},|l_m + l_{m-1}|\}} \frac{2}{\pi} . \frac{l_u f_v(v_u) dv_u}{\sqrt{4l_{m-1}^2 v_m^2 - (l_{m-1}^2 + v_m^2 - l_m^2)^2}} \\ &= \int_{\max\{0,|l_m - l_{m-1}|\}}^{\min\{2v_{\max},|l_m + l_{m-1}|\}} \frac{2}{\pi} .T(v_m) f_v(v_m) \end{aligned} \tag{12}$$

According to the speed probability density distribution function in Eq. (3), the one-step state transition probability density function can be further obtained.

$$\begin{aligned} P_{ij} &= \Pr\{l_m \in R_j | l_{m-1} \in R_i\} \\ &= \int_{(i-1)\varepsilon}^{i\varepsilon} \int_{(j-1)\varepsilon}^{j\varepsilon} \int_{\max\{0,\left|l_m - l_{m-1}\right|\}}^{\min\{2v_{\max},\left|l_m + l_{m-1}\right|\}} \frac{2}{\pi} \cdot \frac{l_u f_v(v_u) dv_u}{\sqrt{4l_{m-1}^2 v_m^2 - (l_{m-1}^2 + v_m^2 - l_m^2)^2}} dl_m dl_{m-1} \end{aligned} \tag{13}$$

3.2 Multi-CDS Scheduling

Typically, construction of CDS is a NP-hard problem, in which only approximately effective solutions can be found. In this paper, we design an improved construction algorithm of CDS based on the minimum panning tree (MST) method, by incorporating a series of time-related information. As a result, we can find a most effective CDS at each time slot for a better scheduling. The CDS constructing algorithm is shown in Algorithm 1.

Initially, the CDS with the maximum energy among the constructed CDSs is selected as the current on-duty CDS. With the results of selection, the current CDS on duty is substituted by the to-be-on-duty CDS through replacing the non-common nodes of these two sets, as shown in Fig. 4.

Algorithm 1. MST-CDS

```
Require: The set of MCDS at t-th time slot, cds
Ensure: $G^t(V^t, E^t), t$
Initial Calculate the node weight matrix $\mathbf{W_1}$ and edge weight matrix $\mathbf{W_2}$
$\{V_{bn}\} \leftarrow \phi$ //$V_{bn}$ is the set of MCDS at current time slot
for $num\{$ V$_{bn}\} < N$ //$V_{bn}$ is the number of MCDS do
    $T \leftarrow \{V_{bn}^k,\ v_k,\ W\}$ //$v_k$ is the starting node
    $v_m \leftarrow v_k$, $V_{bn} \leftarrow v_m$
    for $m \leftarrow num\{N_{node}\} < N$ //$N_{node}$: node traversal number do
        SET $U = MAX\{W[u]\}$ //$\{u \in N_1(v_m)\}$
        if $num\{U\}$ euual to 1 //only one node with maximum weight then
            $V_{bn} \leftarrow U$
        else
            $V_{bn} \leftarrow MAX\{W(\langle u, v\rangle)\}$
        end if
    end for
end for
if $V_{bn}^i \not\subset V_{bn}^j$ then
    DELETE $V_{bn}^i$
end if
```

Here, we define an energy value of each CDS, denoted as $E_{(cds_i)}$, which is exactly a specific value of lowest residual energy of certain node in the

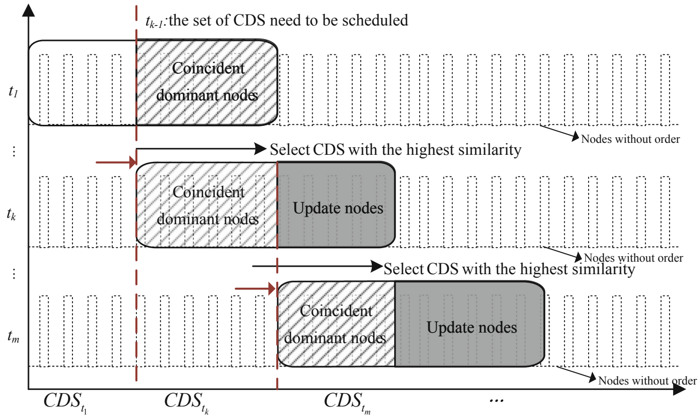


Fig. 4. The scheduling procedure of CDSs.

CDS. In particular, all the backup CDSs will be filtered for being the on-duty CDS according to the energy rule. If there exists imminent disconnections predicted by the proposed Markov model, the scheduling algorithm will find another CDS in the candidate set most similar with the current on-duty CDS. In specific, we define a metric of similarity between two CDSs, denoted by $\frac{CDS_m \cap CDS_{ct}}{CDS_m}$. Furthermore, we define K_{TS} as the maximum similarity between the backup CDS set and the current CDS on duty, expressed as $K_{TS} = Max\{\frac{CDS_1 \cap CDS_{ct}}{CDS_1}, \frac{CDS_2 \cap CDS_{ct}}{CDS_2}, \ldots \frac{CDS_m \cap CDS_{ct}}{CDS_m} \ldots\}$.

The pseudo-code multi-CDS scheduling algorithm is detailed in Algorithm 2. In particular, $E_{\max}\{S_{cds}\}$ represents the certain CDS with the highest energy. Typically, we set two specific rules for triggering the scheduling procedure, respectively as:

(a) a certain type of network changes described in Sect. 2 occurred.
(b) the connection probability of certain edge in $G\left(V^t, E^t\right)$ is lower than the pre-determined threshold.

3.3 Communication Scheme

The entire process of constructing and scheduling CDSs requires a sequence of information exchanges, i.e., results of link predictions, identity numbers of CDSs, and routing lists, etc. Once the FANET starts, each node will periodically broadcast its own relevant information and updates the neighbor and dominator lists. In particular, these routing information will be used for constructing CDS and updating the status of local node. As long as the topology changes, the information about neighbor nodes is required to be updated by the local node. The complete procedure of signaling interaction is shown in Fig. 5.

For the dominate node, we propose a specific data structure, including a group of information such as neighbor list, one-hop dominator list and parent node identity, as shown in Fig. 6.

Algorithm 2. Multi-CDS Scheduling

```
Require: $S_{cds}$,$G^t\left(V^t, E^t\right)$, $t$, $t_d$ //detection period
Ensure: Scheduling of $S_{cds}$ and nodes
 1: Broadcast nodes' information to vote $CDS_{ct}$
 2: $CDS_{ct} \leftarrow E_{\max}\{S_{cds}\}$, $\{\ \mathrm{S}_{cds}^{bc}\}\ \leftarrow S_{cds} \backslash CDS_{ct}$
 3: while $num\left\{V^t\right\} > 1$ // no nodes in the network do
 4:    DETECT changes that need to be maintained
 5:    DETECT failed CDS in $S_{cds}$,UPDATE $S_{cds}$
 6:    CASE1: changes in $V\{CDS_{ct}\}$ or $E\{CDS_{ct}\}$ break
 7:    GOTO 14
 8:    CASE2: $p(u \rightarrow v) < p_\Theta$ // $< u, v > \in E\{CDS_{ct}\}$
 9:    GOTO 14
10:    CASE3: domintees leave the network
11:    GOTO 18
12:    CASE4: domintee $v_i$ loses connection with $v_i.parent$
13:    GOTO 18
14:    if $MIN\left\{CDS_{ct} \cap \mathrm{S}_{cds}^{bc}\right\} > 1$ then
15:       $CDS_{bc} \leftarrow E_{\max}\{MIN\left\{CDS_{ct} \cap \mathrm{S}_{cds}^{bc}\right\}\}$
16:       UPDATE nodes belong to $CDS_{ct} \cap CDS_{bc}$ Break
17:    end if
18:    Wait until there are dominators in $N_2 v_i)$
19:    if there are dominators in $N_1(v_i)$ then
20:       Add $u \leftarrow E_{\max}\{N_1(v_i)\}$ into $CDS_{ct}$ UPDATE
21:    end if
22:    return $S_{cds}$ $CDS_{ct}$ //Broadcast and update
23: end while
```

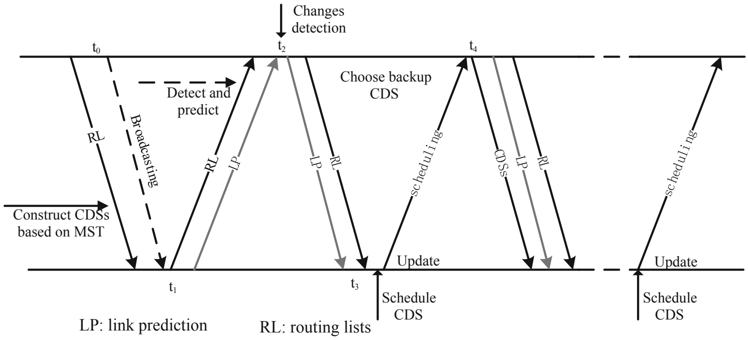


Fig. 5. The signaling sequence.

4 Numerical Analysis

In the simulations, we exploit SYN-boid mobility model for generating a group of topological snapshots in multiple time slots with various number of nodes. The relevant parameters are shown in Table 2, for comparison with the single CDS algorithm (SCDS) in different scenarios.

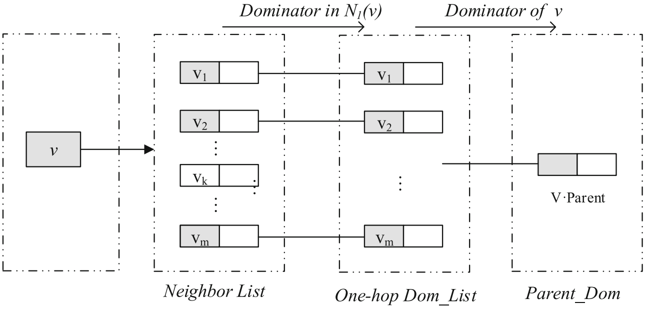


Fig. 6. Data structure.

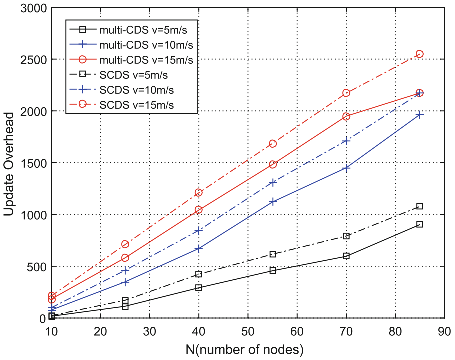
Table 2. Simulations parameters.

Parameter	Parameter description	Value
N	Number of nodes	10/25/...70/85
T	Total time slots	400
v	Maximum velocity	5/10/15/20 m/s
R_c	Communication radius	90/100/110 m
L_i	Initial position	400 m × 400 m
R_r	Repulsion perception radius	90 m

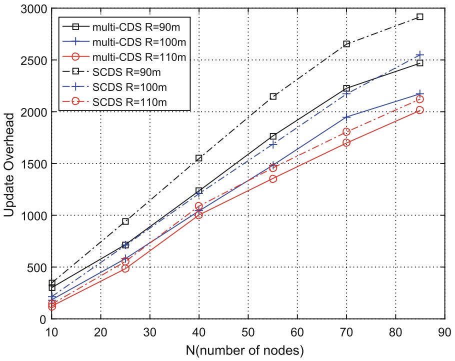
Typically, a single-CDS algorithm implements the maintenance process in an event-triggered manner. In specific, a backup node is firstly selected for a backbone node according to the rules described above. Once this backbone node fails, its status is updated. However, this method has a non-trivial probability of failure due to the disconnected nature of the sparse topology. In particular, we use the update overheads and the amounts of successful update as two respective metrics for the comparative experiments.

Figure 7 shows the performance comparison of two algorithms under different scenarios with respect to the update overheads during the lifetime of the network. In particular, the used topology is generated by changing the maximum operating speed of node under a certain communication radius. As can be seen in Fig. 7(a), as the speed of node movement increases, the instability duration of topology increases and corresponding update overheads increase, since more nodes are required to be scheduled in order to maintain the normal operation of CDS. Besides, the overheads of proposed algorithm are smaller than the single-CDS (SCDS) algorithm, which indicates that the multi-CDS scheduling algorithm based on link prediction can effectively reduce the update overheads.

Generally, changes of node's communication radius will also affect the topology stability. As the results, we can get the comparison of two algorithms under different communication radium, as shown in the Fig. 7(b). It is seen that the update overheads of the proposed algorithm are significantly less than the SCDS



(a) Under different velocity.



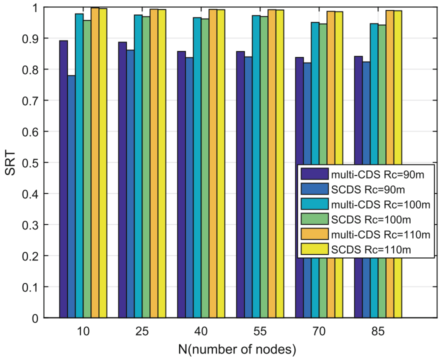
(b) Under different communication radius.

Fig. 7. Update overheads.

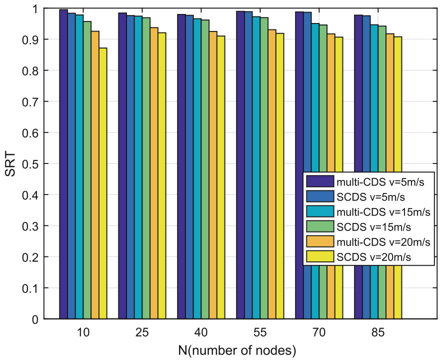
algorithm. Combined with the previous simulations, we can conclude that the proposed algorithm has smaller update overheads and smaller impacts on maintaining a highly dynamic network backbone network.

Typically, the success rate (SRT) is directly related to the network connectivity and the success selection of backup node. From Fig. 8(a), it is seen that the success update rates of the proposed algorithm is slightly higher than SCDS, as is apparent under low communication radius. In another word, the proposed algorithm has obvious advantages with heavily topological changes. As the communication radius increases, both algorithms have higher probabilities of successful maintenance.

By changing the maximum velocity of node, we observe the success probability of two algorithms to maintain the backbone network. As can be seen from Fig. 8(b), both algorithms have approximate probabilities of success update with small node's speeds. Once the speed develops to be large, the success rate of the



(a) SRTs under communication radius.



(b) SRTs under different velocity.

Fig. 8. Successful update rates.

proposed algorithm is higher than that of SCDS. Combined with the previous results, we can conclude that the multi-CDS algorithm has advantages in highly dynamic networks.

5 Conclusion

In this paper, we propose a link prediction based multi-CDS scheduling algorithm to maintain the connectivity of the virtual backbone network. The link prediction algorithm could reduce the impact of the maintenance phase on the connectivity. Moreover, the proposed algorithm can maintain the stability of backbone network effectively. Through simulations results, we find that under different motion scenarios, the performance of proposed algorithm is better than SCDS algorithm. We mainly compare two aspects of update overheads and maintenance success rate. The algorithm proposed in this paper has better performance in both aspects.

Acknowledgments. The authors would like to express their high appreciations to the supports from the National Natural Science Foundation of China (61871426) and Basic Research Project of Shenzhen (JCYJ20170413110004682).

References

1. Büchter, K.D.: Availability of airborne ad-hoc communication network in global air traffic simulation. In: International Symposium on Communication Systems, Networks and Digital Signal Processing (2016)
2. Bujari, A., Calafate, C.T., Cano, J.C., Manzoni, P., Palazzi, C.E., Ronzani, D.: Flying ad-hoc network application scenarios and mobility models. Int. J. Distrib. Sens. Netw. **13**(10), 155014771773819 (2017)
3. Kuo, T.W.: On the approximability and hardness of the minimum connected dominating set with routing cost constraint (2017)
4. Ugurlu, O., Tanir, D., Nuri, E.: A better heuristic for the minimum connected dominating set in ad hoc networks. In: IEEE International Conference on Application of Information and Communication Technologies, pp. 1–4 (2017)
5. Dash, D.: Restoring virtual backbone of wireless sensor network on sensor failure. In: International Conference on Recent Advances in Information Technology, pp. 29–34 (2016)
6. Zhu, Q., Zhou, R., Zhang, J.: Connectivity maintenance based on multiple relay UAVs selection scheme in cooperative surveillance. Appl. Sci. **7**(1), 8 (2016)
7. Wang, J., Kodama, E., Takata, T.: Construction and maintenance of K-Hop CDS in mobile ad hoc networks. In: IEEE International Conference on Advanced Information Networking and Applications, pp. 220–227 (2017)
8. Wu, C.H., Chen, H.S., Dai, W.H.: Obstacle-avoiding connectivity restoration based on power adjustment and node's movement in disjoint mobile sensor networks. In:International Conference on Information System and Artificial Intelligence, pp. 115–119 (2017)
9. Chen-Ming, M.A., Wang, W.L., Hong, Z.: Distributed construction for (k, m)-fault tolerant connected dominating set in wireless sensor network. Comput. Sci. (2016)
10. Choi, T.J., Chang, W.A.: Artificial life based on boids model and evolutionary chaotic neural networks for creating artworks. Swarm Evol. Comput. (2017)