A Multi-MCDS scheduling based virtual backbone network construction and maintenance algorithm in FANET

Xiaohan Qi, and *Zhihua* Yang*

Communications Engineering Research Centre, Harbin Institute of Technology Shenzhen Graduate School, China

Abstract. Constructing a virtual backbone network can efficiently concentrate most routing and forwarding operations of whole network into several dominating nodes in the virtual backbone sub-net, which is recently developing into an attracted candidate solution for the architectures of terrestrial mobile ad hoc networks. In a flying ad hoc network (FANET), however, high dynamics of nodes and scenarios produce considerably large challenges on the topology control and maintenance due to rapidly time-varying property of topology. In this paper, therefore, we proposed a link prediction-based minimal connected dominating set (MCDS) dependent topology control mechanism to achieve efficient maintenances of connectivity in the network. In particular, the proposed algorithm could provide a stable and effective virtual backbone sub-net in a fast changing topology of FANET, by flexibly scheduling multiple MCDS with a very efficient method. The simulation results shows that, compared with typical single CDS method, the proposed algorithm presents excellent performances in obviously dynamic environments with respect to fewer update times and higher rate of successful updates.

1 Introduction

Typically, a flying ad hoc network (FANET)[1-2] is a self-organizing dynamic network system, which is composed of multiple unmanned aerial vehicle (UAV) nodes in the absence of infrastructure or central control entity. In the network, each node plays the role of a host and a router at the same time. In many complex flight scenarios, such as obstacles avoidance and low-altitude surveillance, which result in a frequently intermittent connectivity of direct communication due to quite a lot of types of clutters, also determine the multi-hop scheme as a main approach of inter-node communication.

With the increasing nodes in FANET, however, multi-hop communications between nodes will cause a lot of communication overheads by allowing considerable relaying operations. In these years, a generally exploited method by constructing a virtual backbone network is proposed for mobile ad hoc network (MANET), which can forward data with reduced energy consumption and prolonged network lifetime in a stably global connectivity in a slightly dynamic environment. In particular, currently, a minimum connected

* Corresponding author: yangzhihua@hit.edu.cn

dominating set (MCDS) [3] is wildly used for effectively constructing a virtual backbone network (VBN) in MANET. In a FANET, however, relative motion with high speed and fast changing scenarios between nodes, far beyond complicated more than general MANETs, produce considerably large challenges on the construction and maintenance of a VBN due to rapidly time-varying property of topology. For example, a fixed connected dominating set could not ensure permanent connectivity of VBN since partial nodes in the CDS may be failed due to exhausted energy or disrupted link. Therefore, constructing an effective CDS should have the following requirements. Firstly, the number of nodes in the CDS is as small as possible. Besides, once a CDS is required to be updated or rebuilt, nodes as few as possible should be replaced for prolonging network lifetime.

In this paper, we propose a connected dominating partition (CDP)[4-5]-based VBN construction algorithm, which could effectively maintain connectivity of VBN even with a time-varying topology, such as a swarming mode of FANET. In particular, the proposed algorithm designed a periodically planning mechanism by scheduling multiple CDSs calculated in prior by the algorithm according to the current status of topology. Compared with a single CDS method, the proposed algorithm could maintain much stable connectivity of VBN in dynamic scenarios with obviously reduced overheads in term of the amounts of replaced nodes.

2 Problem formulation

2.1 Mobility model

In this paper, we adopt a boid [6]-based mobility model(SYN-boid) to describe the motional property of one FANET node, which presents a typical behavior of a UAV swarming team. In a boid model, each node is moving in a random state at the initial stage. During the flight, each node will make motion by exchanging a certain quantity of information with each other with the following typical rules:

Cohesion rules. Node moves towards the centre of its neighbour nodes.

Separation rules. Node moves by avoiding collisions between nearby neighbours.

Alignment rules. Node matches the velocity with its neighbours.

Due to the limited field of view for individual node, it can only move towards the calculated direction through the three rules. In order to avoid flight confusions with extra adjustment costs, nodes can compare the directions of neighbors with themselves when calculating the flight direction. Therefore, we introduce a parameter syn_i into the velocity calculation to describe the synchronization degree of the *i* th individual neighbor node.

$$V(t+1)=V_{max}e^{\beta[syn_i(t)-1]} \tag{1}$$

where $syn_i=\frac{|\sum_{j=1}^{m}direc1_{ij}|}{\sum_{j=1}^{m}|direc_{ij}|}$. It represents synchronization degree of individual node, within a range between 0 to 1. In particular, a large value of syn_i means a better synchronization. When all the neighbor nodes in a certain node's communication radius move in the same motion, the value of syn_i of this node equals to 1.

2.2 Network Model

Define a FANET of swarming UAV system as an un-directional graph, which is represented by $G(V^t,E^t)$. In the graph, V^t represents the set of flight nodes at time *t*, denoted by $V^t=\{v_1,v_2,v_3,\ldots,v_n\}$. On the other side, E^t represents the connection between two nodes, denoted by $E^t=\{e_1,e_2,e_3\ \ldots\ e_n\}$. Typically, a connected dominating set is a subgraph of V^t. In this paper, we will construct multiple CDSs for a virtual backbone network, which needs to be updated with the changing topology. For example, we describe the update of CDS at adjacent time slots in Figure 1.

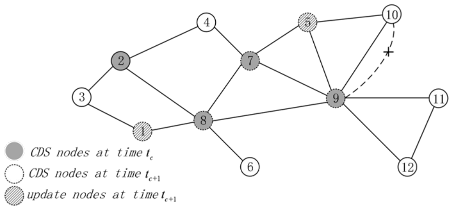

Fig. 1. Updates of CDS in a time-varying topology.

According to the definitions of CDS, we have three following observations:

a) There are a number of minimum dominating sets in a connected graph.

b) If a failed node $v_k \notin V_{cds}^t$ and all the links $\{e = v_i^t v_j^t \mid v_i^t, v_j^t \in V_{cds}^t\}$ are effective, the current CDS is still effective.

c) Both CDS and MCDS problems are NP-Complete problems [7].

Since two problems above cannot be solved in polynomial time, therefore, we employ an approximated solution to find the minimum connected dominated sets. The symbols deployed in the algorithm are shown in Table 1.

Table 1. Symbols descriptions.

Symbol	Description
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_{cds_k}	Nodes in CDS_k
E_{cds_k}	Links of nodes in CDS_k
$N_1(v)$	Node lists in $v's$ one-hop range
$N_2(v)$	Node lists in $v's$ two-hop range
$Domlist(v)$	Dominator nodes in $N_1(v)$, $N_1(v) \wedge CDS_k$
CDS_{ct}	The CDS of graph at current time
$N\_node(CDS_i, CDS_i)$	Number of nodes in $CDS_i \bigcap CDS_i$

2.3 Problem Definition

Given a directed graph of $G^t(V^t, E^t)$, we attempt to build a backbone network with less nodes and maintain it with less update times. In the construction phase, we adapt a *MST_CDS* [8] algorithm to get the backbone, which has been proved to have the shortest path. In the maintenance phase, we should detect the changes of the graph at every time interval of τ . When a topology change is detected, the current CDS needs to be updated. We hope that the update will be completed in a short time. Here, we define a total update time cost $T_{\_\cos t}$ as the total numbers of updates over a successive of snapshots:

$$T_{\_\cos t} = \sum_{t=1}^{i} \sum_{j=1}^{n_{BN}} \sum_{k=1}^{N-n_{BN}} C_{\_node}(v_{k,j}) \text{ , } 1 \le i \le T, 1 \le j,k \le N \tag{2}$$

$$C_{\_node}(v_i) = \begin{cases} 0 & v_i \text{ will be updated} \\ 1 & v_i \text{ won't be updated} \end{cases} \tag{3}$$

where *t* is the number of time slots sampled with interval time τ , and n_{BN} signifies the number of backbone nodes. Here, $C_{\_node}(v_{k,j})$ is a flag on whether node $v_{k,j}$ needs to be updated. It is noted that, $T_{\_\cos t}$ is one important metric of the algorithm performance, namely, the total update cost.

3 Algorithm description

3.1 Multi-MCDS construction

In a FANET network, in order to maintain network connectivity, a general approach is to find those smaller connected dominating sets with corresponding maintenance algorithms.

In this paper, we use a distributed minimum spanning tree algorithm to traverse the connected dominating set, as shown in Algorithm 1. For example, we chose 40 random nodes to generate a MCDS, shown in Figure 2.

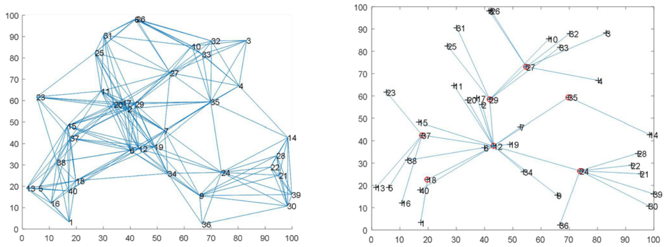

Fig. 2. Construction of MCDS **(a)** 40 random nodes' connected graph; **(b)** corresponding backbone subset.

Considering effective degree and energy of nodes, we define a weight of nodes. For a node v_i, its corresponding weight is $W^t(v_i) = \alpha * d_t(v_i) + \beta * E_t(v_i)$, in which α and β are two different factors.

```
Algorithm1. MST_CDS
OUTPUT: The set of MCDSs at t-th time slot, $S^t_{cds}$
Initialize: Calculate all edge weights in the graph
for $x = 1:N$
    $v = v^x_{st}$
    do
        Finding set $U = \arg\max[w^t(N^t_1(v))]$ ,where $\{v \in V_{NB}, u \notin V_{NB} \mid u \in U\}$
        If (number of nodes in set U equal to 1) //the only node in U is $u_1$
            $u \leftarrow u_1$
        else let $u = \arg\max(E^t(u_k)), k = 1:size(U)$ , $u_k \in U$
        end if
        $V^x_{cds} \leftarrow u; E^x_{cds} \leftarrow <u,v>$
    until $V^x_{cds} = V$
    if $D(u) = D(v) = 1$ and $<u,v>$ exist
        Delete $u$ and $v$
    end if
end for
```

3.2 Multi-MCDS schedule

In the second phase, we find a group of MCDSs based on MST_CDS, which is denoted by $S_{cds} = \{CDS_1, CDS_2, ...CDS_k\}$ at t_c time slot. In order to reduce the reconfiguration time of backbone network, we propose a Multi-MCDS scheduling algorithm. Let

$Em^{t_c}(CDS_i)=E^{t_c}(v)$ be the minimum energy of CDS_i, in which v is the node with minimum energy in CDS_i. For S_{cds}, $ESm^{t_c}(S_{cds})$ represents the maximum energy of S_{cds}, $ESm^{t_c}(S_{cds})=\arg\max(Em^{t_c}(CDS_i))$

In order to reduce the renewal time of the backbone network, we use the parameter LT to describe the connection time of E_{cds_k} .The parameter of LT is predicted from current status of two nodes, which includes velocity, location and direction of movements, as shown in Figure 3.

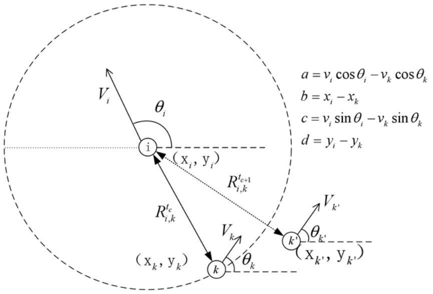

Fig. 3. Connected time prediction.

$$R_i^2 = \left(aLT_k^i(t)+b\right)^2 + \left(cLT_k^i(t)+d\right)^2 \Rightarrow$$

$$LT_k^i(t) = \frac{\sqrt{\left(a^2+c^2\right)R_i^2-\left(ad-bc\right)^2}-\left(ab+cd\right)}{a^2+c^2},\left(k \in N_1(i)\right) \tag{4}$$

Typically, the current CDS needs to be updated under the following conditions.

(1) Changes in $E_{CDS_{ct}}$ or $V_{CDS_{ct}}$ occurred, $E_{CDS_{ct}}$ is the set of links of current CDS, $V_{CDS_{ct}}$ is the set of nodes of current CDS.

(2) There are no changes detected in $E_{CDS_{ct}}$ or $V_{CDS_{ct}}$. That is, for node $v_i \in CDS_{ct}$, $N_1(v_i)\bigcap CDS_{ct}=\phi$.

(3) The number of CDSs in $S_{cds}<2$. In this case, the network topology have changes greatly, then CDS construction algorithm will be used for re-searching S_{cds} until there are no CDSs in the graph.

Algorithm 2 Multi-MCDS Scheduling algorithm

INPUT: S_{cds} , t_c , $G^t(V^t,E^t)$

OUTPUT: *scheduling of* S_{cds}

```
initialize: $CDS_{ct} \leftarrow ESm^{t_c}(S_{cds})$ ,
While 1
    if $v_i \in V_{CDS_{ct}}$ failed
        $\{CDS_k\}=\arg\{v_i \in CDS_k\}$
        $S_{cds} \leftarrow S_{cds} \setminus CDS_k$
        $CDS_{ct} \leftarrow \arg\max(N\_node(S_{cds},CDS_{ct}))$
    end if
    if （$LT_v^u(t)<\Theta$）&(CDS$_{ct}$ isn't a backbone) , $v,u \in V_{CDS_{ct}}$
        $\{CDS_k\}=\arg\{(v,u)\in E_{CDS_{ct}}\}$
        $S_{cds} \leftarrow S_{cds} \setminus CDS_k$
        $CDS_{ct} \leftarrow \arg\max(N\_node(S_{cds},CDS_{ct}))$
    end if
// $v_i \in CDS_{ct}$  $\{v_k\} \leftarrow \{N_2(v_i) \mid N_2(v_i)\bigcap CDS_{ct}=\phi\}$
```

```
$\arg\max(W(N_1 v_m))), N_1(v_m) \bigcap CDS_{ct} \neq \phi$
$\{v_m\} \leftarrow \{v_m \mid N_1(v_m) \bigcap CDS_{ct} \neq \phi\}$
end if
```

4 Simulation results

In order to evaluate the performance of the Multi-MCDS Scheduling algorithm, we make comparative groups of numerical simulations with respects to nodes update overheads and successful update rate, with a typical maintenance algorithm based on single CDS. In the simulations, the parameters of which are shown in Table 2.

Table 2. Simulations Parameters.

Parameter	Parameter description	Value
$\boldsymbol{N}$	Number of nodes	20,30 … 80
T	Simulation time slots	120
τ	time slots interval	3
v	Velocity of nodes	10/20 m/s
$\boldsymbol{R_c}$	Communication range	75/90/110 m

Initially, algorithm will randomly generate a group of nodes in a range of $200m \times 200m$ two-dimension space. Under the proposed SYN-boid model, algorithm will obtain a series of topological snapshots with a fixed sampling rate.

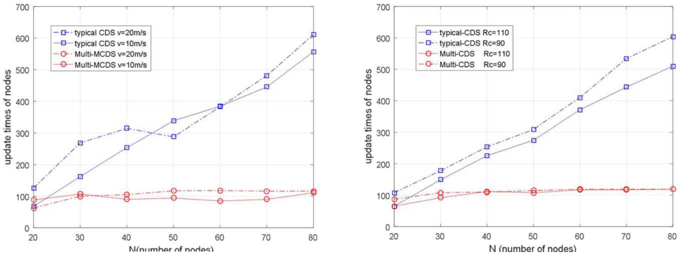

Fig.4. (a) Update times of nodes with different velocity.**(b)** Update times of nodes with different communication range.

As shown in Figure 4(a), with the increasing nodes, the typical CDS algorithm takes higher update time cost than the proposed Multi-MCDS algorithm. It can be easily seen that, the total update time cost of the proposed algorithm is significantly lower than typical CDS algorithm. In particular, the results show that the increase of moving velocity will result in a slight rise of update time cost, especially in the environment of more nodes.

For comparisons, we choose two groups of communication ranges of Rc=90 and Rc=110 during 120 time slots. As shown in Figure 4(b), the increase of communication range Rc can obviously decrease the update time cost of the network in the typical algorithm. It is mainly because larger Rc increases the amounts of available links in the graph.

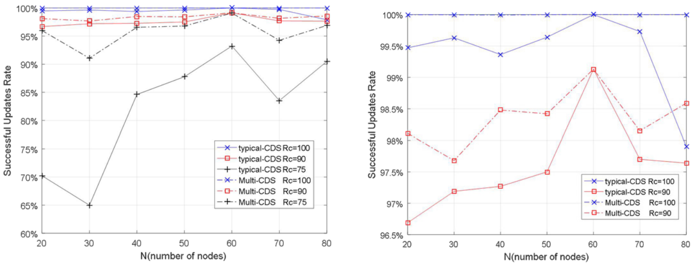

Fig. 5. (a) Successful updates rate under different communication range. (b) Local details.

We found that in Figure 5(a), the simulation under *Rc=75* performs differently with others. Especially, its successful updates rate increases a lot after decreasing *Rc*. It is mainly because the synchronization components of node v_i under the SYN-boid model develops smaller with the decrease of *Rc* until its neighbors move synchronously.

In Figure 5(b), it is obviously indicated that the proposed algorithm has a better performance of successful updates rate. With the increase of *Rc*, the network will get a higher successful updates rate, which is almost equal to 1 by using the proposed algorithm.

5 Conclusion

In this paper, we proposed a multi-MCDS algorithm to construct and maintain virtual backbone network by scheduling multi CDSs in the network. The simulations results show that, compared with the typical CDS construction and maintenance(TCCM) algorithm, the update times of nodes of the proposed algorithm is much less than TCCM, and the successful update rate is higher than TCCM with the increase of number of nodes in the network.

Acknowledgments

The authors would like to express their high appreciations to the supports from the Basic Research Project of Shenzhen (JCYJ20170413110004682 and JCYJ20150403161923521).

References

1. Dong Zhong,Yian Zhu,Tao You,Jie Kong(2014).Topology Control Mechanism Based on Link Available Probability in Aeronautical Ad Hoc Network[J] . *Journal of Networks* . 2014 (12)
2. Ozgur Koray Sahingoz. Networking Models in Flying Ad-Hoc Networks (FANETs): Concepts and Challenges[J] . *Journal of Intelligent & Robotic Systems* .pp.513-27
3. Hongwei Du, Weili Wu, Qiang Ye, Deying Li, Wonjun Lee and Xuepeng Xu (2013). CDS-Based Virtual Backbone Construction with Guaranteed Routing Cost in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(4). pp.652-61.
4. R.Misra,C.Mandal(2009), Rotation of CDS via connected domatic partition in ad hoc sensor networks . *IEEE Transactions on Mobile Computing*. pp.488-99.
5. Tuo Shi, Siyao Cheng, Zhipeng Cai, Yingshu Li, Jianzhong Li(2017).Exploring Connected Dominating Sets in Energy Harvest Networks.*IEEE/ACM Transactions on Networking*,25(3).pp.1803-17
6. Reynolds C W (1987) Flocks, herds, and schools: A distributed behavioral model. *Acm Siggraph Computer Graphics* 21(4):25-34
7. FengZou,YuexuanWang, WeiliWu,"New approximations for minimum-weighted dominating sets and minimum-weighted connected connected dominating sets on unit disk graphs", *Theoretical Computer Science*, 2009, Vol.**412** (3), pp.198-208.
8. Fu, D., Han, L. Yang, Z. and Jhang, S. 2016 A Greedy Algorithm on Constructing the Minimum Connected Dominating Set in Wireless Network. *International Journal of Distributed Sensor Networks*, 12(7), p.1703201