

Received April 4, 2020, accepted April 20, 2020, date of publication April 23, 2020, date of current version May 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989614

# SIDR: A Swarm Intelligence-Based Damage-Resilient Mechanism for UAV Swarm Networks

MING CHEN<sup>1</sup>, (Member, IEEE), HUIBIN WANG<sup>2,3</sup>, CHIH-YUNG CHANG<sup>4</sup>, (Member, IEEE), AND XIANGLIN WEI<sup>5</sup>, (Member, IEEE)

<sup>1</sup>College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

<sup>2</sup>Command and Control Engineering College, Army Engineering University of PLA, Nanjing 210016, China

<sup>3</sup>College of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China

<sup>4</sup>Department of Computer Science and Information Engineering, Tamkang University, New Taipei City 25137, Taiwan

<sup>5</sup>The 63rd Research Institute, National University of Defense Technology, Nanjing 210000, China

Corresponding author: Huibin Wang (wanghuibin@chzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772271 and Grant 61379149, and in part by the Research Foundation of Education Bureau of Anhui Province, China, under Grant KJ2017B15.

**ABSTRACT** Unmanned Aerial Vehicle (UAV) swarm networks have been presented as a promising paradigm for conducting monitoring, and inter-connecting tasks in unattended or even hostile environments. However, harsh deployment scenario may make the UAVs susceptible to large-scale damage, and thus degrades the connectivity and performance of the network. None of existing technologies can effectively re-organize the surviving UAVs in a severely damaged UAV swarm into a unified UAV Swarm Network (USNET), this paper presents and analyzes the damage-resilience problem of USNETs for the first time, and put forwards a Swarm Intelligence-based Damage-Resilient (SIDR) mechanism. First, a damage model of USNETs and several metrics are defined before the problem is formally formulated. Second, the SIDR mechanism is detailed based on comprehensively utilizing the storage, communication, positioning, and maneuvering capabilities of UAVs. Third, a potential-field-based solution to the proposed SIDR mechanism is presented, aiming to recover a USNET rapidly and elastically. At last, an evaluation environment is built on the OMNeT++ platform, and the proposed SIDR mechanism is implemented. Extensive simulations are conducted in both dynamic and static scenarios. Simulation results demonstrate that SIDR outperforms the existing algorithms in terms of resilience capability, convergence time and communication overhead. Even if a USNET is divided into multiple disjoint subnets with arbitrary shape, SIDR can aggregate the surviving nodes into a connected network while the network is still flying along the flight path during the recovery process.

**INDEX TERMS** UAV, swarm intelligence, damage model, damage-resilient mechanism, connectivity.

## I. INTRODUCTION

In recent years, Unmanned Aerial Vehicles (UAVs) have been widely recognized as promising entities to conduct tasks in dangerous, dirty and dull environments with low cost and high flexibility. A UAV swarm network (USNET) composed of multiple UAVs can adapt to different kinds of tasks in unattended or even hostile environments although one single UAV has limited storage, communication, and computation capability [1], [2]. However, there are still many technical

The associate editor coordinating the review of this manuscript and approving it for publication was Youqing Wang.

problems to be solved in designing and implementing a UAV swarm.

In a UAV swarm, UAVs collaborate with each other through the wireless links among them to accomplish specific tasks. Here, we focus on those UAV Swarm Networks (USNETs) working in the ad hoc mode without a Ground Control Station (GCS). Many factors will make maintaining USNETs a challenging task, such as the dynamics of UAV movement, the uncertainty of network topology, the low reliability of UAVs, and high damage-rate in harsh environments [3]. For instance, a military UAV swarm that makes a foray into the enemy territory is difficult to maintain

continuous connection with the GCS, and it may suffer severe damage. The goal of this paper is to design a damage-resilient mechanism for a severely damaged USNET, which is usually divided into multiple disjoint subnets or isolated nodes due to the massive damage or failure of UAVs, in harsh environments. To our best knowledge, little attention has been paid to this problem for now. There are some conflict detection and resolution (CDR) studies related to this paper in the field of air traffic management [4], but CDR is not the focus of this paper.

Three challenges are investigated in designing our damage-resilient scheme. First, the network will be divided into several disjoint subnets or isolated nodes if a large number of nodes are damaged. How to depict the damage model for a USNET? Secondly, for a severely damaged network, the traditional computational search or optimization methods cannot be utilized to restore the connectivity of the network because global information is not available. How to restore network connectivity by leveraging UAV's mobility and storage capability in the absence of global information? Thirdly, how to reduce the computational and communication overhead of the damage-resilient mechanism?

Swarm intelligence is adopted to help addressing these challenges. Each UAV is treated as an agent in the UAV swarm, and it will adjust its behaviors autonomously according to certain principles when needed based on its storage, communication, computing, positioning, and mobility capabilities. The collaborative movements of the surviving UAVs in a distributed and self-organized way can help the UAV swarm restore connectivity and cope with the degradation caused by damage. First, a damage model of USNETs and several metrics are defined before the problem is formally formulated. Second, a Swarm Intelligence-based Damage-Resilient (SIDR) mechanism is detailed based on comprehensively utilizing the storage, communication, positioning, and maneuvering capabilities of UAVs. Third, a potential-field-based solution to the proposed SIDR mechanism is presented, aiming to recover a USNET rapidly and elastically. At last, an evaluation environment is built on the OMNeT++ [5] platform, and the proposed SIDR mechanism is implemented. Extensive simulations are conducted in both dynamic and static scenarios.

Our contributions in this paper are threefold.

- **Formulation the damage model of USNETs.** The term of well-working USNET is defined at first, and a formal description of the goals and constraints of the damage-resilient problem is presented. To the author's best knowledge, this is the first work to consider the damage-resilient problem of severely damaged USNETs.
- **Designing a novel damage-resilient mechanism.** The proposed mechanism leverages UAV's mobility and storage capability to recover the damaged network. Even if a USNET is divided into multiple disjoint subnets with arbitrary shape, SIDR can rapidly re-organize the surviving nodes into a unified network autonomously,

and the network still satisfies the track constraint during the recovery process.

- **Faster recovery and lower communication overhead.** This paper presented a novel solution to the proposed SIDR mechanism to recover the USNET elastically. There is no existing research focusing on the network with dynamic coverage area, the proposed SIDR mechanism is compared with existing mechanisms introduced in [6] and [7] in static scenario. Theoretical analysis and evaluation results show that the proposed SIDR mechanism outperforms existing work in terms of convergence time and the number of sent messages. Moreover, it has the characteristics of low computational complexity.

The remainder of this paper is organized as follows. Section II summarizes the related work. Section III presents the damage model of USNETs and formulates the problem. Section IV details the proposed SIDR mechanism. In section V, extensive simulation tests are presented to verify the proposed mechanism and evaluate its performance. Finally, Section VI presents the conclusion.

## II. RELATED WORK

UAV network is a relatively new research field, and it is a very special kind of network with many technical challenges [3], [8]. Although UAV is often used as a mobile node to restore the connectivity of WSNs [9] and there have been some studies on the damage-resilient problem in WSNs and other fields [6], [7], [10]–[18], there are still no reports on the problem of how to recover severely damaged USNETs.

The categorization of the damage-resilient strategies is summarized in Fig. 1. There are two strategies to solve the damage-resilient problem of the network: Proactive and Reactive [19].

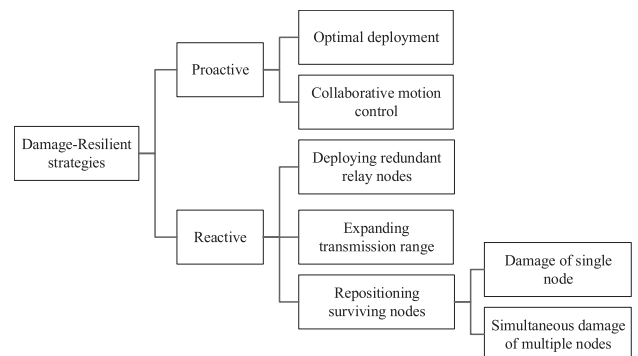


FIGURE 1. Classification of the damage-resilient strategies.

### A. PROACTIVE STRATEGY

The proactive strategy reduce the probability of network partitioning by improving or maintaining network connectivity through optimal deployment or collaborative motion control of the nodes, thereby improve the damage-resilience of the network. For example, Han *et al.* [20] presented an algorithm to improve the MANET connectivity by smart

deployment and movement of UAV. Most of the existing damage-resilient mechanisms of USNETs adopt the proactive strategy [21]–[23], which focuses on how to maintain the connectivity of USNETs. Ajorlou *et al.* [21] proposed a class of distributed potential-based control laws for avoiding the disconnection of the edge in the information flow graph. Dutta *et al.* [22] presented a decentralized controller for multiple UAVs to make a target-centric formation while maintaining the given algebraic connectivity. Esposito *et al.* [23] proposed a potential-based control law to guide a swarm of robots from the initial position to the final position, while preserving the desired links for the duration of the motion. However, the work mentioned above only studied how to maintain network connectivity when the nodes are well working, and cannot deal with the network disconnection caused by the damage of mass nodes.

## B. REACTIVE STRATEGY

The reactive strategy focuses on the connectivity restoration of disjoint subnets, and it can be classified into three categories: deploying redundant relay nodes between disjoint subnets, expanding transmission range of the nodes to merge disjoint subnets, and repositioning the surviving nodes to restore connectivity.

### 1) DEPLOYING REDUNDANT RELAY NODES

Lee *et al.* [18] proposed a Connectivity Restoration with Assured Fault Tolerance (CRAFT) algorithm to restore a partitioned WSN and form a bi-connected inter-partition topology, which is tolerant to a single node failure. The goal of CRAFT is minimizing the maximum path length between pairs of partitions and deploying the least count of relay nodes. Park *et al.* [9] considered the route recovery problem of using UAVs as relay nodes to connect with partitioned terrestrial networks in post-disaster scenarios. This category of methods takes advantages of the maneuverability and flexibility of the UAV, but deploying redundant relay nodes will significantly increase the cost of USNETs due to the relatively high cost of UAV nodes. Moreover, the redundant backup nodes may be damaged at the same time when the network is severely damaged.

### 2) EXPANDING TRANSMISSION RANGE

This category of methods usually requires extra hardware. In [24], uni-directional antennas are used to expand the communication range of the nodes, thereby improving the connectivity of WSNs. Tian *et al.* [25] proposed a connectivity recovery algorithm for UAV networks, which uses cooperative communication technology to establish long-distance communication link between partitioned network parts to reduce movement of nodes. Once the cooperative communication links cannot be established, nodes can proactively move to better places for establishing the links. However, it is difficult for the UAV swarm node to carry extra hardware devices such as uni-directional antennas due to its limited payload, thus this category of methods may not be applicable.

### 3) REPOSITIONING SURVIVING NODES

The technology of recovering damaged networks with surviving resources has been studied in WSNs and this category of methods is relatively more suitable for USNETs. According to the number of the damaged nodes that can be tolerated, this category of methods can be divided into two subcategories: The first can only deal with the problem of network partitioning caused by the damage of single node [10]–[15]. The second can tolerate simultaneous damage of multiple nodes [6], [7], [16], [17].

The first subcategory methods restore connectivity by moving surviving node to the location of the damaged cut-vertex node. The connectivity restoration methods proposed in [10]–[13] have similar idea, which can be summarized as finding out the cut-vertex nodes in the network first, then if the cut-vertex node is detected to be unavailable, restoring connectivity by cascaded movement of the related nodes. However, the cascaded movement causes lots of communication overhead because every moving node broadcasts a message to its neighbor before relocation. Sharma *et al.* [14] proposed a Zone Based Failure detection and Recovery (ZBFR) scheme, which considered both connectivity and coverage. The recovery process strives to recover the coverage and connectivity jointly by recursively relocating some mobile nodes and probing backup nodes. Mi *et al.* [15] proposed an Obstacle-avoidance Connectivity Restoration Strategy (OCSR), which restore connectivity by choosing a backup node for each possible cut-vertex nodes and driving the backup node towards the location of the failed node when a possible cut-vertex node fails. Unlike other work, OCSR considered node dynamics during the execution of recovery process. However, the above cut-vertex-based methods cannot handle the simultaneous failure of multiple nodes, while USNETs working in harsh environments may have a large number of nodes damaged at the same time.

The second subcategory methods restore connectivity by moving nodes/partitions to a pre-defined meeting point or negotiating a recover solution at the meeting point by negotiator nodes. Joshi *et al.* [16] presented a distributed Resource Constraint Recovery (RCR) approach in the case of surviving mobile nodes are insufficient to form a stable inter-segment topology. When the network is partitioned into multiple segments, each segment populates a relay node to the meeting point (Assume that each segment has at least one mobile node). Then the relay nodes are divided into stationary relay nodes and Mobile Data Collectors (MDCs) based on pre-determined criterion (such as remaining energy). The MDCs are used to provide intermittent connectivity among the segments. The problem solved in [17] is similar to that in [16], but the scheme proposed in [17] considered the shape of the network segments, thus reduced the travel distance of MDCs. In [6], a distributed Autonomous Repair (AuR) algorithm is presented to handle the problem of network partitioning due to the failure of multiple nodes. AuR mimics inter-molecular interaction to spread out the partition in the

**TABLE 1.** Comparison between the proposed SIDR and existing mechanisms.

Studies	Network type	Without extra hardware	Number of damaged node	Considering Network delay	Coverage Area	Considering node dynamics	Path tracking
[6]	WSNs	✓	Multiple	×	Static	N/A	N/A
[7]	WSNs	✓	Multiple	×	Static	×	N/A
[9]	Ad-hoc network	✓	Multiple	N/A	Static	×	N/A
[10]–[13]	WSANs	✓	Single	×	Static	×	N/A
[14]	WSNs	✓	Single	✓	Static	×	N/A
[15]	MANET	✓	Single	×	Static	✓	N/A
[16], [17]	WSNs	✓	Multiple	✓	Static	×	N/A
[18]	WSNs	✓	Multiple	N/A	Static	N/A	N/A
[24]	WSNs	×	Multiple	N/A	Static	N/A	N/A
[25]	UAV Networks	×	Multiple	×	Dynamic	×	×
SIDR	UAV Swarm	✓	Multiple	✓	Dynamic	✓	✓

direction of loss in order to have the chance to connect with other partitions, and the partition will be moved in cascade manner towards the meeting point if it does not meet other partitions. The partition repeat the AuR process until it reaches the meeting point or is connected to the node located as the meeting point. AuR scheme can handle the simultaneous failure of multiple nodes, but it asks that when a node moves its neighbors should stay still, thus prolonging the connectivity restoration time. Besides, every moving node has to send its position to its neighbor before relocating, which increases the communication overhead. Shrivastav *et al.* [7] proposed an approach for restoring the connectivity of WSNs by using Round-Table Negotiation (RTN). The goal of RTN is minimization of the time to reconnect, alongside minimization of deployed node count and total distance traveled. The idea of RTN is to select a node from each partition that is closest to the meeting point as the negotiator. Then these negotiators are moved to the circular area around the meeting point (named round table) to negotiate the shortest reconnection paths and replacement nodes. Finally, the negotiators return to their initial locations and guild the nodes of the partition to move to the desired position to restore connectivity. The RTN approach requires many communication and computation iterations, which results in relatively high time complexity and communication overhead. Besides, it does not consider the impact of network delay.

Compared with this paper, most of the studies mentioned above assume that the coverage area of the network is static, but the coverage area of USNETs will change dynamically due to the needs of tracking the flight path when performing missions. To the author's knowledge, this is the first work to consider the connectivity restoration problem of the network with dynamic coverage area. Additionally, most of the existing studies ignore the effects of communication factors such as network delay and packet loss, but these factors have a great impact on the recovery mechanism, especially the wireless links are susceptible to the environment or co-frequency signals. Because frequent communication may lead to a large number of collisions, resulting in high delay or packet loss, this paper optimizes it from the perspective of reducing message complexity. Moreover, because the encountered subnets

will be merged during the execution of recovery process, the algorithm must consider the node dynamics and control the motion of the nodes in real time, and the impact of routing also needs to be considered due to that it takes a while to establish routing among encountered subnets. Besides, the main goal of most existing studies is to minimize the travel distance of nodes to reduce energy consumption. However, because the energy consumed by the UAV nodes in the hover state and the motion state is not much different [26], reducing the travel distance does not make much sense for USNETs. Therefore, the goal of this paper is to minimize the recovery time and communication overhead.

Table 1 compares the characteristics of previous related studies and the proposed SIDR mechanism. It is shown that the proposed SIDR mechanism exhibits all good characteristics.

### III. DAMAGE MODEL AND PROBLEM FORMULATION

This section proposes the damage model of USNETs and formulated the problem investigated in this paper.

#### A. DAMAGE MODEL OF USNETs

The USNET considered in this paper has pre-defined network structure and a flight path (including a series of waypoints) according to its mission. A distributed election algorithm, such as Bully [27], is adopted to determine the master node. Similarly, if the USNET is divided into several subnets, each subnet will automatically elect a master node to take charge of the subnet. The master node is responsible for a variety of functions, such as perceiving the states of other nodes in the subnet, calculating the location of the subnet, and aggregating subnets.

Let time-varying undirected graph  $G(t) = \{U(t), E(t)\}$  be the USNET that performs a certain mission, where  $U(t) = \{u_i \mid i = 1, 2, \dots, n\}$  denote  $n$  UAV nodes of the USNET at time  $t$ ,  $E(t) = \{e_{ij} \mid u_i \in U(t), u_j \in U(t)\}$  denote the bidirectional broadband wireless links between nodes at time  $t$ . It is assumed that each node carries a positioning module such as Global Positioning System (GPS) to get its current position, and the node can move to a specific location independently with the help of the aircraft. Moreover, with the



help of the sensors, the node can sense the external situations and does not collide with other nodes while moving. Let  $q_i(t) \in \mathbb{R}^3$  denote UAV node  $u_i$ 's position at time  $t$ ,  $d_{ij}(t) = \|q_i(t) - q_j(t)\|$  denote the Eulerian distance between two nodes  $u_i$  and  $u_j$  at time  $t$ . The edge  $e_{ij} \in E(t)$  exists if and only if  $d_{ij}(t) \leq R$ , where  $R$  is the transmission range of a UAV node.

In a well-working USNET, all nodes are connected while the whole network moves along the flight path. Here, we define a few terms and symbols as follows before diving into problem formulation.

Let  $c_{i,j}(t)$  be a Boolean variable that represents whether  $u_i$  and  $u_j$  are neighbors or not. That is,

$$c_{i,j} = \begin{cases} 1 & \text{if } e_{ij} \in E(t) \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

**Definition 1:** (1-hop neighboring set): Let  $N_i(t)$  be the set of neighbors of UAV  $u_i$  at time  $t$ . That is,

$$N_i(t) = \{u_j \mid c_{i,j} = 1\} \quad (2)$$

**Definition 2:** (Connected subnet): A set  $S(t)$  is said to be a connected subnet at time  $t$  if  $S(t) \subseteq U(t)$  and  $\forall u_i \in S(t)$  can reach other nodes  $u_j \in S(t)$ .

**Definition 3:** (Maximum connected subnet): A set  $S(t)$  is said to be a maximum connected subnet if there does not exist another connected subnet  $S'(t)$  such that  $S'(t) \supset S(t)$ . Let  $\mathbb{S} = \{S_i(t) \mid i = 1, 2, \dots, m\}$  be the set of  $m$  maximum connected subnets,  $\cup_{i=1}^m S_i(t) = U(t)$ . For the simplicity of expression, the subnets mentioned below all refer to maximum connected subnets.

Let time  $t_1 < t_2 < \dots < t_n$  denote a discrete time series,  $f_p(t_n) \in \mathbb{R}^3$  denote the pre-defined waypoint at time  $t_n$ . The USNET needs to fly along the waypoints as a whole during the mission. Assume that the USNET flies in a straight line at a uniform velocity between two pre-defined waypoints, a continuous function of the pre-defined flight path  $f_p(t) \in \mathbb{R}^3$  can be obtained from the sequence of waypoints, as shown in (3).

$$f_p(t) = f_p(t_{k-1}) + \frac{(t - t_{k-1})[f_p(t_k) - f_p(t_{k-1})]}{t_k - t_{k-1}} \quad (3)$$

where  $t_{k-1} \leq t \leq t_k$ .

**Definition 4:** (Distance between a subnet and the waypoint): Let  $\delta_i(t)$  denote the distance between a subnet  $S_i(t)$  and the waypoint at time  $t$ , it is defined as the minimum distance between all nodes  $u_j \in S_i(t)$  and the waypoint at time  $t$ . That is,

$$\delta_i(t) = \min_{u_j \in S_i(t)} \|q_j(t) - f_p(t)\| \quad (4)$$

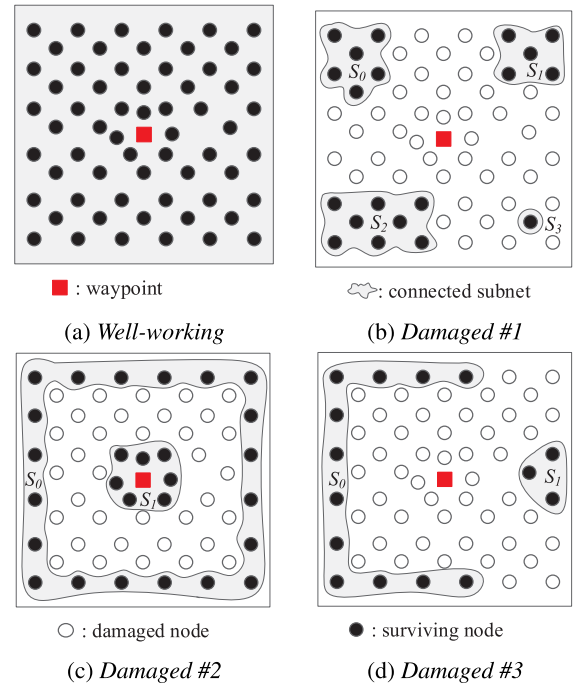
**Definition 5:** (Well-working USNET): A USNET is said to be a well-working USNET at time  $t$  if it satisfies the following conditions:

- 1)  $U(t)$  is the maximum connected subnet includes all surviving nodes,

- 2) The distance between  $U(t)$  and the waypoint at time  $t$  is not larger than  $\frac{R}{2}$ , that is,

$$\min_{S_i(t) \in \mathbb{S}} \delta_i(t) \leq \frac{R}{2} \quad (5)$$

A well-working USNET means that all surviving nodes are interconnected, and the distance between any node and the waypoint is not larger than  $R/2$ . USNET is in *well-working* state when it is initialized, and it is divided into several disjoint subnets with arbitrary shape after severe damage. The state that the damaged USNET is in is referred to the *damaged* state. Fig. 2a gives an example of a well-working USNET, where a black solid circle denotes a surviving node, a hollow circle denotes a damaged node, a gray shaded area denotes a connected subnet composed of surviving nodes, and the red solid square denotes the current waypoint. Fig.2b, Fig.2c and Fig. 2d show some possible structures of the severely damaged USNET.



**FIGURE 2.** An example of a USNET in a harsh environment.

## B. PROBLEM FORMULATION

When the network is severely damaged, the nodes can move to the appropriate location at velocity  $v_i(t)$  to aggregate multiple disjoint subnets into a connected network rapidly, where  $v_i(t) \in \mathbb{R}^3$  is a vector that denotes the velocity of node  $u_i$  at time  $t$ . The dynamics of  $u_i$  is governed by the following formula:

$$\begin{cases} \dot{q}_i(t) = v_i(t) \\ 0 \leq \|v_i(t)\| \leq V_{max} \end{cases} \quad (6)$$

where  $V_{max} \in \mathbb{R}$  denote the maximum flight speed of a UAV. This paper assumes that the maximum flight speed

is determined according the external forces on the UAV, i.e. wind and other forces have been taken into account in the calculation of  $V_{max}$ . Meanwhile, it is assumed that the external forces on all nodes in the same subnet are basically the same, so the maximum flight speed of each node in the same subnet is almost no difference; thus the nodes in the subnet will not fall behind when flying in formation.

High damage-resilience of a USNET refers to the ability that all surviving nodes can reconstruct the USNET after a great damage happens. Recover time  $T_{recovery}$  is a key metric to measure the damage-resilient ability of a USNET, and it refers to the time of recovering a network from the *damaged* state to *well-working* state (satisfying the conditions of *Definition 5*).

Let  $t_{damaged}$  denote the time when a USNET is damaged,  $t_{recovery}^i$  denote the time when the  $i$ th subnet is recovered to *well-working* state, the time taken by a USNET to complete the recovery is:

$$T_{recovery} = \max_{S_i(t) \in S} (t_{recovery}^i - t_{damaged}) \quad (7)$$

The goal of this paper is to design a distributed control scheme for UAV nodes based on swarm intelligence. In other words, each surviving node is required to move to a specific location at an appropriate velocity, so that the USNET can be recovered to *well-working* state. The goal of this paper is:

*Goal:*

$$v(t) = \arg \min_{v(t)} T_{recovery} \quad (8)$$

The following constraints must be satisfied when meeting the goal shown in (8).

The first constraint is the restoring time  $T$ , i.e. the time needed to restore connectivity among all surviving nodes after the damage.  $T$  is related to the size of the USNET, coverage area, and maximum flying speed of a UAV node. This constraint is shown in (9).

(1) *T-bounded connected constraint*

Given times  $t_1 \leq t_2 \leq \dots \leq t_{k-1}$ , for  $\forall u_{i-1}, u_{i-k} \in U(t)$ ,  $\exists u_{i-2}, u_{i-3}, \dots, u_{i-k-1} \in U(t)$ , such that

$$\begin{cases} \prod_{j=1}^{k-1} c_{i-j, i-j+1}(t_j) = 1 \\ |t_{k-1} - t_1| < T \end{cases} \quad (9)$$

The second constraint requires that all surviving nodes to keep track of the flight path during the recovery process. The constraint is expressed in (10).

(2) *Track constraint*

Given times  $t_{k-1} \leq t \leq t_k$ , for  $\forall u_i \in U(t)$ , it always has a velocity component  $v_f^i(t)$  to track the flight path. That is,

$$v_f^i(t) = \frac{f_p(t_k) - f_p(t_{k-1})}{t_k - t_{k-1}} \quad (10)$$

In order to enable UAV nodes to perform the operations of network recovery while tracking the flight path, this paper makes the following assumption:

*Assumption 1:* The speed of tracking the flight path is less than the maximum flight speed  $V_{max}$ . That is,  $\|v_f^i(t)\| < V_{max}$ .

#### IV. SWARM INTELLIGENCE-BASED DAMAGE-RESILIENT MECHANISM

When a large number of USNET nodes are damaged, the surviving nodes may form multiple disjoint subnets. At this point, the USNET may be partially damaged or fragmented. Traditional topology control methods are difficult to aggregate surviving nodes into a unified network. Fortunately, USNET nodes usually have the capabilities of storage, wireless communication, computing, positioning and mobility. Therefore, this paper proposes a Swarm Intelligence-based Damage-Resilient (SIDR) mechanism.

Each UAV node is required to store the mission, network structure, resources, and track information of the USNET in its storage space. This makes all agents have a consensus, which is also the basis for the intelligent emergence. The capability of wireless communication enables a UAV to continuously search for surviving nodes or subnets around it, and constantly merge subnets. The capability of computing enables a UAV to determine the best route to reconstruct the USNET in the shortest time while following the flight path. The capability of mobility enables the agents to move to an appropriate position. Therefore, the combination of these capabilities can be used to emerge the swarm intelligence of damage-resilient in a distributed manner.

The proposed SIDR mechanism consists of 3 phases:

- 1) *Well-work phase:* by utilizing the capability of storage, each node memorizes the structure, configuration and track information of the previous USNET received from the master node. All of the swarm nodes track the flight path of the USNET in a unified direction and speed.
- 2) *Damage identification phase:* the master node identifies whether the network is damaged or not, and the network recovery process will be initiated if the network is damaged. If only an individual node is damaged and the node does not affect the connectivity of the network, the master node will adjust it according to the policy (this paper does not discuss this issue), and the aggregation process will not be initiated.
- 3) *Network aggregation phase:* the master node of each subnet is in charge of adjusting the flying behaviors of the nodes in the subnet. The flying behavior is decided by two factors: one is to move towards the current waypoint, and the other is to follow the pre-defined path. In the process of moving towards the waypoint, the subnet will merge with the encountered subnets and eliminate redundant master node. The USNET will re-enter to the well-working phase if the aggregation phase is finished.

The following subsections first introduces the three phases of the SIDR mechanism, and the potential-field-based solution to network aggregation is discussed in detail. Then the

termination time of recovery process is discussed. Finally, the algorithms are analyzed.

### A. WELL-WORKING PHASE

In the initialization stage of a USNET, it has a unique master node decided by pre-election or other methods. The master node is responsible for perceiving the resources and locations of the other nodes, guiding the USNET to fly along the flight path, and informing all nodes of the information that needs to be shared, such as mission information, network structure, flight path, etc.

The master node sends polling messages to all slave nodes in the USNET every  $T_{poll}$  time while performing a mission. The polling message includes the current state of the resources and configurations. Each slave node enclose its state information into the acknowledge message when receiving a polling message. Thus, the master node can capture the latest global information of the network. Besides, the slave nodes can also actively report its information to the master node using *TRAP* message. Through "*POLL & TRAP*" operations, all nodes in the network reach a consensus on the node set of *well-working* state  $S_w$  and the flight path  $f_p$ .

Moreover, all nodes should keep track of the flight path in *well-working* phase, thus the velocity of the node  $u_i$  is equal to  $v_f^i(t)$ , i.e.  $v_i(t) = v_f^i(t)$ .

### B. DAMAGE IDENTIFICATION PHASE

Once a large number of nodes are damaged, it is necessary to identify the damage rapidly and initiate the recovery process. This section presents an algorithm for accomplishing damage identification.

Let *RTO* be the *Retransmission Time Out* of the polling operation. The master node will retransmit the polling message if it does not receive the anticipated acknowledge message from a slave node within *RTO* time. Let  $k > 0$  be the maximum number of retransmits. The master node will mark the slave node as damaged if no acknowledgement message received after trying  $k$  times. If a slave node does not receive any polling message from the master node in  $T_{poll} + k \times RTO$ , it will spontaneously initiate some election algorithm (such as the Bully) to determine the new master node of the current subnet. Each subnet will have a unique master node after the distributed election process. The newly elected master node will collect the node information of the subnet, and calculate the moving direction and speed of the subnet according to the locations of the nodes and the historical information of the USNET.

The master node will update the node set  $S(t)$  of the current subnet after finishing a complete polling process or receiving a *TRAP* message, and calculate the distance between the subnet and the waypoint. A deviation happens if the distance is greater than  $\frac{R}{2}$ . In this case, the next action will be derived based on the number of damaged nodes, which can be expressed as  $|S_w| - |S(t) \cap S_w|$ . It is noted that the number of damaged nodes mentioned here refers to the number of disconnected nodes, which is not equal to the number of actual

damaged nodes. For instance, the disconnection of multiple nodes may be caused by the damage of a cut-vertex node. If  $|S_w| - |S(t) \cap S_w| = 1$ , the damaged node does not affect the connectivity of the network. If the damaged node is the master node, other slave nodes will initiate a distributed election process, and then the newly elected master node will restart the damage identification process. In order to avoid initiating the high cost aggregation process, the distance between the subnet and the waypoint can be adjusted by the master node, such as moving the subnet to the waypoint as a whole, or selecting a surviving node to replace the damaged node. If  $|S_w| - |S(t) \cap S_w| > 1$ , the network may be partitioned. In this case, the master node switches to the *damaged* state and records the time  $t_{damaged}$ , then initiates the *network aggregation process* (see *Algorithm 2*). The damage identification process performed by the master node is shown in *Algorithm 1*.

---

#### Algorithm 1 Damage Identification Process Performed by the Master Node

---

**Input:** *systemStatus*,  $S(t)$ ,  $S_w$ ,  $f_p(t)$

**Output:** *systemStatus*

```

1 if systemStatus != damaged then
2   isDamaged ← true;
3   damageCount ← 0;
4   for  $u_k \in S_w$  do
5     if  $u_k \notin S(t)$  then
6       | damageCount ++;
7     else if  $\|q_k(t) - f_p(t)\| \leq \frac{R}{2}$  then
8       | isDamaged ← false;
9     end
10  end
11 if isDamaged then
12   if damageCount ≤ 1 then
13     | Adjust the distance between this subnet and
14     | the waypoint;
15   else
16     | systemStatus ← damaged;
17     |  $t_{damaged}$  ← currentTime ;
18     | Initiate network aggregation process;
19   end
20 end
```

---

In *Algorithm 1*, *systemStatus* represents current state of the USNET. In Step 1, the master node checks whether the current system is in the *damaged* state, the damage identification process will be initiated only if the current system is not in the *damaged* state. Step 4-10 count the number of damaged nodes, and check whether the current subnet deviates from the flight path. Step 11-19 give the processing method when the subnet deviates from the flight path. If the number of damaged nodes is not greater than one, the master node adjusts the distance between the subnet and the waypoint. Otherwise, the master node sets the state of the current system

to be *damaged* and record the current time, and then initiates the *network aggregation process*.

### C. NETWORK AGGREGATION PHASE

An aggregation algorithm is needed to combine all subnets into a connected network once a damage is identified. This section first introduces the basic idea of the algorithm design, illustrates several lemmas of the aggregation process, and then details a potential-field-based network aggregation algorithm.

#### 1) BASIC IDEA OF NETWORK AGGREGATION

Network aggregation refers to the process that all subnets approach the current waypoint. A USNET may be divided into several disjoint subnets with arbitrary shape after a severe damage, as shown in Fig. 2. In this paper, the potential field method is adopted to solve the network aggregation problem. Fig. 3 gives an example of potential-field-based network aggregation. The USNET shown in Fig. 3 is divided into five disjoint subnets, each of which can be represented by a virtual node. Let  $\vec{F}_f$  denote the force of the virtual node to track the flight path,  $\vec{F}_a$  denote the centripetal force for aggregating subnets, i.e. the attraction of the waypoint to the subnet. The velocity of the subnet can be determined by the joint force of  $\vec{F}_f$  and  $\vec{F}_a$ . Let  $v_f \in \mathbb{R}^3$  and  $v_a \in \mathbb{R}^3$  denote the velocity components generated by  $\vec{F}_f$  and  $\vec{F}_a$ , respectively. Then the velocity of the subnet is  $v_f + v_a$ . Because the flight speed cannot exceed  $V_{max}$ , the following formula should be satisfied:

$$\|v_f + v_a\| \leq V_{max} \quad (11)$$

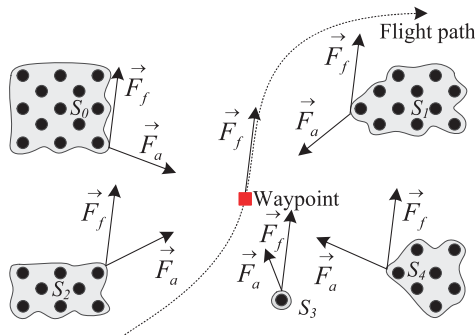


FIGURE 3. An example of potential-field-based network aggregation.

**Lemma 1** gives sufficient conditions for aggregating arbitrarily partitioned subnets.

*Lemma 1:* All subnets will be merged into a connected subnet if the distances from all subnets to the waypoint are not greater than  $\frac{R}{2}$ .

*Proof:* For  $\forall S_1(t), S_2(t) \in S(t)$ , assume that  $u_1 \in S_1(t)$ ,  $u_2 \in S_2(t)$  are the nodes which are closest to the waypoint in  $S_1(t)$  and  $S_2(t)$ , respectively. If the distances from  $S_1(t)$ ,  $S_2(t)$  to the waypoint are not greater than  $\frac{R}{2}$ , according to **Definition 4**, we have  $\delta_1 = \|q_1(t) - f_p(t)\| \leq \frac{R}{2}$  and  $\delta_2 = \|q_2(t) - f_p(t)\| \leq \frac{R}{2}$ . Let  $d_{12}$  represents the distance between

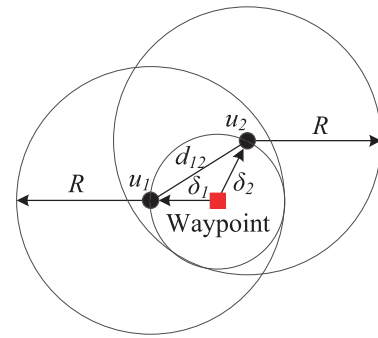


FIGURE 4. Merging of two subnets.

$u_1$  and  $u_2$ . As shown in Fig. 4,  $d_{12} \leq \delta_1(t) + \delta_2(t) \leq R$ , which indicates that  $u_1$  and  $u_2$  can communicate with each other directly, therefore  $S_1(t)$  and  $S_2(t)$  will be merged into a connected subnet. This completes the proof.

#### 2) POTENTIAL-FIELD-BASED NETWORK AGGREGATION ALGORITHM

Since each surviving UAV node has stored the pre-defined flight path, the merging of subnets can be achieved by aggregating the subnets to the waypoint while tracking the flight path. Specifically, each subnet  $S_i$  is regarded as an entity. Each master node in each subnet determines the velocity of its subnet, and slave nodes in the subnet follow the master node in a *Leader-Follower* way. Our previous researches [28], [29] have studied how UAV nodes follow the leader to achieve flocking flight, as long as the UAV swarm is properly designed and implemented, there will be no collision between UAV nodes inside the same subnet.

Before diving into the algorithm design, we first analyze the centripetal force of aggregating subnets and the force of tracking the flight path in the potential field. The basic idea is that the master node  $u_i^j$  of the subnet  $S_i$  collects the locations of all nodes in the subnet, and finds out the node  $u_{si}$  in the subnet  $S_i$  which is closest to the waypoint at time  $t$ . Then let the location  $q_{si}(t)$  of the node  $u_{si}$  represents the location of the subnet, and define the potential function  $\phi^i(t)$  according to the locations of the subnet and the waypoint. The velocity of the current subnet  $v^i$  is calculated according to  $\phi^i(t)$ , as shown in (12).

$$v^i = \dot{q}_{si}(t) = -\frac{\partial \phi^i(t)}{\partial q_{si}(t)} \quad (12)$$

According to Fig. 3, two forces are applied to each subnet: attractive force  $\vec{F}_f$  caused by tracking the next waypoint, and the centripetal force  $\vec{F}_a$  brought by network aggregation. Therefore, the potential function shown in (13) is defined to generate the forces required to recover the USNET.

$$\phi(t) = \phi_f(t) + \phi_a(t) \quad (13)$$

where  $\phi_f(t) \in \mathbb{R}$  represents the potential function of tracking the flight path, and  $\phi_a(t) \in \mathbb{R}$  represents the potential function of aggregating subnets.



The potential function of tracking the flight path  $\phi_f(t)$  is:

$$\phi_f(t) = \sum_{i=1}^m \phi_f^i(t) \quad (14)$$

where  $\phi_f^i(t) \in \mathbb{R}$  represents the potential function of the subnet  $S_i$  to track the flight path, and it is defined as:

$$\phi_f^i(t) = -[q_{si}(t)]^T \cdot \lambda(t) \quad (15)$$

where  $\lambda(t) \in \mathbb{R}^3$  represents the velocity of tracking the flight path at time  $t$ . Let  $t_{k-1} \leq t \leq t_k$ , it can be obtained that

$$\lambda(t) = \frac{f_p(t_k) - f_p(t_{k-1})}{t_k - t_{k-1}} \quad (16)$$

If the location of the next waypoint is the same as that of the current one,  $\phi_f^i(t)=0$ . This function enables the subnet to move along the flight path, thus satisfying the *Track constraint*.

The centripetal force generated by  $\phi_a(t)$  enables all subnets approach the current waypoint, and shortens the distance between each subnet and the waypoint. According to **Lemma 1**, all subnets will be merged into a connected network if the distances from all subnets to the waypoint is not greater than  $\frac{R}{2}$ . Let  $q_{si}(t)$  be the nearest location of the subnet  $S_i$  to the current waypoint, and  $q_{xi}(t)$  be the location of the intersection of the circle with the waypoint  $f_p(t)$  as its center and  $\frac{R}{2}$  as its radius and the line connecting  $q_{si}(t)$  and the waypoint, as shown in Fig. 5.  $\phi_a(t)$  is defined as:

$$\phi_a(t) = \sum_{i=1}^m \phi_a^i(t) = \sum_{i=1}^m \alpha^i(t) h_a^i(t) \quad (17)$$

where  $\phi_a^i(t) \in \mathbb{R}$  is a weighted potential function of aggregating subnets, and the weight  $\alpha^i(t) \geq 0$ . The velocity of the subnet can satisfy  $\|v^i(t)\| \leq V_{max}$  by adjusting  $\alpha^i(t)$ .  $h_a^i(t) \in \mathbb{R}$  is the potential function before weighting, that is,

$$h_a^i(t) = \begin{cases} \frac{[q_{si}(t)-q_{xi}(t)]^T \cdot [q_{si}(t)-q_{xi}(t)]}{2} & \text{if } \delta_i(t) > \frac{R}{2} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

When the distance between the subnet  $S_i$  and the waypoint  $\delta_i(t) \leq \frac{R}{2}$ ,  $h_a^i(t) = 0$ .

The potential function enables all subnets to move toward the current waypoint. When two disjoint subnets meet, they will be merged into one single subnet and a new Master will be elected based on the adopted election algorithm, then the newly elected Master will recalculate the potential of the merged subnet. As long as the aggregation speed is properly controlled, there will be no collision between subnets. The reasons are as follows:(1) the UAV nodes in the same subnet remain relatively static and the nodes are connected to each other by broadband communication links, so they have good communication conditions under normal circumstances and the communication delay mainly depends on the processing delay (tens of microseconds), the master node can immediately notify the slave nodes in the subnet to adjust their speed when it is detected that the distance between nodes

is less than a certain threshold; (2) this paper focuses on researching the small UAVs with limited flight speed and long communication range, two encountered subnets will be merged into one subnet before they collide with each other, once the merge is complete, the nodes know the distance between the nearest nodes and adjust the aggregation speed accordingly.

The velocity  $v^i(t)$  of the subnet  $S_i$  can be decomposed into two velocity components: the velocity component  $v_f^i(t)$  generated by the potential function of tracking the flight path, and the velocity component  $v_a^i(t)$  generated by the potential function of aggregating subnets, as shown in Fig. 5.

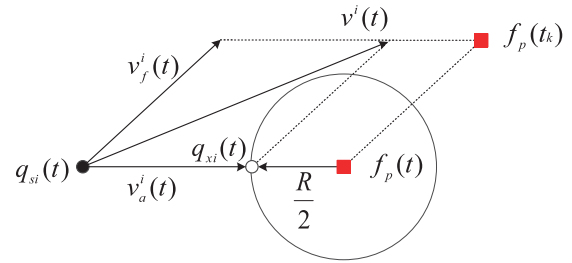


FIGURE 5. Velocity components of the subnet  $S_i$ .

In order to enable each subnet to track the pre-defined flight path, the velocity of each subnet to track the waypoint should be consistent. When the speed  $\|v^i(t)\|$  of the subnet  $S_i$  is more than  $V_{max}$ ,  $\|v^i(t)\|$  can be reduced by reducing the size of the velocity component  $v_a^i(t)$ . Formula (11) and **Assumption 1** enables the subnet to generate centripetal force of aggregating subnets while tracking the flight path. In order to aggregate the dispersed subnets into a connected subnet rapidly, it is necessary to maximize the value of  $\alpha^i(t)$ , as shown in (19).

$$\begin{cases} v^i(t) = v_f^i(t) + v_a^i(t) \\ 0 \leq \|v^i(t)\| \leq V_{max} \\ \alpha^i(t) = \arg \max(\|v^i(t)\|) \end{cases} \quad (19)$$

According to the above analysis results, the velocity of each subnet to recover the network can be derived. This paper does not consider the inertia effect, because the proposed SIDR mechanism is implemented by calling the API of the UAV's flight control system, which is beyond the scope of this paper. **Theorem 1** gives the distributed control scheme of each subnet.

**Theorem 1:** Consider a USNET satisfying **Assumption 1**, where each UAV node knows the pre-defined flight path  $f_p(t)$ , and the maximum flight speed is  $V_{max}$ . When a USNET is divided into multiple disjoint subnets, all subnets will aggregate into a unified network asymptotically and track the flight path if each subnet  $S_i$  flies at the velocity shown in (20).

$$v^i(t) = \begin{cases} \lambda(t) & \text{if } \delta_i(t) \leq \frac{R}{2} \\ \lambda(t) + \alpha^i(t)q_{xs}(t) & \text{otherwise} \end{cases} \quad (20)$$

where the value of  $\alpha^i(t)$  and  $q_{xs}(t)$  are shown in (21) and (22), respectively.

$$\alpha^i(t) = \arg \max_{\alpha} (\|v^i(t)\|) = -\frac{q_{xs}(t) \cdot \lambda(t)}{\|q_{xs}(t)\|^2} + \frac{\sqrt{[q_{xs}(t) \cdot \lambda(t)]^2 - \|q_{xs}(t)\|^2 [\|\lambda(t)\|^2 - V_{max}^2]}}{\|q_{xs}(t)\|^2} \quad (21)$$

$$q_{xs}(t) = q_{xi}(t) - q_{si}(t) \quad (22)$$

*Proof:* see Appendix I.

**Theorem 1** indicates that when the subnets fly at the velocity shown in (19), they will approach asymptotically and reconstruct into a unified network, and the *Track constraint* will be satisfied during the recovery process. In order to ensure that all surviving nodes can be aggregated into a connected network, the recovery process needs to wait for  $T$  time to terminate. The termination time  $T$  will be discussed in **Theorem 2** in Section IV-D. When the master node of each subnet initiates the aggregation process, the value of  $T$  will be set according to the current size and coverage area of the USNET.

When the time spent in the aggregation process exceeds  $T$ , the recovery process will be terminated, and the node set of *well-working* state  $S_w$  corresponding to the current UAV swarm will be updated. The process of network aggregation initiated by the master node of subnet  $S_i$  is shown in **Algorithm 2**.

---

**Algorithm 2** Potential-Field-Based Aggregation Process Performed by the Master Node of the Subnet  $S_i$

---

**Input:**  $S(t), f_p(t), V_{max}, R$

**Output:** A *well-working* USNET

```

1 Calculate  $T$  according to Theorem 2;
2 repeat
3    $u_{si} = \arg \min_{u_j \in S_i(t)} \|q_j(t) - f_p(t)\|$ ;
4   if  $\|q_{si}(t) - f_p(t)\| \leq \frac{R}{2}$  then
5      $v^i(t) \leftarrow \lambda(t)$ ;
6   else
7     Calculate the intersection's location  $q_{xi}(t)$ ;
8      $\alpha^i(t) \leftarrow \arg \max_{\alpha} (\|v^i(t)\|)$ ;
9      $v^i(t) \leftarrow \lambda(t) + \alpha^i(t)(q_{xi}(t) - q_{si}(t))$ ;
10  end
11  Guilds the nodes in  $S_i$  to fly at velocity  $v^i(t)$ ;
12  if  $S_i$  is encountered with other subnets then
13    Merge subnets and elects a new Master;
14    if current node  $\neq$  new Master then
15      return;
16    end
17  end
18 until  $(currentTime - t_{damaged}) \geq T$ ;
19  $systemStatus \leftarrow well-worked$ ;
20 Update  $S_w$ ;

```

---

In **Algorithm 2**, Step 1 calculates the termination time  $T$  of the recovery process according to **Theorem 2**. Step 3 finds out the node in  $S_i$  that closest to the waypoint at time  $t$ . Step 4-11 calculate the velocity of the current subnet according to **Theorem 1**, and guides the nodes in  $S_i$  to aggregate to the current waypoint at the same velocity. Step 12-17 deal with the problem of merging subnets in aggregation process. If  $S_i$  is encountered with other subnets in the aggregation process, they will merge into a new subnet and elects a new master node, which continues to perform the aggregation process. Step 19-21 describe the termination conditions and tail-in work of the recovery process. When the time spent in the aggregation process exceeds  $T$ , the recovery process is terminated and the relevant information is updated.

**D. TERMINATION TIME OF RECOVERY PROCESS**

This section analyses the theoretical upper bound  $T$  of the termination time of the recovery process, that is, the master node that initiates the recovery process needs to wait for at most  $T$  time to ensure that the network is recovered to the *well-working* state.

The time  $T$  spent in the network recovery process includes the time required to identify damage and aggregate subnets, which are given in **Lemma 2** and **Lemma 3**, respectively.

**Lemma 2:** When a USNET suffers from severe damage, the damage can be identified at most  $T_{identify}^{max}$  time, as shown below.

$$T_{identify}^{max} = T_{poll} + 2k \times RTO + \xi \quad (23)$$

where  $\xi > 0$  is the maximum time spent in the election process, which is determined by the selected election algorithm.

*Proof:* see Appendix II.

**Lemma 3:** When the master node identifies the damage and initiates network aggregation process, it takes at most  $T_{aggregate}^{max}$  time to aggregate all surviving nodes into a connected subnet and the distance between the subnet and the waypoint is less than  $\frac{R}{2}$ .  $T_{aggregate}^{max}$  is shown as follows:

$$T_{aggregate}^{max} = T_{move}^{max} + RTT + \tau \quad (24)$$

where  $T_{move}^{max}$  is defined in (25), and it represents the maximum time to move a subnet to the circular area around the waypoint, and  $\tau > 0$  represents the maximum time required for merging subnets,  $\tau$  is an empirical value associated with the current routing protocol and network size.

$$T_{move}^{max} = \frac{\max_{u_i \in U(t_{damaged})} \|q_i(t_{damaged}) - f_p(t_{damaged})\| - \frac{R}{2}}{V_{max} - \|v_f\|} \quad (25)$$

*Proof:* see Appendix III.

**Theorem 2** can be obtained from **Lemma 2** and **Lemma 3**, and it gives the theoretical upper bound of the time spent in the recovery process.

**Theorem 2:** When the recovery process is initiated by USNET nodes, the network can be recovered to the *well-working* state in time  $T$ , which is defined as:

$$T = T_{poll} + T_{move}^{max} + 2k \times RTO + RTT + \xi + \tau \quad (26)$$

*Proof:* When a USNET suffers from a severe damage, the time it takes to recover the network include the time  $T_{identify}$  required to identify damage and  $T_{aggregate}$  to aggregate subnets. From **Lemma 2** and **Lemma 3**, it can be known that:  $T = T_{identify}^{max} + T_{aggregate}^{max} = T_{poll} + T_{move}^{max} + 2k \times RTO + RTT + \xi + \tau$ , which is in consistent with (26). That is, the network is recovered to the *well-working* state within  $T$  time after a damage happens. This completes the proof.

## E. ALGORITHM ANALYSIS

According to **Theorem 1** and **Theorem 2**, the proposed algorithms will converge in a certain time, satisfying *T-bounded connected constraint* and *Track constraint*. Next, the time complexity and message complexity of the proposed algorithms are analyzed.

First, the time complexity of the proposed SIDR mechanism is analyzed. The time complexity of the well-working phase is simply  $O(1)$ . In the damage identification phase, **Algorithm 1** is proposed. The following discusses the complexity of each statement in the algorithm. The time complexity of steps 1-3 is  $O(1)$ , the time complexity of steps 4-10 is  $O(n^2)$ , and the time complexity of steps 11-19 is  $O(1)$ . As a result, the time complexity of the damage identification phase is  $O(n^2)$ . Next, the time complexity of **Algorithm 2** proposed in the network aggregation phase is analyzed. The time complexities of step 1 and steps 2-18 are  $O(n)$ , the time complexity of steps 19-20 is  $O(1)$ . As a result, the time complexity of the network aggregation phase is  $O(n)$ . It can be obtained from the above analysis that the time complexity of the proposed SIDR mechanism is  $O(1) + O(n^2) + O(n) = O(n^2)$ . Compared with the related work, the time complexity of AuR [6] is  $O(n^2)$ , which is the same as that of the proposed SIDR mechanism; the time complexity of RTN [7] is  $O(n^3)$ , which is higher than that of the proposed SIDR mechanism.

Next, the message complexity of the proposed SIDR mechanism is analyzed. In the well-working phase, the message complexity of polling is  $O(n)$ . The message complexity of the damage identification phase is mainly related to the adopted election algorithm. Let  $C_{election}$  denote the message complexity of the election algorithm. In the network aggregation phase, the message complexities of the master node guiding the movement of the slave nodes in the subnet is  $O(n)$ , and the message complexity of merging subnets is  $O(n)$ , thus the message complexity of network aggregation phase is  $O(n)$ . In summary, the message complexity of the proposed SIDR mechanism is  $O(n) + C_{election} + O(n) = \max\{O(n), C_{election}\}$ . If Bully algorithm is adopted, then  $C_{election} = O(n^2)$ , and the message complexity of the SIDR mechanism is  $O(n^2)$ . Therefore, the message complexity of the proposed SIDR mechanism is mainly related to the adopted election algorithm, which can be optimized by adopting an election algorithm with lower message complexity, but this is beyond the scope of this paper. Compared with the related work, the message complexities of AuR and RTN are  $O(n^2)$ , which is the same as that of the proposed SIDR mechanism.

## V. SIMULATION AND PERFORMANCE EVALUATIONS

We build a simulation environment in OMNet++ simulator to test and verify the proposed mechanism and algorithms. The nodes are evenly distributed at a designated density  $D$  in a specific square area with the same altitude. The circular area with the initial waypoint as its center and  $\frac{R}{2}$  as its radius can cover the geometric center of the USNET and at least one node. When all nodes reach a consensus on the information of flight path and topology, the nodes begin to be damaged with a specific probability.

Table 2 summarizes the simulation parameters used in the simulations.

**TABLE 2.** Simulation parameters.

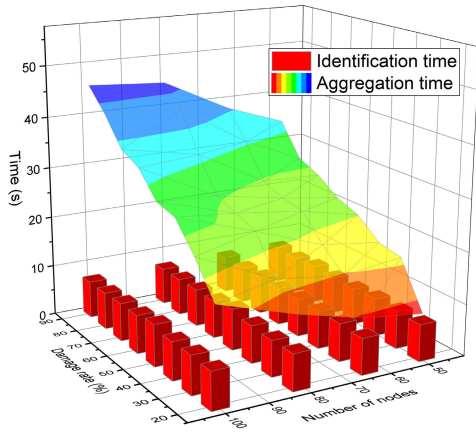
Parameter	Value
Simulation Area(length×width×height)	10 km × 10 km × 2 km
Routing Algorithm	OLSR [30]
Data rate of link	54 Mbps
Transmission range( $R$ )	200 m
Maximum flight speed ( $V_{max}$ )	20 m/s
Node density ( $D$ )	45 nodes/km <sup>2</sup>
Poll interval ( $T_{poll}$ )	5 s
Retransmission Timeout( $RTO$ )	1 s
Round Trip Time( $RTT$ )	0.1 s
Merging Time ( $\tau$ )	5 s
Retransmission times ( $k$ )	3
Distributed election algorithm	Bully [27]

This paper tests the correctness and performance of the proposed SIDR mechanism under varying damage rates(The proportion of randomly damaged nodes to all nodes). Firstly, a dynamic test scenario of UAV swarm is set up according to the damage model, and the SIDR mechanism is tested in terms of convergence time, fitness between the pre-defined flight path and the trajectory of the nodes, and communication overhead. The so-called dynamic scenario refers to the coverage area of the network is changed dynamically due to the need of tracking the flight path. Static scenario refers to that the coverage area of the network remains unchanged. The existing work on damage-resilient of the network mainly focuses on static scenarios, such as AuR [6] and RTN [7]. Therefore, this paper can only compare the proposed SIDR mechanism with the existing work for static scenarios in terms of convergence time of the recovery process and the number of sent messages.

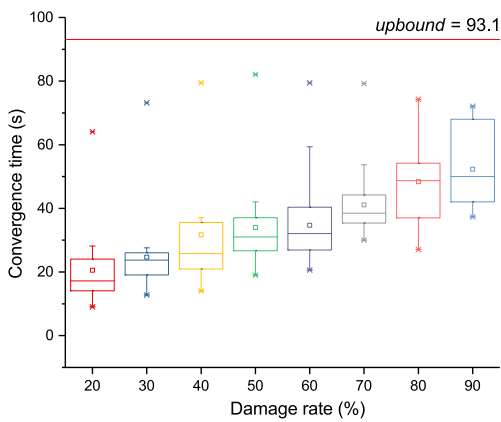
Next, this paper evaluates the performance of the proposed SIDR mechanism in dynamic and static scenario, respectively.

### A. DYNAMIC SCENARIO

The first scenario is used to verify whether the proposed SIDR mechanism can recover the damaged USNET to the *well-working* state within time  $T$ . Firstly, we test whether the SIDR mechanism satisfies *T-bounded connected constraint*. The flight path of the test is set as a straight line and  $\|v_f\| = 10m/s$ . The number of nodes  $n \in \{49, 64, 81, 100\}$ . The damage rates vary from 20% to 90%. Fig. 6a gives the identification time and aggregation time of the proposed



(a) Identification and aggregation time with varying number of nodes and damage rates



(b) Dispersion of the convergence time with varying damage rates when the number of nodes is 81

FIGURE 6. Convergence time.

SIDR mechanism. As can be seen from Fig. 6a, the time of identifying damage at varying number of nodes and damage rates is almost unchanged, which verifies the conclusion of the Lemma 2. Overall, when the number of nodes is the same, the aggregation time increases with the increase of the damage rate. This is because the number of surviving nodes decreases with the increase of the damage rate, which causes the surviving nodes to have a lower probability of encountering other nodes during the aggregation process, resulting in the increase of the moving distance. When the damage rate is the same, the aggregation time increases with the number of nodes, especially when the damage rate is higher than 30%. This is because the coverage of the network expands with the increase of the number of nodes, which leads to an increase in the distance from the nodes to the aggregation point, thus consuming longer moving time. However, when the damage rate is low, the nodes only need to move a short distance to encounter other nodes, in this case, most of the aggregation time is used to restore routing between subnets.

Fig. 6b gives the dispersion of the convergence time with varying damage rates when the number of nodes is 81. It can be seen that with the increase of the damage rate, the average convergence time of the recovery process increases, but in the worst case, the convergence time increases first and then decreases. The worst-case convergence time reaches the maximum when the damage rate is 50%. This is because when the damage rate is not more than 50%, the maximum distance between the subnet and the waypoint increases with the increase of the damage rate, which leads to the increase of the time  $T_{move}^{max}$  to move the subnet to the circular area around the waypoint. When the damage rate is more than 50%, the maximum distance between the subnet and the waypoint remains unchanged in the worst case, but the time of merging subnets  $T_{merge}$  decreases slightly with the number of surviving nodes decreasing. In order to verify the conclusion of the Theorem 2, the theoretical upper bound of the convergence time is first calculated according to the Theorem 2, that is,  $T = T_{poll} + T_{move}^{max} + 2k \times RTO + RTT + \xi + \tau = 93.1$ . As can be seen from Fig. 6b, the convergence time at varying damage rates will not exceed the theoretical upper bound, which is consistent with the conclusion of Theorem 2. In other words, SIDR mechanism satisfies *T-bounded connected constraint*.

Fig. 7 illustrates an example of the subsets' aggregation process. Fig. 7a shows the status of a USNET after a severe damage. At  $t = 20s$ , the USNET is divided into 6 disjointed subnets. Fig. 7b gives the aggregation process of the USNET, and its ordinate represents the distance between the subnet and the waypoint. When  $t \approx 28s$ , the subnets identify the damage and initiate the network aggregation process. When  $t \approx 39s$ , subnets  $S_1$  and  $S_3$  merge into a new subnet  $S_{1,3}$ . Subnets  $S_2$  and  $S_{1,3}$  become a new subnet  $S_{1,2,3}$  when  $t \approx 42s$ . At  $t \approx 43s$ , subnets  $S_4$  and  $S_5$  merge into a new subnet  $S_{4,5}$ . When  $t \approx 47s$ , subnets  $S_{1,2,3}$  and  $S_{4,5}$  merge into a new subnet  $S_{1,2,3,4,5}$ . Finally, at  $t \approx 55s$ , subnets  $S_0$  and  $S_{1,2,3,4,5}$  merge into the final subnet  $S_{0,1,2,3,4,5}$ . As can be seen from Fig. 7b, the distance between the subnets and the waypoint is decreasing linearly, and the distance between the final merged subnet and the waypoint is no more than  $\frac{R}{2}$ , which satisfies the definition of a *well-working USNET*.

The second scenario is used to verify whether SIDR mechanism satisfies *Track constraint*, where the number of nodes is 81 and the initial waypoint is located at the geometric center of the USNET. In order to demonstrate the trajectory of the surviving nodes clearly, all nodes except the four nodes at the corner are damaged when  $t = 20s$ . Fig 8 gives the results of tracking a straight line and a comb line. It can be seen from Fig. 8 that when a damage happens, the subnets will keep track of the flight path while aggregating to the waypoint, which shows good fitness between the pre-defined flight path and the trajectory of the nodes and obviously satisfies *Track constraint*. It's remarkable that the comb line is just for demonstration, Dubins path is usually used in practice.

The third scenario is used to test the communication overhead of SIDR mechanism. Fig. 9 gives the communication overhead to recover the network with varying damage rates



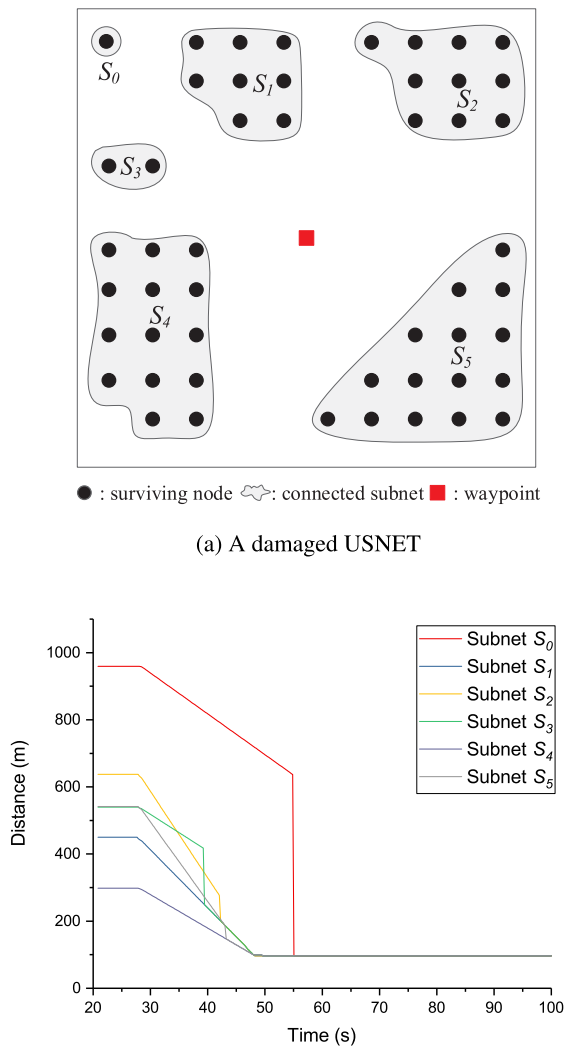
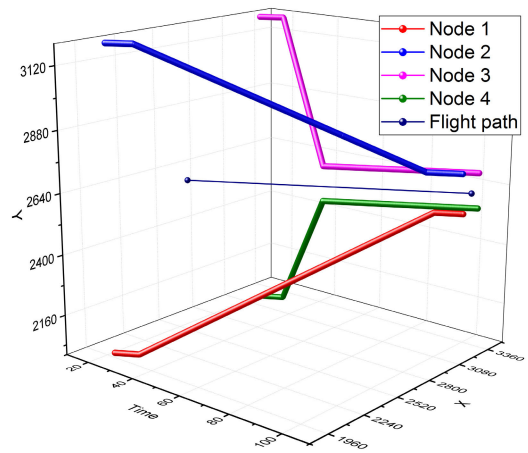
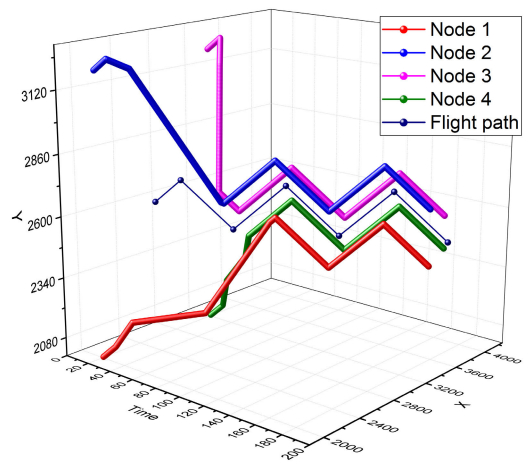


FIGURE 7. An example of aggregation process.

when the number of nodes is 81, and its parameter settings are the same as Fig. 6b. As can be seen from Fig. 9, the communication overhead of the damage identification phase decreases with the increase of the damage rate. This is because the overhead of the damage identification phase mainly includes the overhead of electing a new master node, and the election overhead decreases with the number of surviving nodes. The communication of network aggregation phase mainly includes the overhead of polling and merging subnets. Polling overhead is positively correlated with the recovery time and the number of surviving nodes, and the overhead of merging subnets is positively correlated with the number of partitions and surviving nodes. As the damage rate increase, Fig. 6b shows that the recovery time is on the rise, Fig. 9 shows that the number of partitions increases first and then decreases, and it reaches the maximum when the damage rate is 70%. Owing to the above factors, the communication overhead of



(a) Tracking of a straight line



(b) Tracking of a comb line

FIGURE 8. Flight path tracking.

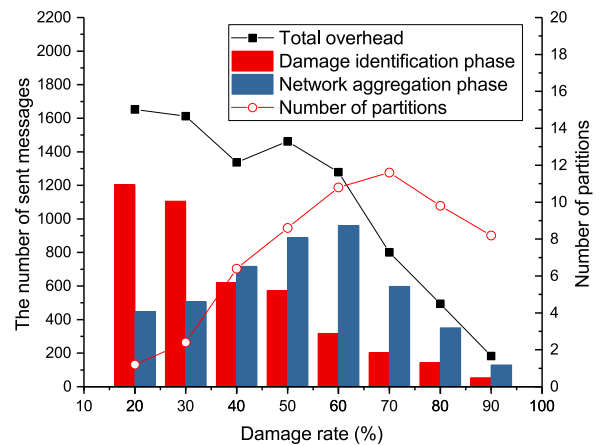


FIGURE 9. Communication overhead and number of partitions with varying damage rates when the number of nodes is 81.

the aggregation phase increases first and then decreases with the increase of the damage rate, and it reaches the maximum when the damage rate is 60%, as shown in Fig. 9. The total

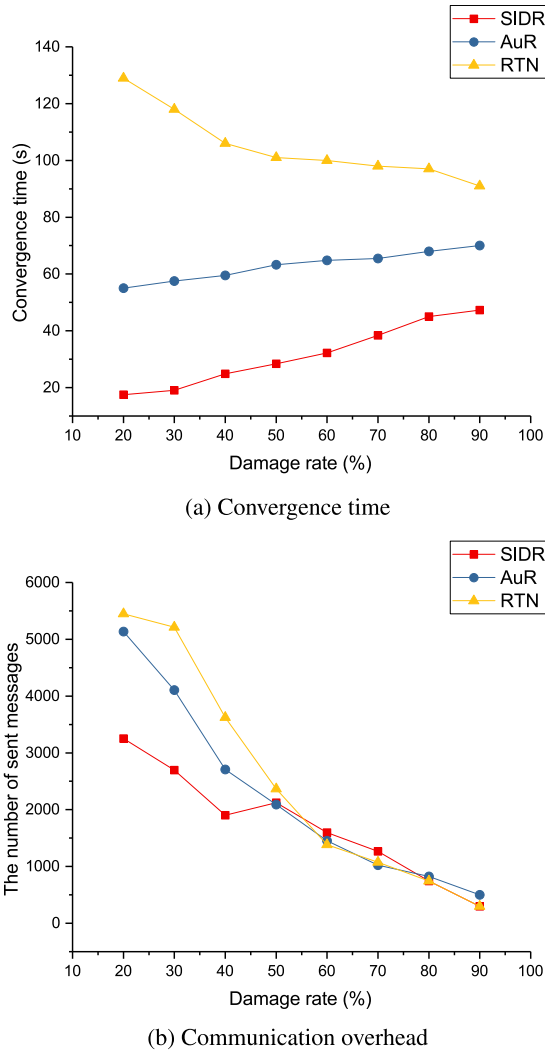


FIGURE 10. Comparison of SIDR, AuR, and RTN mechanisms.

communication overhead includes the overhead of the damage identification phase and the network aggregation phase. As can be seen from Fig. 8, with the increase of the damage rate, the total communication overhead generally shows a downward trend.

The test results show that SIDR mechanism can satisfy the requirements of damage-resilient in dynamic scenarios and converge in a specific time, and it verify the theoretical analysis of this paper.

**B. STATIC SCENARIO**

The performance of SIDR mechanism is compared with AuR and RTN in terms of convergence time and the number of sent messages in static scenario. The number of nodes  $n = 100$ , and the meeting point is located at the geometric center of the network. The nodes are damaged with the probabilities of 20%-90%. Fig. 10a compares the convergence time of the three mechanisms with varying damage rates, it can be seen that the convergence time of the SIDR mechanism is the shortest, followed by AuR mechanism, and that of RTN

mechanism is the longest. This is because the Negotiator nodes in RTN mechanism need to move to the meeting point to negotiate the recover strategy and then return to their respective partitions, resulting in longer recovery time. Fig. 10b compares the communication overheads of the three mechanisms with varying damage rates, it can be seen that when the damage rate is less than 50%, the SIDR mechanism sends the fewest messages, followed by AuR, and the RTN mechanism sends the largest number of messages. When the damage rate is more than 50%, the number of messages sent by the three mechanisms is approximately the same.

**VI. CONCLUSION**

UAV swarm is a valuable technology for accomplishing complex and sometimes dangerous tasks in unattended or even hostile environments. However, UAV swarm technology will not be practical if the problem of resilient recovery of severely damaged USNETs is not solved. This paper first presents the damage problem of USNETs, and formulates the damage model. Secondly, a novel swarm intelligence-based damage-resilient (SIDR) mechanism is proposed for severely damaged USNETs, it can aggregate disconnected nodes into a single network rapidly. Thirdly, the potential field solution method of SIDR mechanism is given. Through distributed autonomous control of UAV nodes, the severely damaged USNET can be recovered rapidly and elastically. This paper theoretically analyses the convergence, time complexity and message complexity of the proposed SIDR mechanism, and builds a simulation environment based on the OMNeT++ platform. The proposed model and mechanisms are evaluated by a series of tests, and simulation results are consistent with the results of theoretical analysis. Performance evaluations reveal that the proposed SIDR mechanism outperforms the existing work in terms of convergence time and communication overhead in static scenarios. Next, we will further study the performance optimization and practical technologies of the SIDR mechanism, such as how to further optimize the convergence time of recovery algorithm, how to reduce the delay of establishing routing in the process of merging sub-nets, how to implement the SIDR mechanism in practical UAV swarm.

**APPENDIXES**

**APPENDIX I. PROOF OF THEOREM 1**

The proof is divided into two parts. We first prove that UAV swarm will aggregate into a unified network asymptotically.

From (15), (17), and (18), the velocity components shown in (27) and (28) can be derived respectively.

$$v_f^i(t) = -\frac{\partial \phi_f^i(t)}{\partial q_{si}(t)} = \lambda(t) \tag{27}$$

$$v_a^i(t) = -\frac{\partial \phi_a^i(t)}{\partial q_{si}(t)} = \begin{cases} 0 & \text{if } \delta_i(t) \leq \frac{R}{2} \\ -[\alpha^i(t)(q_{si}(t) - q_{si}(t)) + \dot{\alpha}^i(t)h_a^i(t)] & \text{otherwise} \end{cases} \tag{28}$$

According to (27), the velocity component  $v_f^i(t)$  of each subnet to track the flight path is only related to the velocity of the waypoint, but not to the specific subnet, so all subnets keep synchronization in the direction of the flight path. Therefore, it is only necessary to prove that all subnets are approaching to the waypoint asymptotically while tracking the flight path.

The potential function  $\phi_a^i(t)$  satisfies all the criteria of a Lyapunov function because it is continuous and positive definite. When  $\delta_i(t) > \frac{R}{2}$ , it can be obtained that

$$\begin{aligned} \dot{\phi}_a^i(t) &= \nabla \phi_a^i(t) \cdot v_a^i(t) \\ &= -[\alpha^i(t)(q_{si}(t) - q_{xi}(t)) + \dot{\alpha}^i(t)h_a^i(t)] \\ &\quad \cdot [\alpha^i(t)(q_{si}(t) - q_{xi}(t)) + \dot{\alpha}^i(t)h_a^i(t)] \leq 0 \end{aligned} \quad (29)$$

Thus, the circular area with the waypoint  $f_p(t)$  as its center and  $\frac{R}{2}$  as its radius is the equilibrium point of the dynamic system because  $\phi_a^i(t)$  is strictly decreasing before the subnet  $S_i$  reaching the circular area.

Therefore, the equilibrium point must be asymptotically stable, and all subnets will be stabilized to the common equilibrium [23], i.e., all subnets will approach to the waypoint. In addition, two encountered subnets will be merged into one subnet, so it can be proved that UAV swarm will aggregate into a unified network asymptotically.

Next we prove that the speed of the subnet is not more than the maximum flight speed, i.e.  $\|v^i(t)\| \leq V_{max}$ .

When  $\delta_i(t) \leq \frac{R}{2}$ , it can be obtained from (20) that  $v^i(t) = \lambda(t) = v_f^i(t)$ . In addition, **Assumption 1** shows that  $\|v_f^i(t)\| \leq V_{max}$ . Therefore, when  $\delta_i(t) \leq \frac{R}{2}$ ,  $\|v^i(t)\| \leq V_{max}$  holds.

When  $\delta_i(t) > \frac{R}{2}$ , substituting (22), (27), (28) into (19) yields:

$$\|v^i(t)\| = \|v_f^i(t) + v_a^i(t)\| = \|\lambda(t) + \alpha^i(t)q_{xs}(t)\| \leq V_{max} \quad (30)$$

Formula (30) can be transformed into a one-variable quadratic inequality of  $\alpha^i(t)$ , as shown in (31).

$$\begin{aligned} \|q_{xs}(t)\|^2 [\alpha^i(t)]^2 + 2\lambda(t)^T \\ \cdot q_{xs}(t)\alpha^i(t) + \|\lambda(t)\|^2 - V_{max}^2 \leq 0 \end{aligned} \quad (31)$$

The maximum value of  $\alpha^i(t)$  which satisfies (31) is shown in (21). Therefore, when  $\delta_i(t) > \frac{R}{2}$ ,  $\|v^i(t)\| \leq V_{max}$  holds. This completes the proof.

## APPENDIX II. PROOF OF LEMMA 2

Let  $T_{identify}$  denote the time required to identify damage. When USNET is damaged, the worst case is that the subnet does not contain a master node or the master node has been damaged. In this case,  $T_{identify} = T_{detect} + T_{elect} + T_{perceive}$ , where  $T_{detect}$  denote the time required to detect damaged nodes,  $T_{elect}$  denote the time required to elect master node, and  $T_{perceive}$  denote the time to perceive the subnet information. According to Section IV-B, the maximum time to detect damaged nodes is  $T_{detect}^{max} = T_{poll} + k \times RTO$ .

When the network is divided into many disjoint subnets, the time for each subnet to detect the damage of master node are approximately consistent, thus the time for each subnet to initiate the distributed election process is approximately the same. Let  $\xi$  denote the maximum election time, if Bully algorithm is adopted,  $\xi = 2 \times RTO + RTT$ , where  $RTT$  represents *Round Trip Time*. When a node is elected as the master node, it needs to perceive the information of the current subnet to determine whether the current network is damaged or not. The maximum time spent in the process of perceiving information is  $T_{perceive}^{max} = k \times RTO$ .

Therefore, the maximum time to identify the damage of network is  $T_{identify}^{max} = T_{detect}^{max} + T_{elect}^{max} + T_{perceive}^{max} = T_{poll} + 2k \times RTO + \xi$ . This completes the proof.

## APPENDIX III. PROOF OF LEMMA 3

The time required to aggregate subnets includes the time  $T_{move}$  to move subnet, the transmission time  $T_{tran}$  required for the master node to guild the movement of the nodes in the subnet, and the time  $T_{merge}$  to merge subnets. Because the node does not know which nodes outside the current subnet are surviving, the maximum moving distance  $d_{move}^{max}$  of the subnet can only be calculated based on the historical information, and its value is the farthest distance between all nodes and the waypoint at time  $t_{damaged}$  minus  $\frac{R}{2}$ , as shown below.

$$d_{move}^{max} = \max_{u_i \in U(t_{damaged})} \|q_i(t_{damaged}) - f_p(t_{damaged})\| - \frac{R}{2} \quad (32)$$

**Theorem 1** shows that the minimum flight speed of the node in the direction of aggregating subnets is the maximum flight speed  $V_{max}$  minus the speed of tracking the flight path  $\|v_f\|$ , thus the maximum time required to move the subnet to the circular area with the waypoint  $f_p(t)$  as its center and  $\frac{R}{2}$  as its radius is  $\frac{d_{move}^{max}}{V_{max} - \|v_f\|}$ , which is consistent with (25). Besides, the distance between the subnet and the waypoint is not greater than  $\frac{R}{2}$  after the movement is completed.

$T_{tran}$  represents the time required to send the decision information of the master node down to the slave nodes in the subnet, and its value is 1  $RTT$  time. Two encountered subnets will be merged into one subnet and elect the master node with larger ID as the new master node. The time of merging subnets  $T_{merge}$  is mainly related to the adopted routing protocol and network size. This paper uses the empirical value  $\tau$  to represent the maximum value of  $T_{merge}$ .

Because the processes of merging subnets and transmission is simultaneous with the moving process, it can be concluded from the above analysis that the maximum time required to aggregate subnets is  $T_{aggregate}^{max} = T_{move}^{max} + RTT + \tau$ . This completes the proof.

## REFERENCES

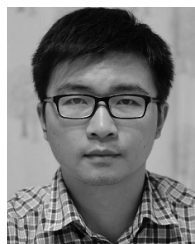
- [1] A. Bürkle, F. Segor, and M. Kollmann, "Towards autonomous micro uav swarms," *J. Intell. Robotic Syst.*, vol. 61, nos. 1-4, pp. 339-353, 2011.

- [2] A. Hocraffer and C. S. Nam, "A meta-analysis of human-system interfaces in unmanned aerial vehicle (UAV) swarm management," *Appl. Ergonom.*, vol. 58, pp. 66–80, Jan. 2017.
- [3] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.
- [4] J. Tang, "Conflict detection and resolution for civil aviation: A literature survey," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 10, pp. 20–35, Oct. 2019.
- [5] A. Varga, "Omnet++," in *Proc. Modeling Tools Netw. Simulation*. Berlin, Germany: Springer, 2010, pp. 35–59.
- [6] Y. K. Joshi and M. Younis, "Autonomous recovery from multi-node failure in wireless sensor network," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 652–657.
- [7] S. Shrivastav and D. Ghose, "Round-table negotiation for fast restoration of connectivity in partitioned wireless sensor networks," *Ad Hoc Netw.*, vol. 77, pp. 11–27, Aug. 2018.
- [8] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.
- [9] S.-Y. Park, D. Jeong, C. S. Shin, and H. Lee, "DroneNet+: Adaptive route recovery using path stitching of UAVs in ad-hoc networks," in *Proc. IEEE GLOBECOM Global Commun. Conf.*, Dec. 2017, pp. 1–7.
- [10] A. A. Abbasi, K. Akkaya, and M. Younis, "A distributed connectivity restoration algorithm in wireless sensor and actor networks," in *Proc. 32nd IEEE Conf. Local Comput. Netw. (LCN)*, Oct. 2007, pp. 496–503.
- [11] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, "Distributed recovery from network partitioning in movable Sensor/Actor networks via controlled mobility," *IEEE Trans. Comput.*, vol. 59, no. 2, pp. 258–271, Feb. 2010.
- [12] M. Younis, S. Lee, and A. A. Abbasi, "A localized algorithm for restoring internode connectivity in networks of moveable sensors," *IEEE Trans. Comput.*, vol. 59, no. 12, pp. 1669–1682, Dec. 2010.
- [13] Y. Zhang, J. Wang, and G. Hao, "An autonomous connectivity restoration algorithm based on finite state machine for wireless sensor-actor networks," *Sensors*, vol. 18, no. 2, p. 153, 2018.
- [14] K. P. Sharma and T. P. Sharma, "ZBFR: Zone based failure recovery in WSNs by utilizing mobility and coverage overlapping," *Wireless Netw.*, vol. 23, no. 7, pp. 2263–2280, Oct. 2017.
- [15] Z. Mi and J. Y. Yang, "Obstacle-avoidance connectivity restoration for mobile sensor systems with local information," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 6395–6399.
- [16] Y. K. Joshi and M. Younis, "Restoring connectivity in a resource constrained WSN," *J. Netw. Comput. Appl.*, vol. 66, pp. 151–165, May 2016.
- [17] W. Lalouani, M. Younis, and N. Badache, "Interconnecting isolated network segments through intermittent links," *J. Netw. Comput. Appl.*, vol. 108, pp. 53–63, Apr. 2018.
- [18] S. Lee, M. Younis, and M. Lee, "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance," *Ad Hoc Netw.*, vol. 24, pp. 1–19, Jan. 2015.
- [19] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel, "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *Comput. Netw.*, vol. 58, pp. 254–283, Jan. 2014.
- [20] Z. Han, A. L. Swindlehurst, and K. Liu, "Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3533–3546, Sep. 2009.
- [21] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2828–2833, Dec. 2010.
- [22] R. Dutta, L. Sun, and D. Pack, "A decentralized formation and network connectivity tracking controller for multiple unmanned systems," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 2206–2213, Nov. 2018.
- [23] J. M. Esposito and T. W. Dunbar, "Maintaining wireless connectivity constraints for swarms in the presence of obstacles," in *Proc. IEEE Int. Conf. Robot. Autom. ICRA*, May 2006, pp. 946–951.
- [24] S. Shankar and D. Kundur, "Towards improved connectivity with hybrid uni/omni-directional antennas in wireless sensor networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops*, Apr. 2008, pp. 1–4.
- [25] W. Tian, Z. Jiao, M. Liu, M. Zhang, and D. Li, "Cooperative communication based connectivity recovery for UAV networks," 2018, *arXiv:1808.08846*. [Online]. Available: <http://arxiv.org/abs/1808.08846>
- [26] D.-K. Phung and P. Morin, "Modeling and energy evaluation of small convertible UAVs," *IFAC Proc. Volumes*, vol. 46, no. 30, pp. 212–219, 2013.
- [27] H. Garcia-Molina, "Elections in a distributed computing system," *IEEE Trans. Comput.*, vols. C-31, no. 1, pp. 48–59, Jan. 1982.
- [28] M. Chen, F. Dai, H. Wang, and L. Lei, "DFM: A distributed flocking model for UAV swarm networks," *IEEE Access*, vol. 6, pp. 69141–69150, 2018.
- [29] F. Dai, M. Chen, X. Wei, and H. Wang, "Swarm intelligence-inspired autonomous flocking control in UAV networks," *IEEE Access*, vol. 7, pp. 61786–61796, 2019.
- [30] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proc. IEEE Int. Multi Topic Conf., INMIC Technol. 21st Century*, Dec. 2003, pp. 62–68.



**MING CHEN** (Member, IEEE) received the Ph.D. degree from the Institute of Communication Engineering, Nanjing, China, in 1991.

He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing. His research interests include network architecture, UAV networks, network measurement, and future networks.



**HUIBIN WANG** was born in Anqing, China, in 1986. He received the B.S. degree in computer science and technology from Anqing Normal University, Anqing, China, in 2007, and the M.S. degree in computer application technology from Liaoning Technical University, Huludao, China, in 2010. He is currently pursuing the Ph.D. degree with the Army Engineering University of PLA, Nanjing, China.

He is also a Lecturer with the College of Computer and Information Engineering, Chuzhou University, Chuzhou, China. His current research interests are in UAV networks and distributed computing.



**CHIH-YUNG CHANG** (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Zhongli, Taiwan, in 1995.

He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei City, Taiwan. His current research interests include the Internet of Things, wireless sensor networks, ad hoc wireless networks, and long-term evolution broadband technologies.



Dr. Chang was an Area Chair of the IEEE International Conference on Advanced Information and Applications (AINA) 2005, the Taiwan Academic Network (TANET) Conference 2000, TANET 2010, the IEEE International Symposium on Wireless IP (WisCom) 2005, the IFIP International Conference on Embedded And Ubiquitous Computing 2005, the IEEE International Conference on Information Technology: Research and Education 2005, and the IEEE AINA 2008, and the Program Co-Chair of the IEEE International Workshop on Multimedia Network Systems and Applications 2005, the Mobile and Ubiquitous Technologies Enhanced Learning Conference 2006, the Workshop on Wireless, Ad Hoc, and Sensor Networks 2007, and the ACM International Workshop on Sensor, Ad Hoc, and Mesh Networks 2008. He has served as an Associate Guest Editor for several SCI-indexed journals, including the *International Journal of Ad Hoc and Ubiquitous Computing*, from 2011 to 2014, the *International Journal of Distributed Sensor Networks*, from 2012 to 2014, *IET Communications*, in 2011, *Telecommunication Systems*, in 2010, the *Journal of Information Science and Engineering*, in 2008, and the *Journal of Internet Technology*, from 2004 to 2008.



**XIANGLIN WEI** (Member, IEEE) received the bachelor's degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2007, and the Ph.D. degree from the PLA University of Science and Technology, Nanjing, in 2012. He is currently working as a Researcher with The 63rd Research Institute, National University of Defense Technology, Nanjing. His research interests include mobile edge computing, wireless network optimization, and the Internet of Things.

He has served as an Editorial Member of many international journals and a TPC Member of a number of international conferences. He has also organized a few special issues for many reputed journals.

...