Abstract—In a multi-agent system, agents coordinate to achieve global tasks through local communications. Coordination usually requires sufficient information flow, which is usually depicted by the connectivity of the communication network. In a networked system, removal of some agents may cause a disconnection. In order to maintain connectivity in agent removal, one can design a robust network topology that tolerates a finite number of agent losses, and/or develop a control strategy that recovers connectivity. This paper proposes a decentralized control scheme based on a sequence of replacements, each of which occurs between an agent and one of its immediate neighbors. The replacements always end with an agent, whose relocation does not cause a disconnection. We show that such an agent can be reached by a local rule utilizing only some local information available in agents’ immediate neighborhoods. As such, the proposed message passing strategy guarantees the connectivity maintenance in arbitrary agent removal. Furthermore, we significantly improve the optimality of the proposed scheme by incorporating δ-criticality (i.e. the criticality of an agent in its δ-neighborhood).

I. INTRODUCTION

OVER the last decade, advances in networking and computing technologies along with new manufacturing techniques have created a new paradigm shift towards multi-agent systems (MAS) in engineering applications. A MAS involves a set of agents. In most cases, some information flow among the agents via local communications. Recently, there is a significant interest in using multiple unmanned vehicles over large areas for target tracking (e.g. [1], [2]), environmental monitoring (e.g. [3], [4]), persistent surveillance (e.g. [5], [6]), formation and coverage (e.g. [7], [8]), and several others. In these missions, it is often desirable to have a connected communication network. However, an agent removal may cause a disconnection.

Multi-agent health management problems emphasize that there is an inherent possibility of agent removal in a MAS due to failure or refueling (e.g. [9]–[11]). In such cases, connectivity maintenance can be achieved through proactive (e.g. [12]–[14]) or reactive (e.g. [15], [16]) approaches. In proactive approaches, a robust network topology is designed a priori to mission such that the network can tolerate a finite number of agent losses. In reactive approaches, a control strategy is developed such that the network self-repairs itself in the case of agent removal. Note that relying only on proactive approaches, by adding redundant edges to the graph for strong connectivity, can be impractical in missions that have limited resources for communication and a possibility of losing large number of agents.

Recently, there is a growing interest in the recovery processes for connectivity maintenance of networked systems. These processes can be characterized as centralized or decentralized with respect to the available information of the overall or local network structure, respectively. In a large scale system, the availability of the overall network is not a realistic assumption. Therefore, a decentralized strategy is preferable over a centralized one due to practicality and scalability concerns.

This paper introduces a decentralized recovery scheme that is applicable to any scale of networked systems. The decentralized scheme is based on a sequence of replacements occurring between an agent and one of its neighbors. Each agent is assumed to have a unique ID, which is known to its immediate neighbors. Before an agent leaves the group (e.g. due to reaching to a critical power threshold), it creates a message with its individual ID and passes it to one of its neighbors as a request for that neighbor to replace itself. Whenever an agent receives a message, it adds its own ID to the bottom of the message and sends it to one of its other neighbors, whose ID is not included in the message, i.e. a neighbor who has not received the message earlier. Accordingly, some consecutive replacements are executed. A message passing strategy is similar to token-based techniques used in various algorithms, such as in [17] and [18], to record the nodes visited by the token. In this paper, we use such a token-based idea, and we show that the resulting replacement sequences guarantee connectivity in any agent removal. The proposed strategy is a decentralized scheme that leverages only local information (e.g. agent IDs) for the connectivity maintenance, hence it is applicable to networks of any scale.

The organization of this paper is as follows: Section II presents some related work in the literature. Section III depicts some mathematical preliminaries. Section IV motivates and defines the problem. Sections V and VI elaborate on the replacement control problem and introduce the message passing strategies. Section VII presents the Monte Carlo simulations for the analysis of proposed control scheme and its comparison to an optimal (minimum number of replacements) centralized method. Finally, Section VIII concludes the paper.

II. RELATED WORK

Recently, a great amount of interest has been devoted to the analysis of multi-agent systems via the graph theory. In
these studies, the nodes of a graph represent the agents (such as robots, sensors or individuals), and edges represent the direct interactions between them. For such a representation, a fundamental graph property related to the system robustness is graph connectivity (e.g. [19]–[21] and the literature cited within). As such, the robustness of a system is related to the total number of edges/nodes, whose removal will cause a network disconnection. For the graph theoretical connectivity control of mobile systems against edge failure, the literature is including, but not limited to, optimization based connectivity control (e.g. [21]), continuous feedback connectivity control (e.g. [22]), and control based on the estimation of the algebraic connectivity (e.g. [23]). In these studies, the authors mainly consider uncertainty in edges and assume a constant number of nodes.

Maintaining connectivity against the removal of multiple agents is a more challenging problem than maintaining connectivity against multiple edge removal [12]. In the last few years, there has been a significant interest in addressing agent loss problem in networked systems. In [12] and [13], the main focus is on the design of robust network topologies that can tolerate a finite number of agent removals. In [13] and [24], the authors propose self-repair strategies that create new connections among the neighbors of the failing agent. In addition, a connectivity maintenance strategy based on decentralized estimation of algebraic connectivity is presented in [25]. Based on their estimations, agents increase or decrease their broadcast radii for satisfying connectivity requirements.

Different from the previous studies, [15], [26], [27] and [16] consider mobile agents and propose some agent movements for connectivity restoration of wireless sensor networks in agent failure. In [15], a distributed control algorithm is introduced for connectivity maintenance. Before any failure, the algorithm runs and identifies all critical agents, whose failure will cause network disconnection. Then, it assigns required actions to each agent in advance. The studies in [26] and [27] differ from [15] by maintaining connectivity through some agent relocations initiated by the failing agent. In [27], the authors present a centralized algorithm as an alternative to the decentralized scheme given in [26], which is not always feasible in general graphs. Finally, the authors of [16] use the shortest path routing table in their algorithm, and they propose a distributed recovery mechanism that maintains the network connectivity with minimal topology change, i.e. not increasing the length of the shortest path between any arbitrary two agents after the reconfiguration.

In this paper, we present a decentralized recovery mechanism to maintain network connectivity in arbitrary agent removal. The replacement control problem has been initially introduced in [28], and replacements by minimum degree neighbors have been presented as a solution. Here, we generalize the connectivity maintenance scheme as the message passing strategy, and we show that this method maintains connectivity even in the case of agents sharing minimum amount of information, i.e. only node IDs. Moreover, we show that utilizing δ-criticality information in the message passing strategy significantly improves the optimality of the solution.

III. MATHEMATICAL PRELIMINARIES

An undirected graph, \( G = (V, E) \), consists of a set of nodes, \( V \), and a set of undirected edges, \( E \). A \( k \)-length path, \( p \), is a sequence of nodes \( (p_0, p_1, \ldots, p_k) \) such that any \( \{p_i, p_{i+1}\} \in E \). Here, \( p_i \) is the \( i \)-th element of \( p \), which corresponds to a node \( v_j \). A path in \( G \) is called simple if it does not have any repeated nodes. The distance between any two nodes is equal to the length of the shortest path between them. The diameter of \( G \), \( \Delta \), is defined as the largest distance between any two nodes of \( G \). An undirected graph, \( G \), is connected if there exists a path between any two nodes of the graph.

In a graph, the unweighted adjacency matrix, \( A \in \mathbb{R}^{n \times n} \), is

\[
A_{ij} = \begin{cases} 
1 & \text{if } (v_i, v_j) \in E, \\
0 & \text{otherwise}.
\end{cases}
\]  

(1)

The neighbor set, \( \mathcal{N}_v \), of \( v_i \) is the set including all adjacent nodes that are connected to \( v_i \):

\[
\mathcal{N}_v = \{v_j \mid (v_i, v_j) \in E\}.
\]  

(2)

The degree of \( v_i \) is the number of nodes adjacent to \( v_i \), in other words the cardinality of \( \mathcal{N}_v \).

IV. PROBLEM FORMULATION

Given a networked system with no control strategy for the connectivity maintenance, the overall network will eventually become disconnected as more agents are removed. In fact, it is possible to observe graph disconnection after the removal of a small set of randomly selected agents. We demonstrate this claim by conducting Monte Carlo simulations with Erdős-Rényi graphs [31]. Let a connected graph, \( G \), have 50 nodes that have a probability of 0.04 to connect with other nodes. Suppose that a randomly selected node and its incident edges are iteratively removed until the occurrence of the first graph disconnection. When the graph is disconnected for the first time, we record the corresponding iteration number \( i_{\text{dis}} \). Then we repeat this process for 1000 cases that are initiated with different random graphs. The results of the simulations are displayed as an empirical cumulative distribution function of \( i_{\text{dis}} \) illustrated in Figure 1. To provide insight into the randomly generated graphs, Figure 1 also displays the average degree distribution. As it is seen from the figure, the removal of 10 nodes most likely cause disconnection in random graphs with 50 nodes and an approximate average degree of 2.5.

As the degrees of nodes increase in a graph, the robustness to agent failure increases. However, there is a compromise between improving the robustness and increasing the total edge number. While a robust graph is tolerant to a finite number of agent removals, it involves many redundant edges that increase the overall communication cost of the network. Alternative to designing a network with a large number of redundant links, a control strategy can be utilized to maintain connectivity. Such a control strategy may provide self-reconfiguration on the graph whenever an agent removal occurs. For static networked systems, the removal of an agent will trigger to have new connections in the neighborhood of the removed agent. For
mobile agent networks and robotic networks, the removal of an agent will induce some agent movements until the connectivity is maintained.

In many distributed systems, network connectivity plays an important role in achieving a desired system performance. For example, connectivity is required for the propagation of local data to achieve coordinated tasks. As such, formation of connected networks is emphasized in spacecraft studies (e.g., [32], [33]), where multiple spacecrafts synchronize their attitudes with each other via local interactions. Alternatively, a peer-to-peer network (e.g., Gnutella) is a distributed system, where individual computers communicate directly with each other and share information and resources without using centralized servers. In these systems, network protocols are designed to achieve various objectives, one of which is the ability to operate in a dynamic environment (e.g., [34], [35]). In this manner, when a host computer leaves a network, utilizing a connectivity maintenance strategy avoids network partitionings and prevents the performance degradation. Moreover, a group of sensors can be deployed in an area of interest to achieve distributed estimation in a harsh environment. In these systems, the network topology is dynamic due to the possibility of sensor failures or efficient energy management. As such, the sensors may increase or decrease their sensing radii to achieve desired connectivity requirements [25]. Finally, connectivity is crucial in surveillance missions, where a group of heterogeneous agents or unmanned aerial vehicles (UAV) operate around a base. For example, an efficient task assignment, which also satisfies a connected communication network, provides agents to stream back the surveillance data back to the base [36]. Similarly, a group of UAVs can monitor a desired area to track radar data and a base process the individually gathered data to estimate the position of a target [37]. In such problems, a UAV can go further away from the group to investigate unmonitored areas, or an agent can return to the base. The removal of an agent/UAV may cause a disconnection in the communication network, which leads to a base not to collect data from the disconnected agent(s)/UAV(s).

In this study, we consider an undirected connected graph, \( G \), where the nodes of \( G \) represent the required interactions if a pair of agents is assigned on the corresponding nodes (tasks). Furthermore, the edges of \( G \) correspond the required interactions if a pair of agents are assigned on the corresponding nodes (tasks). In this setting, the communication network of agents is the sub-graph of \( G \) based on the agent assignments. Here, our main objective is to maintain the connectivity of the communication network in the arbitrary agent removal by properly assigning agents to the feasible points. One way to maintain the network connectivity in these problems is to replace the removed agent by one of the remaining agents. For instance, if the removal of an agent causes disconnection in the network, then one of its neighbors may replace it to recover the connectivity. If the replacement also causes a disconnection, then another replacement is also required. In this manner, the replacements can be executed until a connected network is obtained. Accordingly, we introduce the replacement control problem as follows:

**Replacement Control (RC) problem**: Given a set of agents with a connected communication network, design a decentralized control scheme such that the agents realize minimum number of node replacements to maintain connectivity in the presence of agent removal.

**V. Message Passing Strategy**

For any solution of the RC problem, the sequence of replacements needs to end with a noncritical node since the removal of such nodes does not require any replacements.

**Definition 1. (Node Criticality)** A node, \( v_i \), is critical in \( G \) if the graph, \( G' = G - (v_i, E_i) \), obtained by removing \( v_i \) and \( E_i \) is disconnected; otherwise, \( v_i \) is noncritical.

Note that a connected graph always has a finite number of noncritical nodes [38].

**Proposition 1. (Existence of noncritical nodes)**: [38] Let \( G \) be a connected undirected graph. Suppose that each of its nodes has a degree at least \( k \). Then \( G \) has at least \( k+1 \) noncritical nodes.

For any connected graph, there are always at least two noncritical nodes, and the goal of the RC problem is to find one. The following remark presents a condition for a trivial noncritical node in a graph.

**Remark 1.** Given a connected graph \( G = (V, E) \), let \( v_i \in V \) be a leaf node such that \( |N_{v_i}| = 1 \). Then, \( v_i \) is noncritical in \( G \) because any simple path involving \( v_i \) either starts or ends with \( v_i \). Hence, its removal will not cause a disconnection between any two nodes.

In the RC problem, a replacement is assumed to occur in between a node and one of its neighbors. Therefore, the sequence of replacements can also be defined as a path from the removed node to a noncritical node.

**Remark 2.** Let \( G \) be a connected undirected graph. It follows from Proposition 7 that there always exists a path from any node in \( G \) to a noncritical node.

Note that a centralized controller can solve the RC problem by finding a shortest path between the removed node and a
Definition 2. (Maximal simple path) Let $\mathcal{G} = (V, E)$ be a connected undirected graph, and let $\mathcal{N}_i$ denote the neighbors of $v_i \in V$. Suppose that $p = (p_0, p_1, \ldots, p_k)$ is a simple path with a length of $k$. Then $p$ is a maximal simple path if $\mathcal{N}_{p_k} \subseteq \{p_0, p_1, \ldots, p_k\}$.

Theorem 1. Given a connected undirected graph $\mathcal{G}$, a maximal simple path on $\mathcal{G}$ always ends with a noncritical node.

Corollary 1. A sequence of replacements along a maximal simple path, $(p_0, p_1, \ldots, p_k)$, on $\mathcal{G}$, such that $p_0$ represents any arbitrary removed node and any $p_{i+1} \in \mathcal{N}_{p_i} \setminus \{p_0, p_1, \ldots, p_i\}$, maintains the graph connectivity.

Proof. The maximal simple path $(p_0, p_1, \ldots, p_k)$ is the replacement path where $p_0$ is any arbitrary removed node and for $0 \leq i \leq k-1$ any $p_i$ is replaced by $p_{i+1}$. After the replacements are realized, the graph will have a new structure as if $p_k$ is removed from the system. From Theorem 1, we know that $p_k$ is noncritical so its removal does not cause any disconnection in $\mathcal{G}$.

In light of the preceding facts, we introduce a decentralized connectivity maintenance scheme called message passing strategy (MPS). Let $p_0$ be any arbitrary node that will be removed from $\mathcal{G}$. The objective of MPS is to find a sequence of replacements, which is initiated by $p_0$ and ending with a noncritical node, by using only some local information. In this manner, the replacements will result in a graph reconfiguration as if the final node in the replacement sequence, which is noncritical, is removed from $\mathcal{G}$ instead of $p_0$.

The outline of MPS is as follows: Before the removal of $p_0$, first $p_0$ creates a message including its own node ID as $\{p_0\}$ and checks whether it is a leaf node. If it is a leaf node, then it is noncritical (from Remark 1) and its removal will not cause a disconnection. Otherwise, it selects a node, $p_1$, from $\mathcal{N}_{p_0} \setminus \{p_0\}$. Then, $p_0$ sends the message to $p_1$, which will replace $p_0$. In this respect, whenever a node, $p_i$, receives a message, $\{p_0, \ldots, p_{i-1}\}$, from $p_{i-1}$, before $p_i$ replaces $p_{i-1}$, it adds its individual node ID to the bottom of the message as $\{p_0, \ldots, p_{i-1} \cup \{p_i\}\}$, and it sends the message to one of its neighbors from the set $\mathcal{N}_{p_i} \setminus \{p_0, \ldots, p_i\}$. Eventually, the message resulting configuration, it is guaranteed to observe the removal of a noncritical node (e.g. $v_4$) instead of the removal of an arbitrarily removed node (e.g. $v_0$).

Algorithm 1: Message Passing Strategy (MPS)

Input: An arbitrary node, $p_0$, from $\mathcal{G}$
Output: Connectivity maintenance in the removal of $p_0$
Assumption: Each node shares its unique node ID with its neighbors.

1. initialization: $p_1 \leftarrow p_0$; $\mathcal{N}_{p_1}$, message $\leftarrow \{p_0\}$
2. if $|\mathcal{N}_{p_1}| = 1$:
3. no replacements required;
4. else:
5. while $\mathcal{N}_{p_i}$ message $\neq \emptyset$
6. $p_{i+1} \leftarrow v$ s.t. $v \in \mathcal{N}_{p_i}$ message;
7. $p_i$ sends message to $p_{i+1}$;
8. $p_i$ replaces the second last node in the message;
9. $p_i \leftarrow p_{i+1}$; $\mathcal{N}_{p_i}$, message $\leftarrow \{p_{i+1}\}$;
10. end while
11. end if

Proposition 2. The message obtained from MPS results in a set of ordered nodes, which represents either a leaf node or a maximal simple path.

Proof. The message obtained from MPS is either $\{p_0\}$ or $\{v_0, v_1, \ldots, v_k\}$. If it is $\{p_0\}$, then $|\mathcal{N}_{p_0}| = 1$ implying that $p_0$ is a leaf node. If the message is $\{v_0, \ldots, v_i, \ldots, v_k\}$, it involves consecutive pairs of nodes, $(p_i, p_{i+1}) \in E$, thus the message always represents a path in $\mathcal{G}$. Additionally, the message never involves repeated nodes because each $p_i$ selects $p_{i+1}$ from $\mathcal{N}_{p_i} \setminus \{p_0, \ldots, p_i\}$. Thus, the path is always simple. Finally, MPS stops whenever $\mathcal{N}_{p_i} \setminus \{p_0, \ldots, p_{i-1}, p_i\} = \emptyset$.

Corollary 2. MPS always stops at a noncritical node. Hence, MPS guarantees connectivity maintenance in the removal of any arbitrary node from $\mathcal{G} = (V, E)$.

Proof. Let $p_0 \in V$ be any arbitrary node that will be removed from $\mathcal{G}$. If $p_0$ is a leaf node, MPS stops at $p_0$, and the connectivity maintenance is an immediate result. Otherwise, $p_0$ generates a message as $\{p_0\}$, and the message is modified as $\{p_0, \ldots, p_i\}$ whenever it is received by $p_i \in V$. Let $N + 1$ be the total number of nodes in $\mathcal{G}$. In this respect, as $i \rightarrow N$, $\{p_0, \ldots, p_i\} \rightarrow \{p_0, \ldots, p_N\} = V$. Eventually, there exist an instant $k = i \leq N$, at which $\mathcal{N}_{p_k} \subseteq \{p_0, \ldots, p_k\}$. From Theorem 1, $p_k$ is a noncritical node because it satisfies $\mathcal{N}_{p_k} \setminus \{p_0, \ldots, p_k\} = \emptyset$. Consequently, MPS always stops at a noncritical node. Moreover, from Corollary 1, the replacements based on MPS always guarantee connectivity maintenance because the graph is reconfigured as if $p_k$ is removed from $\mathcal{G}$ instead of $p_0$.

An illustration for MPS is displayed in Figure 2 where there is an initially connected graph with 7 nodes. As it is seen from Figure 2(b), the removal of $v_0$ will create a disconnection in the graph. If each node runs MPS, then a replacement path is generated as $\{v_0, v_2, v_4\}$ such that $v_2$ replaces $v_0$, and $v_4$ replaces $v_2$. Note that $\{v_0, v_2, v_4\}$ is not the only replacement path, i.e. $\{v_0, v_1, v_3\}$. Consequently, the system reconfigures itself to maintain connectivity, and, in the resulting configuration, it is guaranteed to observe the removal of a noncritical node (e.g. $v_4$) instead of the removal of an arbitrarily removed node (e.g. $v_0$).
A. Performance of MPS

Given a networked system, reactive schemes for connectivity maintenance result in some changes in the graph topology. While maintaining the graph connectivity, an important aspect is not to cause significant changes in the graph properties such as the total number of edges or the maximum node degree. Note that the total number of edges and the maximum node degree can be directly related to the overall communication cost, whose increase is not desirable for a networked system containing agents with limited power capacity.

**Proposition 3.** A sequence of replacements along a maximal simple path, \( (p_0,p_1,\ldots,p_k) \), on \( G \), such that every \( p_{i+1} \in \mathcal{N}_{p_i} \setminus \{p_0,p_1,\ldots,p_i\} \), guarantees no increase in the total number of edges and maximum node degree in the presence of any arbitrary node removal.

**Proof.** Let \( p = (p_0,p_1,\ldots,p_k) \) be the replacement path, \( G^* \) be the new graph structure after the replacements. Then, this corollary is proven in two parts: (1) In the removal of an arbitrary node, \( p_0 \), \( p \) results in \( G^* \), which corresponds to the removal of \( p_k \) and its adjacent edges from \( G \). As a result, the total number of edges decrease as the agents are removed. (2) Let \( p_0 \) in \( p \) be the agent that has the maximum degree \( d_{\text{max}} \) in \( G \). If \( p_0 \) is removed, then \( p_1 \) replaces \( p_0 \). Now, if \( k = 1 \), then \( p_1 \) is the noncritical node that will not be replaced. As a consequence, the degree of \( p_1 \) becomes \( d_{\text{max}} - 1 \) after the replacement. If \( k \neq 1 \), then \( p_1 \) will be replaced by \( p_2 \). Hence, the degree of \( p_1 \) becomes \( d_{\text{max}} \) after the replacements. In both cases, \( p_1 \) becomes the node with the maximum degree in \( G^* \) after replacing \( p_0 \). Finally, in the removal of an arbitrary node, which does not correspond to the maximum degree node \( \tilde{v} \), either no replacements occur in the neighborhood of \( \tilde{v} \), or the replacements in the neighborhood of \( \tilde{v} \) may cause at most one reduction in \( d_{\text{max}} \). As a result, the maximum node degree in \( G^* \) becomes either \( d_{\text{max}} \) or \( d_{\text{max}} - 1 \).

The optimal solution satisfying the minimum number of replacements for the RC problem can be obtained by a centralized controller by finding the shortest path between the removed node and a noncritical node on the graph. Note that such a centralized controller requires the complete information about the graph. The objective of MPS is to solve the RC problem only by using some local and partial information. Due to utilizing limited information, MPS may not necessarily guarantee the optimal solution for any graphs. In this section, we will discuss the performance of MPS for various graph structures.

**Proposition 4.** In any undirected connected graph, \( G = (V,E) \), the maximum number of replacements that can occur via MPS is \((|V| - 1)\).

**Proof.** From Proposition 3, MPS results in a message that is the sequence of replacements represented as a maximal simple path. \( p \). Let \(|p| \geq |V| + 1 \), then at least one node appears multiple times in \( p \), thus \( p \) is not simple. This is a contradiction, hence \(|p| \leq |V| \) implying an upper bound for the number of replacements as \(|V| - 1 \).

**Definition 3.** A tree graph is an undirected graph in which any two nodes are connected by exactly one simple path.

**Proposition 5.** In tree graphs, \( G = (V,E) \), the maximum number of replacements that can happen via MPS is \((\Delta - 1)\), where \( \Delta \) is the diameter of \( G \).

**Proof.** Note that a noncritical node in a tree graph is always a leaf node, and a critical node always has a degree of 2. In this manner, the diameter of a tree graph corresponds to the length of the longest maximal simple path. Let \( \{p_0,p_1,\ldots,p_{\Delta-1},p_{\Delta}\} \) denote to the longest maximal simple path. In this path, both \( p_0 \) and \( p_{\Delta} \) are leaf nodes (noncritical), and the nodes in between are critical. If \( p_0 \) is the removed node, then MPS does not initiate replacements. If \( p_1 \) is the removed node, then the maximum number of replacements based on MPS may occur along the sequence \( \{p_1,\ldots,p_{\Delta-1},p_{\Delta}\} \) resulting in \((\Delta - 1)\) replacements.

**Definition 4.** A biconnected graph is a connected graph that does not have any critical nodes.

**Proposition 6.** In biconnected graphs, MPS cannot achieve optimal solution for connectivity maintenance.

**Proof.** In a biconnected graph, \( G = (V,E) \), each node is noncritical. However, based on MPS, any arbitrary node, \( v \in V \), that will be removed from \( G \) always initiates the replacements. In this manner, the graph is reconfigured with \(|V| - 1 \) node replacements even though the removal of \( v \) does not cause any disconnection in \( G \).

Note that MPS may not always result in the minimum number of replacements in agent removal. For instance, if the removed agent is not a leaf node, but noncritical, MPS still initiates the sequence of replacements as depicted in Proposition 3. From a centralized perspective, understanding the criticality of a node is feasible. However, the node criticality may not be determined locally. As shown in Figure 3, let \( G_1 \) and \( G_2 \) be an infinite cycle and infinite path graphs, respectively. Suppose that any node in \( G \) knows its \( \delta \)-hop neighborhood. Let \( v_0 \) be the removed node. As seen from Figure 3, \( v_0 \) is noncritical in \( G_1 \), but critical in \( G_2 \). Note that for any finite \( \delta \), \( v_0 \) has the same neighborhood in \( G_1 \) and \( G_2 \).
hence it can not differentiate its criticality by just looking at its $\delta$-neighborhood.

![Graphs with infinite nodes](image)

Fig. 3. Examples to graphs with infinite nodes.

Since the criticality cannot be always determined locally, MPS may sometimes initiate a sequence of replacements when a noncritical node is removed. Such an optimality gap is due to the limitation of local information in the computations. However, it is important to emphasize that, for any undirected connected graph, connectivity maintenance in the presence of any node removal is guaranteed by MPS by using only some local information. As depicted, MPS always stops at a noncritical node. Thus, the graph reconfigures itself as if a noncritical node is removed from the system instead of an arbitrary node, $p_0$. In this respect, $p_i$ selects the node, $p_{i+1}$, which will replace itself, from $N_{p_i} \setminus \{p_0, \ldots, p_i\}$. Here, a question arises as which node from $N_{p_i} \setminus \{p_0, \ldots, p_i\}$ should be selected to increase the efficiency of MPS. For instance, a random selection scheme requires very little information to be shared among nodes, or a node selection based on the minimum degree, [28], may capture the leaf node neighbors. Consequently, as the information possessed by a node and shared in the neighborhood increases, the solution approaches the optimal solution.

VI. $\delta$-CRITICALITY MPS

In this section, we introduce a variant of MPS, which uses $\delta$-hop criticality information for each node. Here, the $\delta$-hop criticality is defined as Definition 5.

**Definition 5.** ($\delta$-hop criticality) A node, $v_i$, is $\delta$-hop critical if the subgraph, induced by the $\delta$-neighborhood of $v_i$, is disconnected by the removal of $v_i$; otherwise, $v_i$ is $\delta$-hop noncritical.

**Remark 3.** Let $G = (V,E)$ be a connected graph, and let $v_i \in V$ be $\delta$-hop noncritical. Then, $v_i$ is noncritical in $G$. Suppose that a simple path, $p^m$, connects any two arbitrary nodes $v_n, v_m \in V$ and includes $v_i$ as an intermediate node. In $p^m$, $v_i$ appears between two of its neighbors. In the removal of $v_i$, there exist another path, $p'$, consisting of some nodes within $\delta$-hops of $v_i$ since $v_i$ is $\delta$-hop noncritical by definition. Hence, the removal of $v_i$ does not cause a disconnection between $v_n$ and $v_m$ because $v_i$ can be replaced by $p'$. Consequently, a $\delta$-hop noncritical node is always a noncritical node in $G$.

In light of Remark 3, $\delta$-hop criticality is used in MPS as in Algorithm 2. In this respect, each node knows whether itself and immediate neighbors are $\delta$-hop critical.

In $\delta$-criticality MPS, whenever a node, $p_i$, receives a message, it adds its own individual ID likewise MPS. Then, it selects a neighbor from the candidate set, $N_{p_i} \setminus \{p_0, \ldots, p_i\}$, based on $\delta$-criticality. In the case, where the candidate set does not contain a $\delta$-hop noncritical node, $p_i$ selects a random node from the candidate set.

It has been shown in Remark 3 that a $\delta$-noncritical node is globally noncritical in $G$. Now, a question arises as when a $\delta$-critical node assures global criticality. In this respect, Proposition 7 presents a sufficient condition that guarantees global node criticality by relating $\delta$ to a graph structure.

**Definition 6.** A chordless cycle in $G$ is a cycle such that no two nodes of the cycle are connected by an edge that does not itself belong to the cycle.

**Proposition 7.** Let $c_{\text{max}}$ be the length of the longest chordless cycle in $G$. If $\delta \geq \frac{c_{\text{max}}}{2}$, then a $\delta$-critical node is globally critical in $G$.

**Proof.** Let $v$ be a noncritical node in $G$, and let $N^\delta$ be the $\delta$-neighborhood of $v$ for some $\delta \geq \frac{c_{\text{max}}}{2}$, where $c_{\text{max}}$ is the length of the longest chordless cycle in $G$. Suppose that $v$ is a $\delta$-critical node, then the graph, $G'$, induced by the nodes in $N^\delta$ is disconnected. Now, since $v$ is noncritical, there exist a shortest path between the nodes $(u,w) \in N^\delta$, which are not connected in $G'$ but connected in $G - v$. Moreover, there always exist a shortest path, $(u,p^*,w)$, where no elements on $p^*$ is connected to $v$ (in other words, no elements on $p^*$ is in $N^\delta$). Note that $(u,p^*,w,v,u)$ is a chordless cycle and its length, $c$, cannot be larger than $c_{\text{max}}$. However, $v$ does not know the existence of such a path, so $c \geq 2\delta$, which is a contradiction because $2\delta \geq \frac{c_{\text{max}}}{2}$.

**Corollary 3.** If $\delta \geq \frac{c_{\text{max}}}{2}$, then the replacement sequence generated via MPS involves only one noncritical node, which is the last node on the replacement sequence.

**Proof.** Based on Algorithm 2, a message travels from a $\delta$-critical node to a neighboring $\delta$-critical node until finding a $\delta$-noncritical node. In the case of $\delta \geq \frac{c_{\text{max}}}{2}$, Proposition 7 shows that a $\delta$-critical node is globally critical. Hence, the replacement sequence generated via MPS contains only one noncritical node, which is the last node on the sequence.

**Remark 4.** Suppose that there is no chordless cycle in $G$. A $\delta$-critical node for any $\delta \geq 1$ is globally critical in $G$ because...
is a tree graph where each noncritical node is a leaf node.

**Remark 5.** If $\delta$ is selected properly based on the graph topology (based on Proposition 2 and Remark 2), a $\delta$-critical node is always a critical node. Hence, a resulting replacement sequence does not contain any redundant replacements since only the final node of the sequence is non-critical.

As it is seen, if $\delta$ is selected properly such that a $\delta$-critical node is globally critical, then $\delta$-MPS does not cause any unnecessary agent replacements to maintain connectivity, and the resulting sequence approaches the optimal solution. Note that avoiding any unnecessary replacements is crucial for networked systems with limited power capacity. In some graph structures, there might not necessarily exist a unique replacement sequence. As such, if $\delta$-noncritical nodes are beyond the immediate neighborhood of a node $v$, then $v$ selects a $\delta$-critical neighbor randomly for its replacement. Due to the randomized nature of selecting replacing agent, $\delta$-MPS may not always guarantee the shortest path to a noncritical node.

In Figure 4, a line graph involving 7 nodes is presented. In this example, let $\delta = 1$, then $v_1$, $v_0$, $v_3$, $v_4$, $v_5$ are 1-critical nodes whereas $v_2$ and $v_6$ are noncritical nodes. Since the graph does not contain a chordless circle, from Remark 3, a 1-critical node is globally critical. Here, we illustrate that $\delta$-MPS may result in a sequence of necessary replacements, but a longer route, for connectivity maintenance. In this manner, assume that $v_0$ is removed from the graph. From the perspective of $v_0$, selecting $v_1$ or $v_3$ is indifferent because $v_0$ can only see $N^{\delta=1}$, which contains the highlighted nodes. Hence, $\delta$-MPS results in either $(v_0,v_1,v_2)$ or $(v_0,v_3,v_4,v_5,v_6)$ as a replacement sequence.

![Fig. 4. In the case of $v_0$ is removed, 1-MPS generates either $(v_0,v_1,v_2)$ or $(v_0,v_3,v_4,v_5,v_6)$ as a replacement sequence.](image1)

Consequently, the optimality of $\delta$-MPS depends on the lengths of the shortest paths from a removed node to one of the noncritical nodes. If the lengths does not significantly vary from each other, then the solution of $\delta$-MPS is close to the optimal one.

**VII. SIMULATION STUDIES**

In order to elaborate on the performance of the message passing strategy, the Monte Carlo simulations are conducted to understand how close MPS is to the centralized solution. For the simulations, we use the MATLAB simulation environment and consider the canonical scenario depicted in the following section.

**A. Canonical Scenario**

Consider an indoor reconnaissance and surveillance mission, where a set of robots (i.e, small unmanned vehicles) gather data from critical points and share the gathered data to increase their situational awareness. In such a mission, an undirected connected graph, $\mathcal{G}$, associates with the critical viewpoints of the environment (i.e. the viewpoints are the feasible points that a robot can be located on) as in Figure 5, where the nodes are the viewpoints and the dashed edges represent the communication links if two robots are located on the corresponding nodes.

![Fig. 5. An indoor environment associated with an undirected connected graph, whose nodes represent the critical viewpoints and edges represent the possible communication capability if agents are located on the corresponding nodes.](image2)

In this scenario, let each robot have limited energy capacity and different energy consumption. Then, it is likely to observe that the robots have variable energy levels. In this respect, a robot leaves the group when it reaches an energy threshold. Here, the removal of an agent may cause a disconnection in the communication network. For example, let 13 agent be assigned to points such that a connected communication network is obtained as in (a) of Figure 6. Then the removal of $v_4$ causes a disconnection in the communication network as in (b) of Figure 6. Consequently, the objective in this scenario is to maintain a connected communication network among the remaining agents with minimum agent replacements.

![Fig. 6. (a) Thirteen agents assigned to the feasible tasks and having a connected communication network. (b) The removal of $v_4$ causes four partitioning in the communication network.](image3)

**B. Monte Carlo Simulations**

Based on the canonical scenario, the objective of the Monte Carlo simulations is to provide a statistical understanding for the optimality of MPS and $\delta$-criticality MPS with respect to the centralized solution. In all simulations, we consider 20 assignments, which associate with a randomly generated undirected connected graph. Initially, we assume 20 agents individually assigned to each node. At an instant, a randomly
selected arbitrary agent is removed from the network. To maintain the connectivity, we solve the RC problem via centralized controller, MPS, and $\delta$-criticality MPS.

The results of the Monte Carlo simulations show that the optimality of MPS significantly improves as $\delta$-criticality information is incorporated to the decision scheme for the replacements. For example, as illustrated in Table I, an increase in $\delta$ leads to the total number of replacements induced from $\delta$-criticality MPS to approach the number of replacements resulted from the centralized solution. Note that the huge cost difference between MPS and $\delta$-criticality MPS is due to the fact that MPS is using a randomized decision mechanism for the replacements while $\delta$-criticality MPS incorporates $\delta$-criticality of a node in the decision mechanism. Furthermore, the optimality of MPS also varies with respect to the graph topology. For example, in the case of the graph diameter decreases and the average node degree increases, the performance of MPS degrades while the performance of $\delta$-criticality MPS improves greatly.

### TABLE I

Based on 500 simulations, mean cost $^*$ of various strategies for connectivity maintenance in graphs with 20 nodes

| Mean diameter | Mean avg. node degree | Centralized | MPS | $1$-criticality | MPS | $2$-criticality | MPS |
|---------------|-----------------------|-------------|-----|----------------|-----|----------------|-----|
| 6.7           | 4.3                   | 0.186       | 7.224| 0.312          | 0.240|                |     |
| 4.0           | 7.5                   | 0.024       | 12.274| 0.068          | 0.300|                |     |

$^*$ number of agent replacements

Fig. 7. Distributions pertaining to the graph properties and replacement solutions. As $\delta$-criticality is utilized, the number of replacements for preserving connectivity approaches to the number induced by a centralized controller.

Fig. 8. Distributions pertaining to the graph properties and replacement solutions. As the graph diameter decreases and the average node degree increases, the replacement sequence driven by MPS diverges from the centralized solution.

Fig. 9. Distributions pertaining to the graph properties and replacement solutions. As the graph diameter decreases and the average node degree increases, the performance of $\delta$-criticality MPS improves greatly and converges to the performance of centralized controller.

---

removed agent. The benefits of the proposed control scheme are guaranteeing the connectivity maintenance by using only some local information and not increasing the total number of edges and the maximum node degree of a network as the agents are removed.

The optimality gap of the proposed strategy has been discussed through Monte Carlo simulations by comparing the performance of the proposed decentralized strategy with respect to the centralized solution, which results in the minimum number of replacements. While the message passing strategy maintains the graph connectivity even in the case of replacements by randomly selected agents, it has been observed that incorporating $\delta$-criticality information to the decision mechanism significantly improves the resulting performance. As such, a variant of MPS has been introduced as $\delta$-criticality MPS, which demonstrates a significant performance improvement.

---

VIII. CONCLUSIONS

In this study, the connectivity issue of networked systems in the presence of agent removal has been discussed, and a decentralized connectivity maintenance strategy, which is applicable to any scale of network, has been proposed. We showed that the message passing strategy (MPS) proposed in this paper maintains the graph connectivity for any initially connected network until the removal of all agents. This is achieved by a sequence of replacements initiated by the
Some interesting aspects requiring further investigation include the development of a strategy for simultaneous agent removals, a throughout study for a more general communication problem involving the delays and uncertainty, and the introduction of a more general mission, in which some of the removed agents return to the mission area.

REFERENCES

[1] S. Martinez and F. Bullo, “Optimal sensor placement and motion coordination for target tracking,” Automatica, vol. 42, no. 4, pp. 661–668, 2006.

[2] R. Olfati-Saber, “Distributed tracking for mobile sensor networks with information-driven mobility,” in American Control Conference, 2007. ACC ’07, 2007, pp. 4606–4612.

[3] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Frattoni, and R. E. Davis, “Collective motion, sensor networks, and ocean sampling,” Proceedings of the IEEE, vol. 95, no. 1, pp. 48–74, 2007.

[4] B. White, A. Tsourdos, I. Ashokaraj, S. Subchan, and R. Zbikowski, “Contaminant cloud boundary monitoring using network of UAV sensors,” Sensors Journal, IEEE, vol. 8, no. 10, pp. 1681–1692, 2008.

[5] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, “Decentralized cooperative aerial surveillance using fixed wing miniature UAVs,” Proceedings of the IEEE, vol. 94, no. 7, pp. 1306–1324, 2006.

[6] B. Bethke, J. P. How, and J. Vian, “Multi-UAV persistent surveillance with communication constraints and health management,” in Proceedings of the AIAA Guidance, Navigation and Control Conference, Chicago, IL, 2009.

[7] J. Fax and R. Murray, “Information flow and cooperative control of vehicle formations,” IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1465–1476, 2004.

[8] X. Li and Y. Xi, “Distributed connected coverage control for groups of mobile agents,” International Journal of Control, vol. 83, no. 7, pp. 1347–1363, 2010.

[9] J. Redding, Z. Dydek, J. P. How, M. A. Vavrina, and J. Vian, “Proactive planning for persistent missions using composite model-reference adaptive control and approximate dynamic programming,” in American Control Conference (ACC), 2011, pp. 2332–2337.

[10] M. Valenti, B. Bethke, J. P. How, D. P. de Farias, and J. Vian, “Embedding health management into mission tasking for UAV teams,” in American Control Conference, (ACC). IEEE, 2007, pp. 5777–5783.

[11] M. Valenti, D. Dale, and J. How, “Mission health management for 24/7 persistent surveillance operations,” in AIAA Guidance, Navigation, and Control Conference, 2007.

[12] S. A. Motavalli, C. Yu, and B. D. Anderson, “Robustness to the loss of multiple nodes in the localizability of sensor networks,” in IFAC WC, 2011.

[13] T. H. Summers, C. Yu, and B. Anderson, “Addressing agent loss in vehicle formations and sensor networks,” International Journal of Robust and Nonlinear Control, vol. 19, no. 15, pp. 1673–1696, 2009.

[14] A. Y. Yazgooglu, M. Egerstedt, and J. S. Shamma, “Decentralized degree regularization for multi-agent networks,” in 52nd IEEE Conference on Decision and Control. IEEE, 2013.

[15] K. Akkaya, A. Thimmappuran, F. Senel, and S. Uludag, “Distributed recovery of agent failures in wireless sensor and actor networks,” in IEEE Wireless Communications and Networking Conference, WCNC 2008. IEEE, 2008, pp. 2480–2485.

[16] A. A. Abbasi, M. Younis, and U. Baroudi, “Recovering from a node failure in wireless sensor-actor networks with minimal topology changes,” IEEE Transactions on Vehicular Technology, vol. 62, no. 1, pp. 256–271, 2013.

[17] A. Petcu and B. Faltins, “Distributed generator maintenance scheduling,” in Proceedings of the First International ICSC Symposium on Artificial Intelligence in Energy Systems and Power: AESP 06, 2006.

[18] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, “Deployment and connectivity repair of a sensor net with a flying robot,” Experimental Robotics IX, pp. 333–343, 2006.

[19] M. Ji and M. Egerstedt, “Distributed coordination control of multi-agent systems while preserving connectedness,” IEEE Transactions on Robotics, vol. 23, no. 4, pp. 693–703, 2007.

[20] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, “Flocking while preserving network connectivity,” in 46th IEEE Conference on Decision and Control. IEEE, 2007, pp. 2919–2924.