New Selection Strategies of Actor's Substitute in DARA for Connectivity Restoration in WSANs

Riadh Saada(✉), Yoann Pigné, and Damien Olivier

Normandie Univ, UNIHAVRE, LITIS, 76600 Le Havre, France
{riadh.saada,yoann.pigne,damien.olivier}@univ-lehavre.fr

Abstract. Wireless Sensor and Actor Networks are used in many dangerous applications. When performing their tasks, actors may fail due to harsh environments in which they are deployed. Some actors are cut-vertices within network. Their loss breaks its connectivity and disrupts its operation accordingly. Therefore, restoring network connectivity is crucial. DARA is among the most popular connectivity restoration schemes. It performs multi-actor relocations in order to replace a failed cut-vertex by one of its neighbors, based on the lowest degree. In this paper, we propose new selection strategies of actor's substitute in DARA, in which the substitute selection is based on the nature of the links with neighbors rather than on the degree. Our approaches improve the performance of DARA by reducing the number of relocated actors in the recovery process by 24% on average compared to its original selection strategy. The proposed strategies are validated through simulation experiments.

Keywords: Topology management · Fault tolerance · Connectivity restoration · Wireless sensor and actor networks

1 Introduction

Wireless Sensor and Actor Networks (WSANs) [1] are used to replace or assist humans in many hazardous situations such as fire extinguishing and rescue of victims within hostile or unknown environments. A WSAN is composed of two categories of elements: sensors and actors. Sensors are usually small devices characterized by low cost and limited resources in computation, communication and energy. They are present within the network in abundance. The main duty of sensors is to probe their surroundings by collecting data about the supervised area and to report it to one or several actors, which react when necessary by performing appropriate tasks. Actors are more powerful devices. They could be mobile, they usually have advanced computation and communication capabilities, as well as a significant onboard energy, thus, they could be relatively more expensive. This is why their number within networks is fewer compared to sensors.

Sensors and actors can be subject to failure for many reasons. For instance, they can undergo external attacks from the fact that they are deployed unattended in harsh environments. They can also fail due to energy depletion or

Y. Zhou and T. Kunz (Eds.): ADHOCNETS 2016, LNICST 184, pp. 38–49, 2017.
DOI: 10.1007/978-3-319-51204-4_4

simply because of internal malfunctions [2]. Upon failing, an actor loses all its communication links. In most applications, actors must operate in a structured and collaborative manner for better efficiency in tasks' realization. Therefore, they need to interact with each other in order to share information and coordinate their actions. Thus, it is mandatory that they stay all the time reachable from each other. In other words, it is necessary to maintain a connected inter-actor network. An actor can be a cut-vertex in the inter-actor network topology. The failure of such actor splits the network into two or many disjoint partitions and affects its connectivity. The loss of network connectivity will have a detrimental impact on its performance. Indeed, actors belonging to different sub-networks will no longer be able to communicate, and thus, information exchange between them as well as their actions' coordination will be interrupted. As a result, the overall network operation will be severely disrupted.

To deal with this situation, the inter-actor network must integrate mechanisms of resilience, so that it continues to perform its tasks normally even if some actors fail. Many contemporary fault tolerance techniques in WSANs use topology management methods [3]. One of the most popular of them is DARA [4] (Distributed Actor Recovery Algorithm). DARA is a distributed connectivity restoration technique which performs coordinated multi-actor relocation in order to replace a failed actor by a healthy one into the inter-actor network. The connectivity restoration in DARA is thus a self-healing process, exploiting the mobility of operational actors within the network. The main objective of DARA is to reduce the number of involved actors in the recovery process. For this aim, a failed actor's substitute is chosen based on the lowest degree (least number of neighbors). In this paper, we propose new selection strategies of failed actor's substitute in DARA based on the nature of links with its neighbors rather than their number. Our selection schemes improve the performance of DARA in reducing the number of relocated actors during the recovery process by 24% on average, compared to its original selection strategy. We validated our approaches through simulation experiments.

The rest of paper is organized as follows: next section describes the system model. In Sect. 3, we review related works. An analysis of DARA is given in Sect. 4. In Sect. 5, we identify a shortcoming in DARA's selection strategy and propose solutions in order to remedy it. In Sect. 6, we evaluate our proposed approaches through simulation experiments. Section 7 concludes the paper.

2 System Modeling

As mentioned earlier, a WSAN is composed of two types of elements: sensors and actors. Sensors probe their surroundings and send regularly their sensed data to actors. On the other hand, actors collect information from sensors in their neighborhood and collaborate between them to perform one or several tasks. Figure 1 illustrates an example of WSAN. The system model includes the following assumptions:

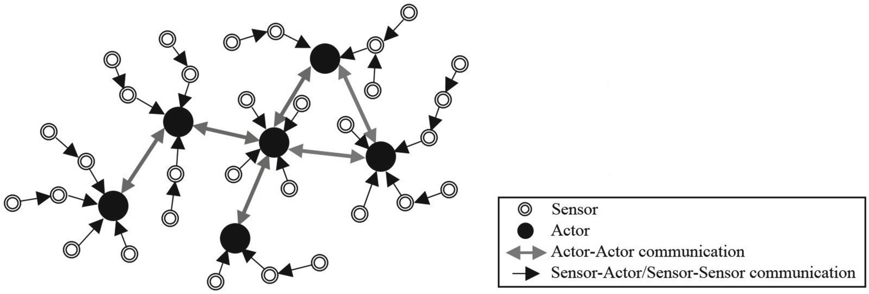

Fig. 1. Representation of a WSAN with a connected inter-actor network.

1. Each actor is identified by a unique identifier. A_i denotes the actor identified by identifier i. The set of all actors within the network is designated by $\mathcal{A}$.
2. Actors are mobile.
3. Actors are able to recognize their positions using localization techniques like GPS.
4. The communication range of an actor corresponds to the maximum Euclidean distance that its radio can reach. We denote it by r.
5. The radio range of all actors is identical, finite and significantly smaller than the dimensions of the deployment area.
6. Two actors can communicate if they are at range from each other.
7. Communications between actors are symmetric.
8. Actors are randomly deployed in an environment of interest and form a connected inter-actor network after a discovery step.

The following definitions are used in this paper:

Definition 1 (One-Hop Neighbors). *Let* $A_i \in \mathcal{A}$.
1-Hop-Neighbors(A_i*) or simply Neighbors(*A_i*) is the set of actors that are directly reachable from* A_i.

Definition 2 (Two-Hop Neighbors). *Let* $A_i \in \mathcal{A}$.
2-Hop-Neighbors(A_i*) is the set of actors that are reachable from* A_i *through* A_f*, where* $A_f \in$ *Neighbors(*A_i*).*

Definition 3 (Adjacent Siblings). *Let* A_i,$A_f \in \mathcal{A}$ *such as* $A_f \in$ *Neighbors(*A_i*).*
Adjacent-Siblings(A_i,A_f*) is the set of actors that are neighbors of both* A_i *and* A_f*. Mathematically: Adjacent-Siblings(*A_i,A_f*) =* $\{A_k$ — $A_k \in$ *Neighbors(*A_i*)* $\wedge$ $A_k \in$ *Neighbors(*A_f*)*$\}$

Definition 4 (Dependents). *Let* A_i,$A_f \in \mathcal{A}$ *such as* $A_f \in$ *Neighbors(*A_i*).*
Dependents(A_i,A_f*) is the set of actors that are neighbors of* A_i *(without* A_f*) but not neighbors of* A_f*. Mathematically: Dependents(*A_i,A_f*) =* $\{A_k$ — $A_k \in$ *Neighbors(*A_i*)* $\wedge$ $A_k \notin$ *Neighbors(*A_f*)* $\wedge$ $A_k \neq A_f$*)*$\}$*. It is easy to see that: Dependents(*A_i,A_f*) = Neighbors(*A_i*) − Adjacent-Siblings(*A_i,A_f*) −* $\{A_f\}$.

3 Related Works

Failure tolerance in WSANs is divided in two large categories: tolerating failure of one actor at a time and tolerating failure of multiple actors simultaneously. Our work lies within the first category. We focus on actors' failures because sensors are supposed available in abundance, as previously mentioned. When a cut-vertex actor fails, affecting network connectivity, the most obvious solution is to replace it by a new one. This solution can take a lot of time and can be dangerous if the operation is performed by humans within harsh environments. Contemporary connectivity-centric fault tolerance methods for WSANs are based on network topology management. In the literature, they are grouped into three classes: proactive techniques, reactive techniques and hybrid techniques.

Proactive techniques strive to anticipate the failure by taking some precautions at setup, allowing the network to continue operating normally in case of losing one or more actors. For example, the authors in [5] deploy a k-connected topology in which the failure of $k-1$ actors does not break the network connectivity.

Reactive techniques aim to perform network connectivity restoration as soon as a cut-vertex failure is detected, in real time, and in a distributed manner for most algorithms. The recovery process involves available healthy actors within the network. These actors being mobile, the idea is to reposition them to the appropriate locations in order to restructure the network's topology, so that it becomes connected again. Reactive techniques are more suitable than proactive ones for failure tolerance in WSANs. Indeed, WSANs are asynchronous by nature and may be dynamic. Therefore, the recovery process must be a network self-healing using adaptive schemes [3]. This category contains a wide variety of algorithms. DARA [4] is a reference algorithm within the research community working on this field. Many proposed approaches are compared against it for assessment. DARA performs a coordinated multi-actor relocation in order to replace a failed cut-vertex. To do this, it only requires to maintain an updated list of one-hop and two-hop neighbors, its mechanism is detailed in the next section. RIM [6] adopts another strategy: when an actor fails (cut-vertex or not), all its direct neighbors move toward its position in order to establish a connectivity between them. The process is repeated in cascade as long as the movement of neighbors causes further broken links. RIM shrinks the network topology inward, affecting network coverage. Compared to DARA, it involves much more actors in the recovery process and generates overhead messaging. However, its advantage is in splitting the load between actors. Indeed, it has been proven in [6] that each actor travels a maximum distance of $r/2$. VCR [7] and LDMR [8] are variants of RIM. Some algorithms consider a secondary objective in addition to the principal one that consists in restoring network connectivity. In return, they introduce additional assumptions or consume more resources. LeDiR [9] has as a secondary objective, the preservation of shortest-path length between any pair of actors using a shortest-path routing table. C^3R [10] has the auxiliary aim of maintaining network coverage: when a cut-vertex fails, its direct neighbors coordinate to establish a schedule in order to replace it in turns, during a time interval. Excessive movements of actors consume a lot of energy, therefore, this solution is considered temporary. RACE [11] is an interesting recent work based on DARA. It restores network connectivity while minimizing its coverage loss. For that, it needs additional information about actors' criticality, which is provided by the method developed in [12].

Hybrid techniques are a compromise between the two previous categories. They anticipate the failures by assigning backups to cut-vertices at setup in a proactive manner. However, the recovery process is triggered when a cut-vertex fails like reactive approaches. PADRA [13] and DCRS [14] operate this way.

For a complete state of the art on the subject, we recommend the reader a very interesting survey available in [3].

4 DARA Analysis

DARA [4] is a distributed connectivity restoration Algorithm. As a previous knowledge, it only requires that actors are aware of their one-hop and two-hop neighbors. For this, they have to maintain updated one-hop and two-hop tables. The tables must contain degrees, positions and identifiers (IDs) of neighbors.

Actors periodically report their presence to direct neighbors by sending heartbeat messages. When an actor fails and can no longer communicate, its direct neighbors detect the failure by missing the heartbeat messages. If the failed actor is a cut-vertex, DARA is launched locally on each of its neighbors.

The main idea of DARA is to replace a failed cut-vertex by an appropriate actor among its direct neighbors. When a cut-vertex actor fails, its direct neighbors can no longer communicate between them. However, they know each other thanks to their two-hop tables. These neighbors are all considered as potential candidates to replace the failed actor A_f. Nevertheless, they must elect the most suitable of them (the Best Candidate, BC) to restore the connectivity by moving to the position of A_f. The potential candidates have the same information on each of them, so, they will come to the same result. The selection strategy of BC in DARA is based on the following criteria:

1. *Least actor degree*: The authors assume that moving an actor having few neighbors minimizes the number of involved actors in the recovery process (we will show in the next section that this is not always true).
2. *Closest proximity to* A_f: in the case of actors which have the same degree, DARA favors the nearest one to A_f in order to minimize the traveled distance.
3. *Highest actor* ID: in the case of actors having the same degree and are equidistant to A_f, DARA decides between them by picking the one with biggest ID.

Figure 2a illustrates an inter-actor network in which a cut-vertex A_f fails, dividing the network topology in two disconnected subnetworks. A_1, A_2, A_3 and A_4 initiate the recovery process as soon as they detect the failure. They execute DARA simultaneously in order to determine BC. A_2 has the highest degree and is then excluded. A_1, A_3 and A_4 have the same degree. A_4 is the farthest among them from A_f. So, it is also excluded. A_1 and A_3 have the same degree and are equidistant to A_f. Thus, A_3 is selected as BC based on the highest ID and moves to the position of A_f as shown in Fig. 2b. Upon its arrival to destination, it broadcasts a RECOVERED message.

When BC leaves its position, it may cause another network partitioning as depicted in Fig. 2b. In this case, its neighbors must perform DARA again, exactly as if BC had failed, in order to replace it by the most suitable of them. The process is repeated recursively until full connectivity restoration. However, this time, the recovery process does not concern all the neighbors of BC. Indeed, BC can have two types of neighbors according to their previous relations with A_f: the siblings (actors in the set Adjacent-Siblings(BC,A_f), see Definition 3) and the dependents (actors in the set Dependents(BC,A_f), see Definition 4). Siblings preserve direct links with BC after relocation (for example A_2 in Fig. 2b). Dependants can either preserve indirect links with BC (such as A_8 in Fig. 2b) or find them detached from it (like A_9 in Fig. 2b).

Dependants can check whether they conserve indirect links with BC after relocation or not, as explained in the following. Before leaving its position, BC sends a MOVING message to all its dependent neighbors, in order to inform them about its departure. BC integrates also in this message, the list of its sibling with A_f. When a dependent receives the message, it verifies if it is related to one of BC's siblings or not, using its neighborhood tables. If such link exists, the dependent remains connected to BC after relocation, otherwise, it concludes that it is detached from it based on limited two-hop knowledge.

BC's neighbors concerned by the cascade are its detached dependents, so that they reestablish their broken links with it. In our previous example, A_9 moves to the position of A_3 and completes the recovery process, as shown in Fig. 2c. For more details about DARA's mechanism, please refer to article [4].

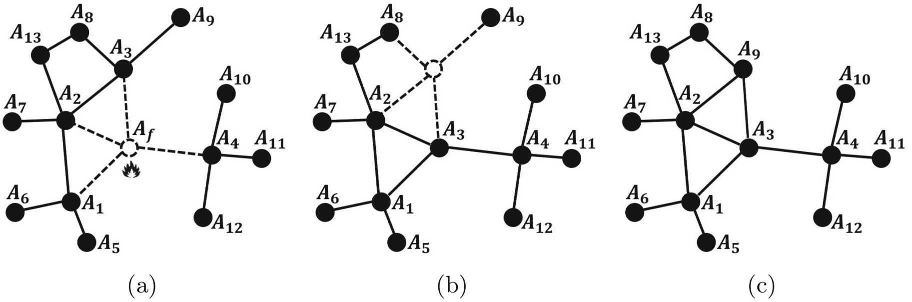

Fig. 2. An execution example of DARA. (a) represents a failure of A_f. In (b), A_3 replaces A_f. In (c), A_9 replaces A_3 and completes the recovery process.

5 Our Approach

The first selection criterion of A_f's substitute in DARA is based on the least degree of its neighbors. Abbasi et al. in [4] have motivated this choice by the

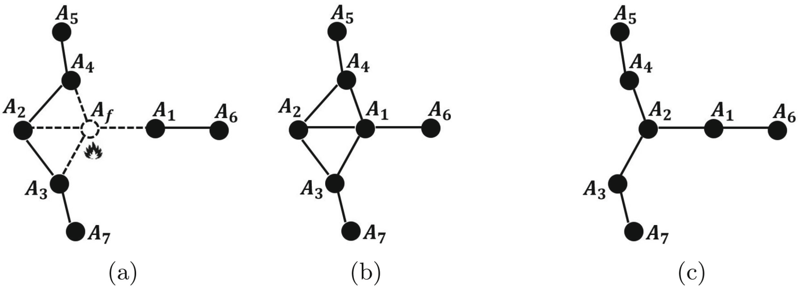

Fig. 3. An example in which DARA does not select the best substitute for A_f. (a) represents the failure of A_f. In (b), A_1 and A_6 move according to selection strategy of DARA. Alternatively in (c), the displacement of A_2 is sufficient to restore connectivity in (a).

fact that it limits the number of cascaded relocations in the recovery process. We will show through the next example that this is not always true.

In Fig. 3a, the cut-vertex A_f fails. In Fig. 3b, A_1 replaces it and A_6 replaces A_1 according to selection strategy of DARA. In Fig. 3c, the selection of A_2 as BC rather than A_1 limits the number of involved actors in recovery process, even if it has higher degree. Moreover, in our previous analysis of DARA, we have shown that neighbors of BC that may cause cascaded relocations are its dependents, when they are not related to any of its siblings with A_f. Based on these observations, we realized that it is not the candidates' degree that influences the cascades but the nature of links with their neighbors (dependency links). Indeed, in Fig. 3a, A_1 has one dependent while A_2 does not.

In order to reduce the number of involved actors in DARA, we propose three new selection strategies of a failed actor's substitute. Our selection strategies focus on the nature of links with neighbors rather than their number. They are presented in the following:

- **Strategy 1:** select the candidate with least number of dependents. This choice is motivated by the fact that dependents are the actors that may cause cascades as mentioned earlier. Therefore, we propose to minimize their number. In case of a tie, break it with distance and highest ID like in DARA.
- **Strategy 2:** select the candidate with highest number of siblings. We believe that the higher the number of siblings, the more likely dependents are related to one of them, and thus, stop relocations. In case of a tie, break it with distance and highest ID like in DARA.
- **Strategy 3:** is a compromise between the two previous ones. Here, we favor the candidate which number of dependents is null, if exists, because no cascade will be trigged in this case. Otherwise, we pick the candidate with highest number of siblings. In case of a tie, break it with distance and highest ID like in DARA.

To use our selection strategies, each potential candidate A_p needs to calculate the number of its dependent neighbors and the one of its siblings. It must also do the same for the other candidates which are its two-hop neighbors through A_f. For this purpose, we propose a distributed algorithm that does not require any other information than the one available in the tables of DARA. Our algorithm begins by counting the number of siblings. Then, it infers the number of dependents from formula (1), which is easy to demonstrate.

$$Number\,of\,Dependents \;=\; Degree - Number\,of\,Siblings \;-\; 1 \qquad (1)$$

Indication: *the neighbors' set of a candidate is composed of its siblings with* A_f*, its non siblings with* A_f *(its dependents) and* A_f*.*

The calculation of siblings' number by the potential candidate A_p is based on the four following rules:

- **Rule 1 (initialization):** A_p initializes its number of siblings as well as those of other candidates to zero.
- **Rule 2:** if A_p is neighbor of another candidate $A_{p'}$, it means that both are siblings with A_f. Therefore, A_p increases by 1, its number of siblings and the one of $A_{p'}$ (For example, A_2 and A_4 in Fig. 3a).
- **Rule 3:** if two candidates A_{p_1} and A_{p_2} other than A_p are neighbors, it means that both are siblings with A_f. If A_p is adjacent to at least one of them, it can infer the link between them by consulting its two-hop table. Thus, A_p increases by 1 the number of siblings of A_{p_1} and the one of A_{p_2} (for example, A_3 can detect the link between A_2 and A_4 in Fig. 3a).
- **Rule 4:** if two candidates A_{p_1} and A_{p_2} other than A_p are neighbors, it means that both are siblings with A_f. If A_p is not adjacent to any of them, it needs a three-hop vision to detect their link, which is not provided by its tables. However, A_p is aware of the positions of A_{p_1} and A_{p_2} through its two-hop table. Thus, A_p can verify whether they are neighbors or not by calculating the distance between them. If Distance(A_{p_1},A_{p_2}) $\leq r$ (r is the communication range) then A_{p_1} and A_{p_2} are neighbors. Therefore, A_p increases by 1 their respective siblings' numbers (For example, A_1 detects the link between A_2 and A_4 in Fig. 3a).

6 Validation

The effectiveness of our approaches is validated through simulations. In this section, we describe the simulation environment and discuss the obtained results.

6.1 Simulation Environment

To evaluate our approaches, we developed a simulation environment similar to that presented in the original article of DARA [4]. For this, we used the JAVA programming language and the GraphStream[1] library. We have randomly generated connected inter-actor networks with different density levels, in an area of 1000 m × 600 m. The test consists in varying the number of actors within the network topology from 20 to 100. Their communication range is fixed at 100 m. Each of these simulation steps is run for 1000 different network topologies in which we treat the failures of all cut-vertices independently. Average measures on these 1000 topologies are then reported. Our selection schemes are assessed and compared with the original version of DARA based on the following metrics:

(1) *Number of relocated actors:* it is the number of actors involved in the recovery process. This metric evaluates the extent of the connectivity restoration process.
(2) *Total traveled distance:* it is the distance traveled by all actors during the recovery process. This metric assesses the efficiency of the different connectivity restoration strategies. It is considered as an indicator of energy consumption and network reconstruction time.
(3) *Number of exchanged messages:* this metric counts all sent messages during the recovery process. It is also an indicator of energy consumption.

6.2 Discussion

Our simulation results are depicted in Figs. 4, 5 and 6. For the original version of DARA, we got the same curve shapes as those presented in the articles [4,6], which shows the consistency of our results compared to previous works. As in the articles [4,6], increasing the network density reduces the number of cascades in the recovery process, because high connectivity degree promotes the presence of siblings in the network's topology rather than dependents. We observe the same behavior in all our proposed approaches as shown in Fig. 4. Table 1 compares the performance of the different strategies with the original version of DARA.

As previously explained, the dependents are those that may cause cascades, so we aim in our first selection strategy to minimize their number. As expected, this choice reduces the number of relocated actors by 14% compared to the original version of DARA (see Table 1). It also decreases the total traveled distance during the recovery process by 21%, and the number of exchanged messages by 4%. Decreasing the total traveled distance in recovery process improves the network reconstruction time allowing it to resume operations more quickly. It also reduces the energy consumption and extends the network lifetime accordingly. Decreasing the number of exchanged messages allows also to save energy.

Our second strategy which aims to maximize the number of siblings in the selection of BC obtains better results than the first one in terms of actors' relocation (−17%) and total traveled distance (−26%, see Table 1). The reason is: maximizing the number of siblings increases the probability that dependents are related to one of them, and thus, avoid cascades. However, this strategy increases the number of exchanged messages by 87% compared to the original version of

---

[1] http://graphstream-project.org/.

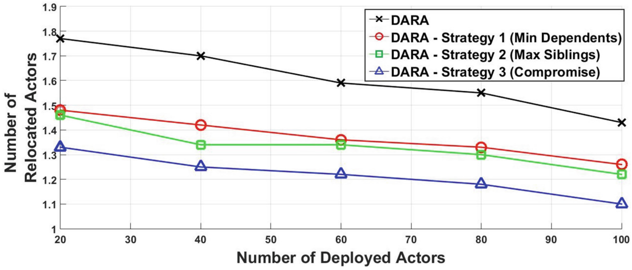

Fig. 4. Number of relocated actors with varying number of actors ($r = 100\,\text{m}$).

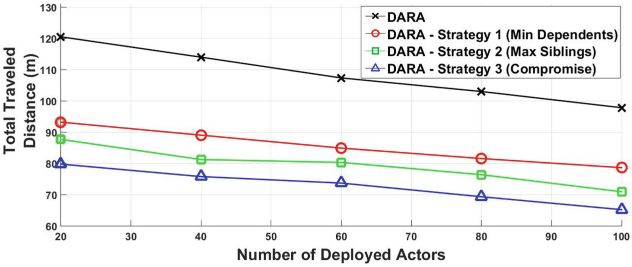

Fig. 5. Total traveled distance with varying number of actors ($r = 100\,\text{m}$).

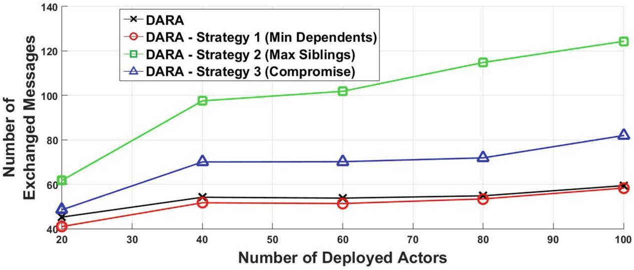

Fig. 6. Number of exchanged messages with varying number of actors ($r = 100\,\text{m}$).

Table 1. Performance analysis of our strategies compared to the original version of DARA.

Metrics	Strategy 1 (Min dependents)	Strategy 2 (Max siblings)	Strategy 3 (Compromise)
Number of relocated actors	−14.80%	−17.16%	−24.37%
Total traveled distance	−21.26%	−26.94%	−32.98%
Number of exchanged messages	−4.36%	+87.11%	+28.12%

DARA for the following reason: BC is selected based on the highest number of siblings. After displacement of BC, its siblings must send messages to their own neighbors in order to update their two-hop tables. Furthermore, the number of siblings increases accordingly with the network density, which increases the number of sent messages compared to the original version of DARA and the first selection strategy where BC is chosen based on a minimization criterion.

Our third and last selection strategy favors the candidate with no dependents, if exists. Otherwise, it maximizes the number of siblings in the selection of BC. Indeed, the second strategy only focuses on the number of siblings and ignores the case when a candidate has no dependents. Nevertheless, this case is very important because it stops the cascades and ends the recovery process immediately. This is what motivated the idea of the third selection strategy which outperforms all previous ones in terms of cascades limitation (-24%), as well as total traveled distance (-32%), while alleviating the number of messages introduced by the second strategy up to 28% more than the original version of DARA. We know that for relatively powerful actors, the cost of sending a message in terms of energy is negligible compared to the cost of their physical displacement. Therefore, this strategy can be a good compromise for energy preservation in WSANs.

7 Conclusion

DARA [4] is one of the most popular connectivity restoration schemes in WSANs which does some topology control in order to replace a failed cut-vertex by a healthy actor into the network. In this paper, we identified a shortcoming in its substitute selection strategy and proposed new ones that improve its performance. Our approaches was evaluated and compared with the original version of DARA through simulation experiments. As future works, we plan to apply our selection strategies on RACE [11] algorithm which relies on DARA's mechanism in order to restore network connectivity while minimizing its coverage loss. We believe that our approaches are able to improve its performance. We also plan to do real experimentations of our methods on mobile robot networks in collaboration with robotics researchers in our laboratory.

Acknowledgment. The project is co-financed by the European Union with the European regional development fund (ERDF) and by the Haute-Normandie Regional Council.

References

1. Akyildiz, I.F., Kasimoglu, I.H.: Wireless sensor and actor networks: research challenges. Ad hoc Netw. **2**(4), 351–367 (2004)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Comput. Netw. **38**(4), 393–422 (2002)

3. Younis, M., Senturk, I.F., Akkaya, K., Lee, S., Senel, F.: Topology management techniques for tolerating node failures in wireless sensor networks: a survey. Comput. Netw. **58**, 254–283 (2014)
4. Abbasi, A.A., Younis, M., Akkaya, K.: Movement-assisted connectivity restoration in wireless sensor and actor networks. IEEE Trans. Parallel Distrib. Syst. **20**(9), 1366–1379 (2009)
5. Han, X., Cao, X., Lloyd, E.L., Shen, C.C.: Fault-tolerant relay node placement in heterogeneous wireless sensor networks. IEEE Trans. Mob. Comput. **9**(5), 643–656 (2010)
6. Younis, M.F., Lee, S., Abbasi, A.A.: A localized algorithm for restoring internode connectivity in networks of moveable sensors. IEEE Trans. Comput. **59**(12), 1669–1682 (2010)
7. Imran, M., Younis, M., Said, A.M., Hasbullah, H.: Volunteer-instigated connectivity restoration algorithm for wireless sensor and actor networks. In: 2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS), pp. 679–683. IEEE (2010)
8. Alfadhly, A., Baroudi, U., Younis, M.: Least distance movement recovery approach for large scale wireless sensor and actor networks. In: 2011 7th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 2058–2063. IEEE (2011)
9. Abbasi, A.A., Younis, M.F., Baroudi, U.A.: Recovering from a node failure in wireless sensor-actor networks with minimal topology changes. IEEE Trans. Veh. Technol. **62**(1), 256–271 (2013)
10. Tamboli, N., Younis, M.: Coverage-aware connectivity restoration in mobile sensor networks. J. Netw. Comput. Appl. **33**(4), 363–374 (2010)
11. Haider, N., Imran, M., Younis, M., Saad, N., Guizani, M.: A novel mechanism for restoring actor connected coverage in wireless sensor and actor networks. In: 2015 IEEE International Conference on Communications (ICC), pp. 6383–6388. IEEE (2015)
12. Imran, M., Alnuem, M.A., Fayed, M.S., Alamri, A.: Localized algorithm for segregation of critical/non-critical nodes in mobile ad hoc and sensor networks. Procedia Comput. Sci. **19**, 1167–1172 (2013)
13. Akkaya, K., Senel, F., Thimmapuram, A., Uludag, S.: Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility. IEEE Trans. Comput. **59**(2), 258–271 (2010)
14. Guizhen, M., Yang, Y., Xuesong, Q., Zhipeng, G., He, L., Xiangyue, X.: Distributed connectivity restoration strategy for movable sensor networks. Chin. Commun. **11**(13), 156–163 (2014)