Semantic Clustering in Wireless Sensor Networks

Atslands Rego Rocha, Igor Leão Santos, Luci Pirmez, Flávia C. Delicato, Danielo G. Gomes, José Neuman Souza

Semantic Clustering in Wireless Sensor Networks

Atslands Rego da Rocha[1] , Igor Leão dos Santos[2], Luci Pirmez[2], Flávia C. Delicato[3], Danielo G. Gomes[1] and José Neuman de Souza[1]

[1] Federal University of Ceará - Brazil, [2] Federal University of Rio de Janeiro - Brazil,
[3] Federal University of Rio Grande do Norte - Brazil,
{atslands, danielo, neuman}@ufc.br, igorlsantos@poli.ufrj.br, luci@nce.ufrj.br,
flavia.delicato@dimap.ufrn.br

Abstract. Wireless Sensor Networks have critical resource constraints and minimizing resources usage is crucial to extend the network lifetime. Energy saving in WSNs can be achieved through several techniques, such as topology control and clustering, to provide a longer lifetime and scalability to the network. In this paper we propose a semantic clustering model based on a fuzzy inference system to find out the semantic neighborhood relationships in the network. As a case study we describe the structural health monitoring domain application which has been used to illustrate and verify the proposed model.

Keywords: Wireless sensor networks, semantic clustering, fuzzy logic system.

1 Introduction

A typical wireless sensor network (WSN) consists of spatially distributed nodes with sensing, processing, storing and communicating capabilities that cooperatively monitor environmental conditions. Due to critical resource constraints of the sensor networks it is important to design techniques which minimize their resources usage and consequently extend their lifetime.

Clustering has been used in ad-hoc networks and WSNs as an effective technique for achieving extended network lifetime and scalability [1]. The general idea is to perform the cluster formation based on the received signal strength, and to use local cluster heads (CHs) as routers of data gathered by the sensors in their clusters towards the sink node. Several works on WSNs have been developed in the context of clustering [2]. However, although there is often some correlation between sensor nodes grouped in a same cluster, a semantic correlation is not frequently exploited.

In this paper we propose a semantic clustering for heterogeneous WSNs in order to minimize the communication resource usage and energy cost. Our proposal is based on the computation of semantic neighborhoods relationships by finding correlations between information from sensor nodes. Sometimes neighbor nodes sense areas that are not related at all, or neighbor nodes provide measurements that are not correlated. For example, in the airport security applications [3], sensor nodes both do video and audio processing and communicate with their neighbors nodes in order to share a global view of the sensoring environment. However, nodes fixed on different sides of the same wall observe different areas of environment that are not related at all. Therefore, since closely located sensors generated data that are not semantically correlated, they should not exchange information, since this would be a waste of network resources.

In our work, the set of sensor nodes which are semantically correlated among each other is grouped into *semantic clusters*. The semantic clustering is a service provided by a semantic middleware for WSNs described in our previous work [4]. A fuzzy system is responsible to establish the relationships of the semantic neighborhoods. Fuzzy inference systems match two of the most challenging requirements [5] of WSNs: (i) they are simple and can be executed on limited hardware and (ii) they accommodate imprecise data. The ability of handling imprecise data is desirable since individual data from sensor nodes often are inaccurate due to calibration problems, environmental noise, wireless transmission loss, faulty sensors among other itens.

In this work, our proposal is applied in the Structural Health Monitoring (SHM) domain. Extending the network lifetime is crucial to enable SHM systems to perform a high density sensing of the monitored environment, so that sampled data can be reliably analyzed.

2 Related Works

There exist some approaches in WSN that address semantic clustering techniques or structural monitoring issues, but not both in conjunction. Bouhafs et al. [6] propose a semantic clustering algorithm for energy-efficient routing in WSNs that allows data aggregation. Our work differs from it since the neighborhood relationships are inferred by using fuzzy logic in the CHs.

Ulieru and Madani [7] present a bridge monitoring application of WSNs using both agent technology and ubiquitous computing. The authors use an optimization strategy inspired in ant colonies to control the network topology based on context information. In contrast, we use semantic clustering in order to extend the lifetime of network. Moreover, our method allows reclustering when environmental changes.

Bruckner et al. [3] show how a network of smart sensor nodes can be established by using high-level semantic knowledge that is gathered by loopy belief propagation between sensors with overlapping sensing areas. The semantic neighborhood relationships are inferred by smart nodes using statistical analysis of the shared environment. The authors do not use clusterization at all.

Perianu et al. [1] present a method for spontaneous clustering of mobile wireless sensor nodes based on a common context, namely Tandem, which allows reclustering in case of topological or contextual changes. Our method allows saving communication and energy resources by (i) turning on/off sensor nodes which are not involved in monitored variables and (ii) controlling messages are sent between semantic neighbors and sink nodes.

3 Methodology

In our work, there are two phases of clustering: a physical clustering and a semantic clustering. At the network start up process, a physical clustering of sensor nodes is done. The physical organization is hierarchical and consists of two levels. The upper level is composed of CHs that do not perform any sensing tasks, but perform both processing on data received by the sensors and inter-cluster communication. The lower level is composed of sensors that are responsible for collecting the data and are subordinated to one of the CHs. For the physical clustering phase, algorithms such as LEACH [8], among other protocols [2], can be used.

The semantic organization[1] is also hierarchical and consists of two levels. The upper level is composed of semantic collectors. We define a semantic collector as a node that is in charge of making a report containing all data received by the sensors that are semantically correlated and sending this report to the sink node. The lower level is composed of sensor nodes that are semantically correlated to each others and are subordinated to one of the semantic collectors.

We designed a methodology (Fig. 1) to perform the semantic clustering phase that is performed after the physical clustering previously explained. Our proposed methodology is applied in three steps: (i)creating Low Level Symbols (LLSs) for each sensor input; (ii) performing fuzzy system to calculate the semantic neighborhood relationships of the network; (iii) (re)grouping in semantic clusters the set of sensor nodes which are semantically correlated to each others. Step 1 is processed locally in each sensor node, while Steps 2 to 3 are processed locally in each CH.

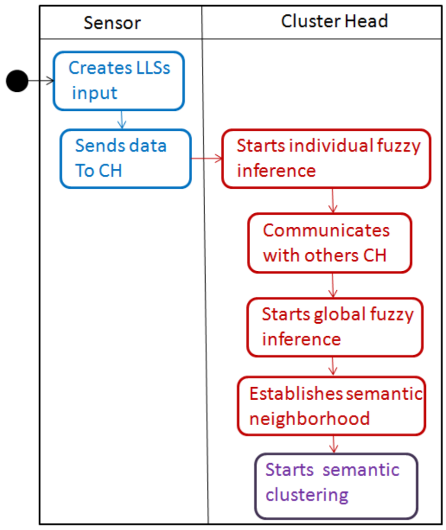

Fig. 1. Methodology (UML Diagram).

[1] The concept *cluster head* do not exist in the semantic organization. Thus, whenever we refer to the concept *cluster head* in this document, we are referring to the physical clustering.

In the first step, each sensor node creates LLS to represent mono-modal symbols, as in [3]. A sensor input variable, such as acceleration, temperature, humidity is considered a mono-modal symbol. The several achieved LLSs are represented in XML format. Thus, the LLSs specify the variables which can be analyzed in the environment in order to support a decision making process. For example, in a SHM domain, a semantic correlation can be performed between LLSs such as acceleration, temperature and stress in order to detect possible damages in the civil structure. In the next step, a fuzzy system is responsible to establish semantic neighborhoods relationships by finding correlations between information from sensor nodes. The fuzzy inference system is divided in two phases: (i) an individual fuzzy inference process is performed in each CH considering only the sensor nodes correlation that belong to its cluster and (ii) a global fuzzy inference process is performed in each CH considering the neighbors CHs "opinion" about their sensor nodes correlation. The information correlation is gathered when the CHs communicate each other. In the last step, the set of sensor nodes which are semantically correlated to each others is (re)grouped in semantic clusters.

4 Semantic Neighborhood Process

We use a rule-based distributed fuzzy inference system for WSN similar to the work described in [5] to establish the semantic neighborhood relationships. Briefly, a fuzzy logic system is composed of four basic components: fuzzifier interface, knowledge base, inference process and defuzzifier interface. The fuzzifier maps crisp inputs into fuzzy sets by using the membership functions. A membership function gives the degree of similarity of a crisp input to the fuzzy set. Next, the fuzzified values activate the rules of knowledge base that are provided by experts or extracted from numerical data. Next, the fuzzy inference process combines the rules to obtain an aggregated fuzzy output. Finally, the defuzzifier interface maps the fuzzy output back to a crisp number that can be used for making decisions. The fuzzy inference system uses two types of inputs: individual observations of the sensor nodes (individual fuzzy inference) and neighborhood observations (global fuzzy inference).

The individual fuzzy inference process is explained as following. The fuzzy logic system starts whenever a CH notices that one or more sensors inside its cluster are "candidates" to become semantic neighbors. The sensors are considered "candidates" to become semantic neighbors when their data satisfy a domain rule related to the event monitored by the WSN. For example, in the SHM domain, if there is any relevant change in the modal frequency values of a sensor node, the sensor node will be considered a "candidate" to become a semantic neighbor. In this example, the domain rule is "acceleration is higher than a specified threshold”. The fuzzifier utilizes as input data the aggregated data (crisp input) of the sensors that satisfy a domain rule. The fuzzifier can also utilize as input data some sensor's data that were processed by the CH. The fuzzifier maps crisp inputs into fuzzy sets by using the trapezoidal membership functions. The most common membership functions are trapezoidal and triangular. Triangular membership functions are used whenever there is a single element that has the pertinency value equal to 1 (one). The trapezoidal membership functions are used if there are several (> 1) elements that have the pertinency value equal to 1. We use the max-min inference since it proves to be computationally fast in the system implementation [5]. Each CH stores their fuzzy inference results. The CHs communicate to each others (one-hop) by sending and receiving messages containing their individual fuzzified results about semantic neighborhood relationships inside their respective clusters. Every CH which receives these messages from neighbor CHs performs the fuzzy inference using the neighbors fuzzified observations (global fuzzy inference). It is used a sigma-count factor [9], a measure of fuzzy cardinality that allows to generalize counting techniques, in the quantification of neighborhood observations:

$$\sum \mathrm{Count}(F) = \sum_i \mu F(xi)\,. \tag{1}$$

where a fuzzy set F is characterized by a membership function $\mu F(x_i)$ which gives the degree of similarity of x to F. F is a property of interest related to the sensor nodes observations, e.g. "acceleration value is high" or "humidity level is low". Finally, X = {x1, ..., xn} is the set of neighbors.

A fuzzy majority quantifier is used to get a fuzzified indication of the consensual neighborhood opinion. A fuzzy majority is defined as a fuzzy linguistic quantifier. In our case, we use the *most* [10] quantifier to characterize the fuzzy majority, i.e., the consensual neighborhood opinion in order to take a more accurate decision:

$$\mu_{most}\left(\frac{\sum \mathrm{Count}(F)}{|X|}\right) = \mu_{most}\left(\frac{\sum_i \mu F(xi)}{n}\right). \tag{2}$$

$$\mu_{most}(x) = \begin{cases} 0 & \text{if } x \leq 0.3\,; \\ 2x - 0.6 & \text{if } 0.3 < x < 0.8\,; \\ 1 & \text{if } x \geq 0.8\,. \end{cases}$$

Next, the fuzzified values activate the inference rules. The fuzzy inference system incorporates both the fuzzified individual observations and the quantified neighborhood observations in the rules. For example, concerning a SHM domain:
IF acceleration is High AND stress/strain is High AND
most(accelerationtNeigh) is High AND most(stress/strainNeigh) is High
THEN SemanticNeighborDecision is High

The fuzzy inference system combines the rules to obtain an aggregated fuzzy output. Finally, the defuzzifier maps the fuzzy output back to a crisp number which it is used for making decisions related to the semantic neighborhood. To reduce the computational complexity, we use a simple defuzzification method denoted as *maximum* method in order to produce a crisp output.

5 Semantic Clustering

The following steps illustrate an overview of the proposed method:

1) A new semantic cluster may be composed of the semantically correlated sensor nodes which are identified in the messages sent by CHs, although those sensor nodes may be either in the same physical cluster or not.
2) In a given neighborhood, the CH that has the highest number of semantic neighbors in its cluster is elected as a semantic collector.
3) The semantic collector is responsible for sending the reports to the sink nodes. The reports may contain both the agregated values of the semantic neighbors, the semantic collectors IDs associated to the detected damage and the semantic neighbors IDs.

6 Case Study: Structural Damage Detection, Location and Extent Estimation

The main goal of a SHM application is to assess the integrity of the monitored structures performing damage detection, localization and extent estimation tasks. Our proposed method is based on the Sensor-SHM algorithm [11]. Sensor-SHM is a distributed algorithm for WSNs which performs damage detection, localization and extent determination in civil structures, making use of the shifts in the structure's modal frequencies. Sensor-SHM is described as following.

6.1 Sensor-SHM

There is a cluster formation[2] at Sensor-SHM algorithm start up process. The CHs are determined and each CH is aware of its CH neighbors. CHs do not perform sensing tasks, and are at a higher level in the network. The lower level is composed of sensors responsible for collecting the signatures of the structure and are subordinated to one of the CHs. Sensor nodes are identified by the index *i*, in the lower level of the network, while CHs are identified by the index *j*, from the higher level of the network.

The presence of damage in a structure may cause shifts in all of its modal frequencies, at a given sensor node location. Then, the perceived change depends on the position of the sensor node, if it is close to the damage site or not. Briefly, the Sensor-SHM algorithm is explained as following:

1) Each sensor node individually collects acceleration values from its position on the civil structure in the time domain. A Fast Fourier Transform (FFT) is then performed on the collected acceleration measurements and a simple method is used in order to extract the modal frequencies shown in the frequency spectrum generated by the FFT algorithm. A vector $\varpi_{i,t}$ (3) of M extracted modal frequencies represents the signature of the structure, considering a sensor *i* in the data collection stage t. Thus, each sensor node *i* obtains a $\varpi_{i,t}$ vector in each data

[2] Semantic clustering is not used in the original Sensor-SHM algorithm.

collection stage *t*. Every sensor node sends to its CH the $\varpi_{i,t}$ vector at each sensing stage *t*.

$$\overrightarrow{\omega_{i,t}} = \begin{bmatrix} \omega_{i,t}^{1} \\ \vdots \\ \omega_{i,t}^{M} \end{bmatrix}. \tag{3}$$

2) Each CH analyzes incoming signature vectors from each sensor node contained in its cluster, in order to notice any modal frequency change. Each CH compares the modal frequency vectors $\varpi_{i,t}$ sent at the current data collection stage by the sensor nodes with a modal frequency vector $\varpi_{i,0}$ sent by each sensor *i* in the first data collection stage, containing the initial signature of the structure for each location. Each $\varpi_{i,0}$ means a signature from a healthy state of the civil structure, which means it has no damage or undesired perturbations. The comparison is made through the absolute value of the vector $\Delta\,\varpi_{i,t}$, which stores the subtraction of the actual values with the ones from the undamaged state, for each mode of vibration. The initial signature of the structure of each sensor node is sent to the CH by the sensor nodes. If there is a relevant change in the modal frequency values in $\varpi_{i,t}$ related to those of $\varpi_{i,0}$, considering a certain amount of tolerance threshold specified at the network start up process, the algorithm assumes the possibility of damage presence in the monitored structure. The tolerance threshold vector $\overrightarrow{T_i}$ is defined for each sensor node. The threshold values depend on the sensor node position on the civil structure and are defined by a structure specialist.
3) A $D_{i,t}$ coefficient is calculated for each sensor node *i*. For the first five modal frequencies, see the $D_{i,t}$ formal definition:

$$D_{i,t} = \overrightarrow{A_i}\ \overrightarrow{\Delta\omega_{i,t}} = \left[A_i^1\, A_i^2\, A_i^3\, A_i^4\, A_i^5\right] \begin{bmatrix} \Delta\omega_{i,t}^{1} \\ \Delta\omega_{i,t}^{2} \\ \Delta\omega_{i,t}^{3} \\ \Delta\omega_{i,t}^{4} \\ \Delta\omega_{i,t}^{5} \end{bmatrix}. \tag{4}$$

The $D_{i,t}$ coefficient value means how close a sensor node *i* is to the detected damage site. The $\overrightarrow{A_i}$ vector is composed by weights associated to each modal frequency shift. The sink node informs the A_i values to the CHs at the network start up. However, the A_i values can be modified during the operation of the network. Since changes in the higher modal frequencies mean there are changes in the local modes of vibration, the $\overrightarrow{A_i}$ vector is responsible for identifying the sensor nodes which are closest to the damage position, once higher weight values should be set to the higher modes of vibration. Therefore, in the network startup process, the highest weight values are associated with the highest modes of vibration and are stored in the CH of each sensor node *i*. Thus, the sensor nodes contained in a cluster formation which is closest to the damage position have the highest $D_{i,t}$ coefficients of the whole network.

4) A $C_{j,t}$ coefficient is performed through the sum of all $D_{i,t}$ coefficients for all *k* sensor nodes contained in a cluster *j* of the network:

$$C_{j,t} = \sum_{i=1}^{k} D_{i,t} \, . \qquad \textbf{(5)}$$

The $C_{j,t}$ coefficient means how close to the detected damage site the cluster is, as a whole.

5) In the network startup process, a L_j threshold value is set in the CHs, which depends on the structure local attributes where each cluster formation is installed. Whenever $C_{j,t} \geq L_j$, the CH sends to its neighbor CHs (one-hop) a message including its $C_{j,t}$ coefficient value. This action avoids a large number of false damage detection notifications, because in a real damage detection situation it is expected that some neighbor CHs also detect the same event and also send their $C_{j,t}$ coefficients to the neighbor CHs.
6) Every CH which receives the $C_{j,t}$ coefficients from neighbor CHs compares those values to its $C_{j,t}$ coefficient. In a given neighborhood area, the CH which has the highest $C_{j,t}$ coefficient value is considered a collector CH. The collector CH is responsible for making a report containing all $\varpi_{i,t}$ values of the neighbor CHs and send it to the sink node, emitting an alert notification to the engineers.

Therefore, the damage localization and extent determination are determined by the sensor positions which its CH has $C_{j,t} \geq L_j$. In multiple damage detection case, or in case of a large area of damage situation, the occurrence of multiple collector CHs is expected. These collectors may send multiple reports from different positions in the structure to the sink node.

6.2. Semantic Clustering in SHM

Sensor-SHM algorithm uses the cluster position in the structure in order to localize and estimate the extension of detected damage. However, depending on the sensors position in a cluster *j*, some sensors can be closer to the detected damage site than other sensors in the same cluster. Such fact can impact the $C_{j,t}$ coefficient used by the Sensor-SHM algorithm since this coefficient is composed of the $D_{i,t}$ values of the sensor nodes contained in a cluster *j*. And as we previously mentioned, the $D_{i,t}$ coefficient denotes how close a sensor node *i* is of the detected damage site. So, that fact consequently can affect the Sensor-SHM method results related to this aspect.

We applied our proposed clustering in the SHM domain, using the Sensor-SHM algorithm as a base and our proposal as an enhancement to this algorithm aiming at achieving best performances regarding the energy consumption thus extending the WSN lifetime. Whenever a damage is detected, the network organization is modified based on semantic correlation between the sensors. The sensors which are closest to the damage site are denoted semantic neighbors. Such sensors are easily identified since they have the highest $D_{i,t}$ coefficient values of the network. However those sensor nodes may be in the same physical cluster or not (considering the physical clustering).

A damage detection in a civil structure is illustrated in Figure 2. Using the Sensor-SHM algorithm, possibly the clusters C, D, E and F have the highest $C_{j,t} \geq L_j$ coefficients values in the network, since they are close to the detected damage. Sensor-SHM determines the damage area (represented by blue dashed lines) by the clusters positions which $C_{j,t} \geq L_j$. However, we can observe that in the clusters C, D, E and F, some sensors are closer to the damage site than other sensor nodes. In this same example, using our clustering method, the semantic neighborhood is composed of sensors node with ID 13, 14, 17, 18 ,21, 25 and 30 that are grouped in a semantic cluster. The damage area is only determined by the semantic cluster position in the structure (represented by yellow dashed lines). Thus, considering the crosslayer nature of the WSNs systems, using both Sensor-SHM and our proposed semantic clustering method we can improve the precision of the damage localization and extent estimation tasks. Briefly, our method works as following:

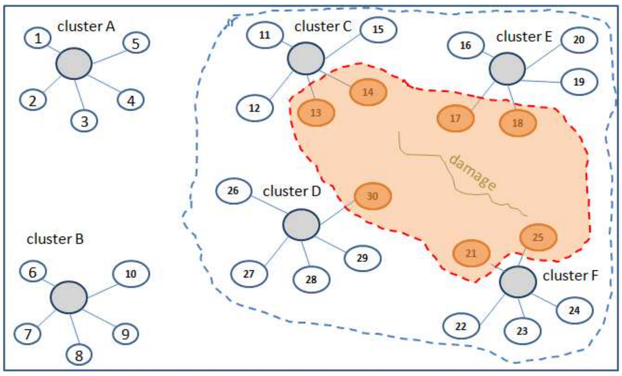

Fig. 2. Semantic Clustering.

1) At the network start up process, a physical clustering scheme (explained in section III) is performed. Using the Sensor-SHM algorithm for damage detection, each sensor node *i* obtains vectors $\varpi_{i,t}$ (3) in each data collection stage *t* and sends them to its CH. If the CH notices a relevant modification in the frequency vectors $\varpi_{i,t}$, it assumes the possibility of a damage presence in the observed structure. A $D_{i,t}$ coefficient (4) is also calculated for each sensor node *i* contained in a physical cluster *j* which presents a relevant modification in the frequency vectors.

2) Next, our fuzzy system starts whenever a CH notices that one or more sensors inside its cluster are "candidates" to become semantic neighbors because they satisfy the domain rule:

$$\overrightarrow{\Delta\omega_{i,t}} \geq \overrightarrow{T_i} = \begin{bmatrix} \Delta\omega_{i,t}^{1} \\ \Delta\omega_{i,t}^{2} \\ \Delta\omega_{i,t}^{3} \\ \Delta\omega_{i,t}^{4} \\ \Delta\omega_{i,t}^{5} \end{bmatrix} \geq \begin{bmatrix} T_i^1 \\ T_i^2 \\ T_i^3 \\ T_i^4 \\ T_i^5 \end{bmatrix}. \tag{6}$$

3) In the individual fuzzy inference process, a partial set of semantic neighbors is selected in each physical cluster. The bases for the fuzzy logic system are built from data generated in extensive simulations. One linguistic variable was labeled as *coefficient* (Fig. 3), representing the $D_{i,t}$ coefficient of sensor nodes. For this linguistic variable, two labels are defined as Low (L) and High (H) in order to represent the two simulated coefficients. For the coefficient variable, the universe of discourse was defined considering the closed interval of real numbers between zero and the largest simulated value for this variable. Regarding the membership function, that determines the shapes that represent each fuzzy set, the trapezoidal shape was chosen for this variable. Similar to [12], the boundaries of the shape are defined using a adopted confidence interval (95%). The value that represents the adopted confidence interval is used for each fuzzy set on the left side (lower boundary) and the smallest simulated value whose membership degree in the next fuzzy set is equal to 1, on the right side (upper boundary). In the start up process of the network, a Ld_j threshold value (Fig. 3) is informed to the CHs, which depends on the structure local attributes where each sensor node is installed. The Ld_j threshold is specified to determine if a *sensor node* is close to the damage site or not.

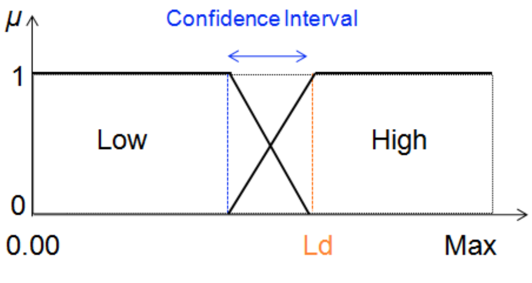

Fig 3. Membership Functions for the Coefficient Variable.

4) The CH sends to its neighbor CHs a message including both the semantic neighbors IDs into a physical cluster *j* and the aggregated data related to the $D_{i,t}$ coefficients of each sematic neighbor. Every CH which receives these messages from neighbor CHs updates the fuzzy inference using the neighbors observations (global fuzzy inference). A fuzzy rule is also defined:

   IF coefficient is High AND nmessages is High AND
   most(coefficientNeigh) is High AND most(nmessagesNeigh) is High
   THEN SemanticNeighborDecision is High

   where *nmessages* is a linguistic variable that represents the number of received messages by the CHs. This variable avoids a large number of false damage detection notifications, because it is expected that some neighbor CHs also detect the same event.

5) The semantically uncorrelated sensor nodes should not exchange information since that would waste resources of the network. So, uncorrelated nodes can be temporarily turned off.

6) A new semantic cluster is composed of all detected semantic neighbors and the semantic collector election is performed.The semantic collector might calculate the $CS_{j,t}$ coefficient value:

$$CS_{j,t} = \sum_{i=1}^{k} D_{i,t}\,. \tag{7}$$

$CS_{j,t}$ is based on $D_{i,t}$ coefficient values of the detected semantic neighbors and it denotes how close the semantic cluster is to the detected damage site. The semantic collector may also send the reports to the sink node including the $\varpi_{i,t}$ values of the semantic neighbors.

Whenever a new damage is detected, both a new election of the semantic neighbors and a new semantic clustering are perfomerd. In cases of multiple damages or large damages, it is possible the occurrence of multiple semantic collectors which can send to the sink node multiple reports from the different positions in the structure. The sink node has a time history of the $\varpi_{i,t}$ values of all semantic neighbors.

7 Conclusion

We propose a semantic clustering for WSNs. One important benefit of our proposal is to allow the network self-organize according to semantic correlation between sensor nodes. Another important benefit is to reduce the number of sensors which are monitoring the environment and consequently save resources such as processing, communication and energy in order to extend the lifetime of the network.

We applied the proposed semantic clustering algorithm in a SHM study case. As future work, it will be interesting to evaluate the semantic clustering algorithm for all detected semantic neighbors of the network and to investigate parameters such as the node degree, transmission power, battery level, processor load, i.e. metrics for choosing the optimal clustering structure as described in [1]. Moreover, we intend in a subsequent work to further explore the process of clustering formation in order to improve the WSN efficiency and performance. If the cluster is large, there is a large overhead due to control messages and more energy consumption. On the other hand, small cluster formations increase the spatial granularization due to the growth in the number of CH nodes. Accordingly, small clusters reduce the amount of collecting points because the CHs do not gather data from the environment. It will be interesting to consider trade-off between CH size, overhead, and energy saving.

Acknowledgments. This work is partly supported by the CNPq through processes 477226/2007-8 for Luci Pirmez and Flávia C. Delicato; 477229/2009-3 and 306938/2008-1 for Flávia C. Delicato; 481638/2007 for Neuman Souza, Danielo G. Gomes, Luci Pirmez and Flávia C. Delicato. This work is also partly supported by the FINEP through process 1979/09 for Luci Pirmez.

References

1. Marin-Perianu, R.S., Lombriser, C., Havinga, P. J. M., Scholten, J., Troster, G.: Tandem: A context-aware method for spontaneous clustering of dynamic wireless sensor nodes. In: International Conference for Industry and Academia, Springer, (2008).
2. Abbasi, A. A., Younis, M.: A survey on clustering algorithms for wireless sensor networks. Computer Communications 30, 2826–2841, (2007).
3. Bruckner, D., Zucker, G., Simo, J., Herzner, W., Mahlknecht, S.: Semantic neighborhood sensor network for smart surveillance applications. In: 3rd International Conference from Scientific Computing to Computational Engineering, (2008).
4. Rocha, A., Delicato, F., Souza, N., Gomes, D., Pirmez, L.: A semantic middleware for autonomic wireless sensor networks. In: Workshop on Middleware for Ubiquitous and Pervasive Systems, vol. 389. ACM, (2009).
5. Marin-Perianu, M., Havinga, P.: D-fler: A distributed fuzzy logic engine for rule-based wireless sensor networks. In: 4th International Symposium on Ubiquitous Computing Systems. Springer Verlag, (2007).
6. Bouhafs, F. Merabti, M., Mokhtar, H.: A semantic clustering routing protocol for wireless sensor networks. In: Consumer Communications and Networking Conference, pp. 351–355, IEEE Computer Society, (2006).
7. Ulieru, M., Madani, S. A.: An application of industrial agents to concrete bridge monitoring. In: ICINCO-ICSO, (2006).
8. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications, pp. 660–670, (2002).
9. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computers and Mathematics 9, 149–184, (1983).
10. Kacprzyk, J.: Group decision making with a fuzzy linguistic majority. Fuzzy Sets and Systems 18, 105–118, (1986).
11. Santos, I. ; Pirmez, L. ; Lemos, E. ; Vaz, L. ; Delicato, F. C. ; Souza, J. N.: Resource Consumption Analysis for a Structural Health Monitoring Algorithm Using Wireless Sensor Networks. In: 28º Simpósio Brasileiro de Redes de Computadores e de Sistemas Distribuídos, (2010).
12. Pirmez, L., Delicato, F., Pires, P., Mostardinha, F., Rezende, N.: Applying fuzzy logic for decision-making on wireless sensor networks. In: Fuzzy Systems Conference. (2007).