

Redundancy reduction in wireless sensor networks via centrality metrics

Citation for published version (APA):

Mocanu, D. C., Torres Vega, M., & Liotta, A. (2015). Redundancy reduction in wireless sensor networks via centrality metrics. In P. Cui, C. Aggarwal, Z.-H. Zhou, A. Tuzhilin, H. Xiong, & X. Wu (Eds.), *Proceedings 15th IEEE International Conference on Data Mining Workshop : 14-17 November 2015, Atlantic City, New Jersey* (pp. 501-507). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/ICDMW.2015.53>

DOI:

[10.1109/ICDMW.2015.53](https://doi.org/10.1109/ICDMW.2015.53)

Document status and date:

Published: 01/01/2015

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Redundancy reduction in Wireless Sensor Networks via Centrality Metrics

Decebal Constantin Mocanu
Department of Electrical Engineering
Eindhoven University of Technology
Eindhoven, the Netherlands
Email: d.c.mocanu@tue.nl

Maria Torres Vega
Department of Electrical Engineering
Eindhoven University of Technology
Eindhoven, the Netherlands
Email: m.torres.vega@tue.nl

Antonio Liotta
Department of Electrical Engineering
Eindhoven University of Technology
Eindhoven, the Netherlands
Email: a.liotta@tue.nl

Abstract—The advances in wireless communications, together with the need of sensing and controlling various nature or human made systems in a large number of points (e.g. smart traffic control, environmental monitoring), lead to the emergence of Wireless Sensor Networks (WSN) as a powerful tool to fulfill the above requirements. Due to the large amount of wireless devices needed and cost constraints, such networks are usually made by low-cost devices with limited energy and computational capabilities, these further on being subject to easy communication or hardware fails. At the same time, the deployment of such devices in harsh environments (e.g. in the ocean) may lead to uncontrollable redundant topologies which have to be often refined during the exploitation phase of these networks in an automated manner. In the scope of these arguments, in this paper, we take advantage of the latest theoretical advances in complex networks and we propose an automated solution to refine the topology of WSNs by using centrality metrics to detect the redundant nodes and links in a network, and further on to shut down them safely. Our solution may work in both ways, centralized or decentralized, by choosing a centralized or a decentralized centrality metric, this choice being driven by the application goal. The experiments performed on a wide variety of network topologies with different sizes (e.g. number of nodes and links), using different centrality metrics, validate our approach and recommend it as a solution for the automatic control of WSNs topologies during the exploitation phase of such networks to optimize, for instance, their life time.

I. INTRODUCTION

With the emergence of sensors with wireless capability, most of current sensor networks consist of a collection of wirelessly interconnected units, each of them with embedded sensing, computing and communication capabilities [1]. Such sensor networks are referred to as wireless sensor networks (WSNs) [2]. Due to their versatility, WSNs have been employed in a wide range of sensing and control applications [3], such as smart traffic control, environmental monitoring, security surveillance, and health-care [4]. As a consequence of the low cost, low available energy [5] and potential mobility of sensors, they are prone to hardware and communication failures [6], failures which may be detected or solved using smart autonomic solutions [7], [8], [9], [10], [11]. Moreover, the dense and random deployment of sensors introduces a huge amount of redundancy in sensing capabilities and gathered data [12]. In these situations, to identify and turn off the redundant nodes can save energy consumption for the entire network to extend the network lifetime [13]. Moreover, to

increase the life of WSNs by saving energy, or to reduce the number of interferences due to the limited resources, the number of communication links between autonomous sensors must be reduced at minimum [14], [15]. Redundancy reduction requires that the network stays fully connected to let the flow of information pass between any communication points.

At the same time, WSNs may be easily mapped into complex networks [16], [17], graph objects which model real world networks. Then, our assumption, is that the problem of finding redundant nodes or links in a WSN translates into a problem of finding the less important nodes or links in a complex network to safely remove them. The latter one, we argue, can be solved using centrality metrics [17], [18], which are algorithms capable to find the most or the less important (central) nodes or links in a graph, by associating to each element (node or link) a centrality score.

This paper reflects an ongoing work towards a completely distributed redundancy reduction in WSNs. More exactly, the main contribution of this paper is to propose a method to control the topology of WSNs, by using centrality metrics to detect the redundant nodes or links, and to reduce them safely afterwards. Besides that, the method may be applied both in a decentralized or centralized fashion, depending on the nature of the centrality metric used.

The remaining of this paper is organized as follows. Section II provides background knowledge on complex networks and WSNs, for the benefits of the non-specialist reader. Section III details our proposed method for redundancy reduction in WSN. Section IV demonstrates the validity of our approach, while Section V concludes the paper and presents further research directions.

II. BACKGROUND

In this section we present background knowledge useful for the non-specialist reader to understand how to compute centrality metrics in complex networks, and related work on redundancy reduction in WSNs.

A. Complex Networks

Network science [16] is an extension of graph theory in the real world. It studies the hidden relations between the real world networks, such as biological neural networks, power grids, actors and movies, and so on. All these types of networks

are generally named complex networks. Formally, any arbitrary network is an object which contains nodes and edges (links) between nodes. These edges can be directed or undirected. Also, each link may have associated a weight value, in this situation the resulting object being named weighted complex networks. Further on, in this paper we will focus just on weighted and undirected networks because, as we will show in Section III, these can effectively map WSNs. For simplicity, let us introduce the following standard notations: $\mathcal{G} = (V, E)$ a graph, where \mathbf{V} represents the set of all nodes and \mathbf{E} the set of all edges, $n = |V|$ the total number of nodes in the graph, and $m = |E|$ the total number of edges in the graph. Mainly, based on their topological properties, three main classes of networks are studied in the literature, as it is presented further.

1) *Erdős-Rényi Random Graph [19]*: is one of the most studied type of graphs as it represents the very basis of network's theory. It has n nodes, where each pair of nodes may be connected by a link with a probability $p \in [0, 1]$, independently on the other links. A graph constructed using the aforementioned property may have no spatial structures. Moreover, due to the fact that each link is independent, it may be inappropriate to model real-world phenomena with such type of networks. Hence, the scale-free and small-world models, discussed next, are commonly used to model real word networks.

2) *Scale-Free networks*: are networks often observed in natural (e.g. high temperature superconductors [20], protein-protein interactions) and human-made systems (e.g. social networks, power grids, airline networks), in which the degree distribution follows a power law. Barabási et al. in [21] mapped a part of the World Wide Web topology in such a network type. Also, they proposed an algorithm to build artificially such networks using the preferential attachment algorithm (i.e. the node with a high degree, called "hubs" in the literature, have a higher chance to obtain new links when a new node is added in the graph).

3) *Small-World networks*: are graphs in which each node, may be reached from any other node in a very small number of hops. Thus, the shortest path between any two nodes has a length of $\propto \log(n)$. Such construction, is also observed in real-world networks (e.g. power grids, social networks, biological neural networks [22]).

B. Centrality metrics

Given a network topology is important to find the most important nodes or links. This may be done using centrality metrics. Thus, centrality is a measure to assess how important individual nodes or links are in a network and how they can influence their neighborhood or even the full network. Although there is no clear definition in the literature for graph "centrality", we can find different ways to compute different variations, each one being capable to provide a different centrality score for the same element (i.e. node or link). Some of the most used centrality metrics, which achieve a very good performance in the literature, are Betweenness Centrality (BC), Current Flow Betweenness Centrality (CFBC), and Second Order Centrality (SOC), as briefly presented below.

1) *Betweenness centrality*: is one of the most discussed methods to measure the centrality of nodes or links [23], [24].

On short, it is quantifying how a node is lying on the path between other nodes. Formally, for a node $v \in \mathbf{V}$ this can be written as:

$$C_{bc}(v) = \sum_{w,u \in \mathbf{V}} \frac{\sigma_{w,u}(v)}{\sigma_{w,u}} \quad (1)$$

where $\sigma_{w,u}(v)$ represents the number of shortest paths comprised between node w and node u which are passing through node v , and $\sigma_{w,u}$ represents the total amount of shortest paths from w to u . The computational complexity of the original algorithm is $\mathcal{O}(n^3)$, making it unsuitable for large networks. For this reason, an approximation of betweenness centrality was proposed [25] which has a computational complexity of $\mathcal{O}(|n||m|)$ and which we will use in our experiments in Section IV.

2) *Current flow betweenness centrality*: is another way to measure the centrality of network elements (i.e. nodes or links) which is inspired by the behavior of the electric current. In practice, it makes the assumption that for a centrality of a node not only is important the shortest path between any two other nodes which pass through it, but also the consideration off all the other possible paths. The interested reader is referred to [26], [27] for a deeper discussion.

3) *Second order centrality*: is a novel form to measure node's centrality, calculated in a distributed manner, proposed by Kermarrec et al. in [28]. The algorithm is based on a long random walk in the graph, which may start from any randomly chosen node, and runs continuously. After it visits each node for at least three times, in each node the standard deviation of the steps needed by the random walk to return to it is computed and this further on will give the centrality score, as shown in the paper. As proved in [28] the computational complexity of SOC is $\mathcal{O}(n^3)$, but usually a good enough performance is achieved a bit faster (somewhere over $\mathcal{O}(n^2)$).

C. Redundancy reduction in WSNs

Extensive research has been performed on the task of efficiently reduce redundancy, in order to extend the life time of the sensors and, thus, the network itself. WSNs are wireless networks composed of low-cost, low-power, multifunctional sensors and actuators [29].

Coverage parameters easily obtained from the network nodes have been used to reduce redundancy. Example of these approaches are the coverage configuration protocol (CCP) of Xing et al. [30], the perimeter coverage algorithm of Huang et al. [31] or the coverage probability approach of Huang et al. in [13]. However, these approaches suppose prior network and nodes knowledge and start from the premise of a central node (usually the gateway node) which performs the reduction algorithm. In addition, their long execution time makes them impossible to realize in a real-time network environment.

Network tools such as virtualization [32] and clustering [33], [34], [35], [36] have also been used to address the redundancy reduction task [37]. Two main types of virtualization have been proposed in the context of WSN: (1) virtualization of sensors (i.e., virtual sensors) and (2) virtualization of sensor networks (i.e., virtual sensor networks or sensor cloud). Virtual sensors are either collocated on a single physical sensor [38] or combine multiple physical sensors into a virtual one [39].

Collocation facilitates the reuse of a single sensor by multiple WSN applications. Combining sensors allows the sensing data from several heterogeneous sensors to be aggregated to provide applications with indirect measurements of abstract conditions that cannot be directly measured by physical sensors. The virtualization of entire sensor networks has been proposed under multiple alternative names, including virtual sensor networks [40], [38] and sensor cloud [41]. Independently of virtualization research, the grouping of sensor nodes has also been previously studied in the context of WSN routing and is referred to as clustering [42], [12], [43]. In clustering-based routing algorithms, sensors are structured into a hierarchy, which has several advantages [12]. It improves scalability, energy-efficiency by allowing duplicate sensing data to be filtered out at the cluster level, and robustness by handling failures at the cluster level. Existing WSN clustering algorithms do not consider sensor node mobility and exclusively try to remove duplicate sensor data to improve energy-efficiency, rather than exploiting it to improve resilience [12]. Both in the case of clustering and virtualization, the approach is done in a centralized manner.

In the next section we propose to use complex networks centrality techniques to perform the reduction redundancy in WSNs

III. PROPOSED PROCEDURE

In this section we describe our methodology to perform redundancy reduction in WSNs. The main idea is to treat the WSN under scrutiny as a complex network and to take advantages of the centrality metrics used in network science or graph theory to detect the less important nodes or links in the network and to remove them safely.

Thus, the first step is to map the WSN under scrutiny into a weighted undirected graph $\mathcal{G}(\mathbf{V}, \mathbf{E})$, where \mathbf{V} is the set of all nodes, \mathbf{E} is the set of all links, and in which each sensor node is represented by a node $v \in V$, and the link between any two nodes v and w is represented by the edge $(v, w) \in \mathbf{E}$. The weight (cost) C_{vw} of any link $(v, w) \in \mathbf{E}$ may be computed by using local parameters of the specific nodes v and w (e.g. energy level, covered area) and it has to be determined for each WSN under scrutiny in particular or it may be set to 1 for all links if we have a homogeneous network.

After the WSN is mapped into a graph, we can distinguish two cases based on our goal (i.e. nodes or links reduction) and even if the procedure is similar we will treat them separately below for the sake of clarity.

A. Nodes reduction

- Run any of the nodes centrality algorithms (i.e. BC, CFBC, SOC) on $\mathcal{G}(\mathbf{V}, \mathbf{E})$ to obtain the centrality score associated with each node. Please note that BC and CFBC have to be run in a centralized manner, but SOC may be run in a decentralized fashion, directly in the sensor nodes.
- Create a list with all nodes, sorted from the less central one to the most central one, according to the respective scores computed by the centrality algorithm used.

- Start removing¹ the nodes from the WSN under scrutiny according to the ranking list computed earlier, starting from the first element of it and continuing with the next ones until the network reaches the wanted size.

B. Links reduction

- Run any of the links centrality algorithms (i.e. BC, CFBC) on $\mathcal{G}(\mathbf{V}, \mathbf{E})$ to obtain the centrality score associated with each link.
- Create a list with all links, sorted from the less central one to the most central one, according to their respective scores computed by the centrality algorithm used.
- Start removing² the links from the WSN under scrutiny according to the ranking list computed earlier, starting from the first element of it and continuing with the next ones until the network reaches the wanted size.

Further on, in the next section, we demonstrate the robustness of our proposed methodology using three centrality metrics (i.e. BC, CFBC, and SOC) presented in Subsection II-B (bear in mind that other metrics may also offer good results).

IV. EXPERIMENTS AND RESULTS

To demonstrate the performance of our proposed methodology, we performed two sets of experiments (i.e. nodes reduction and links reduction) on the most usual types of complex networks topology (i.e. Erdős-Rényi Random Graphs [19], Scale-Free networks [21], and Small-World networks [44]).

A. Node reduction

In this set of experiments we have analyzed our proposed methodology on Erdős-Rényi Random Graphs, Scale-Free and Small-World networks by using three node centrality metrics, among which two are centralized (BC and CFBC) and one is decentralized (SOC). We compare their performance with a random node removal procedure. Moreover, for each network type, we performed the analyzes on two cases: smaller networks (i.e. 100 nodes and approximately 5 times more links) and bigger networks (i.e. 1000 nodes and approximately 10 times more links). Further on, to have a complete overview of the method behavior, for each case we have generated randomly 10 networks, and we averaged the results. Before we analyze the results of this set of experiments, to exemplify the approach, Fig 1 shows snapshots of a scale-free network which has initially 100 nodes and 496 links. We show the status of the network after 60%, 75%, and 90% of the less important nodes are removed, respectively. The figure reflects the fact that by using any of the centrality metrics presented in Section II-B the remaining network in each snapshot is still fully connected, while by performing random nodes removal the remaining network in any of the snapshots is split in at least three components.

¹Please note that this operation is equivalent with the physical shut down of a wireless node in the WSN

²Please note that this operation is equivalent with the physical shut down of a link in the WSN

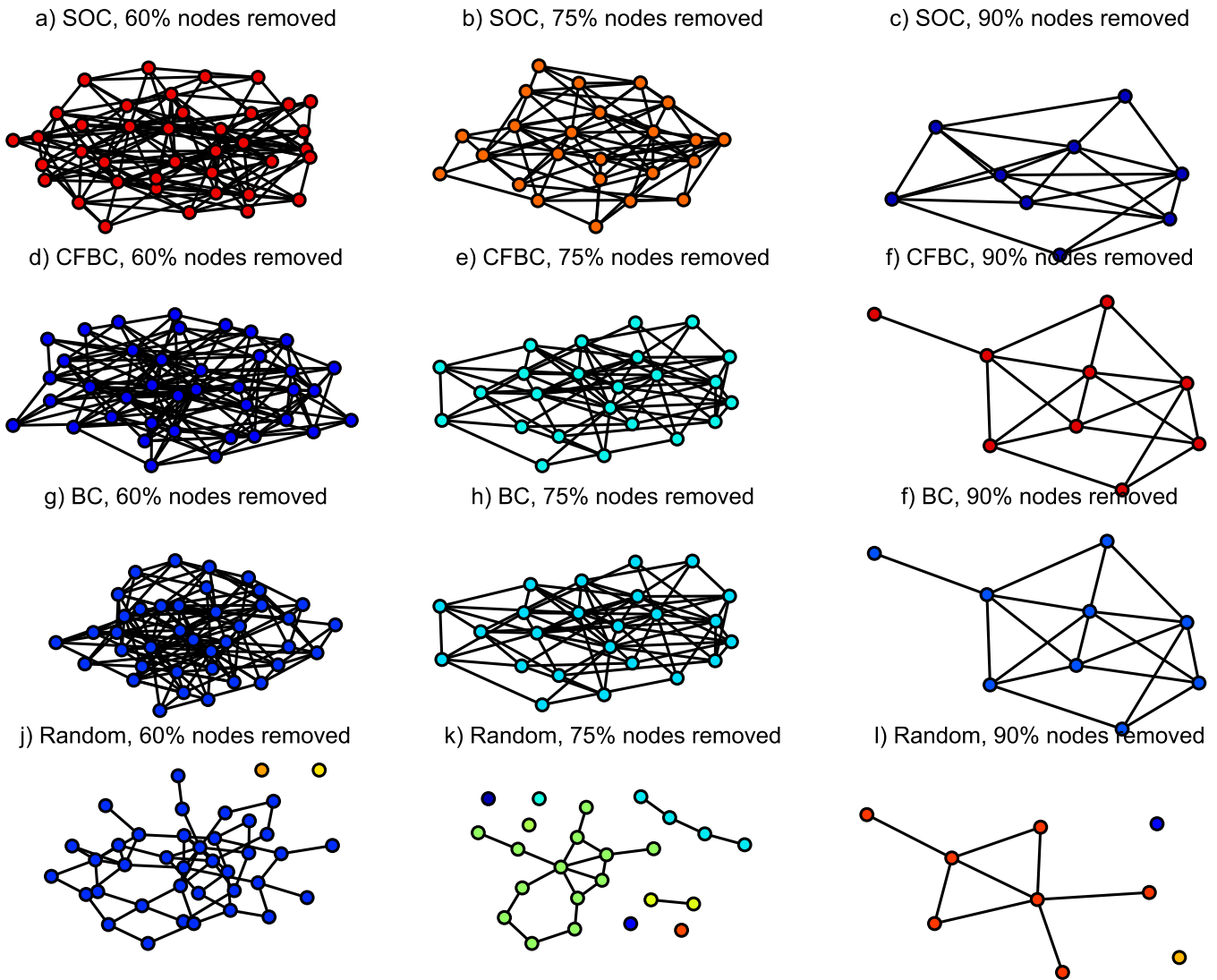


Fig. 1: Snapshots with the resulted network topologies after 60%, 75% and 90% of the less central nodes are removed according with different centrality metrics from a scale-free network with 100 nodes and 496 links.

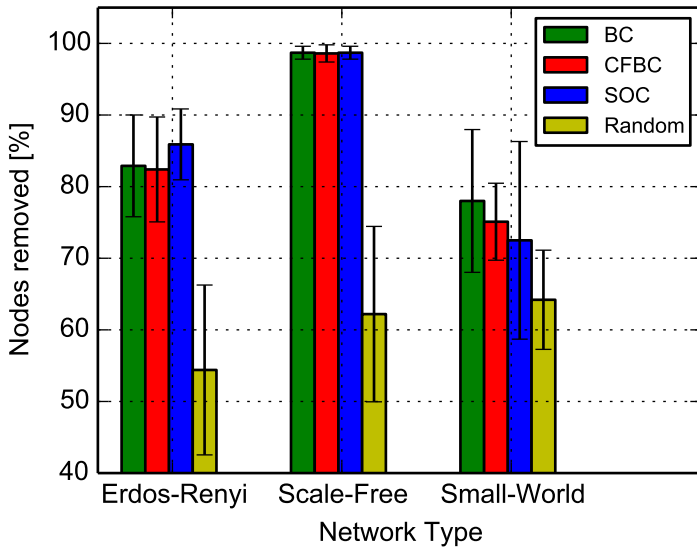
Thus, to summarize our experiments on node reduction, in Fig 2 we show after how many nodes removed (in percentages from the total number of nodes in the initial network) the network is split in two connected components. The results confirm our initial hypothesis. It shows that independently of the network size or the initial topology if the nodes are removed according with the scores given by the node centrality metrics, starting from the less central nodes up to the most central ones, the network will stay completely connected for a much longer time than the case when the nodes are removed randomly. Remarkably, SOC (i.e. a decentralized centrality metric) performs similarly to BC, and CFBC, which are centralized methods. This may be really useful in the context of WSNs to shut down nodes in a completely decentralized manner. More than that, it can be observed that the Scale-Free networks are the most suitable for redundancy reduction using centrality metrics due to the fact that the remaining network is still fully connected even after 97% of the nodes are removed, while by performing random removal, after we

removed approximately 60% of the nodes, the network is split in two connected components.

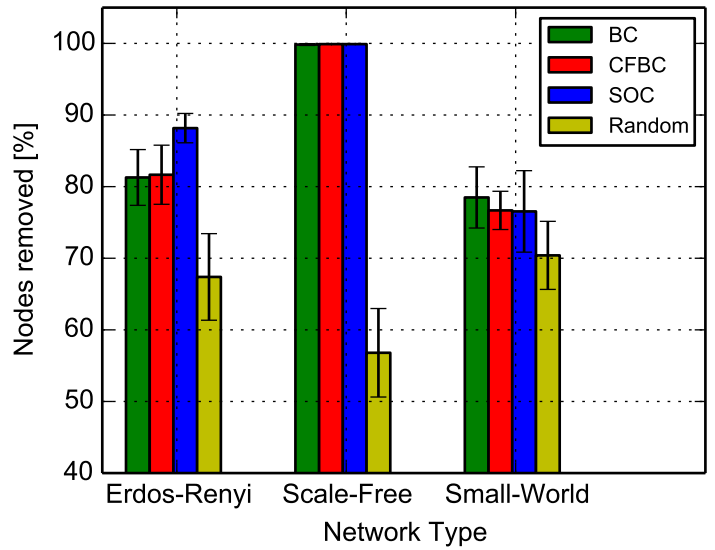
B. Link reduction

In this set of experiments we studied the proposed procedure on the same type of networks as before (i.e. Erdős-Rényi Random Graphs, Scale-Free and Small-World) using two centralized link centrality metrics (BC and CFBC)³ and we compare their performance with random links removal procedure. Similarly to the previous set of experiments, we performed the analyzes on two cases: small networks (i.e. 100 nodes and approximately 500 links) and big networks (i.e. 1000 nodes and approximately 10000 links) on each network topology, and for each case we generated randomly 10 networks. Summarizing this set of experiment on links reduction, in Fig 3 we show after how many links removed (as

³Please note that there is no version of SOC for links in the literature and we let the conceiving of a SOC version for links as further work

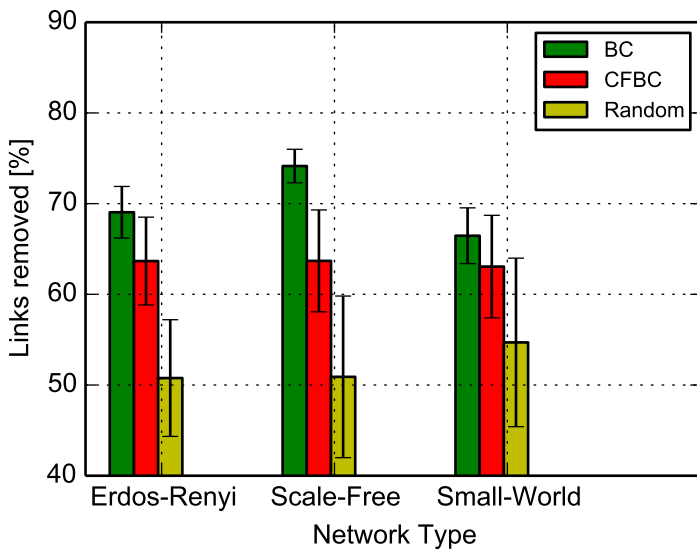


(a) Networks with 100 nodes and around 5 times more links.

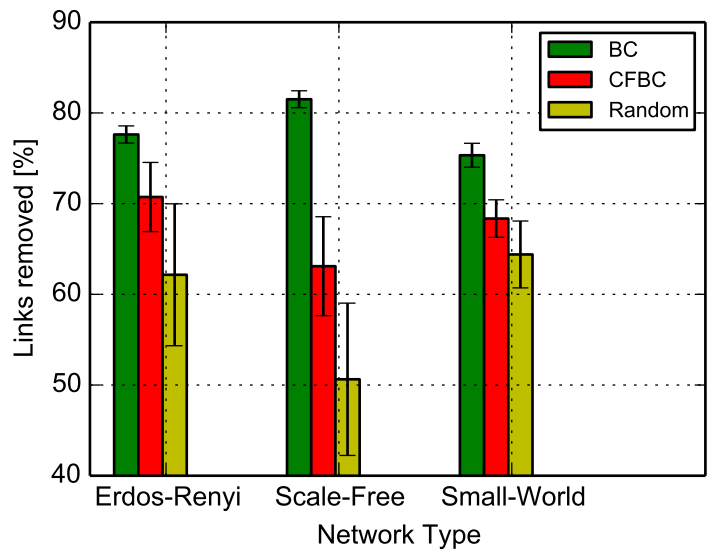


(b) Networks with 1000 nodes and around 10 times more links.

Fig. 2: Nodes removed in percentages from the total number of nodes until the network is split in two connected components. For each network type, the results are averaged over 10 random generated networks and the results are presented with mean and standard deviation.



(a) Networks with 100 nodes and around 5 times more links.



(b) Networks with 1000 nodes and around 10 times more links.

Fig. 3: Links removed in percentages from the total number of links until the network is split in two connected components. For each network type, the results are averaged over 10 random generated networks and the results are presented with mean and standard deviation.

a percentages of from the total number of links in the initial network) the network is split in two connected components. As before, the results confirm our hypothesis, showing that, independently of the network size or the initial topology, if the links are removed according to the scores given by the link centrality metrics, starting from the less important ones up to the most central ones, the network will stay fully connected for a much longer time than in comparison with the case when the links are removed randomly. Please note that in the case of links removal the network can not stay fully connected up to

100% of the links removed because it needs at least a number of links equal to the number of nodes in the network such that all the nodes⁴ in the network are connected. As before, we can observe that the Scale-Free networks are the most prone to redundancy reduction using centrality metrics due to the fact that they remain completely connected for a longer period. Another interesting finding is that the BC metric is the most suitable for link redundancy reduction, as it outperforms

⁴The number of nodes does not decrease during the links reduction procedure.

CFBC in all situations.

V. CONCLUSION

In this paper, we introduced a novel methodology to perform redundancy reduction in wireless sensor networks using concepts from network science (i.e. centrality metrics) in order to optimize the functionality of the WSN under scrutiny. We showed that our methodology is capable to outperform drastically the random reduction of WSNs elements (i.e. wireless nodes, or links) in various experiments, on different types of networks topology, and with different sizes. Moreover, we show how this methodology can be applied in a distributed manner especially for nodes redundancy reduction. As further work, we intend to perform decentralized redundancy reduction also for links and to apply the proposed procedure in a physical WSN environment to quantify the savings, mainly in terms of energy, interference and communication efficiency.

ACKNOWLEDGMENT

This work has been partially supported by: ARTEMIS project ACCUS (Adaptive Cooperative Control in Urban sub-Systems), Grant agreement n. 333020 (<http://projectaccus.eu/>); The European Research Council project BROWSE (Beamsteered Reconfigurable Optical-Wireless System for Energy-efficient communication), Grant agreement n. 291632.

REFERENCES

- [1] J. Li, L. L. Andrew, C. H. Foh, M. Zukerman, and H.-H. Chen, "Connectivity, coverage and placement in wireless sensor networks," *Sensors*, vol. 9, no. 10, p. 7664, 2009.
- [2] S. Kumar Singh, M. P. Singh, and D. K. Singh, "Routing Protocols in Wireless Sensor Networks - A Survey," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 1, no. 2, 2010.
- [3] H. Das, B. Naik, B. Pati, and C. R. Panigrahi, "A survey on virtual sensor networks framework," *International Journal of Grid Distribution Computing*, vol. 7, no. 5, pp. 121–130, 2014.
- [4] S. Galzarano, R. Giannantonio, A. Liotta, and G. Fortino, "A task-oriented framework for networked wearable computing," *Automation Science and Engineering, IEEE Transactions on*, vol. PP, no. 99, pp. 1–18, 2014.
- [5] R. Kotian, G. Exarchakos, D. C. Mocanu, and A. Liotta, "Predicting battery depletion of neighboring wireless sensor nodes," in *Algorithms and Architectures for Parallel Processing*, ser. Lecture Notes in Computer Science. Springer, 2013, vol. 8286, pp. 276–284.
- [6] S. Ou, K. Yang, A. Liotta, and L. Hu, "Performance analysis of of-flooding systems in mobile wireless environments," in *Communications, 2007. ICC '07. IEEE International Conference on*, June 2007, pp. 1821–1826.
- [7] A. Liotta, "The cognitive NET is coming," *IEEE Spectrum*, vol. 50, no. 8, pp. 26–31, Aug. 2013.
- [8] N. Qadri and A. Liotta, "Analysis of pervasive mobile ad hoc routing protocols," in *Pervasive Computing*, ser. Computer Communications and Networks, A.-E. Hassanien, J. H. Abawajy, A. Abraham, and H. Hagra, Eds. Springer London, 2010, pp. 433–453.
- [9] S. Galzarano, G. Fortino, and A. Liotta, "A learning-based mac for energy efficient wireless sensor networks," in *Internet and Distributed Computing Systems*, ser. Lecture Notes in Computer Science, G. Fortino, G. Di Fatta, W. Li, S. Ochoa, A. Cuzzocrea, and M. Pathan, Eds. Springer International Publishing, 2014, vol. 8729, pp. 396–406.
- [10] H. Bosman, G. Iacca, H. Wortche, and A. Liotta, "Online fusion of incremental learning for wireless sensor networks," in *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*, Dec 2014, pp. 525–532.
- [11] M. Vega, S. Zou, D. Mocanu, E. Tangdiongga, A. Koonen, and A. Liotta, "End-to-end performance evaluation in high-speed wireless networks," in *Network and Service Management (CNSM), 2014 10th International Conference on*, Nov 2014, pp. 344–347.
- [12] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors (Basel, Switzerland)*, vol. 12, no. 8, pp. 11 113–53, Jan. 2012.
- [13] S. Huang and X. Zhao, "Redundant Nodes Elimination in Wireless Sensor Networks," in *4th International Conference on Information Computing and Applications (ICICA)*, 2013.
- [14] D. Geelen, G. van Kempen, F. van Hoogstraten, and A. Liotta, "A wireless mesh communication protocol for smart-metering," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, Jan 2012, pp. 343–349.
- [15] A. Liotta, D. Geelen, G. van Kempen, and F. van Hoogstraten, "A survey on networks for smartmetering systems," *International Journal of Pervasive Computing and Communications*, vol. 8, no. 1, pp. 23–52, 2012.
- [16] M. E. J. Newman, *Networks: An Introduction*. Oxford University Press, 2010.
- [17] X. Fu, W. Li, and G. Fortino, "Empowering the invulnerability of wireless sensor networks through super wires and super nodes," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, May 2013, pp. 561–568.
- [18] D. Mocanu, G. Exarchakos, and A. Liotta, "Node centrality awareness via swarming effects," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, Oct 2014, pp. 19–24.
- [19] P. Erdős and A. Rényi, "On random graphs. I," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.
- [20] M. Fratini, N. Poccia, A. Ricci, G. Campi, M. Burghammer, G. Aeppli, and A. Bianconi, "Scale-free structural organization of oxygen interstitials in La₂CuO_{4+y}," *Nature*, vol. 466, no. 7308, pp. 841–844, 2010.
- [21] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [22] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, no. 393, pp. 440–442, 1998.
- [23] L. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, pp. 35–41, 1977.
- [24] U. Brandes, "On variants of shortest-path betweenness centrality and their generic computation," *Social Networks*, vol. 30, no. 2, pp. 136–145, 2008.
- [25] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, no. 163, 2001.
- [26] M. Newman, "A measure of betweenness centrality based on random walks," *Social networks*, vol. 27, no. 1, pp. 39–54, 2005.
- [27] U. Brandes and D. Fleischer, "Centrality measures based on current flow," in *Proceedings of the 22Nd Annual Conference on Theoretical Aspects of Computer Science*, ser. STACS'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 533–544.
- [28] A.-M. Kermarrec, E. L. Merrer, B. Sericola, and G. Trdan, "Second order centrality: Distributed assessment of nodes criticality in complex networks," *Computer Communications*, vol. 34, no. 5, pp. 619–628, 2011.
- [29] D. Kumar Gupta, "A review on wireless sensor networks," *Network and Complex Systems*, vol. 3.
- [30] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Trans. Sen. Netw.*, vol. 1, no. 1, pp. 36–72, Aug. 2005.
- [31] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, ser. WSNA '03. New York, NY, USA: ACM, 2003, pp. 115–121.
- [32] D. Mocanu, G. Santandrea, W. Cerroni, F. Callegati, and A. Liotta, "Network performance assessment with quality of experience benchmarks," in *Network and Service Management (CNSM), 2014 10th International Conference on*, Nov 2014, pp. 332–335.
- [33] S. Sivavakeesar, G. Pavlou, C. Bohoris, and A. Liotta, "Effective management through prediction-based clustering approach in the next-

- generation ad hoc networks,” in *Communications, 2004 IEEE International Conference on*, vol. 7, June 2004, pp. 4326–4330 Vol.7.
- [34] D. Pettinger and G. Di Fatta, “Space partitioning for scalable k-means,” in *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*, Dec 2010, pp. 319–324.
- [35] D. Pettinger and G. Di Fatta, “Scalability of efficient parallel k-means,” in *E-Science Workshops, 2009 5th IEEE International Conference on*, Dec 2009, pp. 96–101.
- [36] G. Di Fatta, F. Blasa, S. Cafiero, and G. Fortino, “Epidemic k-means clustering,” in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, Dec 2011, pp. 151–158.
- [37] C. Bohoris, A. Liotta, and G. Pavlou, “Evaluation of constrained mobility for programmability in network management,” in *Services Management in Intelligent Networks*, ser. Lecture Notes in Computer Science, A. Ambler, S. Calo, and G. Kar, Eds. Springer Berlin Heidelberg, 2000, vol. 1960, pp. 243–257.
- [38] Z. Khalid, N. Faisal, and M. Rozaini, “A Survey of Middleware for Sensor and Network Virtualization,” *Sensors*, vol. 14, no. 12, pp. 24 046–24 097, Dec. 2014.
- [39] S. Kabadayi, A. Pridgen, and C. Julien, “Virtual Sensors: Abstracting Data from Physical Sensors,” in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM’06)*. IEEE, Jun. 2006, pp. 587–592.
- [40] M. M. Islam, M. M. Hassan, G.-W. Lee, and E.-N. Huh, “A survey on virtualization of Wireless Sensor Networks.” *Sensors (Basel, Switzerland)*, vol. 12, no. 2, pp. 2175–207, Jan. 2012.
- [41] S. Madria, V. Kumar, and R. Dalvi, “Sensor Cloud: A Cloud of Virtual Sensors,” *IEEE Software*, vol. 31, no. 2, pp. 70–77, Mar. 2014.
- [42] V. Kumar, S. Jain, and S. Tiwari, “Energy Efficient Clustering Algorithms in Wireless Sensor Networks: A Survey.” *International Journal of Computer Science Issues*, vol. 8, no. 5, pp. 259–268, 2011.
- [43] C. Ragusa, A. Liotta, and G. Pavlou, “An adaptive clustering approach for the management of dynamic systems,” *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 12, pp. 2223–2235, Dec 2005.
- [44] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks.” *Nature*, vol. 393, no. 6684, pp. 409–10, 1998.