A Simple Clustering Strategy for Wireless Sensor Networks

Waltenegus Dargie, Senior Member, IEEE and Jianjun Wen

Abstract—Organising nodes into efficient clusters in wireless sensor networks facilitates data aggregation and command dissemination. But clustering is a complex and costly process, since it has to be carried out in a distributed and periodic manner. In this paper we propose a simple clustering strategy employing an adjacency matrix which encodes nodes' neighbourhood and connectivity in a network. Our approach enables the assignment of cluster heads for multiple rounds in a single step thereby limiting the cost of cluster head election and child node association.

Index Terms—Adjacency matrix, clustering, node, wireless sensor network, Katz Index, topology

I. INTRODUCTION

Recent advances in low-power wireless communications, mobile robotics, wireless sensor networks, and Unmanned Aerial Vehicles (drones) promise the deployment of Cyber-Physical Systems (CPS) in remote and inaccessible areas for various operations. Examples include toxic gas detection, assessment of damage following a natural or man-made disaster, and pandemics, such as the COVID-19 [1]. The mobile robots and drones can be used to deploy the sensor nodes and, once the nodes establish a network, to interface the network with a remote base or control station.

In this respect, the way the sensor nodes organise themselves to establish a network and the efficiency of the network in fulfilling its purpose are critical aspects. This is because:

1) One's knowledge of the area of interest depends on the quality of information that can be extracted from the environment, which in turn depends on how quick and efficient the nodes are in delivering this information.
2) The mobile robots and drones operate with exhaustible batteries. This necessitates highly efficient interactions and movements.

Broadly speaking, a wireless sensor network may have either a flat or a hierarchical topology [2]. In a flat topology network, all nodes play the same roles, namely, sensing, data processing, and packet forwarding. Flat topology networks are relatively easy to set up and robust to node failure, but they can also be highly inefficient since routes are usually defined by first flooding the network with discovery packets [3].

In a hierarchical topology there are two types of nodes: child nodes and cluster heads. The child nodes are grouped into different clusters and in each cluster a single node is designated as a cluster head. The responsibility of a cluster head is to (1) define a communication and sleeping schedule for each node, (2) aggregate the data from its child nodes (for example, by applying MIN, MAX, MEAN operations), and transfer the data to a remote base station, either directly or in collaboration with other cluster heads [4]. Apparently, the assignment of cluster heads is more intensive than the child nodes' as a result of which they are likely to exhaust their energy more quickly than the child nodes, potentially causing the entire network to prematurely fragment [5]. In order to prevent this from happening, the underlying protocol assigns this role to other nodes on a regular interval [6].

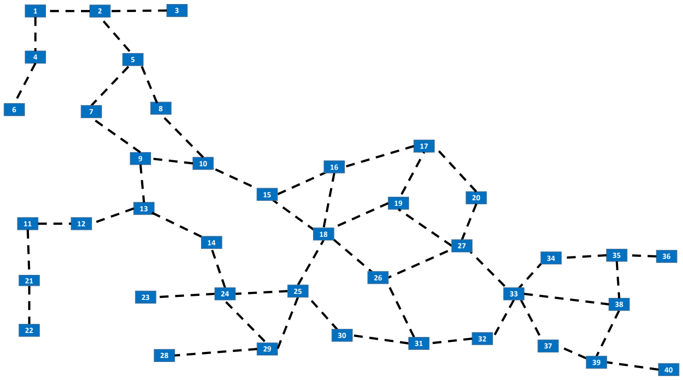


Fig. 1: A spontaneously deployed WSN.

Hence, in a hierarchical-topology network, the nodes can be found either in a cluster formation or in a steady state. The first phase is communication and computation intensive whereas the second is resource efficient [7]. A hierarchical topology makes sense if the energy saved in the steady state far outweighs the energy cost of the first state. In general, which topology is more suitable for a particular application depends on many factors, including the size of the network, the expected lifetime of the network, the availability of global knowledge such as the number of clusters needed to maintain a fully connected network.

In this paper we propose an algorithm which drastically cuts the cost of cluster formation. It is based on the computation of a binary adjacency matrix signifying node neighbourhood. For stationary nodes our approach deterministically identifies the cluster heads and the associated child nodes for each round thus avoiding the need (as well as the associated cost) for nodes to advertise their candidacy to be cluster heads and, once the cluster heads are elected, to express their membership.

II. NODE RELEVANCE

In a self-organising network, cluster heads are selected in several steps dealing with advertisement of candidacy, election (voting), and node association (cluster formation) [7], [8]. These steps must be repeated at the beginning of a new epoch (round). Our clustering algorithm determines cluster heads for multiple rounds in one go relying only on the **Relevance Index** of nodes. The **Relevance Index** of the nodes is computed from the adjacency matrix describing the physical topology of the network.

In [10], we propose a model to measure the relevance of nodes in a wireless sensor network. Accordingly, node relevance is defined as a measure of how well-connected nodes are in a network. This aspect

Manuscript first submitted on 12 March 2020.

W. Dargie and J. Wen are with the Technische Universität Dresden, Faculty of Computer Science, 01062, Dresden, Germany.
E-mail: {waltenegus.dargie, jianjun.wen}@tu-dresden.de

This work has been partially funded by the German Research Foundation in the context of the RoReyBan project (DA 1121/7-1) and by the German Federal State of Saxony as part of the "SNIFFBOT: Sniffing Dangerous Gases with Immersive Robots" project under grant agreement number 100369691.

takes into consideration not only the direct links nodes establish with their immediate neighbours but also how well-connected their neighbours and the neighbours of those neighbours are, and so on. To illustrate this, consider the network displayed in Fig. 1. We can use a binary adjacency matrix **C** to express its topology (ref. to Equation 1). The element $c_{ij} = 0$ if there is no direct link between nodes i and j, otherwise, $c_{ij} = 1$. Moreover, if the wireless channel is taken as symmetric, then $c_{ij} = c_{ji}$, otherwise, $c_{ij} \neq c_{ji}$. For our case, we assume a symmetrical channel. Notice that the column sums of **C** give the degree of each node (i.e., the number of single-hop links the nodes establish with their peers).

$$\mathbf{C} = \begin{array}{c} \\ N1 \\ N2 \\ N3 \\ N4 \\ N5 \\ N6 \\ N7 \\ N8 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccccccc} N1 & N2 & N3 & N4 & N5 & N6 & N7 & \cdots \end{array} \\ \left[\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & \cdots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right] \end{array} \quad (1)$$

Squaring **C** yields the number of two-hop links the nodes establish with their peers. Likewise, the k-th power of **C** yields the number of k-hop links which can be established between any two nodes in the network. If we normalise the number of sing-hop links a node establishes with its neighbours by $(n - 1)$, where n is the number of nodes in the network, then we have:

$$\mathbf{H} = \frac{\mathbf{C}}{(n-1)} \quad (2)$$

For a fully meshed network, the column sums of **H** yields 1. Similarly:

$$\mathbf{H}^k = \frac{\mathbf{C}^k}{(n-1)^k} \quad (3)$$

Furthermore, if we assume that the wireless channel is lossy with a probability p of successfully transmitting a packet, then the normalised number of single as well as all multi-hop links the nodes can establish with their peers can be expressed as:

$$\mathbf{T} \approx \sum_{k=1}^{\infty} (p\mathbf{H})^k = p\mathbf{H}\,(\mathbf{I} - p\mathbf{H})^{-1} \quad (4)$$

T is a measure of the network's connectedness. The relative connectedness of the individual nodes can be determined by summing the columns of **T**:

$$\mathbf{r} = \mathbf{u}^{\intercal}\mathbf{T} \quad (5)$$

where **u** is a column vector consisting of n unit elements. The m-th element of **r** refers to the **Relevance Index** of the m-th node.

III. Clustering

Designating well-connected nodes as cluster heads ensures fast data collection and command dissemination. However, in case the network is deployed spontaneously or the actual placement of nodes is influenced by physical conditions, the nodes which are well-connected may also be very close to each other. Let us consider the topology of Fig. 1. Table 1 lists the nodes according to their *R-Index* in descending order. If we decide to designate the top five nodes (12.5 % of the nodes) as cluster heads (i.e,, nodes 18, 33, 25, 27, 24), the distribution of cluster heads in the network looks like the one displayed in Fig. 2. This assignment is straightforward, but as can be seen, it makes the association of the remaining nodes as child nodes problematic. Some of the cluster heads, such as nodes 25 and 27, will have a small number of child nodes while others, such as nodes 18 and 24, will potentially have a disproportionally large number of nodes as their child nodes. Similarly, the child nodes will experience different end-to-end latency during command dissemination and data collection. If, for instance, node 1 is associated with node 18, a packet from its parent node will reach it passing 6 intermediate nodes. On the other hand, if node 17 is associated with node 27, packet from its parent node reaches it passing only a single intermediate node.

Table 1: Nodes ordered according to their R-Index

Rank	Node	Rank	Node	Rank	Node	Rank	Node
1	18	**11**	29	**21**	32	**31**	11
2	33	**12**	17	**22**	34	**32**	4
3	25	**13**	10	**23**	37	**33**	21
4	27	**14**	9	**24**	20	**34**	23
5	24	**15**	31	**25**	30	**35**	28
6	19	**16**	13	**26**	14	**36**	40
7	26	**17**	5	**27**	8	**37**	36
8	16	**18**	39	**28**	7	**38**	3
9	15	**19**	35	**29**	12	**39**	6
10	38	**20**	2	**30**	1	**40**	22

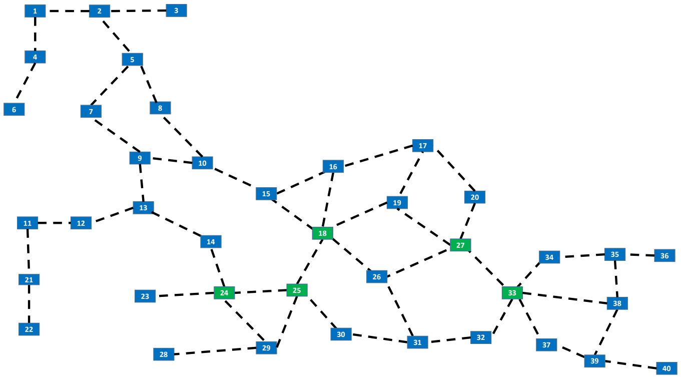


Fig. 2: Designating densely interconnected nodes as cluster heads

In order to ensure that cluster heads are well distributed in the network, one can introduce different constraints to the application of the *R-Index*. For example, one can introduce a minimum of 2-hop distance between any of the cluster heads. This enables cluster heads to associate those nodes within their immediate proximity (1-hop away) as their child nodes, as can be seen in Fig. 3. The decision automatically disqualifies the nodes which are a single-hop away

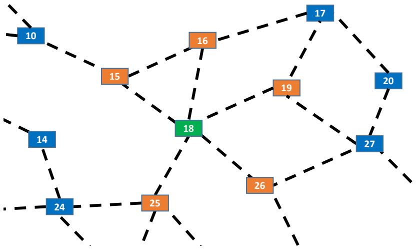


Fig. 3: Associating single-hop neighbours with cluster heads.

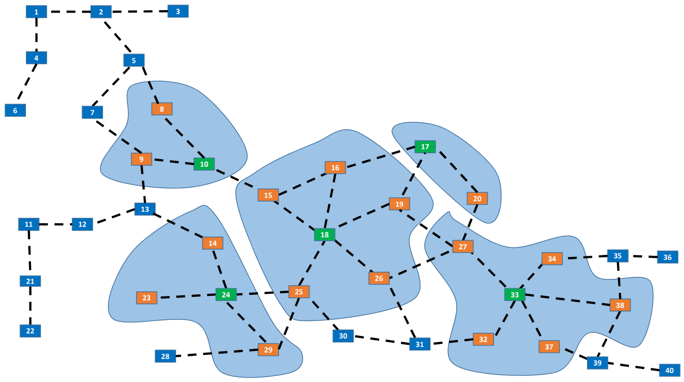


Fig. 4: Assignment of cluster heads based on *R-Index* and a minimum of two-hop distance between cluster heads.

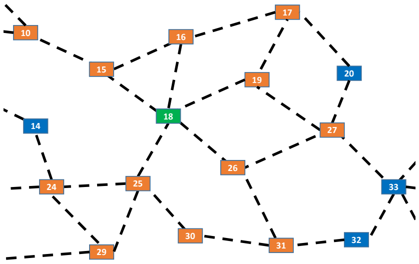


Fig. 5: Associating 2-hop neighbours with cluster heads.

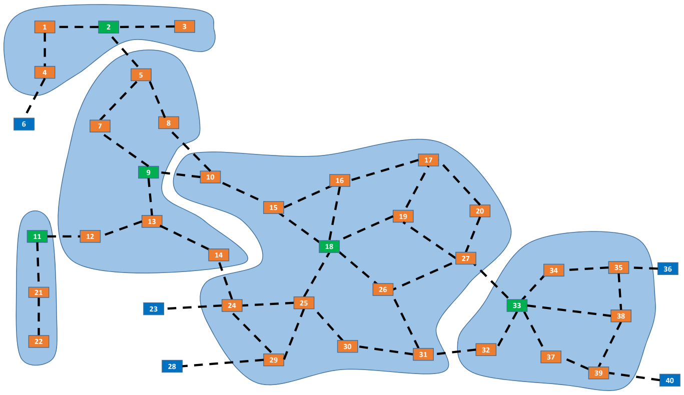
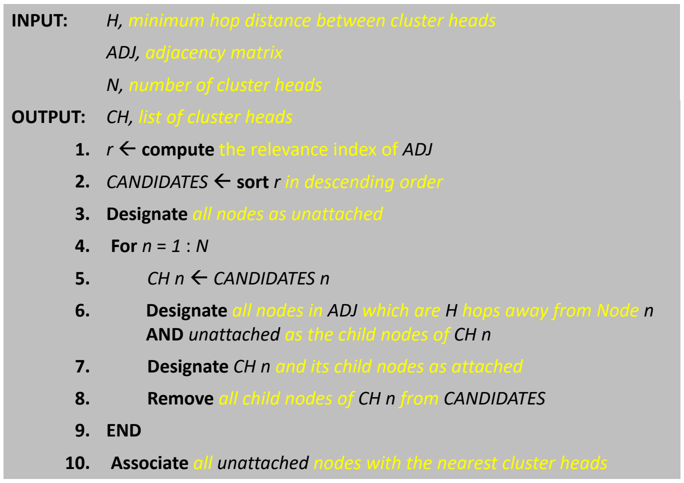


Fig. 6: Assignment of cluster heads based on *R-Index* and a minimum of three-hop distance between candidates.



```
INPUT:    H, minimum hop distance between cluster heads
          ADJ, adjacency matrix
          N, number of cluster heads
OUTPUT:   CH, list of cluster heads
          r ← compute the relevance index of ADJ
          CANDIDATES ← sort r in descending order
          Designate all nodes as unattached
          For n = 1 : N
              CH n ← CANDIDATES n
              Designate all nodes in ADJ which are H hops away from Node n
              AND unattached as the child nodes of CH n
              Designate CH n and its child nodes as attached
              Remove all child nodes of CH n from CANDIDATES
          END
          Associate all unattached nodes with the nearest cluster heads
```

Fig. 7: A partial overview of the clustering algorithm.

from being cluster heads even if they have a high *R-Index*. For example, node 25 is the third most connected node in the network but it is one-hop away from node 18, which is the most connected node in the network. Once node 18 is identified as a cluster head, the same decision automatically disqualifies node 25 from being eligible for the role of a cluster head. The second node eligible for the role is node 33, the third, 24, and so on.

Fig. 4 displays the cluster head assignment and the association of child nodes based on this constraint. Looking at the figure, one can notice that there are nodes which are more than one-hop away from the nearest cluster heads (the blue nodes, constituting 45 % of the nodes). These nodes experience a slightly longer delay in communication compared to the orange nodes. Furthermore, the cluster heads associating them as their child nodes will have a heavier computation load compared to the other cluster heads. To deal with this problem, we can designate more cluster heads. For example, if we increase the number of cluster heads to 8 (20 % of the nodes), we can reduce the number of leaf nodes which are more than 1-hop away from a nearest cluster head to 27.5 %.

Alternatively, we can define the minimum distance between two cluster nodes to be at least 3-hop. This disqualifies the nodes which are within two-hop away from a cluster head from being eligible for the role of a cluster head even though their *R-Index* is high. The assignment begins with the node whose *R-Index* is the highest in the network (node 18) and by associating all nodes within two-hops distance as its child nodes (ref. to Fig. 5). The next candidate cluster head is node 33 because it fulfils both criteria, namely, it (1) has the next top *R-Index* and (2) it is three-hops away from node 18. However, nodes 25, 27, and 24 do not qualify to be cluster heads even though they are ranked third, fourth, and fifth in Tab. 1 according to their *R-Index*[1]. Therefore, the next candidate node for a cluster head is node 9 and so on. Fig. 6 shows the cluster head distribution based on this criteria. Fig. 7 summarises our clustering algorithm.

In the introduction we highlighted that cluster heads are more

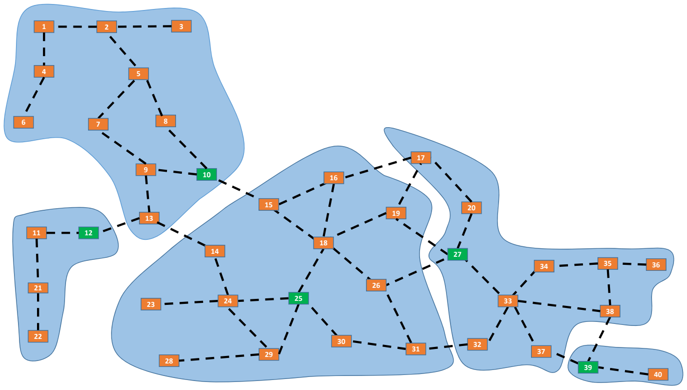


Fig. 8: Second round assignment of cluster heads based on *R-Index* and a minimum of three-hop distance between candidates.

[1] Indeed all the nodes ranked between third and thirteen are not eligible to be cluster heads because they are already associated with nodes 18 and 33 as their child nodes.

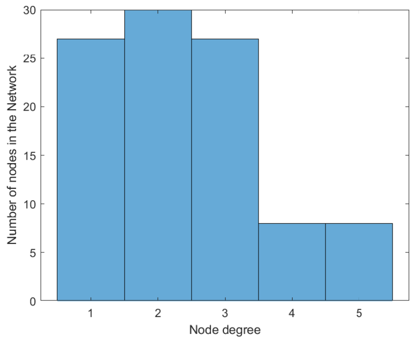


Fig. 9: The distribution of the degree of nodes (in percent) for the simulated networks.

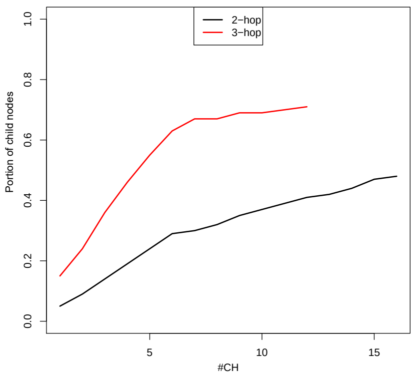


Fig. 10: The portion of child nodes as a function of cluster heads.

likely to exhaust their energy more quickly than child nodes. This necessitates the rotation of the role, so that all nodes exhaust their energy evenly. Our algorithm supports this feature. Thus, in the selection of cluster heads for round n, all nodes which were cluster heads up to round $n - 1$ will be regarded as ineligible. For the other nodes, however, the procedure in Fig. 7 will apply. Fig. 8 displays the clustering assignment for the second round. Notice that all the leaf nodes are associated with the nearest cluster heads even though they were farther than 2-hops away from them.

IV. EVALUATION

In order to evaluate the efficiency of our approach we simulated ten different wireless sensor networks, each time the size of the network was 100 nodes. Fig. 9 displays the aggregate node distribution normalised by the network size. Cluster-based protocols require global knowledge about the size and density of the network in order to determine the number of cluster heads. Moreover, at a local level, the nodes exchange information to advertise their candidacy and to evaluate their proximity with respect to the newly elected cluster heads. The latter is necessary for child nodes to choose their parent nodes. For our case, all this information can be directly obtained from the adjacency matrix alone. So, the only cost associated with our approach is the establishment of the adjacency matrix and its evaluation to determine the *R-Index*.

This said, the efficiency of our approach is influenced by the minimum distance between cluster heads – the shorter the distance, the more cluster nodes are needed to cover the whole network and the less uniform is the distribution of cluster heads in the network. The longer the distance, the fewer cluster heads are needed and the more uniform is their distribution. In both cases, as more and more cluster heads are added, less and less child nodes are associated with them. Fig. 10 exhibits this ratio when the minimum distance between cluster heads is set to 2-hop and 3-hop, respectively. In the first case (black), 15 cluster heads are required to associate about 50 % of the child nodes. In the second case (red), approximately 6 cluster heads are sufficient to associate more than 70 % of the child nodes, but a further addition of cluster heads does not bring any appreciable association.

Our approach complements the approaches proposed by Qing et al. [5] and Neamatollahi et al. [7]. In the former cluster heads are determined in a distributed manner, each node comparing its own energy with the average energy of the network to candidate itself as a cluster head. In the latter, the size of a cluster and the timing of clustering is partly determined by local events (tasks), in a distributed manner.

V. CONCLUSION

In this paper we proposed a simple clustering algorithm based on the evaluation of the adjacency matrix and the introduction of a minimum distance between cluster heads. Our strategy enables the assignment of cluster nodes for multiple rounds in a single step, assuming that the physical placement of nodes remains unchanged. Even in the presence of nodes failure, our strategy requires the recalculation of the adjacency matrix which requires local knowledge only. Our approach contributes to the deployment of wireless sensor networks in remote and inaccessible places. The assignment of cluster heads based on knowledge of their degree of connectivity enables to send mobile robots and drones specifically to these nodes for fast data aggregation and transfer as well as command dissemination.

REFERENCES

[1] Y.-Y. Zheng, Y.-T. Ma, J.-Y. Zhang, and X. Xie, “Covid-19 and the cardiovascular system,” *Nature Reviews Cardiology*, vol. 17, no. 5, pp. 259–260, 2020.

[2] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

[3] M. Hammoudeh, F. Al-Fayez, H. Lloyd, R. Newman, B. Adebisi, A. Bounceur, and A. Abuarqoub, “A wireless sensor network border monitoring system: Deployment issues and routing protocols,” *IEEE Sensors Journal*, vol. 17, no. 8, pp. 2572–2582, 2017.

[4] X. Bai, Z. Wang, L. Sheng, and Z. Wang, “Reliable data fusion of hierarchical wireless sensor networks with asynchronous measurement for greenhouse monitoring,” *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1036–1046, 2018.

[5] L. Qing, Q. Zhu, and M. Wang, “Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks,” *Computer communications*, vol. 29, no. 12, pp. 2230–2237, 2006.

[6] J. Wang, J. Cao, S. Ji, and J. H. Park, “Energy-efficient cluster-based dynamic routes adjustment approach for wireless sensor networks with mobile sinks,” *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3277–3290, 2017.

[7] P. Neamatollahi, S. Abrishami, M. Naghibzadeh, M. H. Y. Moghaddam, and O. Younis, “Hierarchical clustering-task scheduling policy in cluster-based wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 1876–1886, 2017.

[8] S. Bandyopadhyay and E. J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 3, pp. 1713–1723, IEEE, 2003.

[9] W. Dargie, “A quantitative measure of reliability for wireless sensor networks,” *IEEE Sensors Letters*, vol. 3, no. 8, pp. 1–4, 2019.