An Efficient Routing Algorithm for Wireless Sensor Networks based on Centrality Measures

Stephanie Mwika Mbiya, Gerhard P. Hancke, Bruno Silva

Advanced Sensor Networks Research Group, Department of Electrical, Electronic and Computer Engineering, University of Pretoria
Lynnwood Road, Pretoria 0002, Republic of South Africa
u14174406@tuks.co.za, gerhard.hancke@up.ac.za, bruno.silva@up.ac.za

Abstract: A routing algorithm for wireless sensor networks with a random distribution in a target observation area is proposed. In practice, selecting a path to route data from a source node to a destination node in a sensor network is very useful. An investigation is carried out on the combination of centrality measures and a routing algorithm to determine whether this can improve the route selected by the network's decision. Various measures of centrality are used and the network's response is evaluated with regards to the route selected by the network's decision when some nodes fail. It is demonstrated through simulations that controlling sensor nodes efficiently with a high measure of centrality gives a network the ability to resist node failures or attacks. Furthermore, this provides the network with high failure tolerance. In this paper, a routing algorithm that uses centrality measures to select the shortest path (a low-energy path between the source and destination node) is implemented.

Keywords: Graph Theory; Routing algorithm; Shortest path; low cost; Wireless Sensor Network

1 Introduction

The routing problem in wireless sensor networks is to select routing paths between nodes in the network so that data can be forwarded to the nearest node with the smallest distance (e.g. number of hops) within a random network. This also includes establishing connections from source nodes to the base station (BS).

Routing is required in higher level decision making when packets in the network have to be forwarded from their source to their destination via intermediate nodes by using various mechanisms to compute the distances between nodes within the network.

The main challenge of routing is to reduce the energy consumption without compromising the network's reliability, whilst keeping fault tolerance high.

One of the limitations of existing routing algorithms is fault tolerance. Fault tolerance is important as it enables the network to still function even if some nodes are disconnected. In some algorithms, there is considerable overhead involved as messages have to be exchanged between a large number of nodes when selecting routing paths. Algorithms which can select paths with a lower number of messages are preferable.

In literature, when routing algorithms choose a path, the data packet is collected through the network, and the source node sends it to the nearest nodes in the network, randomly. In this paper, we study the case where the data packet has been collected through the routing network but the source node does not send it to all other nodes at random; rather, it chooses which node to send the data packet to. We propose an algorithm where the closest node to the source node is chosen for data transmission.

In the proposed algorithm, the network has to be reliable regardless of connectivity.

This also includes investigating whether routing algorithms based on centrality measures can outperform routing algorithms such as Dijkstra's in wireless sensor networks.

The proposed algorithm computes the smallest number (Snbr) of nodes between the source and the nearest node. The nearest node with Snbr is treated as if it is a source node, and again the process is repeated to find the nearest node that has Snbr. The cost of this new found node becomes the total cost calculated from the original source node. This process is repeated until the destination node is reached. What distinguishes this dynamic approach to routing from the greedy approach is that the former will always lead to the optimal solution, while in the latter case, one is not assured of obtaining the optimal solution although the solution might be satisfactory.

The proposed algorithm uses a dynamic approach as opposed to Dijkstra's algorithm which uses a greedy approach.

- **Contribution:** In this paper, a shortest path routing algorithm is proposed pertaining to the calculation of the shortest distance among the connected nodes in a network. This proposed routing algorithm addresses key issues existent in routing algorithms such as Dijkstra.

  The research investigates choosing Snbr where 0 is the source or a permanent node denoted as S sending information to the connected node, which we call the tentative node and is the shortest distance away. This procedure is repeated for the previous node (which is denoted as permanent node) until the process reaches the destination node.

- The existing routing algorithm such as Dijkstra Algorithm is compared to the proposed algorithm based on centrality measures. However the result shows in the implementation has beter performance in term of releability of nodes within the network can further connect nodes for WSNs and reducing the energy consumption.

2 Network Modelling

2.1 Overview of Centrality Metrics

This subsection starts with some notions required to understand centrality measures. Centrality measures (or metrics) depend on the shortest paths between two nodes. In Figure 1, we observe that although 8 and 5 have the largest measure of betweenness, 0 is top in terms of closeness. This is intuitive, as 6 seems to fill a more central position. On the other hand, 8, whose ranking is above 2 with regards to closeness and degree, takes a second position as far as betweenness is concerned. A routing path to 6 or 2 has to go through 8. This is illustrated in Figure 2.

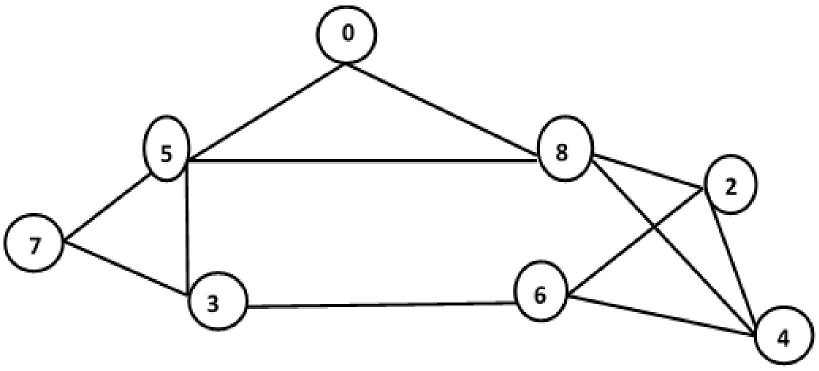

Figure 1
A simple network

Some necessary notions to understand centrality measures include:

- Walk: A walk is a sequence of edges and vertices, where each of the end edges are two vertices which are adjacent.
- Trail: A trail is a walk with no repeated edges.

- Gossip: When gossiping is used as a routing method, data from a node is forwarded to a randomly selected neighbor node sequentially until a packet reaches the destination node

For instance, Figure 2 indicates that route 8, 2, 4, 6, 2 is neither a path nor a trail. Some information, e.g., a gossip, usually accelerates on a trail. Thus, 4 may hear a gossip from 2 as well as 6, where 4 could hear it from 6, but the chance that 6 and 4 can gossip back to each other is not high. Figure 2 (B) indicates that route 8, 2, 4, 6 forms a walk. It is neither a trail nor a path. Communication between node A and node B can be reciprocal; 6 can write a bill to 4 in a single transaction, 4 can return it to 6 in another. Figure 2 (A) and 2 (B) show nodes that have high closeness and betweenness.

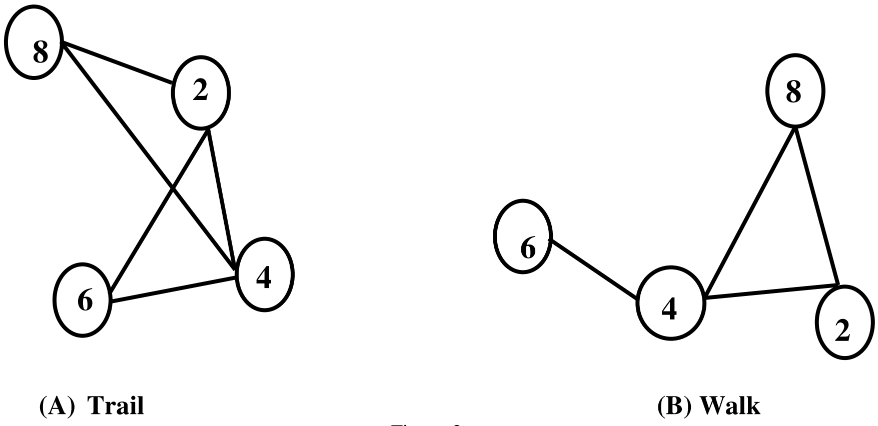

Figure 2

Different nodes with different degrees of closeness

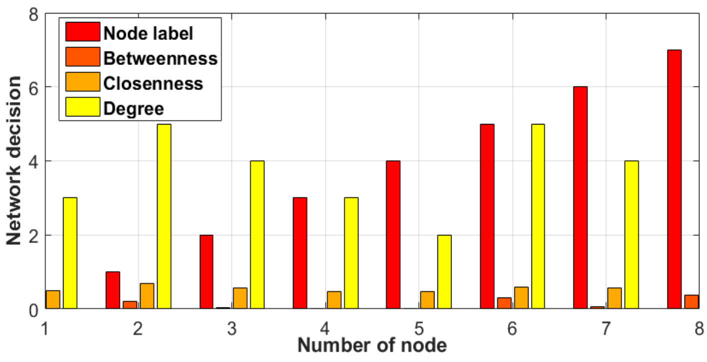

Figure 3

Centrality measures for the network

2.2 Modelling of Centrality Measures

It is possible to have a mathematical representation of a network. In this representation, a network is visualised as a graph of nodes and links. The links are edges that connect the nodes. Therefore, a network can be defined as $G(V, E)$:

$$G(V,E) = \{(v, e): v \in V, e \in E\} \tag{1}$$

in which V is the set of nodes and E the set of links (or edges). The links in the network represent paths of communication. Alternatively, instead of writing $G(V, E)$, we simply use the notation (V, E) [1], [2], [3].

The connection between sensors can be represented by a matrix, called the adjacency matrix. Denote this adjacency matrix by A. The matrix A has order $n * n$ where n is the total number of sensors. An entry of A, a_{ij} is given by:

$$\sigma(p_j, p_i) = \begin{cases} 1, & if\ p_{i\ is\ connected\ to\ p_j} \\ 0, & Otherwise\ . \end{cases} \tag{2}$$

Here, sensors are labelled $1,2,3, \ldots.$ up to n and sensor p_i is represented by the label i. Note that A is symmetric if sensor p_i is connected to sensor p_j, then p_i is also connected to p_i and we assume that a given sensor p_i is not connected to itself. Therefore $a_{ii} = 0$.

In this section, we denote the distance between sensors p_i and p_j by $w(i, j)$. The shortest path between sensors is the path whose number of links that connect sensor nodes is minimal. If there is no path between p_i and p_j then $w(i, j) = \infty$. The network's diameter is calculated by taking the average of the shortest paths between two pairs of nodes [4], [5], [6], [7].

Extensive research has been done on probabilistic graphs. In this study, we concentrate on sensors that are static where they acquire information, execute decisions and exchange data with neighbours. The data must be communicated to a sink node.

Utilizing the mathematical framework in a network, many algorithms have been developed to study networks and measures have been defined. Centrality measures provide information about network features and how data is spread over the network.

- **Connectivity centrality metric**

The degree of centrality of node p_i is simply the number of links connected to it [3], [2], [4]. It is computed from the formula:

$$\deg(p_i) = \textstyle\sum_{p_j \in V} \sigma(p_j, p_i), \tag{3}$$

where,

$$\sigma(p_j, p_i) = \begin{cases} 1, & if\ p_{i\ is\ connected\ to\ p_j} \\ 0, & Otherwise\ . \end{cases} \tag{4}$$

With regards to the matrix A, we can alternatively define $\deg(p_i)$ to be the sum of all entries in row i or column i, i.e.,

$$\deg(p_i)\sum_{j=1}^{n} a_{ij} = \sum_{j=1}^{n} a_{ji}. \tag{5}$$

For a sensor p_i, $\deg(p_i)$ gives an indication of the sensor's impact on how information is communicated in a network. This occurs in such a way that the more connections a sensor has, the higher the probability of making a high contribution to the communication of information. Hence, such sensors are crucial for information transfer. Given a network G with n nodes, one defines the normalized degree of centrality $CD(p)$ for node p as:

$$C_D(p) = \frac{\deg(p)}{n-1}. \tag{6}$$

where one can extend $C_D(p)$ to the entire network.

Let a node p^* be such that $\deg(p^*)$ is the greatest. Furthermore, let X be the component (connected) of G that maximizes the quantity H given by,

$$H = m\sum_{j=1}^{n_x}[C_D(y^*) - C_D(y_j)]. \tag{7}$$

where n_x enumerates nodes in X, y^* the node with highest degree centrality in X, and y_j is a node in X. Hence, the centrality degree of G is the following Quantity C_D,

$$C_D = \frac{\sum_{i=1}^{n} C_D(p^*) - C_D(p_i)}{H}. \tag{8}$$

Note that when G is connected, then H is maximal. A connected graph is a graph where every node is linked to all nodes. In that case, $H = (n-1)(n-2)$, and C_D becomes [2], [3], [8], [9].

$$C_D = \frac{\sum_{i=1}^{n} C_D(p^*) - C_D(p_i)}{(n-1)(n-2)} \tag{9}$$

Another assumption is that $n \geq 3$.

- **Closeness centrality metric**

The closeness centrality metric of a node p_i is the reciprocal of the total number of path lengths that are the shortest distance from the rest of the nodes [3], [4], [5]. In a network that is connected, the centrality of a node p_i is calculated from the status of p_i and the average distance to all other nodes. We denote the status of a node p_i by S_{p_i}. This implies that the status of a node p_i is the ratio of the sum $w(i,j)$ for all nodes p_j to the total number of such possible paths, i.e., $n(n-1)$. Hence, the status of a node p_i is given by [10], [11]:

$$S_{p_i} = \frac{1}{n=(n-1)}\sum_{j=1}^{n} w(ij), \tag{10}$$

The closeness centrality metric Cp_i of a node P_i is the reciprocal of its status. It represents the extent to which the node is able to acquire information through other nodes and relay it. The central node has a high closeness metric because it is the sink. On average, the nodes that have the closest proximity are those which are positioned at a smaller number of hops to other nodes, and form a group of nodes which enable higher information transfer in the network. The capacity of the network is given by:

$$G = \{p_i \in V : C(p_i) = max_{p_i} \in VC(p_j)\}. \tag{11}$$

- **Betweenness centrality metric**

Betweenness centrality measures the ability of a node to take the shortest paths [4]. Nodes that appear on several shortest paths possess higher betweenness centrality. This metric reflects the impact that a node has on others in the network and also shows its importance for information transfer [3], [12], [13], [14]. We denote the betweenness centrality measure of a node v by $B(v)$, given by:

$$B(v) = \sum_{(p,t)\in V_v} \frac{\sigma_{pt}(v)}{\sigma_{pt}} \tag{12}$$

where $\sigma_{pt}(v)$ enumerates the shortest paths from node p to node t passing through v, and $\sigma_{pt}(v)$ enumerates shortest paths from p to t. Furthermore, V_v is given by:

$$V_v = \{(p,t) \in V^2 : p \neq v \neq t, p \neq t\}. \tag{13}$$

The normalized betweenness of a node v is given by:

$$B_N(v) = \frac{1}{(n-1)(n-2)} \sum_{(pt)\in V_v} \frac{\sigma_{pt}(v)}{\sigma_{pt}}, \tag{14}$$

It follows that nodes with high B_v have a high capability of bridging or disconnecting the network. Therefore, such nodes are crucial for robustness, integrity and continued communication in the network [5], [11].

3 Discussion of the Proposed Algorithm

Dijkstra's algorithm lies in Bellman's Principle of Optimality, and both algorithms are based on an optimization method called dynamic programming. Dijkstra's algorithm is used to compute the minimum cost or shortest path between one node and all other nodes when the vertices of the graph represent nodes and path costs are represented by path distances between pairs of nodes connected by a direct link. The proposed algorithm computes the shortest path between two nodes. It uses a graph consisting of nodes and edges. The cost of each node is calculated from the source by summing up the cost up to that node. For a given source vertex (i.e. node), the algorithm works out all paths and computes the path with the lowest cost, (i.e. the shortest distance between the source node and any other node) [15], [16].

This algorithm offers another method of computing the costs of the shortest paths from a single source node to a single destination node. The goal of both algorithms, the proposed algorithm and Dijkstra's algorithm, is to select the nodes in the shortest path problem. The main difference in the algorithms is that Dijkstra carries the overall information of the network and every node is involved, while the proposed algorithm deals only with the nearest node, and not all nodes are involved as in Djikstra's algorithm [17], [18], [19].

Figure 4 shows a flowchart that illustrates the steps performed by the proposed routing algorithm from start to end. Firstly, the algorithm identifies the source and destination nodes, denoted R1 and R2, respectively, and a cost of 0 is assigned to the source node. Subsequently, another node is selected and labeled as P (or permanent node). This node then becomes a tentative node and another node close to it is selected and labeled as P. This procedure is repeated until the destination node is reached [20].

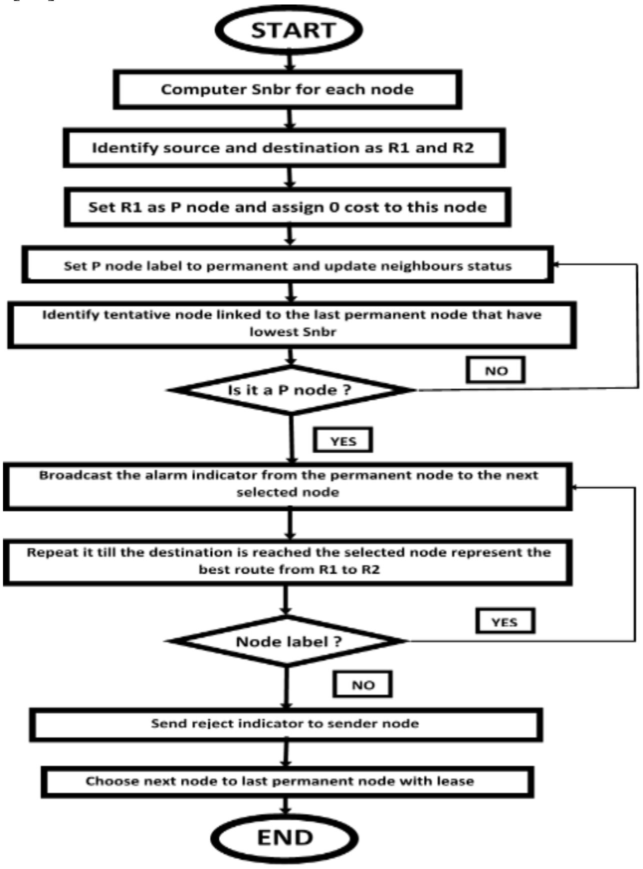
START
Computer Snbr for each node
Identify source and destination as R1 and R2
Set R1 as P node and assign 0 cost to this node
Set P node label to permanent and update neighbours status
Identify tentative node linked to the last permanent node that have lowest Snbr
Is it a P node ?
NO
YES
Broadcast the alarm indicator from the permanent node to the next selected node
Repeat it till the destination is reached the selected node represent the best route from R1 to R2
Node label ?
YES
NO
Send reject indicator to sender node
Choose next node to last permanent node with lease
END

Figure 4
Flow chart of the proposed algorithm

4 Implementation and Results

4.1 Simulation

We investigate the behaviour of a network that finds a route between source and destination nodes, with particular focus on the routing algorithm. We confine the analysis to the case where some nodes fail in the network. The impact of random node losses on network performance is also measured.

A random deployment of n nodes is generated in an $80m \times 80m$ network field. The origin $(0,0)$ is located at the bottom left corner of the square. This square, shown in Figure 5, is in the first quadrant with the x and y-axes. We assume that the node collecting the information from the source node is in location $(4.0, 48.0)$. The node nearest to the sink is the node that connects the sensor network to the sink. All data must be transmitted through this node to reach the sink. Within the network, two nodes are said to be connected as long as the distance between them is smaller than the length between the communicating nodes and the BS (or sink). The state of the node is either 0 or 1. When a node is connected to another node, its state is 1. Otherwise, it is 0. Nodes are randomly distributed. In our simulation, to generate 100 different states, different random seeds were applied. Every simulation is executed 3 times for different network sizes $(i.e.\ 30\ nodes)$, in which the state samples are independent. For each network size, the centrality measures are computed. Table I shows details of the simulation setup.

Table 1

Simulation Setup

Parameter	Value
Deployment area	80m x 80m
Maximum number of nodes	100
Source node	1
Sink node	1
Number of topologies (runs) per experiment	100
The origin	(0,0)
Node distribution	Uniform
Position coordinate	(4.0, 48.0)
Number of simulation runs	3 times for each network size

4.2 Decision Making in the Network

Decision making in the network is the number of failed sensors. The network can be regarded as a collection of sensor nodes executing decisions. A decision that a sensor executes depends on its own decision at a particular time as well as the decisions of its closest neighbors. The formula for the decision of the node p_i at time k is:

$$\text{Deg}\, p_i = \frac{\sum_{i=1}^{N_i} d_i}{N_i} \tag{15}$$

where $Ni = deg(p_i) + 1$, d_i is node i, and 1 is added because the "neighbourhood" includes node p_i itself.

We assume that node i detects an event that it uses in executing its unilateral decision. The methods employed in making decisions that are followed by the nodes are those of decentralized data fusion systems such as those shown in [6]. A few requirements are introduced in the network. Firstly, the data should eventually reach the sink node. The distance from one node to other nodes is determined based on the Snbr, where the Snbr is used to select nodes as relays to forward data.

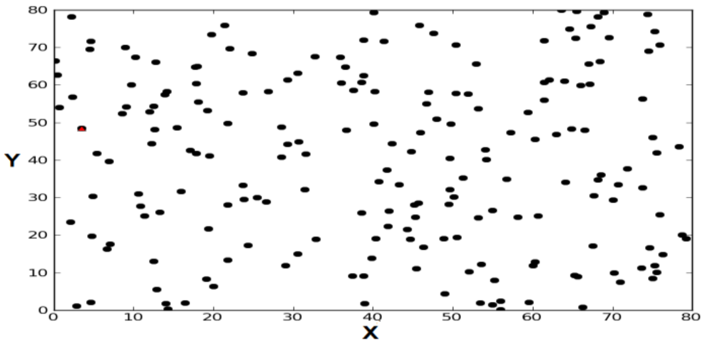

Figure 5
Random network

In the context of a network where there are human agents, it may be the decision maker who decides the next possible action for the group. In this situation, one node or point must decide to stimulate the network into action. Eventually, the decisions made by the individual nodes arrive at this central point. In the simulation, all nodes apart from the sink are displayed. In each case, the node nearest to the sink is the grey node (shown in Figure 5), which takes the role of decision-maker as well as the final link between the sensor network and the action command center.

4.3 Test for Degree, Closeness and Betweenness Metrics

A network of 30 nodes is shown in Figure 6. The gray-colored node is the one nearest to the BS. This reveals that as the sensors tend to be near each other, (observe the betweenness and closeness centrality measures in Figure 7), the sensor disruption has a negative impact on the network. Eventually, this will lead to a higher energy consumption on the entire network as the distance between nodes increases. This shows that with regards to the network's topology, the network may be tolerant to sensor failures, however, higher overall energy consumption can still be seen as the network tries to maintain network decisions.

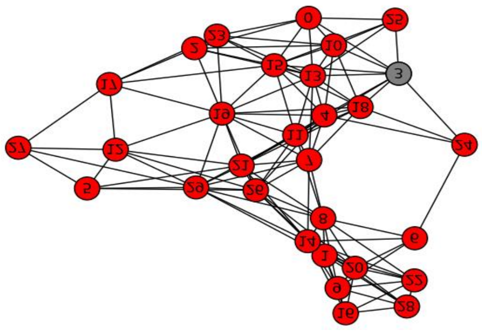

Figure 6
Network of 30 nodes

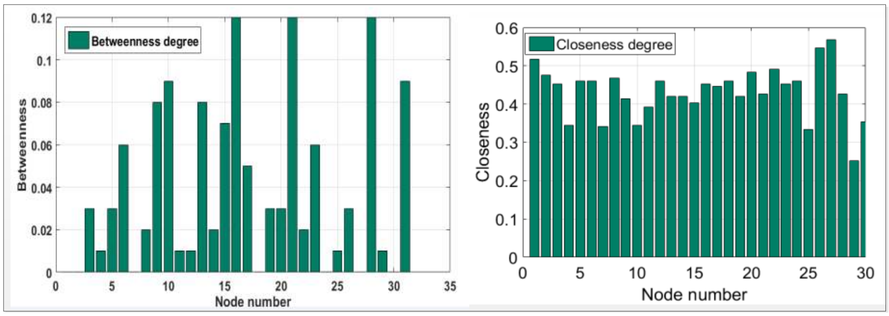

Figure 7
The impact of betweenness and closeness centrality measures of sensors on the network decisions of 30 nodes

The network illustrated in Figure 8 is tested under varying random failure situations. To simulate this, a number of nodes are excluded randomly. For instance, the network decision is still well-respected when a node is excluded. The single node omission is repeated in the network and the average network measures are determined. The goal of exclusions is to execute a network test where some of the sensors are left out. The exclusion of a node is randomly done for a number of sensors that are eliminated per execution.

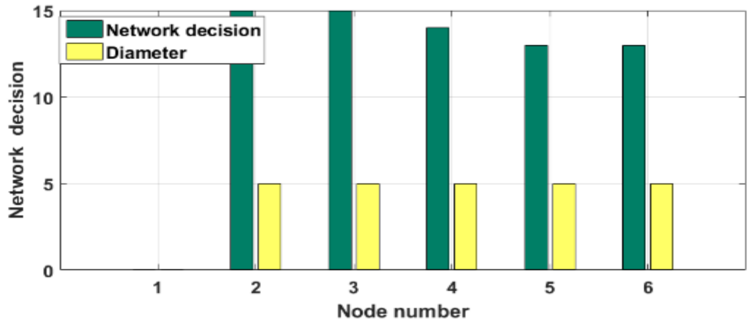

Figure 8
Network decision in terms of the percentage of failed sensor nodes for 30 nodes network test

Note that the objective is to determine the impact of random failures (or attacks) on the network and to determine the stability of the network's decision-making in the event that nodes are excluded. Finally, for every network, the impact of the exclusions is determined.

4.4 Results

This section evaluates the behaviour of the proposed algorithm in a network and presents the results. In this experiment, the proposed algorithm based on centrality measures is used to find the shortest path (i.e. path with the lowest energy consumption) in a network. The algorithm used to determine the shortest path is implemented in MATLAB.

The first step is to calculate the time it takes to find a route between source and destination nodes. The individual paths between nodes are traced until the overall path reaches the target (i.e. destination) node. In the proposed algorithm, the execution period is defined as time elapsed between event detection at the source node and data delivery to the destination node. However, the deployment of the specific node closest to the sink node has to be considered and it is because of this requirement that a multitude of routing algorithms exist. Centrality measures help reduce the time taken to select intermediate routing nodes between the source node and destination nodes, which is equivalent to minimizing the shortest distance from one node to another. The algorithm expands the first nodes connected to the intermediate node with small number (Snbr) to ensure that energy consumption and congestion are reduced when data is transmitted.

The general steps to select a forwarding node in the proposed algorithm are as follows. In the network graph representation, a path can be found between a starting point and end point in the graph. Firstly, a vertex from the graph is chosen as the starting point. The degree of an edge (i.e node) is the number of vertices connecting said edge to the adjacent edges. The current node attempts to find an adjacent node with a small number. This is repeated for every adjacent node, until the node with the smallest number is selected, and that node becomes the current node. This procedure is repeated until the destination node is reached. Then, all paths from the beginning to the end vertex in the graph are found and lastly it is determined whether the graph is connected. If one considers the starting point (i.e. source node) to be the same for both the proposed algorithm and Dijkstra's, Dijkstra's algorithm needs to visit all nodes in the network before it is able to select a path, whilst the proposed algorithm visits only the nearest nodes when selecting the shortest path from source to destination.

A comparison between the running (i.e. execution) times of Dijsktra and the proposed algorithm is shown in Table 2. Each algorithm is executed ten times. The algorithm was also executed for 100 times and the results were found to be statistically similar to 10 times. Only the results for 10 executions are shown here because it's easier to visualize. The execution times in seconds for both algorithms are shown in Table 2, where the execution time for the proposed algorithm (PA) is 0.000118017196655 and the execution time for Dijkstra is 0.000144004821777 for the first execution. The results show that there is a difference in the execution times and the proposed algorithm is faster with regards to the average execution time.

Table 2
Running time for simulation in seconds

Execution number	**Proposed algorithm (10^{-4})**	**Dijkstra's algorithm (10^{-4})**
1	1.2	1.4
2	1.1	2.6
3	2.2	1.5
4	2.7	1.1
5	1.3	4.4
6	1.1	1.7
7	1.3	2.7
8	1.6	1.5
9	1.3	1.3
10	1.9	4.9
Average	1.6	2.3

Figure 9 shows a more clear representation of the data from Table 2 using a bar-stacked graph. It clearly shows that overall, the proposed algorithm executes faster than Dijkstra. Figures 10 shows the execution times for both algorithms over 10 executions. It is seen that the worst-case execution time for Dijkstra's algorithm is 4.67 and for the proposed algorithm it is 2.6, showing that the proposed algorithm is indeed faster.

It has also been shown that the proposed algorithm consistently chooses shorter paths between nodes than Dijsktra. This is attributed to the use of centrality measures.

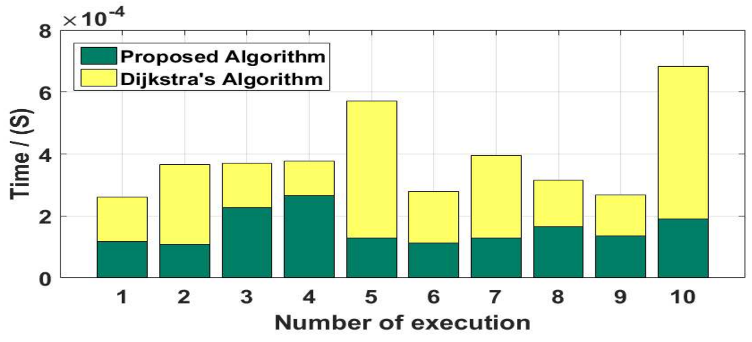

Figure 9

Time response comparison for the bar-stacked chart

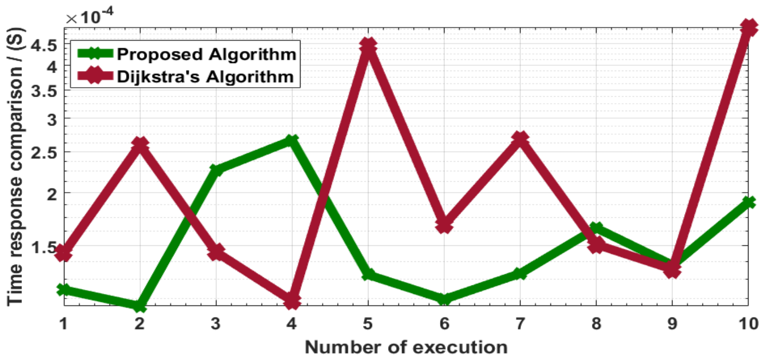

Figure10

Time response comparison for both algorithms

Conclusion

A study on routing using centrality measures has been conducted. The study enabled us to analyze how information is disseminated over a network, from source node to nearest nodes and finally to the base station via the shortest path. The network's performance can be evaluated by utilising the degree, closeness and betweenness centrality measures, as well as the shortest path carrying data from one node to another until the sink nodenis reached. We conducted an analysis on certain characteristics of sensor networks to address different challenges encountered in a network. These include reliability, failure tolerance and robustness. These three characteristics enable the network to perform better. Each of the characteristics has a role to play, especially failure tolerance. Failure tolerance is crucial if a network has to keep making network decisions that are stable even when some nodes are disconnected. Currently, centrality measures are used for many applications in sensor networks. One of these applications is finding a routing path within a sensor network. In this study, an algorithm that finds a routing path with the shortest distance between nodes is implemented. The proposed algorithm lowers the energy consumption and thus increases the network's lifetime. When considering a routing algorithm for sensor networks, resilience against attacks or failure is very important. Using simulations, the impact of nodes on the network's performance via centrality measures has been shown. It was observed that the proposed algorithm is faster than Dijkstra's algorithm in terms of execution time. The results show that the proposed routing algorithm based on centrality measures outperforms Dijkstra's algorithm in terms of connectivity, and centrality measures (i.e. Degree, Closeness and Betweenness) can be used for routing in sensor networks, with the Betweenness centrality measure outperforming the centrality measures. The work presented in this paper can be extended by changing some simulation parameters such as the number of simulation repetitions in order to extend the results. Additionally, this work can be extended by considering other metrics such as packet delivery ratio and congestion level in the network.

References

[1] S. M. Mwika and S. W. Utete, "Sensor networks for detecting events," 2011, unpublished Essay

[2] D. J. Higham, P. Grindrod, and E. Estrada, "People who read this article also read...: Part 1," *SIAM News*, Vol. 44, No. 1, January/February 2011

[3] B. Ruhnau:, "Eigenvector-centrality a node-centrality?" *Social Networks*, Vol. 22, pp. 357-365, 2000

[4] L. Freeman, "A set of measures of centrality based on betweenness, sociometry," *Sociometry*, Vol. 40, pp. 35-41, March, 1977

[5] C. E. Shannon, A mathematical theory of Communication. Urbana: University of Illinois Press, 1948

[6] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor networks," Control Engineering Practice, Vol. 2, No. 5, pp. 849-863, October, 1994

[7] A. Jain and B. Reddy, "Node centrality in wireless sensor networks: Importance, applications and advances," in Advance Computing Conference (IACC), *2013 IEEE 3rd International.* IEEE, 2013, pp. 127-131

[8] E. J. Segovia and P. Vila, "New applications of the betweenness centrality concept to reliability driven routing," in VIII Workshop in G/MPLS Networks, 2009

[9] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in physics*, Vol. 56, No. 1, pp. 167-242, 2007

[10] C. E. Shannon, A mathematical theory of Communication. Urbana: University of Illinois Press, 1948

[11] Z. Chen and H. Qi, "A distributed and shortest-path-based algorithm for maximum cover set problem in wireless sensor networks," *IEEE International Conference on Trust, Security and Privacy in Computing and Communications,* pp. 1224-1228, November 16, 2011

[12] M. Z. Siam, M. Krunz, A. Muqattash, and S. Cui, "Adaptive multi-antenna power control in wireless networks," in Proceedings of the 2006 international conference on Wireless communications and mobile computing. ACM, 2006, pp. 875-880

[13] M. A. Othman and H. A. Sulaiman, "An analysis of least-cost routing using bellman ford and dijkstra algorithms in wireless routing network," International Journal of Advancements in Computing Technology, Vol. 5, No. 10, June, 2013

[14] L. Sitanayah, K. N. Brown, and C. J. Sreenan, "Fault-tolerant relay deployment based on lengthconstrained connectivity and rerouting centrality in wireless sensor networks," in European Conference on Wireless Sensor Networks. Springer, 2012, pp. 115-130

[15] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad-hoc networking environment," Proceedings of IEEE INFOCOM , Anchorage AK, 2001

[16] S.-C. Wang, D. S. Wei, and S.-Y. Kuo, "An spt-based topology control algorithm for wireless ad hoc networks," Computer communications, Vol. 29, No. 16, pp. 3092-3103, 2006

[17] J. Wu and M. Gao, "Stojmenovic. on calculating power-aware connected dominating sets for efficient routing in ad-hoc wireless networks," Proceedings of the 30th Annual International Conference on Parallel Processing, September, 2001

[18] M. Dramski, "A comparison between dijkstra algorithm and simplified ant colony optimization in navigation," Zeszyty Naukowe/Akademia Morska w Szczecinie, pp. 25-29, 2012

[19] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," IEEE wireless communications, Vol. 11, No. 6, pp. 6-28, 2004

[20] V. Rodoplu and T. Meng, "Minimum-energy mobile wireless networks revisited," IEEE Journal of Selected Areas in Communications, Vol. 17, No. 8, pp. 1333-1344, 1999

[21] YJ. Jang, SY. Bae, SK. Lee. "An energy-Efficient routing algorithm in wireless sensor networks". In : Kim T. et al. (eds) Future Generation

information thechnology, Lecture notes in computer Science, Springer, Berlin, Heidelberg. Vol. 7105, pp. 183-189, FGIT 2011

[22] V. Jose, Deepa and G, Sadashivappa. "A novel Energy efficient Routing algoritm for wireless sensor networks using sink mobility." Internation Journal of Wireless and Mobile Networks. No. 6, pp. 15-25, 10.5121/ ijwmn, 2014