Using Hidden Markov Chain for Improving the Dependability of Safety-Critical WSNs

Issam Alnader, Aboubaker Lasebae, Rand Raheem
I.Al-Nader@mdx.ac.uk, A.Lasebae@mdx.ac.uk, R.H.Raheem@mdx.ac.uk
Middlesex University, The Burroughs, Hendon, London NW4 4BT

Abstract. Wireless Sensor Networks (WSNs) are distributed network systems used in a wide range of applications, including safety-critical systems. The latter provide critical services, often concerned with human life or assets. Therefore, ensuring the dependability requirements of Safety critical systems is of paramount importance. The purpose of this paper is to utilize the Hidden Markov Model (HMM) to elongate the service availability of WSNs by increasing the time it takes a node to become obsolete, via optimal load balancing. We propose an HMM-algorithm that, given a WSN, analyses and predicts undesirable situations, notably, nodes dying unexpectedly or prematurely. We apply this technique to improve on the Randomized coverage-based scheduling algorithm (RCS) by C. Lius, a scheduling-based algorithm that has served to improve the lifetime of WSNs. Our experiments show that our HMM technique improves the lifetime of the network, achieved by detecting nodes that die early and rebalancing their load. Our technique can also be used for diagnosis and provide maintenance warnings to WSN system administrators. Finally, our technique can be used to improve algorithms other than the RCS.

1 Introduction

Thanks to their easy deployment, WSNs have been used in a wide range of application systems, from simple detection systems such as humidity detection [1] to safety critical systems such as intrusion detection systems [2][3][4][5]. Our research findings revealed weaknesses in the Randomised Coverage Based Scheduling (RCS) algorithm [6] and bridged the gap in increasing the WSN's lifetime, service continuity, reliability, and availability. Through our analysis, we captured scenarios whereby some nodes are over-worked, while others are under-worked, hence compromising the availability and lifetime of the network. The reason for this is the randomized nature of the RCS algorithm, which makes optimal load balancing for future performance optimization a challenging task. For example, towards the end of the network life-cycle, some nodes are found to be already obsolete ("dead"), which can lead to breaking the WSN into different segregations. This situation can lead to data loss due to the inability to transmit messages across the WSN.

Our main objective here is to work on multi-objective optimisation, i.e., trade-offs between network lifetime (service availability) and network coverage & connectivity. Therefore, the aim is to address the problem of improving the system dependability of safety-critical WSNs, thus achieving their quality of service (QoS) requirements [7] [19]. The RCS algorithm in [6] already works towards optimising network coverage. In this work, we extend network lifetime optimisation by using a Hidden Markov Model or HMM [8] to reason about WSNs. As we recreated the RCS algorithm in our simulation environment, we were able to achieve two goals: (1) avoid duplicating efforts in designing a scheduling algorithm for WSN while RCS already exists, (2) improve on RCS optimisation algorithm by increasing the network's lifetime via optimising the nodes' ON states. Thus, this work acknowledges reaching a state of obsoleteness is an inevitable state for nodes. However, delaying reaching this state is useful and has arguably economic benefits. The work utilised HMM to intelligently extend the usability of nodes to collectively increase the service availability & reliability of the WSN [9].

The authors of this paper acknowledge to the best of their ability that the literature lacks focus on extending the WSN lifetime while maintaining coverage and connectivity in a significant manner [10] [11] [12]. Related work in the literature has focused on extending the last remaining nodes alive, while we aim for extending the collective lifetime of the WSN. Consequently, another objective in this work is to guarantee minimum QoS levels e.g. connected coverage sensor networks, even in undesired situations such as example, when the WSN becomes segregated due to nodes' premature death. The paper is structured as follows; Section 2 reviews related work in the literature, while the proposed new HMM scheduling algorithm with its mathematical representation is introduced in Section 3. Section 4 covers the results and compares the proposed HMM algorithm to the RCS algorithm. Finally, Section 5 is the conclusion drawn from this work.

2 Literature Review

HMM is one of the most common methods used in unsupervised learning. HMM is a hidden state machine in which each state has a certain transition probability to another state. Each transition state generates an observation state, which follows a law of probability associated with the current state. The observations can be discrete or continuous. In the case where they are discrete, each state will be associated with the probability of making the observation of each of the possible discrete symbols. In the case where they are continuous, each state will be associated with a density function (often a Gaussian mixing model) as depicted from the definition of [13]. There are several papers in the literature that use the HMM method to address multiple problems that are related to energy consumption in WSNs. Therefore, the use of the HMM method in WSNs is not new [8] [14]. What follows shed the light on some related work that utilizes HMM methods in WSNs.

The use of HMM methods in WSNs was presented by [15], where a scheduling algorithm that utilized HMM method to observe states for a resource-constrained WSN is proposed. Krishnamurthy, 2002 [15] addresses the problem of data reliability, of obtaining certain physical data, e.g. measurements of a certain process known as signal processing application. The basic principle is to utilize the HMM finite states to locate noisy sensor nodes in the WSN, then select these nodes to send the required data. The solution utilized dynamic stochastic programming which is achieved in two folds: (1) By finding the optimal channel allocation for various components of a measurement vector, for example, when a noise sensor must transmit over a time-shared communication channel with limited bandwidth; (2) By finding the optimal time of measurements of the sensor when the number of possible measurements is limited because of energy constraints [15].

Goudarzi et al., 2010 [16] propose the use of HMM method in combination with particle swarm optimization (PSO) to predict the energy level in WSN. The algorithm uses PSO to select the cluster head Nodes. The proposed method reduces the cost of clustering and improves the performance of the WSN. The optimization problem defined for the best possible energy values can be obtained by solving it with PSO to find the best position that is evaluated from the fitness function. To initialize the PSO all the defined variables are assigned random values in the search space. In each run, each particle generates the best individual position (associated with energy) and global best position to obtain the best possible solution in search space. The communication overhead is reduced due to the use of PSO. However, there is always a trade-off between the cost of finding the energy versus the accuracy [16]. If the cost is higher the accuracy is better and a lower cost leads to lower accuracy. Hence, we have to work between these extremes. This solution introduces an overhead in the WSN as a result of the complexity involved in using the PSO algorithm at the expense of obtaining a good level of the accurate energy solution.

Qihua et al., 2015, [17] propose a scheduling algorithm to extend the network lifetime. The HMM is utilized to model sensor node states in the WSN where a node can be in two states: 0 or asleep and 1 or active. The work in [17] bases its HMM method on two factors that are the energy cost and the errors of the sensor nodes reading. In addition, they consider the use of an actuator that takes the reading of the node's energy and sends that data to the central controller, which manages the entire WSN. Qihua et al.' algorithm [17] seems to improve the lifetime of the WSN, however, the introduction of a coordinator leads to an overhead control message on the WSN.

In summary, to our best knowledge, there are not many works that utilize HMM to address multi-objective optimization problems involving energy consumption, network lifetime, coverage, and connectivity, in the context of the WSNs scheduling approach. In comparison, our work utilizes a very simple HMM algorithm to optimize the energy as well as the position of nodes for effective scheduling and energy management. Hence, our approach utilizes the HMM method to extend the network lifetime and introduces no communication overhead.

3 Problem Formulation

A WSN network is composed of sensors connected through radio links in a given target area, where the function of the WSN is embedded in the sensors [18]. In an HMM, a node/sensor is modeled as a state-transition diagram, indicating the possible states of the node and the transitions from one state to the other. Each transition is assigned a probability for its occurrence. The state of the WSN is then given by the states of all its nodes. The possible states of a sensor vary depending on the application. Typical states include a Transmission state and a Receiving state. Other possible states include Active, Sleep, Relay, Idle, and Fail, amongst others. Different states consume different levels of energy.

Here, we distinguish the ON state, when the sensor is performing some action, from the OFF state, where no energy is consumed. Problem: we are trying to optimize energy consumption in the ON state, by switching the sensor from the (ON, Tx) transmission state to the (ON, Rx) receiving state.

The following are conditions to be considered for switching:

1. The energy level (threshold energy level) of nodes
2. The probability assignment (of transition between states), based on the distance of a node from the sink
   1. The initial probability assignment is proportional to the threshold energy levels and the distance from the sink, for send and receive states.
   2. We call hidden states those that lie between (ON, Tx) and (ON, Rx), e.g., Sleep. The probability of their occurrence is randomly derived from the immediate probability assigned to the parent node such that the sum of the distributed probabilities is equal to the probability assigned to the parent node.

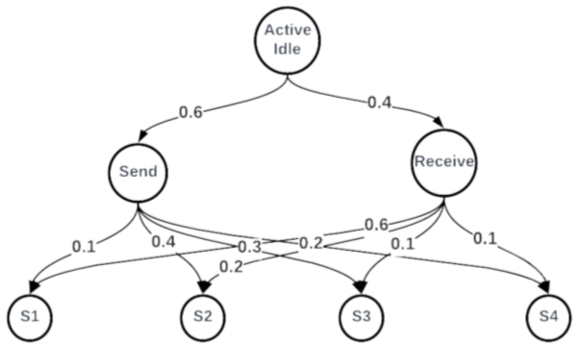

Fig. 1. Balancing Tx (Send) and Rx (Receive) states using HMM

Figure 1 illustrates the state-transition diagram of a simple WSN, with two states, Send and Receive, and four hidden states (chosen for the purpose of illustration), named S1 to S4.

A sensor node initially has two states, but after using HMM a sensor node will be assigned four hidden states (S1, S2, S3, S4) as illustrated in Figure 1 where the greatest probability amongst S1, S2, S3, S4 will be assigned to the sensor node instantaneously. A sansor node starts its operation with an active idel state. This means, as the network progresses in time, e.g. nodes executing their task routine; Tx (Send) and Rx (Receive), the energy of each node reduces, hence probability using HMM is a good fit to keep up with which nodes to transmit and which nodes to receive. As time evolves, the energy of the sensor nodes depletes as result probability of receiving is higher than transmitting because the energy for transmitting is always higher.

The following represents the pseudo code of the proposed HMM Algorithm

1. *At an instant time (t).*
2. *For each (N) set of nodes starting from node*
3. *Assign two states Tx and Rx states for a node*
4. *Do the granularity of each states be (m) states with observations*
5. *Assign the probability for transmit > receive*
6. *Update transition_probability and emission_probability (w.r.t the energy levels)*
7. *Until probability for transmit < receive*
8. *Set the final probability transmit and receive to the maximum of Emission_probability*
9. *Next node*

Let N be the set of the Sensor Nodes over the area to be monitored, j is the minimum number of nodes, jmax the maximum number of nodes, and λ is the binding variable between j and jmax that represent the minimum and the maximum number of Sensor Nodes respectively. In our scenario, j is a none zero variable.

$$N = \{j \geq 1 \,\lambda\, j \leq jmax\} \qquad \textbf{(1)}$$

Similarly, let CHs be the set of the Cluster Head Nodes that are chosen randomly amongst the Sensor Nodes and CHmax is the maximum number of CHs.

$$CHs = \{CH \geq 1 \,\lambda\, CH \leq CHmax\} \qquad \textbf{(2)}$$

Upon the deployment of the nodes in the network, each node is able to set up a link to the other nodes or to the CH accordingly. Through the above discussion, we have found this connectivity parameter:

$$P(a,b) = \{If\ \boldsymbol{a}\ establishes\ a\ connection\ with\ \boldsymbol{b} = \mathbf{1},\ otherwise = \mathbf{0}\} \qquad \textbf{(3)}$$

The a,b ∈ N, and a ≠ b, where b is either a member node or a CH in the network. If b is a CH, then a is a member node. The quality of links is directly related to the received signal strength and the distance. Suppose that c(t) denotes a stochastic process to signify the selected number of CHs at a specific time instant t. As the process of CHs selection starts from the beginning of each round, an integer scale t and discrete-time t+1 instant are selected at the beginning of two successive rounds. We express that r(t) is the round at a time instant t, and x(t) is a stochastic procedure that signifies the period of a scheme at a time instant t, which is x(t)=r(t)mod(1/P) (this equation guides the selection round). We also assume another integer 1/P, which denotes n=1/P. The state space (in this case the state space model represent only those nodes which are connected. Because connected nodes form the network, which is our point of interest.) of this model is:

$$\{0, N\} \cup \{(i, x): i \in [0, N], x \in [1, n-1]\} \quad \textbf{(4)}$$

where i and x are integers; transition state (active nodes) and observation (the networked nodes) state respectively, as this process {x(t), c(t)} holds the Markov property. $n-1$ is number of connections link between the nodes.

We use the bi-directional (a sensor node can receive and send as well) hidden Markov Chain model stationary distribution and one-step transition probabilities from [14] to estimate the Probability Mass Function (PMF) for the nodes and CHs to switch its states as follows:

$$P(CH = L) = \pi(0, N).\left[P(0, N) \to (1, N-L) + f(n-1, L) + \sum x\right] \quad \textbf{(5)}$$

where π denotes the initial probability distribution, P is one step transition probability matrix, and f signifies a factor matrix (is the matrix which is assigned to 1 for active state , 0 otherwise), $\sum x$, is included due to stochastic effect of the wsn , fxi,x∈[1, n−1], and i∈[0,N] are elements of the factor matrix. The sensor nodes can be switching from the (ON, Rx) transmission state to the (ON, Tx) receiving state from one state to another using previously set threshold probabilities.

The HMM optimization method assigns random transition probabilities for switching into these four different states (S1, S2, S3, and S4). As the network progresses in time, the energy of each node reduces, and the HMM algorithm adjusts the probability of reaching the state. As time evolves, the probability of receiving is higher than transmitting and nodes that deplete their energy at a higher rate eventually enter into the Rx state only. Nonetheless, there will remain Tx nodes too, depending on their current predicted energy levels.

4 Results and Discussion

The main objective of this experiment is to analyse the performance of the new proposed HMM algorithms with the original RCS scheduling algorithm. The proposed solution can be applied to any scheduling solution since it alters the state of the ON node at a very particular point in time at a very particular event, e.g., when a node's level of energy reaches a certain level.

In the experiment of the new solution using the MATLAB simulation environment, we set up our experiments in a $100m^2$ area to be monitored. The simulation consists of 100 stationary sensor nodes that are randomly deployed. The base station is located on the far-left edge of the network. All nodes are homogeneous which means they have the same sensing and communication capabilities. Figure 2 illustrates the simulated environment as explained above.

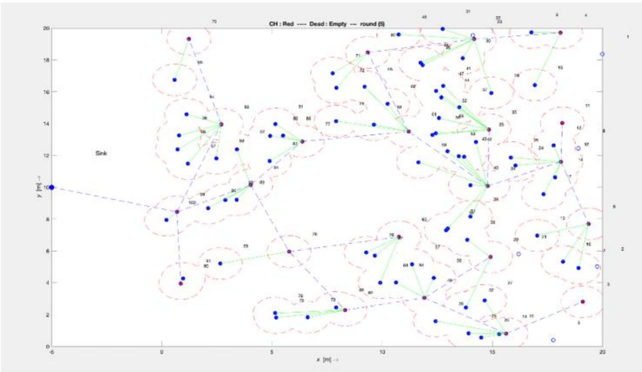

Fig. 2. Simulation Set-up of HMM in Matlab environment

The simulations were performed over 2000 rounds as the upper bound of our simulation experiment. The breaking point of our simulation experiment (when the simulation will completely stop) is when 95% of the sensor nodes have depleted their energy. In each simulation round, an event is detected and reported to the base station. In this experiment, we analyse and compare the performance of the original RCS algorithm against the improved version using the HMM process, in terms of the following metrics: Energy Used, WSN Lifetime, Throughput, Connectivity, Coverage, and Coverability. Below are the diagrams (the output of our experiments) obtained per metric, in each diagram, there are two simulation factors to evaluate the achieved results:

1. RCS, referred to as "scheduling_o";
2. RCS with HMM, referred to as "scheduling_m".

All Used Energy

Figure 3 represents the total energy used during the rounds in the WSN. This metric is concerned with the sum of consumed energy in the WSN during its lifetime of operability (simulation time for the purpose of this work). The X-axis is the number of simulation rounds/times, while the Y-axis represents the energy consumption unit in Joules.

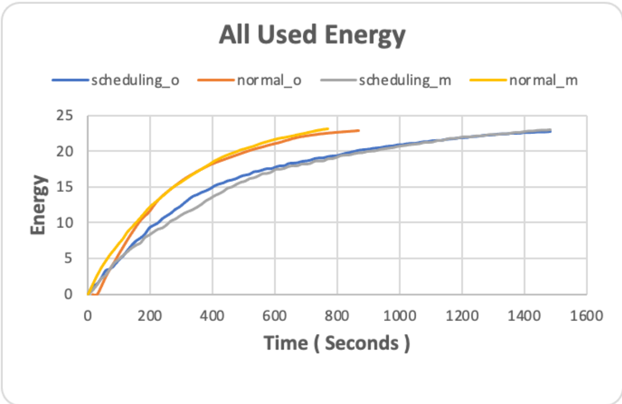

Fig .3. All Used Energy

The performance of the curves in Figure 3 can be interpreted as a metric performance evaluation whereby the less the curvature with respect to Energy levels the better because this shows the WSN achieves the same number of rounds (Time) with less energy. As can be seen in Figure 3, at round 168 the energy consumed by the Original RCS is 7.867 Joules, but the energy consumed by our improved algorithm (using HMM) is 7.133 Joules. Although this seems an insignificant improvement due to the parameters set up in our simulation, if we tuned the energy parameter in our experimentations to a higher value, we would have certainly observed a significant improvement with our proposed HMM-assisted algorithm. Hence, the HMM-assisted algorithm shows better performance with respect to energy consumption, than its RCS peer. Furthermore, we also notice that as time progresses, energy will eventually deplete, then the difference in energy consumption levels between the two algorithms is almost non-existent. This is because there is a positive correlation between the granularity value (of the parameter) and the distance between the curves.

Lifetime of Sensor Nodes

Figure 4 shows the number of live sensor nodes in the WSN. The 'Lifetime of Sensor Nodes' metric refers to how long the sensor nodes will last before their energy deplete. The X-axis is the time referred to the number of Rounds, and the Y-axis is the energy of each sensor node.

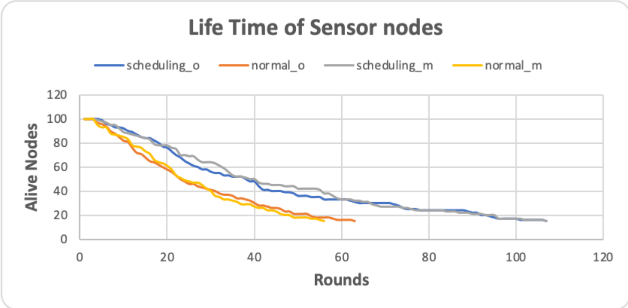

Fig. 4. Lifetime of sensor nodes

We notice from the Y-axis that the remaining number of nodes at the end of the simulation run-time is 15 for both HMM and RCS. Yet, the X-axis shows with HMM the WSN lifetime reaches 107 rounds before the end of the simulation, but RCS lasted 102 rounds only. Despite the small improvement made with the HMM (of 5 rounds only), this can be explained due to the small value allocated to the sensor parameters (number of nodes, energy, communication & sensing ranges, etc.). Note, assigning large parameter values is an unrealistic experimentation practice (Xianglin et al., 2012).

Throughput

Figure 5 represents the throughput utilized by HMM assisted algorithm and its RCS peer in the WSN. The throughput is the number of data packets sent and received in the network. The X-axis is the time which is in rounds, and the Y-axis is the number of data packets transmitted in the network.

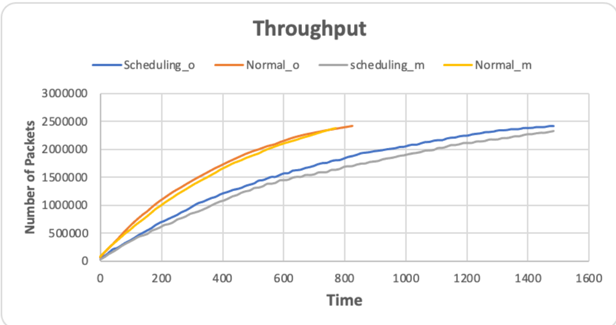

Fig. 5. Throughput

We can notice the distance between the two curves of HMM and RCS algorithms. There are lots of packets generated in the network which indicates that a sizable number of packets are discarded due to collision. The excess generation of packets leads to the wastage of energy and hence, the resultant data transfer is low. With HMM, we managed to transfer fewer packets which results in better performance with respect to throughput. For example, the number of packets generated with HMM at the 1414th round is 2275957, meanwhile, with RCS, the number of generated packets is 2379214. Hence, RCS generated 10,000 more packets than HMM.

Connectivity

Figure 6 reflects the achieved results with respect to connectivity in the WSN. Connectivity is the number of nodes connected at any instance in time. The X-axis is the time in rounds, and the Y-axis is the fraction or percentage of nodes being connected.

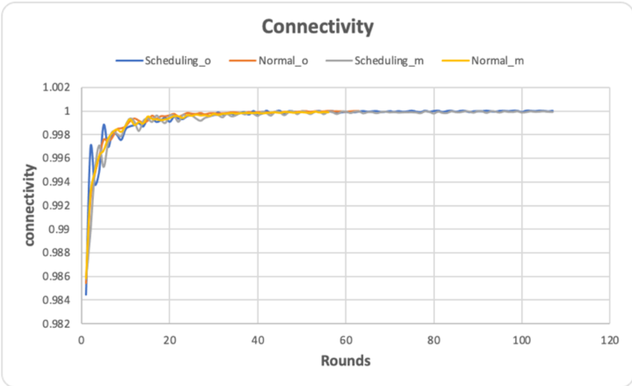

Fig. 6. Connectivity.

The HMM's connectivity performance is similar to the RCS' performance with respect to the fractions of the nodes that are connected. Both algorithms scored equal connectivity values of 99.99%, which indicates that there is almost no scope for improvement over the RCS as far as this metric is concerned. The fluctuations can be attributed to the stochastic nature of the HMM algorithm over the randomization nature of the RCS algorithm.

Coverage

Figure 7 represents the coverage results in the WSN. The coverage metric is defined as the percentage of the area covered by the WSN. The X-axis is the number of rounds in time, and the Y-axis is the ratio of the coverage for the WSN.

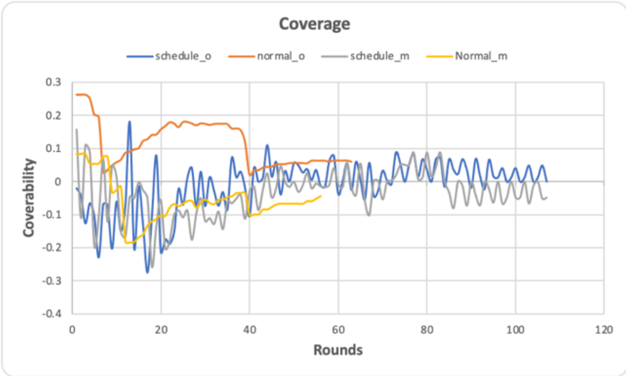

Fig.7. Coverage

If we take the 65th round, for instance, the HMM performance is 0.005, while the RCS algorithm scores 0.05. This indicates that network coverage has been increasedtenfold using the HMM algorithm.

Coverability

Figure 8 represents the achieved Coverability results in the WSN. The X-axis represents the nodes' sectional analysis number, while the Y-axis is the coverage percentage.

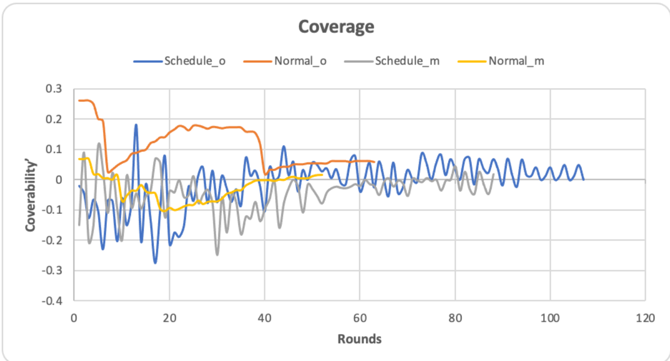

Fig. 8. Coverability

The objective here is to study the network's coverage per section of nodes. For 70 nodes out of a total of 150 nodes, the HMM normal coverage (0.31) > RCS normal coverage (0.27869) and HMM scheduling coverage (0.231885) is also greater than RCS scheduling (0.15254). This trend follows as the number of nodes increases. It is worth noting that the term 'Coverability', while sounding synonymous with 'Coverage', actually refers to the percentage of the coverage per sensor node. Figure 8 explains the result of the experiment. The RCS algorithm was originally tested on the basis of coverability, not coverage, in C Liu's work (Liu et al., 2006). This is understandable because the algorithm is not compared or evaluated with other algorithms with respect to (1) the coverage metric of the extra ON nodes, and (2) network lifetime improvement. In our HMM improved algorithm, we are concerned with improving network lifetime so as to contribute to the safety critical system requirements, and further contribute to the dependability of such systems. The Coverage metric is concerned with all of the network's nodes, while the Coverability metric is concerned with a certain number of nodes and can be used to analyse single nodes only. Therefore, we choose to focus on the Coverage metric in our experimentations.

5 Conclusion

In this work, we introduced a novel improvement to the RCS algorithm, by proposing an HMM algorithm that is based on a probability distribution, unlike the original RCS which is based on random scheduling. Our HMM-based algorithm has increased the network lifetime and also improved the coverage and connectivity. The transition states of the nodes are restricted to four states in our algorithm to avoid computational overheads. The main goal was to predetermine the receive and transmit states' as defined by the algorithm of each node in the WSN in the design time. In this process, we adjust the probability of transition between our receive states and transmit states. As a result, each node knows its receive states' as defined by the algorithm and operates accordingly in the runtime. In network connectivity and path optimality, this prediction is important considering the broadcast collision and channel errors as metrics for quality of service. The RCS algorithm has had several errors due to high throughput values, but our proposed HMM algorithm has mitigated this by the use of path optimality (optimization) to reduce traffic overheads. This is achieved by ensuring both receiving and transmitting nodes at every round of simulation time to ensure the network's operability. In the future, we will be addressing the limitation of the RCS and the HMM algorithms using bioinspired computational and artificial intelligence methods.

Acknowledgement

The authors would like to thank all those who contributed to the completion and success of this work. Also, the Faculty of Science and Technology at Middlesex University played a significant role in backing this work at all stages of the study.

References

[1] A. I. Ambrose, 'Scheduling for Composite Event Detection in Wireless Sensor', *Master Thesis Fla. Atl. Univ.*, 2008, Accessed: Aug. 07, 2015. [Online]. Available:

http://fau.digital.flvc.org/islandora/object/fau%3A2881/datastream/OBJ/view/Scheduling_for_composite_event_detection_in_wireless_sensor_networks.pdf

[2] C. F. García-Hernández, P. H. Ibarguengoytia-Gonzalez, J. García-Hernández, and J. A. Pérez-Díaz, 'Wireless sensor networks and applications: a survey', *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 7, no. 3, pp. 264–273, 2007.

[3] J. C. Knight, 'Safety critical systems: challenges and directions', in *Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on*, 2002, pp. 547–550. Accessed: Oct. 18, 2016. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1007998

[4] T. He *et al.*, 'Energy-efficient surveillance system using wireless sensor networks', in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, 2004, pp. 270–283. Accessed: Jul. 15, 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=990096

[5] Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy, 'A framework for use of wireless sensor networks in forest fire detection and monitoring', *Comput. Environ. Urban Syst.*, vol. 36, no. 6, pp. 614–625, 2012, Accessed: Nov. 05, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0198971512000300

[6] L. Wang, R. Wei, Y. Lin, and B. Wang, 'A clique base node scheduling method for wireless sensor networks', *J. Netw. Comput. Appl.*, vol. 33, no. 4, pp. 383–396, 2010.

[7] A. Avizienis, J.-C. Laprie, B. Randell, and others, *Fundamental concepts of dependability*. University of Newcastle upon Tyne, Computing Science, 2001. Accessed: Nov. 13, 2015. [Online]. Available: http://edi-info.ir/files/Fundamental-Concepts-of-Dependability.pdf

[8] P. Hu, Z. Zhou, Q. Liu, and F. Li, 'The HMM-based modeling for the energy level prediction in wireless sensor networks', in *2007 2nd IEEE Conference on Industrial Electronics and Applications*, 2007, pp. 2253–2258.

[9] F. Sailhan, T. Delot, A. Pathak, A. Puech, and M. Roy, 'Dependable wireless sensor networks', in *3th workshop Gestion des Données dans les Systèmes d'Information Pervasifs (GEDSIP) in cunjunction with INFORSID*, 2009, pp. 1–16. Accessed: Jun. 21, 2015. [Online]. Available: http://cedric.cnam.fr/~sailhanf/publications/gedsip.pdf

[10] A. Cerpa and D. Estrin, 'ASCENT: Adaptive self-configuring sensor networks topologies', *Mob. Comput. IEEE Trans. On*, vol. 3, no. 3, pp. 272–285, 2004, Accessed: Oct. 10, 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1318596

[11] M. Cardei, M. T. Thai, Y. Li, and W. Wu, 'Energy-efficient target coverage in wireless sensor networks', in *INFOCOM 2005. 24th annual joint conference of the ieee computer and communications societies. proceedings ieee*, 2005, vol. 3, pp. 1976–1984. Accessed: Feb. 09, 2016. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1498475

[12] D. Tian and N. D. Georganas, 'A Coverage-preserving Node Scheduling Scheme for Large Wireless Sensor Networks', in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, New York, NY, USA, 2002, pp. 32–41. doi: 10.1145/570738.570744.

[13] U. Castellani, M. Cristani, S. Fantoni, and V. Murino, ‘Sparse points matching by combining 3D mesh saliency with statistical descriptors’, in *Computer graphics forum*, 2008, vol. 27, no. 2, pp. 643–652.

[14] C. Patra and S. Mullick, ‘Calculation of the Duty Cycle for BECA’, *Int. J. Comput. Appl.*, vol. 121, no. 14, 2015.

[15] V. Krishnamurthy, ‘Algorithms for optimal scheduling and management of hidden Markov model sensors’, *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1382–1397, 2002.

[16] R. Goudarzi, B. Jedari, and M. Sabaei, ‘An efficient clustering algorithm using evolutionary hmm in wireless sensor networks’, in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2010, pp. 403–409.

[17] W. Qihua, G. Ge, C. Lijie, and X. Xufeng, ‘Scheduling strategy for Hidden Markov Model in wireless sensor network’, in *2015 34th Chinese Control Conference (CCC)*, 2015, pp. 7806–7810.

[18] J. L. Hill, ‘System architecture for wireless sensor networks’, University of California, Berkeley, 2003. Accessed: Oct. 24, 2015. [Online]. Available: http://www.eps2009.dj-inod.com/docs/09-02-01/system_architecture_for_wireless_sensor_networks.pdf

[19] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr, ‘Basic concepts and taxonomy of dependable and secure computing’, *Dependable Secure Comput. IEEE Trans. On*, vol. 1, no. 1, pp. 11–33, 2004, Accessed: Nov. 18, 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1335465