Mobility Aware Duty Cycling Algorithm (MADCAL) in Wireless Sensor Network with Mobile Sink Node

Craig Thomson*, Isam Wadhaj†, Zhiyuan Tan‡, Ahmed Al-Dubai§
School of Computing
Edinburgh Napier University
Email: *C.Thomson3@napier.ac.uk, †I.Wadhaj@napier.ac.uk, ‡Z.Tan@napier.ac.uk, §A.Al-Dubai@napier.ac.uk

***Abstract*—In Wireless Sensor Networks (WSNs) the use of Mobile Sink Nodes (MSNs) has been proposed in order to negate the ”hotspot” issue. This where nodes closest to the sink node shall run out of energy fastest, affecting network lifetime. However, in using an MSN there remains the issue of nodes that sink passes closest to using most energy. In this study we propose a novel Mobility Aware Duty Cycling Algorithm (MADCAL), for use at the MAC layer. This algorithm provides a lightweight solution, utilising knowledge of the mobility pattern of the sink node in order to determine a dynamic communication threshold, within which static nodes shall awaken to communicate with the sink. This avoids the need for energy-heavy beacon messages. Using the Omnet++ simulation framework and tested on a lightweight implementation of a Carrier-sense multiple access (CSMA) based MAC protocol, we demonstrate a reduction in energy consumption up to 15% and up to 300% increase in frame delivery to the MSN, when compared to the existing duty cycling approach.**

***Index Terms*—Mobile Sink, Dynamic Duty Cycle, WSN.**

I. Introduction

Wireless Sensor Networks (WSN) have gained a lot of attention from industry and academia in recent years. These networks, comprised of small devices with low power and processing capacity, work on a many-to-one multi-hop basis, with the ultimate destination a sink node. This has applications in various areas such as agriculture [1], deep sea oil and gas [2] and others such as disaster recovery [3]. Due to the nature of these networks and often inhospitable environments in which they are located, replacement of batteries within the sensors is problematic. As a result, conservation of power to extend network lifetime is of great significance. Duty cycling, where nodes will sleep when idle, is a common approach to reducing energy consumption within a WSN. However, a challenge inherent in duty cycling is in ensuring overlap of wake-up schedules in order that Neighbour Discovery (ND) may occur between nodes. A further issue in static WSNs is that of energy and routing ”hotspots” [4], which occur near the sink node. This is due to the nodes closest to the sink taking on a greater workload than nodes further away and, therefore, expending a greater amount of energy. This can result in these nodes running out of energy faster and leaving other nodes in the network unable to communicate with the sink node. In effect ending the lifetime of the entire WSN as a functioning entity.

Mobile Sink Nodes (MSNs) may be used in order to more evenly distribute energy consumption amongst nodes, and subsequently increase network lifetime. By moving the sink node, the issue of energy and routing ”hotspots” can be negated to some degree [5][6]. However, in mobilising the sink node, further issues arise with regard to ND and duty cycling. While many studies have taken place to either produce new routing protocols or extend existing work, to account for the use of MSNs, these do not account for ND protocols. There has been considerable research into the issue of ND in mobile IoT devices such as wireless sensors. As such, the more traditional reference to ND as a network layer issue is rendered moot to some degree. In smaller, battery-powered and duty cycled [7] devices, the greater issue is of overlap of wake-up schedules in order that ND may actually happen. To achieve this there have been many proposed approaches, with probabilistic [8][9] and deterministic [10][11][12] algorithms generally utilised in order to either guarantee overlap or produce a high probability of it taking place. There has been shown to be value in both approaches, with a deterministic approach found to be the most common. The issue of long-tail discovery, where a node may not be discovered at all, has proven problematic to the probabilistic approach [13]. This despite it being demonstrated to generally be more efficient than its deterministic counterpart. However, recently other methods have been proposed, such as opportunistic approaches where decisions are made ”on the fly” [14][15][16]. Also, the use of mobility awareness in WSNs where mobility is utilised, has been suggested as a possible approach [17][18][19]. In this regard, approaches such as the prediction of mobility patterns of mobile nodes has been proposed in order to improve routing and data delivery in the network [20]. Thus far though, mobility has not been used in order to influence the duty-cycling of nodes.

Many network layer solutions have been developed for the use of MSNs, with these routing protocols generally utilising energy heavy flood approaches or more energy efficient, but delay-tolerant methods [21][6]. These studies all have validity in terms of network layer packet delivery when a MSN is in use. However, IT is the MAC layer which is responsible for the greatest amount of energy consumption [18]. As such, there is potential in the use of mobility patterns at the MAC layer in relation to channel access, and the duty cycling utilised therein. The aim being for nodes to awaken when there is the greatest chance of transmission being received by the sink, reducing unnecessary expenditure of energy.

In this paper we propose a novel Mobility Aware Duty Cycling Algorithm (MADCAL), to utilise the mobility pattern of the MSN in order to influence the wake-up schedules of static nodes. MADCAL operates at the MAC layer, independent of any routing protocol. The duty-cycling of one-hop nodes to the sink is based on the current location of the MSN and a dynamic communication threshold calculated within each node. This calculation uses the interference range of the node, its distance from the path of the MSN and the speed of the MSN. As such, a demonstrable improvement in energy consumption across the network and number of frames received by the MSN is shown when compared to standard duty cycling with Clear Channel Assessment (CCA) and check interval.

The rest of this paper is organised as follows. Section II gives an overview of related work in this area while section III describes the methodology in use. In section IV the MADCAL algorithm is detailed and in section V evaluation and results are shown. A conclusion and future plans is in section VI.

II. Related Work

In examining related work we highlight studies where sink mobility is utilised in order to affect various aspects of network performance. As such, we do not review particular routing protocols designed for use with sink mobility. In examining the most compelling reason for mobilising sink nodes, that being the vast improvement in energy consumption versus the use of a static sink node, [22] uses a linear mobility model and claims a 500% increase in network lifetime versus a static network. This operates by utilising a sojourn time, which is commonly referenced in other work, where the sink lingers at a node. In this case to limit time until the first node runs out of power. More recent studies have progressed to novel approaches such as to strategically alter the sink trajectory, such as in [23] where trajectory is altered based on node density, with the aim to pass every single node. This utilises a novel approach of a ”Space Filling Curve” [24]. This study suggests a 20% increase in Packet Delivery Ratio (PDR) when a dynamic curve approach is in use. However, given the reliance on node density as a factor, scalability could become problematic. Energy consumption does not appear to be a priority within this study, also, at the MAC layer 802.11 is used as standard, with no reference to this beyond test parameters. As such, whilst this study produces excellent results, the approach is not directly relatable to a MAC layer approach to sink mobility.

Sink trajectory is again a factor in [25]. A real-world application of a MSN, the FarmBeats platform is developed for precision architecture by using algorithms developed to ensure effective path-planning to extend the battery life of drones the MSNs in this scenario. This paper also makes reference to the importance of duty cycling in the base stations but does not base this on the mobility of the MSN. This work is extended in [26] by additionally utilising two residual energy thresholds, one for the entire network and one for the path of the MSN. This is then used to modify the existing FarmBeats [25] algorithms in terms of duty cycling and path selection. The authors claim benefits in terms of processing time and energy performance.

Once again, optimal path planning is in evidence in [4]. One of the main features of this paper is a secondary approach to prioritise emergency broadcasts. One observation to be made in this paper is that the static nodes shall awaken when the sink node is nearby. However, how this happens is not detailed and would appear to be one of the assumptions on which the subsequent work in the study is built. This study demonstrates the benefit of utilising sink mobility to influence network behaviour and results demonstrate improvement in network lifetime and transmission delay. The approach here is to adjust the mobility pattern of the MSN to coincide with the duty cycling algorithm in use. The benefits of this approach are proven as are the potential advantages in linking sink mobility to duty cycling.

[27] propose the eTrail protocol. In this study, the MSN acts as the cluster head, with clusters built as the sink moves. The MSN sends beacons in order to leave a trail of the sinks path, with sensor nodes updating their routing information so that the path to the sink is kept up-to-date for each node. The authors claim minimal communication overhead due to the use of only local broadcasts, with this work located at the network layer. The use of beacons and broadcasts of any kind is something we have looked to eliminate within our own studies given the increase in energy expenditure that comes with their use. However, it is accepted that some may be required in an extension of our work as we progress.

The primary focus of [28] is to address the issue of delay when implementing a MSN. In particular when considering delay sensitive data which should not be subject to the same restrictions placed upon other data. In this case the authors developed a delay-intolerant routing scheme (DRS), implementing a deadline by which data should be received by the sink. When a sensor has emergency data to transmit it will either wait until the sink is within its grid or send immediately to an awake sensor node. Again, this is a study which takes place at the network layer, with regard to the routing scheme implemented. However, reference is made to the probable effectiveness of controlling the wake-up pattern of nodes alongside this scheme.

III. Mobility Aware Duty Cycling

Related work demonstrates a propensity in studies to determine, by various different parameters, an “optimal“ path for the sink node. Whilst these works have each displayed merit in their results, the issue of utilising energy consuming beacon messages in order to keep track of the sink position can be highlighted. This is an approach we have looked to avoid.

We aim to reduce energy consumption amongst the “significant nodes“ the MSN communicates with via one-hop, while also improving the number of frames delivered to the sink node, or at least keeping that figure at a reasonable level. This new mobility-aware duty cycling approach utilises a pre-defined mobility pattern in order that static nodes may calculate the sink position based on network parameters such as sink start position and speed. Resulting in a lightweight algorithm with no network overhead.

When considering the mobility pattern used, we seek to study an environment where all nodes are not treated equally, such as a disaster recovery situation where reaching each node directly is not possible. Many studies aim to have the MSN pass every node and whilst such a mobility pattern has merit it is clear that this is not always possible in a real-world scenario. Therefore, in our tests we have implemented a circular mobility pattern for the MSN, moving around the periphery of the network, as can be observed in Fig 1. When considering network topology, a one-hop grid formation is used. Whilst in future studies we would seek to evaluate the effect of our algorithm on a more random topology, initially it was felt that controlled location of the static nodes makes it easier to observe the effect of the MSN when considering interference range and distance from the path of the sink. Fig 1 shows the network layout, with the start point of the MSN and the clockwise direction of travel.

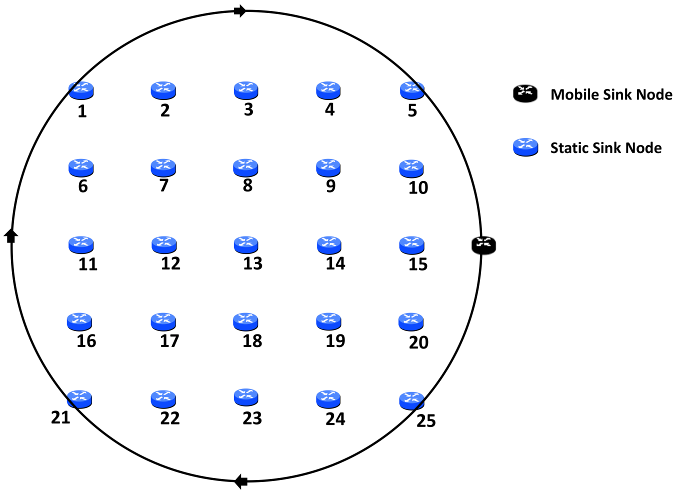

Fig. 1. Network Topology

As can be observed, depending on interference range, certain nodes within the network will be within one-hop of the MSN. These are determined to be ‘significant nodes‘. As such, these nodes replace the afore-mentioned hotspot nodes in a static network, with final responsibility for relaying data to the sink. The aim in moving the sink node in such a way is that more nodes may take on the responsibility of a hotspot, subsequently increasing network lifetime . However, this benefit is negated if nodes compete for channel access and thus, the ability to communicate with the sink node. Therefore, a threshold of communication is required to ensure fair access to the MSN.

The MADCAL algorithm is implemented at the MAC layer, as it is here that is responsible for the greatest energy consumption [29]. The MAC implementation utilised is a lightweight Carrier-sense multiple access (CSMA) implementation which reflects the core functionality of the IEEE 802.15.4 standard [30] [31], using CCA and the transmission of preambles. Fig 2 demonstrates the MAC implementation and where the MADCAL functionality is inserted.

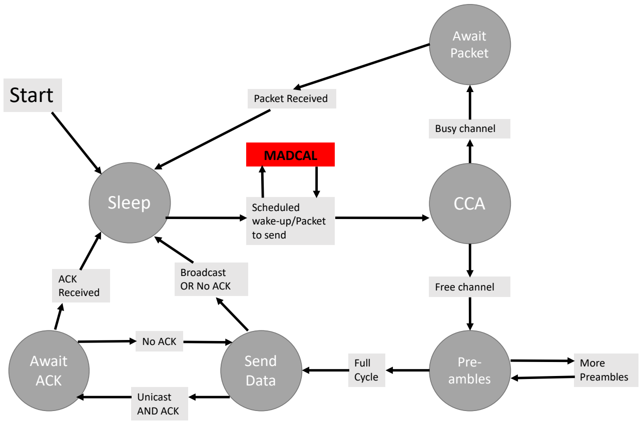

Fig. 2. MAC Implementation

We assume the following properties of the WSN:

1) Static node positions are constant throughout.
2) Static nodes are aware of their own location.
3) Static nodes are unaware of the location of neighbouring nodes, each node implements the MADCAL algorithm independently.
4) Node power levels are consistent.
5) Interference ranges, though variable across tests, are consistent across static nodes and the MSN.
6) Sink speed shall not be less than 2mps or greater than 40mps.

In examining the simulation parameters as in Table 1, the simulation time is calculated to ensure an exact number of circuits of the network by the MSN. As such, with the speed at 2mps (meters per second) the sink shall complete exactly

TABLE I
SIMULATION PARAMETERS

Test Parameters	Values
Number of Static Nodes	25
Playground Size	x=500m y=500m
Grid Size	200m * 200m
Circle Radius	150m
Sink Start Position	x=400m, y=250m
Sink Node Speed (metres per second)	2mps, 10mps, 20mps, 40mps
Simulation Time	942.47779607694s
Interference Distance	77.52m, 69.13m, 62.02, 55.94m
Number of Runs	5
Path-loss Alpha	1.85, 1.9, 1.95, 2
Carrier Frequency	2.4GHz
Maximum Sending Power	1.0mW
Signal Attenuation Threshold	-85dBm
Sensitivity	-75dBm
Transmitter Power	1.0mW
Thermal Noise	-85dBm
Signal to Noise Ratio Threshold	4dB

2 circuits of the network, for 10mps, 10 circuits and so on.

Interference distance is calculated as thus [32]:

$$interferenceDistance = \frac{(\frac{SoL}{Freq})^2 * Power}{(16 * PI^2 * 10^{\frac{SAT}{10}})}^{\frac{1.0}{Alpha}} \qquad (1)$$

Where SoL denotes the speed of light (i.e., 30,000,000 mps); *Freq* stands for the carrier frequency; *Power* indicates the transmitter power; *SAT* is the signal attenuation threshold; and *Alpha* represents the path loss alpha.

Received signals with power below the sensitivity value are ignored. In this case the value was adjusted from -85dBm to -75dBm in order to reduce the number of signals received and thus reduce network failure due to node overload.

All parameters are consistent across all simulation runs apart from the speed of the sink node and the interference distance of the nodes. The path loss alpha is adjusted across four different values, as detailed in the test parameters, this in order to alter the size of the interference distance, which decreases as the alpha value increases.

This study does not seek to test routing protocols. However, in order to ensure final delivery is to the MSN and due to it's heavy load and therefore higher energy consumption, we have utilised the Optimized Link State Routing Protocol (OLSR) protocol [33]. Whilst not usually used in a WSN environment, in this case it was effective in accelerating scenarios and therefore we required a lower simulation time.

IV. MOBILITY AWARE DUTY CYCLING ALGORITHM (MADCAL)

The basic premise of MADCAL is to establish a threshold of communication between each significant static node and the MSN, with significance based on whether the distance from the static node to the path of the sink is less than the node's interference range. In the case of a circular sink mobility pattern as utilised here, this involves establishing the coordinates of the start and end of the portion of the circle circumference where, when reached by the MSN, the particular static node should be awake for communication. This is detailed in Algorithm 1.

With circular mobility this threshold is calculated based on the angle of the closest point to the circular path in relation to the static node – the *circlePoint*. Taking into account the interference range of the node and the radius of the circle, a maximum threshold before and after the circlePoint can be calculated. As demonstrated in Fig 3 in relation to Node 15. However, a simplistic approach such as this would result in a significantly large threshold if the static node is close to the path of the MSN. In this event this node could monopolise communication with the sink for a considerable time, to the detriment of other significant nodes. To negate this, a more dynamic approach to calculating the threshold is required. Firstly, node distance to the sink path is taken into consideration. Secondly, in order to now avoid an extremely small threshold for nodes closer to the path, a factor is utilised. This lessens how much the threshold is reduced based upon the speed of the sink node, which is constant throughout each scenario. It was found, via simulation, that as the sink speed increases a smaller threshold is more efficient. Therefore, as a first approach, static factors have been utilised such that for speeds less than 10mps the threshold cannot be reduced by less than a factor 0.5, for less than 20mps this factor reduces to 0.35, reducing again to no less than 0.25 for less than 40mps. It is envisaged that in future work an extension to this algorithm could be developed to ensure the factor is completely dynamic based on the speed of the sink node.

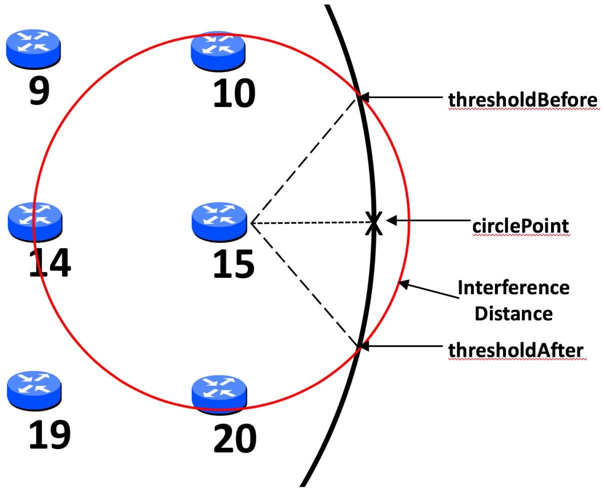

Fig. 3. Illustration of Threshold

Algorithm 2 is designed to be inserted within the existing

Algorithm 1 Communication Threshold

```
procedure INITIALISATION
    set sinkSpeed #constant throughout
    $significantNode \leftarrow false$
    set interDist #the calculated interference distance
    set Circumference
    set firstSinkPos
    set first SinkQuartile
    set distToCircle #shortest distance from static node to circle
    if $distToCircle < interDist$ then
        $significantNode \leftarrow true$
    end if
    if $significantNode$ then
        set circlePoint #closest point on circle to node
        set nodeQuartile #quartile of circle in which circlePoint resides
        set distanceBetweenPoints #distance between firstSinkPos and circlePoint
        set angleOfNode
        $thresholdAfter \leftarrow true$
        $sinkThresholdAfter \leftarrow establishThreshold(sinkRadius, thresholdAfter)$
        $thresholdAfter \leftarrow false$
        $sinkThresholdBefore \leftarrow establishThreshold(sinkRadius, thresholdAfter)$
        set beforeQuartile #previous quartile to threshold
        set thresholdOpposite #opposite point to threshold
    end if
end procedure
function ESTABLISHTHRESHOLD(radius, after)
    $nodeDist \leftarrow (radius - distToCircle)$
    $angleTemp \leftarrow \frac{(radius^2 + nodeDist^2 - interDist^2)}{(2 * radius * nodeDist)}$
    $angleRadians \leftarrow arccos(angleTemp)$
    $angle \leftarrow (angleRadians * (\frac{180}{PI})$
    $factor \leftarrow \frac{distToCircle}{interDist}$
    if $sinkSpeed < 10$ then
        $factorCheck \leftarrow 0.5$
    else if $sinkSpeed < 20$ then
        $factorCheck \leftarrow 0.35$
    else if $sinkSpeed < 40$ then
        $factorCheck \leftarrow 0.25$
    end if
    if $factor < factorCheck$ then
        $factor \leftarrow factorCheck$
    end if
    $angle \leftarrow (angle * factor)$
    if after then
        $threshAngleDegrees \leftarrow (angle + angleOfNode)$
    else
        $threshAngleDegrees \leftarrow (angleOfNode - angle)$
    end if
    $threshAngleRadians \leftarrow \frac{threshAngleDegrees}{(180 * PI)}$
    $threshold.x \leftarrow circleCentre.x + (radius * cos(threshAngleRadians))$
    $threshold.y \leftarrow circleCentre.y + (radius * sin(threshAngleRadians))$
    return Coord threshold
end function
```

MAC code in order to establish the node wakeup time. This is based on calculating the current sink position. This is calculated utilising the sink start position, the size of the circle circumference and the current simulation time. This enables the static nodes to calculate the sink position without the need for beacons or other energy consuming methods such as Global Positioning System (GPS) [34]. The sink position is then compared to the coordinates of the start of the threshold calculated in Algorithm 1, with node wake-up time determined by how long it will take the sink to reach the threshold, or else set to zero in the event the sink is already within the threshold.

Algorithm 2 Threshold Interval

```
procedure SLEEP
    set checkInterval from input
    if $significantNode$ then
        $thresholdTime()$
        if $thresholdReached$ then
            $interval \leftarrow checkInterval$
        else
            $interval \leftarrow timeToThreshold$
        end if
    else
        $interval \leftarrow checkInterval$
    end if
    schedule WAKEUP at simTime + interval
end procedure
function THRESHOLDTIME
    set sinkPos
    set dist #shortest distance from sinkPos to sinkThresholdBefore
    $withinThreshold()$
    if not thresholdreached then
        set arc to distance to sinkThresholdBefore
        set sinkQuartile #current quartile in which sink resides
        $timeToThreshold \leftarrow \frac{arc}{sinkSpeed}$
    else
        $timeToThreshold \leftarrow 0$
    end if
    return timeToThreshold
end function
```

V. Evaluation and Results

A. Simulation Environment and Parameters

Work has been conducted on the OMNeT++ [35] simulation framework, utilising both MiXiM [36] for the network environment and inetmanet [37] for all other factors, including Physical, MAC, Network and Transport layer parameters.

Results were first obtained for the network implementation with a MSN, but with the existing standard duty cycling with CCA and check interval. This makes no allowances for sink mobility. Result metrics are of average energy consumption amongst significant nodes and MAC layer frames received by the sink node. Within this study, significant nodes are 1, 2, 3, 4, 5, 6, 10, 11, 15, 16, 20, 21, 22, 23, 24 and 25.

B. Static Network

As a reference point, tests were conducted with the same simulation time but with the sink node immobile and remaining at the start position of the MSN, next to Node 15, as shown in Figure 1. What was found was unless there is a large interference distance which can encompass more than one node, this one node uses up most energy. However, when there is an overlap of interference distance, this affects the number of frames to reach the sink node due to channel access contention. This highlights the hotspot issue, as one node shall run out of energy far sooner than the others and at that point the network becomes redundant. Even in the event that neighbouring nodes can then take on the role of hotspot when a large interference distance is in use, this only delays the inevitable.

C. Results

Figures 4-7 show average energy consumption in significant nodes. Comparison is between evaluation, utilising standard duty cycling with CCA and check interval in all nodes, and when the MADCAL algorithm is applied to create a dynamic communication threshold in significant nodes.

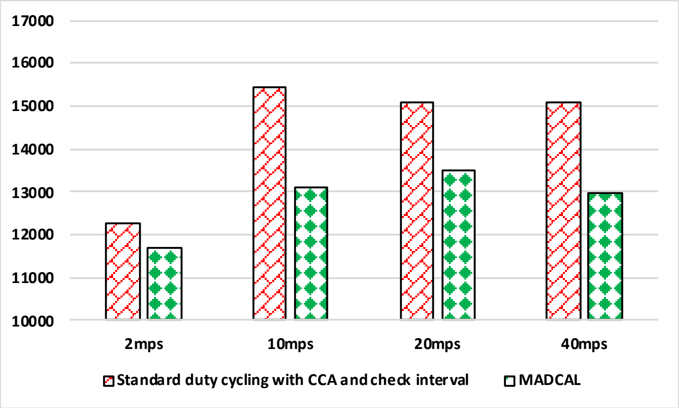

Fig. 4. Average Energy Consumption (mWs), Significant Nodes. Interference Range 77.52m

Fig 4 illustrates considerable energy saving when the MADCAL algorithm is in use. The larger interference range in use here would normally result in considerable overlap of communication between significant nodes and the MSN. However, with a communication threshold established by MADCAL, nodes are less likely to compete for channel access. Hence there is less extraneous energy consumption.

In Fig 5 interference range is reduced to 69.13m, but results are still a significant improvement.

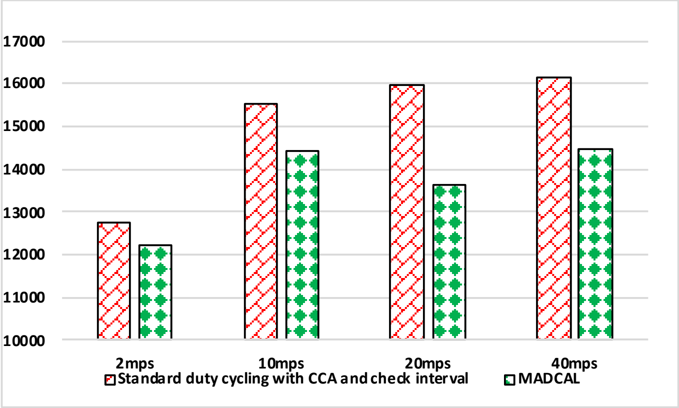

Fig. 5. Average Energy Consumption (mWs), Significant Nodes. Interference Range 69.13m

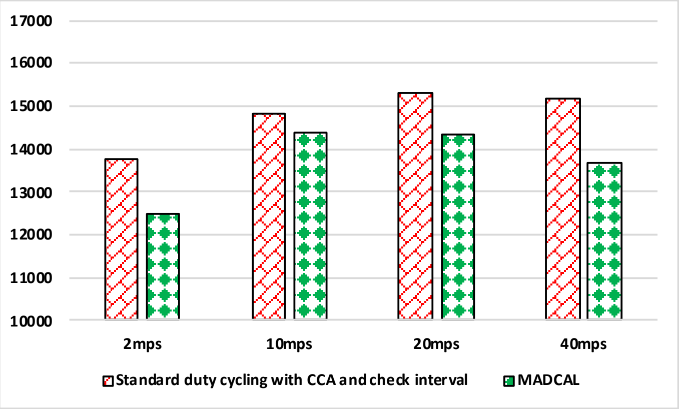

Fig. 6. Average Energy Consumption (mWs), Significant Nodes. Interference Range 62.02m

In Fig 6 it can be seen that improvements are now lessened as the interference range reduces, with communication between the sink and significant nodes now more difficult. However, improvements can still be seen, especially when the sink moves at a slower speed.

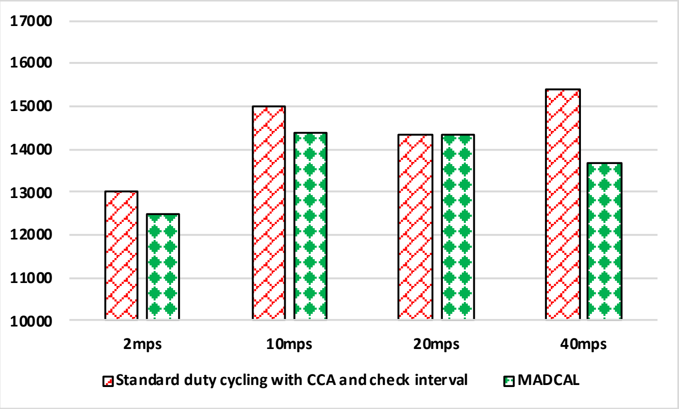

Fig. 7. Average Energy Consumption (mWs), Significant Nodes. Interference Range 55.94m

Fig 7 is significant in that the interference range is now strained to the extent that it is only marginally greater than the distance between nodes and the greatest distance to the sink - 50m. However, despite the reduction in communication overlap when standard duty cycling is in use, the MADCAL algorithm still results in improvements.

Figures 8-11 illustrate the number of MAC layer frames received by the sink during each simulation scenario.

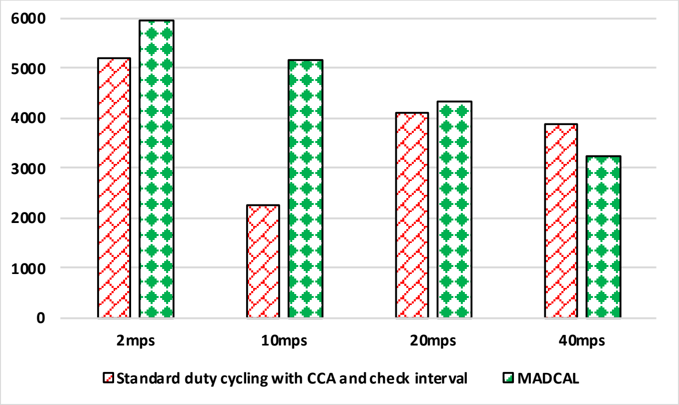

Fig. 8. Sink Frame Reception. Interference Range 77.52m

In observing Fig 8 it can be seen that frame reception is easier to improve upon when the sink mobility is slower. What becomes clear from our studies is that improvements are difficult at sink mobility speeds of 20mps and higher. This shall be taken into account in future studies.

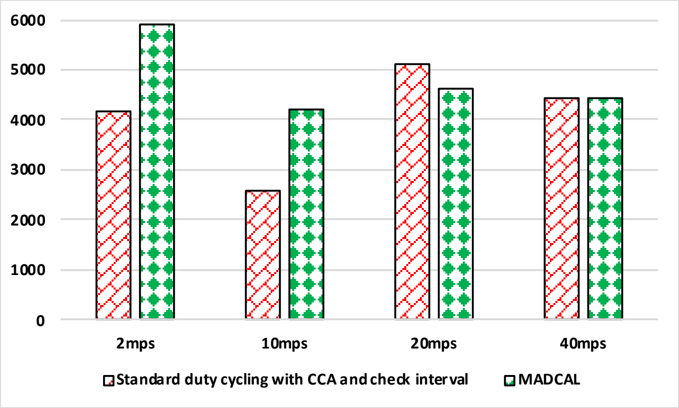

Fig. 9. Sink Frame Reception. Interference Range 69.13m

Figure 9 again shows the benefit of the sink moving more slowly, with frame reception the same or slightly worse for the faster speeds.

In Fig 10 again the benefits are greater when the sink moves more slowly. However, it should be noted that when the interference range is lower that frame reception actually increases overall before MADCAL is even in use.

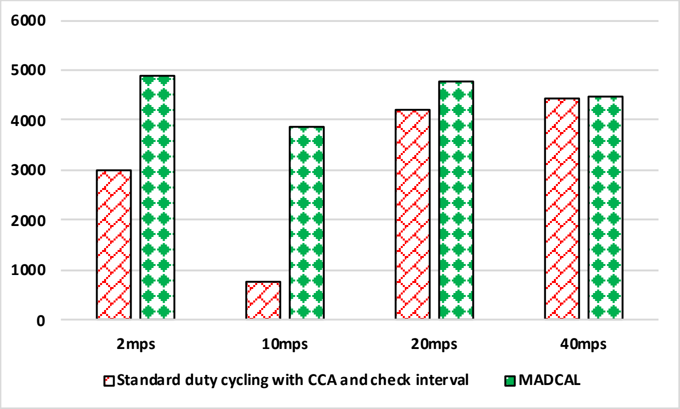

Fig. 10. Sink Frame Reception. Interference Range 62.02m

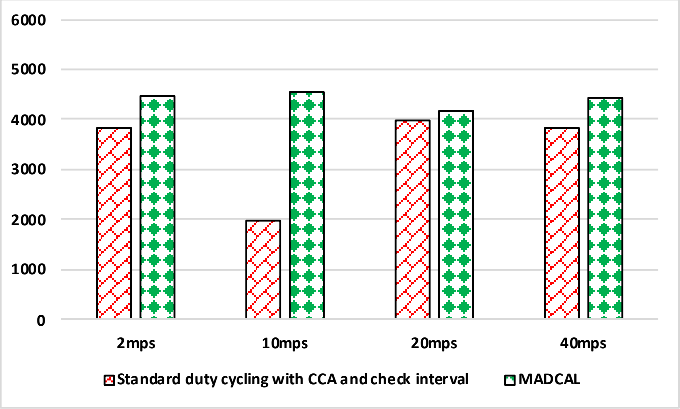

Fig. 11. Sink Frame Reception. Interference Range 55.94m

In Fig 11 benefits once more are greater in lower speeds. But resultantly, MADCAL brings reception up to the same levels for all speeds.

D. *Summary*

A significant improvement in energy consumption can be observed when MADCAL is implemented, especially so when sink mobility is slow. Resultantly, the network lifetime is increased. While improvements are significant at lower speeds, once the MSN speed increases improvements are less clear in terms of frame delivery. However, any differences are marginal and offset by the significant benefits in energy consumption. It can also be argued that improvements in frame delivery are more difficult at faster speeds and that MADCAL is efficient in bringing the same levels of delivery to slower speeds. However, this highlights potential for future study with regard to optimal MSN speed. This research shows that even as speeds reach 40mps (144 kmph), an improvement in energy consumption is possible while frame delivery remains stable.

VI. Conclusion and Future Work

In this paper we propose MADCAL, a dynamic and lightweight duty cycling algorithm for use with MSNs. Results show that when static nodes are aware of their own location as well as the sink start point and speed, energy consumption can be reduced amongst significant nodes. This without the energy consuming exchange of messages. As such, MADCAL provides a crucial first step in this area. In future work, certain issues still need to be addressed such as possible delay in the event of a slow moving sink, which could result in some nodes asleep for longer than may be efficient. We propose extending MADCAL further to receive a minimal amount of information in network messages in order to alter duty cycling for individual nodes. A future development to adjust the factor utilised to control threshold size in a more dynamic way may also be advantageous. We also accept that at some point allowances must be made for a potential alteration of the MSNs path. During our work the effect of sink speed on the ability to receive data also became apparent and we see importance in a study of optimal sink speed.

References

[1] M. A. Uddin, A. Mansour, D. L. Jeune, M. Ayaz, and E. H. M. Aggoune, "Uav-assisted dynamic clustering of wireless sensor networks for crop health monitoring," *Sensors (Switzerland)*, vol. 18, no. 2, 2018.

[2] S. Ali, A. Ashraf, S. B. Qaisar, M. Kamran Afridi, H. Saeed, S. Rashid, E. A. Felemban, and A. A. Sheikh, "SimpliMote: A Wireless Sensor Network Monitoring Platform for Oil and Gas Pipelines," *IEEE Systems Journal*, vol. 12, no. 1, pp. 778–789, mar 2018.

[3] A. Ahmed, K. A. Bakar, M. I. Channa, A. W. Khan, and K. Haseeb, "Energy-aware and secure routing with trust for disaster response wireless sensor network," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 216–237, jan 2017.

[4] X. Tang and L. Xie, "Data Collection Strategy in Low Duty Cycle Wireless Sensor Networks with Mobile Sink," *International Journal of Communications, Network and System Sciences*, vol. 10, no. 05, pp. 227–239, 2017.

[5] L. Saad, C. Chauvenet, and B. Tourancheau, "Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies," in *SENSORCOMM 2011 : The Fifth International Conference on Sensor Technologies and Applications*, no. c, 2011, pp. 128–133.

[6] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Distributed Mobile Sink Routing for Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 877–897, 2014.

[7] L. Chen and K. Bian, "Neighbor discovery in mobile sensing applications: A comprehensive survey," *Ad Hoc Networks*, 2016.

[8] M. Jamalabdollahi and S. A. R. Zekavat, "Joint Neighbor Discovery and Time of Arrival Estimation in Wireless Sensor Networks via OFDMA," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5821–5833, 2015.

[9] L. Chen, Y. Li, Y. Chen, K. Liu, J. Zhang, Y. Cheng, H. You, and Q. Luo, "Prime-set-based neighbour discovery algorithm for low duty-cycle dynamic WSNs," *Electronics Letters*, vol. 51, no. 6, pp. 534–536, 2015.

[10] K. Wang, X. Mao, and Y. Liu, "BlindDate: A neighbor discovery protocol," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 949–959, 2015.

[11] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*. New York, New York, USA: ACM Press, 2008, p. 71.

[12] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" *Mobicom*, p. 185, 2012.

[13] R. Pozza, M. Nati, S. Georgoulas, K. Moessner, and A. Gluhak, "Neighbor discovery for opportunistic networking in internet of things scenarios: A survey," *IEEE Access*, vol. 3, pp. 1101–1131, 2015.

[14] N. Chakchouk, "A Survey on Opportunistic Routing in Wireless Communication Networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2214–2241, 2015.

[15] S. Yang, U. Adeel, Y. Tahir, and J. A. Mccann, "Practical Opportunistic Data Collection in Wireless Sensor Networks with Mobile Sinks," *IEEE Transactions on Mobile Computing*, no. 99, pp. 1–14, 2016.

[16] B. Zhang, Y. Li, D. Jin, and P. Hui, "Adaptive wakeup scheduling based on power-law distributed contacts in delay tolerant networks," in *2014 IEEE International Conference on Communications, ICC 2014*, 2014, pp. 409–414.

[17] F. A. Ghaleb, M. A. Razzaque, and A. Zainal, "Mobility pattern based misbehavior detection in vehicular adhoc networks to enhance safety," in *2014 International Conference on Connected Vehicles and Expo, ICCVE 2014 - Proceedings*, 2015, pp. 894–901.

[18] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noel, "Low-power neighbor discovery for mobility-aware wireless sensor networks," *Ad Hoc Networks*, 2016.

[19] F. Peng and M. Cui, "An energy-efficient mobility-supporting MAC protocol in wireless sensor networks," *Journal of Communications and Networks*, vol. 17, no. 2, pp. 203–209, apr 2015.

[20] A. Hess, E. Hyytia, and J. Ott, "Efficient neighbor discovery in mobile opportunistic networking using mobility awareness," in *2014 6th International Conference on Communication Systems and Networks, COMSNETS 2014*, 2014.

[21] S. Yu, B. Zhang, C. Li, and H. Mouftah, "Routing protocols for wireless sensor networks with mobile sinks: a survey," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 150–157, jul 2014.

[22] Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, vol. 00, no. C, 2005, pp. 1–9.

[23] S. Ghafoor, M. H. Rehmani, S. Cho, and S. H. Park, "An efficient trajectory design for mobile sink in a wireless sensor network," *Computers and Electrical Engineering*, vol. 40, no. 7, pp. 2089–2100, 2014.

[24] A. C. Viana and M. Dias de Amorim, "Sensing and acting with predefined trajectories," in *Proceeding of the 1st ACM international workshop on Heterogeneous sensor and actor networks - HeterSanet '08*, 2008.

[25] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "FarmBeats: An IoT Platform for Data-Driven Agriculture," in *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI '17)*, 2017, pp. 515–529.

[26] R. Dhall and H. Agrawal, "An Improved Energy Efficient Duty Cycling Algorithm for IoT based Precision Agriculture," in *The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018)*, vol. 141. Elsevier B.V., 2018, pp. 135–142.

[27] R. W. Pazzi, A. Boukerche, R. E. De Grande, and L. Mokdad, "A clustered trail-based data dissemination protocol for improving the lifetime of duty cycle enabled wireless sensor networks," *Wireless Networks*, vol. 23, no. 1, pp. 177–192, 2017.

[28] N. Ghosh, R. Sett, and I. Banerjee, "An efficient trajectory based routing scheme for delay-sensitive data in wireless sensor network," *Computers and Electrical Engineering*, vol. 64, pp. 288–304, 2017.

[29] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noel, "Optimizing the handover delay in mobile WSNs," *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, pp. 210–215, 2016.

[30] "IEEE 802.15.4," 2016. [Online]. Available: http://www.ieee802.org/15/pub/TG4.html

[31] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "RFC4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks," 2007.

[32] "IntRange," 2019. [Online]. Available: https://github.com/inetmanet/inetmanet/blob/master/src/underTest/wpan/linklayer/ieee802154/phyLayer/Ieee802154Phy.cc

[33] T. Clausen, "RFC 3626 - Optimized Link State Routing Protocol OLSR," Tech. Rep., 2003.

[34] L. A. Tawalbeh, A. Basalamah, R. Mehmood, and H. Tawalbeh, "Greener and Smarter Phones for Future Cities: Characterizing the Impact of GPS Signal Strength on Power Consumption," *IEEE Access*, vol. 4, pp. 858–868, 2016.

[35] "OMNeT++ Discrete Event Simulator," 2015. [Online]. Available: https://omnetpp.org/

[36] "MiXiM." [Online]. Available: http://mixim.sourceforge.net/

[37] "INETMANET INSTALLATION — Inetmanet Installation Guide."