Efficient Algorithms for MAC Layer Duty Cycling and Frame Delivery in Wireless Sensor Network

by

CRAIG JAMES THOMSON

Thesis submitted in partial fulfillment of the requirements
of Edinburgh Napier University, for the award of

Doctor of Philosophy

School of Computing
Edinburgh Napier University

SEPTEMBER 2020

Dedication

For Wendy, Harvey and Lilly. My crazy wee family.

And to my parents James and Sandra Thomson.

Thanks and I love you all.

Author's declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

SIGNED: .... DATE: ....

Acknowledgements

Doing a PhD has proven to be an at times overwhelming experience. Always challenging, sometimes frustrating, but then ultimately rewarding on a level I have never before experienced. The opportunity to become part of the research community at Edinburgh Napier University has been life changing. To take my work to places such as China and to connect with researchers all over the world and be rewarded for my efforts has been truly unforgettable. None of this would have been possible without the support of my supervision team, friends, colleagues and family.

Firstly to my Director of Studies **Prof Ahmed Al-Dubai**. Thank you for your support throughout my degree. Your technical knowledge and ability to challenge me has been essential. As has your good humour, approachability and friendship. Its my hope that we will work together again as colleagues in the future.

To my second supervisor **Dr Zhiyuan Tan**. You came into my supervision team at a difficult time and with your kind and enthusiastic approach, you have helped me produce the papers that have become the backbone of this research. Your friendship is as valued as your academic support and I can't thank you enough for it.

In addition I have to thank my panel chair **Prof Emma Hart**. Few panel chairs have had as much expected of them and I thank you for all your support throughout. Your insight at my RD meetings in particular forced me to lift my work to another level. But your help in the early part of my degree, when things did not go as planned, is what I'll always remember.

If there is one thing I have learned in this study it is that you cannot do a PhD alone. The support I've had from my two closest colleagues, friends, has been what has kept me going through the good times and bad. To **Peter Aaby**, I would never have even started this degree without your encouragement and I can state categorically I would not be here at the end without you. We've been together since day one, supporting and looking out for each other and as such, I've never felt like this was a solo endeavour. I can't thank you enough for that. You're one of the most generous people I've ever met and I hope a little has rubbed off on me!

To **Isam Wadhaj**, to all intent and purpose my third supervisor and chief collaborator. But much more than that. Your support in helping me understand the simulation framework in this study is what really got my studies moving and resulted in the success in publication that I have had. But more than that, you are quite simply just a good friend. To see the way you work and live, always having time for anyone, has been an inspiration to me and makes me want to be a better person in academia and in life. I thank you from the bottom of my heart for everything.

I've also had support from many other friends and colleagues in the School of Computing at Edinburgh Napier University. It has always been a fun and generous place to work and I can't thank everyone enough.

So finally, to my family. To my parents **James and Sandra Thomson**, who I think will be happier than me to see the end of my PhD! Thank for your love and support, not just over the last 4 years, but the last 50. You have to admit, at least its never been dull!

To my son **Harvey**. Sorry you had a Dad that was stuck in university or in the study at home most of the time for the last 4 years. But always know that I love you and thank you for bringing me down to Earth whenever I was too self-important about my studies!

A quick thank you to my gorgeous cat **Lilly** too for always being available for a cuddle when needed!

And lastly, to my beautiful wife **Wendy**. There isn't enough space here to do this justice. Supportive, patient, tolerant, proud of me even when I wasn't of myself. You were an amazing wife long before this PhD. I may not deserve you but I'm just glad I have you. So I'll just finish the easy way, thank you and I love you.

Craig

List of Abbreviations and Notations

ABBREVIATION	DESCRIPTION
6LoWPAN	**IPv6 over Low-Power Wireless Personal Area Networks**
CCA	**Clear Channel Assessment**
CSMA/CA	**Carrier Sense Multiple Access with Collision Avoidance**
CPU	**Central Processing Unit**
DMEAAL	**Dynamic Mobility and Energy Aware Algorithm**
DTN	**Delay-tolerant Networking**
GPS	**Global Positioning System**
IoT	**Internet Of Things**
IP	**Internet Protocol**
IPv6	**Internet Protocol version 6**
IEEE	**Institute of Electrical and Electronics Engineers**
MA	**Mobility Awareness**
MAC	**Media Access Control**

MADCaDPAL	**Mobility Aware Duty Cycling and Dynamic Preambling Algorithm**
MADCAL	**Mobility Aware Duty Cycling Algorithm**
MiXiM	**mixed simulator**
MSN	**Mobile Sink Node**
mWs	**Milliwatts per second**
ND	**Neighbour Discovery**
NDP	**Neighbour Discovery Protocol**
OFDMA	**orthogonal frequency-division multiple access**
OLSR	**Optimized Link State Routing protocol**
OMNeT++	**Objective Modular Network Testbed in C++**
OR	**Opportunistic Routing**
RFC	**Request for Comments**
RPL	**IPv6 Routing Protocol for Low-Power and Lossy networks**
UAV	**Unmanned Aerial Vehicle**
WBAN	**Wireless Body Area Network**
WF-IoT	**World Forum on the Internet of Things**
WSN	**Wireless Sensor Network**

Abstract

In Wireless Sensor Networks, with small, limited capacity devices now more prevalent, the issue of Neighbour Discovery has shifted. These devices utilise duty cycling methods in order to conserve battery power. Hence, the main issue is now that these devices may be awake at the same time in order to discover each other. When mobility increases complexity further. Rather than attempt to negate the issue of mobility, instead this thesis seeks to utilise a predictable sink mobility pattern in order to influence the duty cycling of static nodes.

Literature demonstrates a move towards Mobility Awareness in Neighbour Discovery in mobile Wireless Sensor Networks. However, there is a gap identified with sink mobility in use. Therefore, this thesis aims to establish to what extent the mobility pattern of a Mobile Sink Node in a Wireless Sensor Network may be exploited at the MAC layer, to influence the performance of static nodes. Such that network efficiency may be improved with energy consumption reduced and balanced across nodes. This study proposes three novel lightweight algorithms, with processing which does not add to the energy consumption within sensor nodes, these being Mobility Aware Duty Cycling Algorithm (MADCAL), Mobility Aware Duty Cycling and Dynamic Preambling Algorithm (MADCaDPAL) and Dynamic Mobility and Energy Aware Algorithm (DMEAAL). These located in the MAC layer of static nodes and utilising knowledge of predictable sink node mobility. This is in order to create a dynamic communication threshold between static nodes on the sink path and the sink itself. Subsequently lessening competition for sink communication between nodes.

In MADCAL this threshold is used to influence the sleep function in order that static nodes only awake and move to Clear Channel Assessment once the sink is within their threshold, improving energy consumption by up to 15%. The MADCaDPAL algorithm takes this approach further, using the threshold to directly influence Clear Channel Assessment and the sending of preambles, as such, closing off the threshold when the sink leaves it. This shows energy consumption lessening by close to 80% with a significant improvement in frame delivery to the sink. Finally, the DMEAAL algorithm utilises previous results to influence energy consumption in real-time by utilising a cross-layer approach, comparing current consumption to optimal target energy consumption and adjusting the threshold for each static node accordingly. This shows benefit in evening out results across nodes, thus improving network lifetime. All algorithms are achieved without the energy-consuming beacon messaging associated with Neighbour Discovery. Analysis and simulation results, tested on a lightweight implementation of a carrier-sense multiple-access-based MAC protocol, show a significant improvement in energy consumption and frame delivery in both controlled and random environments. In utilising a cross-layer approach to access energy consumption in static nodes, is it also shown to be possible to even out energy consumption across nodes by altering the communication threshold in real-time. As such, improving network lifetime by removing spikes in energy consumption in individual nodes.

TABLE OF CONTENTS

List of Tables

Tables **Page**

List of Figures

First Chapter

Introduction

1.1 Introduction

The Internet of Things (IoT) is no longer a future technological advancement and is now a current area of interest, with research performed in many different facets therein. We are now moving away from the original rigid infrastructure of the internet, with the adoption of wireless communications and mobile devices, and onwards to the use of tiny *Smart Objects* [1]. As such, challenges arise regarding how best to fit these small, low-power and low-memory devices into what is now a fast and memory-hungry internetworked world. The IoT has quickly spread into everyday life. Examples of this can already be found within the home, with the roll-out of smart meters [2, 3] and the use of entertainment systems such as the Amazon Echo [4]. The spread of the IoT can be further evidenced by the onset of the *Smart City* [5, 6, 7]. This envisages smart devices utilised in every aspect of city life. Recent events have brought the use of IoT devices to the forefront, such as in the use of robotic vehicles to deliver supplies in China in the midst of the COVID19 pandemic [8]. Therefore, there can also be observed a move away from the devices within the IoT being seen only as part of static networks.

This is particularly pertinent regarding Wireless Sensor Networks (WSNs). Throughout industry and academia WSNs have gained much attention in recent years. Comprising small, low power, limited-capacity devices, with data sent to a sink node on a many-to-one multi-hop basis. Given the environments in which these networks may be located, frequently inhospitable, with applications in many areas such as deep-sea oil and gas [9], disaster recovery [10], and agriculture [11, 12], it may not always be possible to replace the batteries in network devices [13]. Also, given the negative environmental implications in the production of batteries [14], an approach to extend their lifetime significantly would be of benefit. As a result, power conservation techniques which maximise node and, subsequently, network lifetime, take on importance.

Given that battery power within nodes in WSNs is expected to last around ten years, this would be impossible if nodes were constantly active [15]. The low-power and low-memory nature of devices in WSNs have led to approaches such as the nodes in the network saving energy by utilising long periods where the nodes are asleep. This is known as *duty cycling*. Duty is the fraction of time a device is in the active state during which beacon messages may be exchanged in order to establish Neighbour Discovery (ND) [16]. "For example, a device whose duty cycle is 1% activates during one time slot every 100 slots. The duty cycle length is thus 100 slots" [16]. However, whilst of benefit in terms of power consumption, duty cycling gives rise to other issues within WSNs. Heterogeneity of wake-up schedules within nodes in a WSN, as a result of duty cycling, leads to problems with regard to ND, such that it becomes a challenge to ensure these schedules overlap in order that nodes may discover each other and data may be transmitted between them. This issue can then be exacerbated if mobility is also a factor within the network, ND schemes also now having to allow for nodes being within the same vicinity as well as an overlap of wake-up schedule.

1.1.1 Neighbour Discovery with Sink Mobility

As shall be demonstrated in a review of literature, older, standardised approaches may address the issues inherent in ND within a static WSN, there is no allowance made for node mobility. This remains true for the well referenced use of mobility in WSNs that is sink mobility. This in order to combat the particular issue of energy and routing *hotspots* [17] in static WSNs. These hotspots occur near a static sink node as nodes closest to the sink will assume a greater responsibility in terms of routing than leaf nodes and, as a result, consume more energy. Ultimately, the lifetime of these hotspot nodes can be reduced considerably, with the other nodes in the network left unable to communicate with the sink node. Therefore, the WSN itself ceases to function properly. In utilising a Mobile Sink Node (MSN), moving around or across the network, the hypothesis is that energy consumption is spread more evenly amongst the nodes in the WSN. This is due to no particular node being able to take on the role of hotspot for a considerable period of time. As a result, network lifetime can be increased and the hotspot issue may be negated to a certain degree [18, 19].

The mobilising of sink nodes is possible via many different applications such as robots, vehicles, or being located on a person. However, the emergence of Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, provides further possibilities in this area. With studies having taken place to utilise drones in this way, to collect data and provide network connectivity [20] and with applications for use in areas such as dealing with forest fires [21]. As previously stated though, mobility adds another layer of complexity when considering ND and duty cycling. Sink mobility is no exception to this. Whereas in the use of MSNs, studies have taken place to extend existing work or, more specifically, develop new network layer routing protocols, thus far none have taken account of ND protocols. However, the issue of ND with regard to the mobility of devices in general in the IoT, and specifically WSNs, has seen a great amount of research. With ND protocols in WSNs generally more concerned with duty cycling at the Media Access Control (MAC) layer, referencing ND as a network layer issue has been consigned to older, more traditional fixed networks.

In networks consisting of small, battery-powered and duty cycled [16] devices, ND is no longer used to refer to IP addressing and duplicate address detection [22]. Now the greater concern is ensuring that ND is possible at all, with the overlap of wake-up schedules essential in this aim. In order to achieve this in an energy-efficient way, many probabilistic approaches have been proposed [23, 24]. These techniques have been shown to be efficient, however, there is also the possibility that probabilistic methods may result in the log-tail discovery issue. This is where it may be the case that a node is not discovered at all [25]. Resultantly, deterministic [26, 27, 28] algorithms occur more frequently in research. This is despite them being shown to be less efficient than probabilistic methods, but with the advantage of being able to guarantee an overlap, which probabilistic methods cannot claim. Recent studies have shown a move towards the adoption of new methods which may be integrated into ND. Opportunistic approaches involve decisions being made *on the fly* [29, 30, 31]. But it is in the area of mobility awareness in WSNs which is of interest within this study [32, 33, 34]. In this way it is proposed that routing and data delivery in a network may be improved by the prediction of mobile node mobility patterns [35]. However, in terms of influencing duty cycling and the wake-up schedule of nodes, as yet mobility has not been used in this area.

When considering the use of MSNs, many routing protocols have been developed for network layer solutions [36, 37]. These protocols can generally be separated into two categories, those being to implement flooding, which can result in high energy consumption, or delay-tolerant methods which compensate by being far more energy-efficient [38, 19]. However, in more recent studies the use of MSNs has been combined with clustering and optimal sink path determination, such that more energy efficiency may be improved along with network layer routing [39]. Whereas these studies are valid when considering the potential difficulties in ensuring network layer packet delivery when an MSN is implemented in a WSN, it should be noted that it is at the MAC layer where most energy consumption can be found [33].

It shall be demonstrated that there is a propensity for work utilising MSNs to be located at the network layer and with approaches which tend towards a *negation* approach. That being that, despite their perceived benefit in terms of network lifetime, the mobility of a sink node is something to be worked around rather than a metric to be utilised. That existing network layer solutions require to be extended in order to accommodate the sink mobility. However, if the aim is to reduce energy consumption then it would be the MAC layer which requires consideration, especially so when taking into account that this is where duty cycling takes place. Therefore, with sink mobility implemented primarily to conserve energy then it makes sense that a sink mobility approach to work with duty cycling would be of benefit. As such, it is proposed that sink mobility be utilised as a metric at the MAC layer in order to positively influence duty cycling approaches and thus, further conserve energy in a WSN. This implemented without consideration for any particular network layer protocol.

1.2 Thesis Statement

MSNs have been shown to reduce energy and routing hotspots in WSNs, by spreading the load amongst all nodes in a network [18, 19]. The statement of this thesis is that by utilising the mobility pattern of said MSN, the efficiency of duty cycling and MAC layer frame delivery in these environments can be improved. Subsequently, an approach to duty cycling in WSNs which incorporate a MSN may be developed, utilising a communication threshold between static nodes and the sink. With this communication threshold subsequently used to influence communication behaviour between node and sink in order that it takes place only at the appropriate time, again conserving energy. Also, by altering communication between nodes and the MSN based on context, such as the energy levels of nodes, the workload across the network may be balanced out, improving network lifetime.

Heterogeneity of wake-up schedules in a Wireless Sensor Network generally causes issues in terms of delay and, subsequently, energy consumption. The development of a MAC layer solution, to utilise the mobility pattern of the sink node to influence the wake-up schedules of nodes and the communication between the node and sink, would enable more effective and timelier ND and, therefore, more efficient delivery of frames. In turn, utilising energy levels in nodes within the WSN, in order to influence communication with the sink node, would result in a more balanced network performance in terms of MAC layer frame delivery and energy consumption. This context-aware solution would involve the constant monitoring of node energy levels at the MAC layer, adjusting the communication threshold to balance energy consumption. Ultimately, these solutions both individually and combined, would improve the lifetime of static nodes in a WSN and overall network lifetime.

1.3 Research Aims and Objectives

This thesis aims to utilise the mobility pattern of a MSN in order to influence the duty cycle of static nodes in a WSN. Hence, a predictable mobility pattern is used in order that a node may awaken as the sink reaches the static node's vicinity, when there is the greatest chance of any transmission reaching the sink node. The detailed objectives are as follows.

1. Identify requirements and key factors for utilising mobility that influences duty cycling to drive ND in mobile IoT environments.
2. Identify gaps in research of applying sink mobility to influence duty cycling.
3. Identify and implement a suitable MAC implementation for use with a WSN where duty cycling may be altered.
4. Identify a sink mobility pattern suitable for predictable sink mobility in a WSN.
5. Evaluate the current performance of the identified MAC protocol with static and mobile sink node, based on developed network topology and sink mobility pattern.
6. Develop an approach for altering duty cycling within the identified MAC protocol in relation to predictable sink node mobility. Such that wake-up and sleep scheduling schemes for static nodes will be influenced by a communication threshold determined by the MSN position, velocity and direction as well as sink

and static node transmission range. The resultant approach aims to reduce energy consumption across significant nodes which the sink passes, eliminating hotspots, and to increase frame delivery to the sink.

7. Develop an approach to influence MAC layer functionality, such as Clear Channel Assessment (CCA) and preambling, so that communication between the sink and static node ends once the sink moves beyond a predefined threshold. This proposed approach aims to further reduce overall energy consumption and improve frame delivery as well as to negate spikes in energy consumption.

8. Develop an approach to alter interaction between static nodes and the MSN based on a defined context, such as energy levels of static nodes. Such that the communication threshold between the static node and MSN is reactive to the context. With an initial communication threshold altered as time passes. Spikes in energy consumption are anticipated to be levelled so that network lifetime is extended, while retaining average energy consumption.

1.4 Research Questions

Ultimately, leading from the presented thesis statement and in completing the research aims and objectives, this thesis seeks to answer the following research questions:

1. To what extent can the mobility pattern of a MSN in a WSN be exploited at the MAC layer to influence the performance of static nodes, in order to increase efficiency in terms of network performance and energy consumption?

2. To what extent can a threshold built in a static node be utilised for communication with a MSN in order that said communication only occurs within the threshold?

3. To what extent can network performance and energy consumption be improved by this approach?

4. To what extent can nodes in a WSN utilising a MSN be context-aware? Such that said context, such as the energy levels of nodes, may be utilised accordingly to

adjust a communication threshold and ensure even energy consumption across nodes. As such, improving network lifetime.

1.5 Research Methodology

The methodology utilised shall involve the use of simulation to replicate a WSN with a mobilised sink node, due to the difficulty in implementing a network of suitable size on a physical testbed.

1.6 Research Contributions

This thesis proposes three novel dynamic Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC layer algorithms to utilise the mobility pattern of the MSN in order to positively affect the wake-up schedules of static nodes. These algorithms operate at the MAC layer, independent of any routing protocol and use no extra beacons or messages.

- The first algorithm is the Mobility Aware Duty Cycling Algorithm (MADCAL). The duty cycling of one-hop, or *significant*, nodes to the sink is based on the current location of the MSN and a dynamic communication threshold, calculated within each node, independently of all others. This calculation uses the transmission range of the node, its distance from the path of the MSN, and the speed of the MSN. The MSN location in relation to this communication threshold, as well as its speed, is then used in order to influence the sleep procedure within the MAC layer of the significant node. Such that until the MSN is within the node threshold, the node shall not exit its sleep procedure and move to CCA. When subsequently compared to standard duty cycling with CCA and check interval, MADCAL shows improvement in average energy consumption across significant nodes, as well as in the number of frames received by the MSN.

- The second algorithm is the Mobility Aware Duty Cycling and Dynamic Preambling Algorithm (MADCaDPAL). In the development of MADCaDPAL we examine the relationship between the MSN and static sink communication threshold in finer detail. Establishing that whilst the use of MADCAL alone is of benefit, in utilising sink position in relation to the static node to influence the behaviour of CCA and the sending of preambles, that further benefits can be achieved in relation to energy consumption and frame delivery. This whilst also countering the issue of *energy spikes* amongst significant nodes. Where, despite sink mobility, some nodes still take on a greater network load than others.

- The third algorithm is the Dynamic Mobility and Energy Aware Algorithm (DMEAAL), which utilises the threshold in a different way. DMEAAL takes a *real time* approach to energy conservation. In utilising previous results to determine a target, optimum, level of energy consumption, a cross-layer approach is proposed. With current, real-time, battery levels compared to the target energy consumption levels and the threshold adjusted accordingly. In this way, *dead energy consumption* is eliminated in significant nodes. Where the energy consumption of one node is unnecessarily higher than others, when the load could be spread evenly across all significant nodes. As such, the aim in this regard is to even energy consumption across significant nodes, as opposed to reducing energy consumption. Subsequently, increasing network lifetime.

1.7 Thesis Organisation

- **Chapter 2 - Literature Review-** The review of literature starts with a historical examination of the progression of ND, from use in static networks to within IoT environments and finally integrating mobility. This highlights utilising mobility awareness as an area of interest in this regard, leading to the second part of the review. Examining the uses of MSNs in past study, demonstrating a propensity for network layer functionality and development of optimal sink paths. Subsequently, highlighting the gap in research which this thesis aims to fill. That being the utilisation of a MSN with predetermined path to influence duty cycling.

- **Chapter 3 - Mobility Aware Duty Cycling - Testbed-** This chapter outlines the initial approach taken in this thesis and the background to the initial development of Mobility Aware Duty Cycling. This also details the common parameters utilised such as network topology and MAC implementation.

- **Chapter 4 - Mobility Aware Duty Cycling Algorithm (MADCAL)-** This chapter proposes the Mobility Aware Duty Cycling Algorithm (MADCAL). Utilising predictable circular sink mobility, with static nodes calculating the current position of the sink. Each *significant* node on the path of the sink implements a communication threshold. The sink position in relation to this threshold is used to influence static node duty cycling in relation to the MAC protocol SLEEP procedure.

- **Chapter 5 - Mobility Aware Duty Cycling and Dynamic Preambling Algorithm (MADCaDPAL)-** In this chapter the initial MADCAL algorithm is extended to influence the CCA and preambling functions within the MAC protocol. Resultantly, the threshold is now completely *closed* once the sink node reaches its end. The subsequent benefit in energy consumption are shown to be significant.

- **Chapter 6 - Dynamic Mobility and Energy Aware Algorithm (DMEAAL)-** This chapter details a cross-layer, context-aware approach to the use of a communication threshold between MSN and static node. Here, utilising a target energy consumption rate as well as accessing the battery within the node to ascertain current energy levels, the threshold is adjusted according in real-time. As a result, energy levels are shown to even out across significant nodes, extending network lifetime.

- **Chapter 7 - Conclusion and Future Work-** This chapter summarises the work performed in this thesis and the achievements gained, while also highlighting challenges and issues which may lead to future work.

Second Chapter

Literature Review

2.1 Introduction

The challenges inherent in integrating small, low-power and low-memory *Smart Objects* [40], into an internet which continues to grow rapidly in terms of speed and memory requirements, have proven to be vast. With the adoption of small devices, some mobile, the IoT can now be found to be integrated in everyday life. In a short space of time smart meters [2, 3] have been widely accepted within the home in order to more efficiently record energy use. In the area of home entertainment devices such as the Amazon Echo [4] or Sonos [41] products are designed to integrate with cloud applications such as Spotify [42]. Outside the home environment the IoT can be observed to be spreading to city life, with many aspects on the way to becoming interconnected. The *Smart City* [5, 6, 7] concept envisages smart devices becoming part of all aspects of the modern city environment. With this move, these devices can also no longer be seen as part of a static network. With the mobility of these devices and the vast applications appearing for their use therein, there lie many challenges and opportunities, none more so than when considering the use of WSNs. The challenges inherent in networks where small and low-power devices gather data and send it to a sink node are already well documented. These challenges multiply when the nodes are mobilised or, in the case of the focus of this thesis, the sink node itself.

2.1.1 Mobility in WSNs

A well-researched area is that of Wireless Body Area Networks (WBANs). Allowing sensors to be worn or even located within a human body, either in the blood stream or under the skin. In terms of *in body* sensors this can support the early detection of many conditions and illnesses [26]. Wearable body sensors can aid in the monitoring of individuals, this in order to pre-empt any health concerns. This can be of great importance in regard to elderly patient care, with prevention generally seen as of much greater benefit than treatment [26]. In recent years the use of robotics has entered this area [43]. The use of wearable sensors is not restricted to human beings, with applications to be found in agriculture. There has been a noticeable increase in the use of WBANs in recent years for the purpose of cattle monitoring [44][45]. One of the most important, and profitable, uses is in the recording of data in order to measure the optimal gestation period of cows. This done by retrieving the data from sensors that the cows wear constantly [46]. It can be noted, however, that these networks suffer from similar issues as WSNs with regard to the need to conserve energy. This of particular importance considering the applications utilised therein [47].

When examining the various applications of mobility in the IoT, the area of vehicular networks must be given particular consideration. Vehicular networks generally operate in an opportunistic way, where vehicles are merely transmitting or receiving data on a single-hop basis. This may be to inform other vehicles of an accident or to receive data from a roadside router [48]. The area of autonomous vehicles, more commonly referred to in the media as driverless cars, is an area generating great interest. With many companies, including Google [49], currently investing large amounts of money into development in this area. Another area of interest is that of UAVs, more commonly known as drones [50, 51, 20]. These can form entire aerial networks, with work already taking place in an attempt to coordinate communication between drones [52]. There is also great potential for the use of mobility in utilising robots, mobile sensors or sensors located on people in Industrial Wireless Networks [53]. These sensors may function as part of the network itself, alongside static nodes. However, a further possibility exists of mobilising the sink node itself, in order to gather the data from the other nodes in the network, with various different applications for this approach in industry.

2.1.2 Mobile Sink Applications

The theoretical benefits of the utilisation of MSNs are clear when considering the issue of energy hotspots, as well as the problem of transmission bottlenecks near a static sink node.

However, it is important to demonstrate that there are real-world applications to which the use of MSNs may be applied. WSNs can be located in extremely inhospitable environments and consequently, it can be problematic to retrieve data from the nodes. There is also the concern in these environments of failure of the sink node and the inability to successfully replace it. Thus, effectively causing the failure of the entire network, or at least a large part of it. One of the most extreme examples of this is in the area of volcano monitoring. The retrieval of data from sensors within a volcano comes with obvious dangers inherent to both the network and personnel. Therefore, the use of UAVs is proposed to fly over the static sensors in order to collect data [54]. UAVs are a current area of research in regard to their deployment as MSNs, with many different applications mooted such as disaster recovery, agriculture and also in Linear Sensor Networks [55, 56, 51]. This can be in regard to the monitoring of networks along straight line implementations such as roads, bridges and pipelines [57]. UAVs are also expected to play a major role in smart cities with sensor data gathering merely one of many applications of this technology [58]. There is even the proposal of utilising UAVs as a cloud which would then be capable of providing a service, rather than merely performing the functions of a sink node [59].

Sink node mobility, however, is not restricted to UAVs. The sink may be carried by a person or placed on a robot, at ground level. Amongst other applications, this has been proposed for use within theme parks, where the sink would move around the park retrieving information from static nodes. This could help in the case of emergencies where evacuation is necessary or could return valuable information from sensors located on rides [60]. Mobile robots can also be programmed to visit a set of static sensors in order to retrieve data [61]. One novel use of MSNs is to place a sink node within an elevator in the Guangzhou New TV Tower. As the elevator moves and the sink collects data on each floor this can aid in monitoring the health and structural integrity of the building [62]. Mobile sinks are also seen as an important next step in the monitoring of underwater WSNs. The sink may be placed in an automated underwater vehicle to aid in environmental and pollution monitoring as well as the prevention of disasters [63].

2.2 Duty Cycling in Mobile WSNs

Considering the low-power nature of devices in WSNs, duty cycling becomes of great importance in order to conserve energy. Duty cycling is implemented in MAC protocols for WSNs to ensure batteries remain on for very short periods of time, hence using less energy. However, it is accepted that other areas of network performance may have to be compromised in order to utilise duty cycling, such as delay, increased collision rates or greater overhead if synchronisation is required [64]. Duty cycling tends to fall into the categories of synchronous or asynchronous, with synchronisation requiring extra messaging on the network in order to ensure synchronised timing [64].

The effects of synchronisation of duty cycling on the 802.15.4 [65] MAC standard are explored by Choudhury et al. [66]. This demonstrates a benefit in synchronisation in terms of the avoidance of collisions. However, synchronisation is achieved by the sending of beacon frames in order to establish the timing of wake-up schedules of nodes. These beacons therefore add to the load placed on the network. Therefore, it would be of interest of synchronisation of any kind could take place without the need for beacon frames.

However, given the need for control traffic to be placed on the network in order for synchronisation to work, many asynchronous proposals have been developed in response. As such, these do not require nodes to agree timing of wake-up schedules in order to operate. One asynchronous approach of significance historically is the sending of preambles [67]. This originally incorporated into MAC protocols such as S-MAC [68] and B-MAC [69]. This approach requires a preamble to be sent that is at least the length of the frame to come. The initial use of long preambles was eventually usurped by short preamble techniques, due to the wasteful use of channels by longer preambles. Shorter, strobed, preambles are useful from the point of view of being interruptable [64]. Other asynchronous approaches such as on demand wake-up and random duty cycling have also been shown to be of benefit, but preambling is the approach that will be affected by the work in this thesis.

2.3 Neighbour Discovery in Mobile WSNs

Duty cycling causes difficulty with regard to ND and the transmission of data at both Layer 2 and Layer 3 in WSNs, especially if heterogeneous wake-up schedules are utilised. However, in WSNs where devices can be mobile there are a great deal more issues to address. This, along with the lossy nature of nodes in these networks, means the original, multicast-heavy, ND protocol for IPv6 is not suitable for these networks.

Considering the MAC layer being responsible for so much energy consumption [70], the duty cycling of wake-up schedules has become of paramount importance in WSNs. As a result the focus of NDP functionality has shifted when considering mobile WSNs, with ensuring the probability of nodes discovering each other in the first place now of importance. As would be expected, this has resulted in a considerable number of studies in this area.

Historically, when considering NDP, a significant issue has been of synchronicity of wake-up schedules. When examining state-of-the-art approaches to ND in WSNs, the main issue of study is in regard to unsynchronised devices and the ability to guarantee an overlap of communication. In this area the two approaches that dominate research are in the use of *Probabilistic* and *Deterministic* protocols. These can be summarised as thus:

- **Probabilistic** - Exploit statistical properties to guarantee a high probability of temporal overlap of slots. However, overlap is not guaranteed, with the long-tail discovery latency problem a common issue. This means that there remains the possibility, even remote, of neighbouring nodes never actually discovering each other [25]. The base of most probabilistic protocols are the Birthday Protocols suite [71]. These protocols are based on the *Birthday Paradox*. That being the probability that if there are as few as 23 people in a room, then there is at least a 50% chance of two or more having the same birthday [71]. This probability increases with the number of people. The same principle can be applied to the wake-up schedules of neighbouring nodes. The probability of an overlap increasing with the frequency of wake-up slots. This is a simplistic description based on legacy protocols, where more recent work applies these methods in more complex ways.

- **Deterministic** - Use number theory to guarantee a communication overlap. This requires a synchronisation of time slots, however [25]. This generally involves the use of prime numbers or quorums in order to guarantee an overlap. One of the earlier protocols using the *prime* method is *Disco* [27]. When using Disco, nodes will choose a pair of prime numbers, n1, n2. Where the sum of the reciprocal values (1/n1 + 1/n2) should be equal to the required duty cycle. The activation of the node's radio is controlled by the use of a local-counter, incremented by a globally-fixed period. When this value becomes divisible by either of the node's selected prime numbers, the radio is turned on for a single period. This ensures two nodes must have an overlap in radio activation, resulting in a much lower-bound of discovery than a probabilistic approach [27]. Also referenced in regard to Prime-based Deterministic approaches is the use of the *Chinese Remainder Theorem* [72].

Alternatively, original *quorum* based protocols use a two-dimensional array based on the square of a global parameter. For wake-up periods, each host chooses one row and one column, ensuring an overlap of at least two periods for any two nodes [73]. Whilst from a latency viewpoint this ensures a much lower-bound in the worst-case, a Probabilistic approach is much better on average. Also, this two-dimensional approach is restrictive and has been improved upon in later work [28].

Whilst reference is made to Neighbour Discovery, it could be said that the term *Neighbour Discovery* has become somewhat of a misnomer in the area of WSNs, and in particular with regard to mobility. Whilst efficient beaconing is still of importance, in a mobile WSN environment the most important aspect would now appear to be the need for MAC protocols which control duty cycling and wake-up schedules. As without this nodes cannot be awake to discover each other in the first place.

2.3.1 Mobility Awareness/Opportunistic Routing

Within the field of mobility in WSNs there is a great deal of reference to *opportunistic routing* (OR). This can also be referred to as Delay-Tolerant Networking (DTN). In contrast to traditional routing, OR involves decisions being made regarding the routing of a packet without a route ever being guaranteed. This may be involved in industrial applications, disaster recovery or rural areas with poor internet coverage. This is a similar principle to mesh networks in that every router can be involved in forwarding traffic. Routing can be performed dynamically and *on the fly*, no knowledge of network topology is required.

This approach, tailored with mobility in mind, has been shown to be of interest with regard to WSNs. With energy saving of paramount importance, OR lends itself well to the more efficient delivery of packets, therefore lessening expensive re-transmissions, but with the addition of possible delays while opportunistic connections are detected. Depending on the particular application in use, this may be a negative factor when considering this approach. However, it is within the area of OR where reference is initially made to utilising MA [25].

MA has been proposed as a future direction to take in the area of mobility in WSNs. The principle being that if mobility patterns of mobile nodes can be successfully predicted, then the delivery of packets can be made with these patterns in mind, as opposed to the energy consuming process of exchanging beacons. The velocity of nodes may also render current approaches inadequate, given the potential for extremely short connection periods [35]. This is of particular relevance when in connection with opportunistic approaches, where short connection times are assumed. As has been stated, there is a disparate nature to previous studies in this field. Approaches may involve attempting to predict mobility movements of personal devices based upon information gleaned at a social level. This can involve utilising previous movements of the personal device in order to ascertain, to a fair degree of certainty, the future movements it will take. Much of this work, however, does not consider ND, and duty-cycling in particular. The work reviewed here will show legacy studies in the area of MA, whilst aiming to identify potential future work with regard to ND.

2.4 Mobile Sink Node in Wireless Sensor Networks

2.4.1 Introduction

The mobilisation of sink nodes is generally proposed in order to combat the particular issue of energy and routing *hotspots* [17] in static WSNs. These occur near the sink node as nodes closest to the sink will assume a greater responsibility in terms of routing than leaf nodes and, as a result, consume more energy. Ultimately, the lifetime of these hotspot nodes can be reduced considerably, with the other nodes in the network left unable to communicate with the sink node. Therefore, the WSN itself ceases to function properly. In utilising a mobile sink node (MSN), moving around or across the network, energy consumption is spread more evenly amongst the nodes in the WSN. This due to no particular node being able to take on the role of hotspot for a considerable period of time. As a result, network lifetime can be increased and the hotspot issue negated to a certain degree [18, 19]. The mobilising of sink nodes is possible via many different applications such as robots, vehicles or located on a person. However, the emergence of UAVs provides further possibilities in this area. With studies having taken place to utilise drones in this way, to collect data and provide network connectivity [20].

However, while the benefits to be gained from sink node mobility are referenced, subsequent issues arise in the mobilisation of the sink, especially so with regard to routing. As a consequence of this, the majority of studies take a *negation* approach to sink mobility, that being that it is a problem to solve in order that existing approaches may continue to work. Additional work focuses on determining an optimal sink path. In this section of the literature review the various approaches to sink mobility are examined; in order to demonstrate that a gap exists in the use of sink mobility as a metric by which network performance may be improved.

2.4.2 Mobile Sink Mobility Patterns

When considering sink mobility, historically the different mobility patterns can be divided into three main categories [74]:

- **Random Mobility:** Sometimes referred to as unpredictable mobility. Put simply this is where the sink node is attached to something which will not move in a way which can be or is expected to be, controlled. This could be when animal monitoring is involved [46].
- **Controlled Mobility:** In this case it is expected that the sink node is located on a mobile device which may be controlled, such as a car or robot. It would then be expected that mobility would henceforth depend on certain factors such as energy consumption and latency. Nodes within the network could then route data dependant upon the potential future location of the controlled sink node.

- **Predictable Mobility:** This is the simplest mobility pattern to conceptualise and yet as shall be demonstrated in the review of literature, is not extensively used. Basically the path on which the sink node shall travel is pre-determined. This is beneficial in that if the sink node position can be calculated, certain network events may be influenced by this.

These may be further classified depending on the particular implementation. For example, random mobility may be related to wildlife or could be influenced socially if the mobility is of people. Controlled mobility may be influenced geographically, this could particularly relate to the sink node stopping at particular locations to gather data. Predictable mobility may be path constrained, such is if located on a train or bus [75].

When considering the simulation of these patterns, simulation software such as Cooja [76] or OMNeT++ [77] have both been used in literature to simulate sink mobility. In simulating a random pattern the Random Waypoint Model has been proposed [75], although it should be remembered that it true simulation of random movement is unlikely to be attained. The OMNeT++ simulation platform in particular provides many different models for mobility such as tractor mobility, where the node moves in a grid like fashion. With this generally used to attempt to pass every single node in the network [56]. It is also possible to move the sink in straight lines, controlling the angle of movement. Or around the network in a square or circular fashion [77, 78]. The particular mobility model utilised in this thesis and justification for its use shall be detailed later.

2.5 Conclusion

This chapter provides background to the areas addressed within this thesis. The requirement for wake-up scheduling is now at the forefront of ND developments in mobile WSNs, replacing the traditional view of ND as a static, network layer function. With functionality now moving more towards the MAC layer to control duty cycling of nodes. Sink mobility is of interest as it is already a common approach used to spread load across a WSN and negate the *hotspot* issue. Therefore, a research gap appears where implementations utilising sink mobility could use that mobility to influence network behaviour, in particular duty cycling.

Work where MSNs are utilised focus on the desire to establish optimal sink paths through a network. What is also clear is that when examining ND with mobility, even where MA is proposed as a solution, the mobility remains an issue to be negated. Within this thesis the proposal is that with regard to the use of a MSN with ND, this approach be turned *on its head.* Such that, rather than an issue to worked around, the mobility of the sink node becomes a network metric by which other nodes in the network are positively influenced. In influencing ND then this study must be located at the MAC layer as opposed to the many studies which seek to influence network layer behaviour.

As such, when considering the issue of duty cycling in a WSN which implements a MSN, the issues to be addressed can be summarised as thus:

1. **Location of Mobile Sink.** Determination of the location of the MSN requires a lightweight solution as constant re-transmission using, for example, Global Positioning System (GPS) would result in too great an expenditure of energy.

2. **Context.** The particular application for which the MSN is being employed, as well as the environment in which it is to be utilised, may have a direct effect on the mobility. This subsequently affects the wake-up schedule of nodes.

3. **Mobility Pattern vs Heterogeneity.** There are various different models for mobility, which may be affected by various different factors. In short, regarding MSNs, mobility may be predetermined, completely random or reactive to certain factors. The pattern utilised will affect whether it is possible to predetermine optimal wake-up schedules or whether it should be reactive to the position of the MSN. The alternative would be a solution based upon existing Deterministic

and Probabilistic approaches, allowing for heterogeneity of devices and wake-up schedules.

4. **Velocity.** This is directly linked to the mobility pattern of the MSN and again is a factor in determining the ability to predict wake-up schedules or whether approaches which accept heterogeneity should be utilised.

In the following chapter the approach taken and methodology implemented is detailed.

Third Chapter

Mobility Aware Duty Cycling - Testbed

3.1 Introduction

The approach taken in this study is to utilise a MSN with the lesser used predictable mobility pattern in order to positively affect the wake-up schedules of static nodes. All algorithms presented in this thesis operate at the MAC layer and use no extra beacons or messages. The duty cycling of one-hop nodes to the sink is based on the current location of the MSN and a communication threshold, calculated within each node, independently of all others. This calculation uses the transmission range of the node, its distance from the path of the MSN and the speed of the MSN. The rest of this chapter outlines the initial approach and the common parameters utilised such as network topology and MAC implementation. The subsequent chapters shall detail each particular algorithm developed.

3.2 Mobility Aware Duty Cycling

3.2.1 Beacon Messaging

Existing studies demonstrate a propensity to determine, by various different parameters, an *optimal* path for the MSN, which has merit in the results of each particular work. However, a common theme to be found is in the approach taken to keep track of the MSN. The regular exchange of beacon messages in order to facilitate this, however, can have a significant negative effect on the energy consumption of a network. The new mobility-aware duty cycling approach proposed in this thesis utilises a predefined mobility pattern. The hypothesis being that given network parameters such as sink start position, speed and time, each static node is capable of independently calculating the current sink position. This can therefore be achieved without the expensive exchange of messages, resulting in a lightweight algorithm with no network overhead.

3.2.2 Significant Nodes

The first step in this study is to identify the *significant nodes* with which the MSN shall communicate directly as it travels along its path. In effect, these nodes now take on the role of *hotspot*, with responsibility for communication with the sink node spread amongst them. The initial focus of this study is to reduce energy consumption amongst these 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 within a reasonable boundary.

3.2.3 Mobility Pattern

An important factor to consider is that this study does not seek to establish an optimal mobility pattern. The aim of this thesis is to utilise a *predictable* mobility pattern, to establish if pre-knowledge of sink mobility may be used to positively affect duty cycling and therefore, improve energy consumption. Consequently, the particular mobility model in use is not as important as that the pattern is predictable. When considering the mobility pattern, it was decided to use a simple pattern but with an implementation inspired by real-world applications. Therefore, an environment where all nodes are not treated equally, such as a disaster recovery situation where reaching each node directly is not possible. Other studies have aimed to have the MSN pass as many nodes as possible [79, 80, 81], and whilst such an approach has merit, it is clear that this is not always possible in a real-world scenario. For example, a forest fire scenario may render it impossible for a mobile sink to reach every single node.

Therefore, throughout this study the approach is taken to replicate a network scenario where reaching all nodes would not be possible. This is achieved by moving around the periphery of the network in a circular pattern, such that communication with the sink is only possible for the outermost nodes. With internal nodes communicating via multiple hops.

3.2.4 Network Topology

With regard to network topology, two approaches have been taken. Firstly, a one-hop grid formation is used. This in order that results could be observed when controlled location of static nodes was in effect. As a result, it is easier to observe the effect that communication with the MSN has on the static nodes in the network, with both transmission range and distance from the path of the sink node reasonably consistent. Figure 3.1 shows the network layout, with the start point of the MSN and the clockwise direction of travel. In addition a more random topology has been deployed in order if effectiveness of the subsequent algorithms remains when communication between nodes is more *strained*. As such there has been no effort made to spread the nodes in the network in an even fashion, with the intention being to ensure communication with the sink is problematic. This can be seen in Figure 3.2.

As can be observed, depending on transmission range, certain nodes within the network will be within one-hop of the MSN. These are determined to be the aforementioned *significant nodes*, taking the place of *hotspot* nodes in a network where the sink node remains static. Only now the significant nodes take it in turn to have final responsibility for relaying data to the sink. As the ultimate aim is to reduce energy consumption and increase network lifetime, this benefit is negated if only certain nodes have channel access and thus, the ability to communicate with the sink node. Whilst other nodes wait for the channel to become clear before communication with the sink can commence. Therefore, we propose the development of a threshold of communication in order to ensure fair access to the MSN. The determination of significant nodes

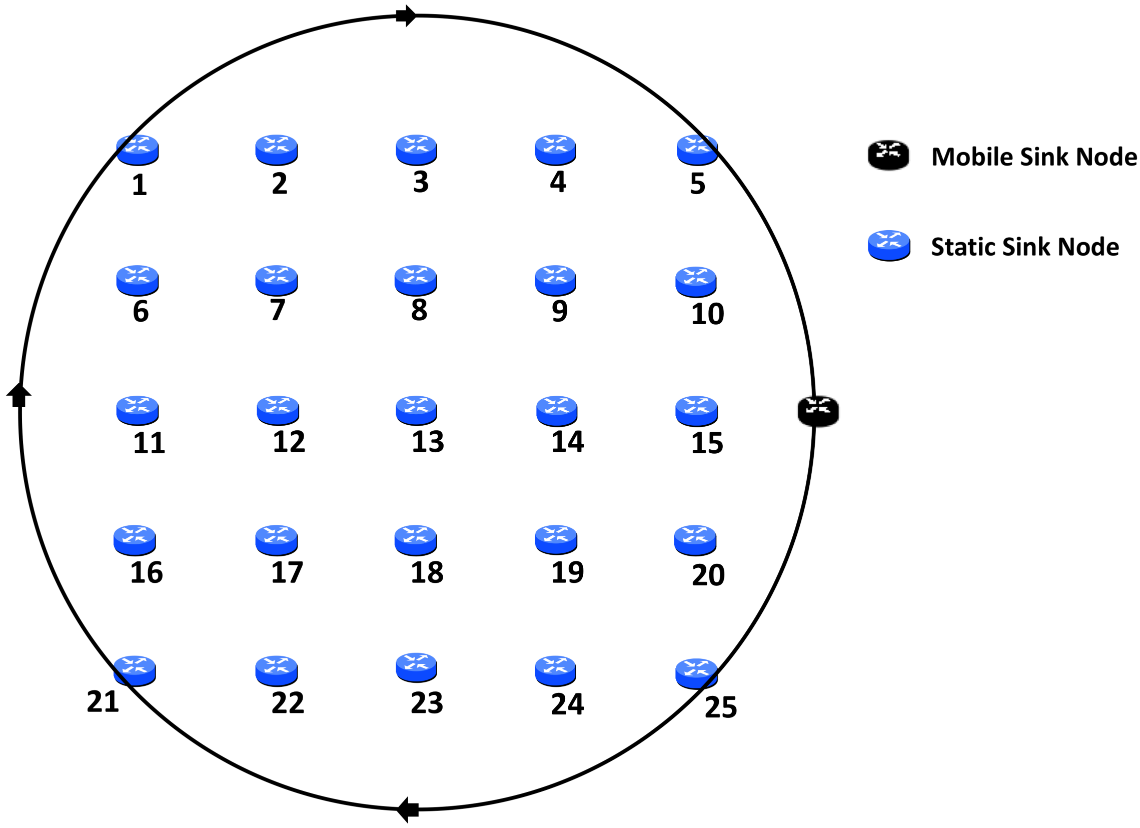


Figure 3.1: Network Topology - Grid

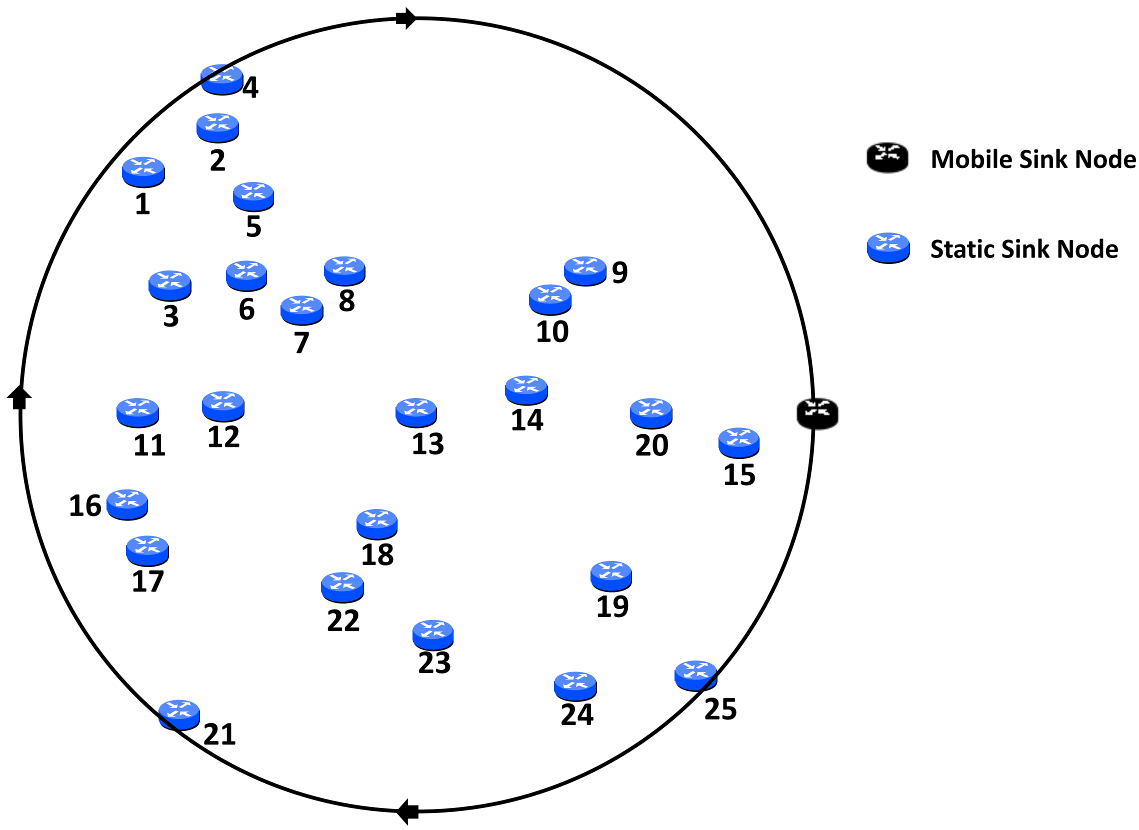


Figure 3.2: Network Topology - Random

and the calculation of the communication threshold is detailed in Chapter 4.

3.2.5 MAC Implementation

It is at the MAC layer where the greatest amount of energy is consumed [82], and it is here where duty cycling takes place. Hence, the three algorithms developed in this study are implemented at the MAC layer, within each static node. The MAC implementation utilised in this study to test the effectiveness of these algorithms is a lightweight CSMA/CA implementation which reflects the core functionality of the IEEE 802.15.4 standard [83, 84]. This uses CCA and the transmission of preambles. Figure 3.3 demonstrates this MAC implementation as utilised within all nodes. Subsequent chapters shall demonstrate where each algorithm is inserted within this implementation for static significant nodes. With nodes deemed not to be significant reverting to the existing duty cycle approach utilising CCA and the sending of preambles.

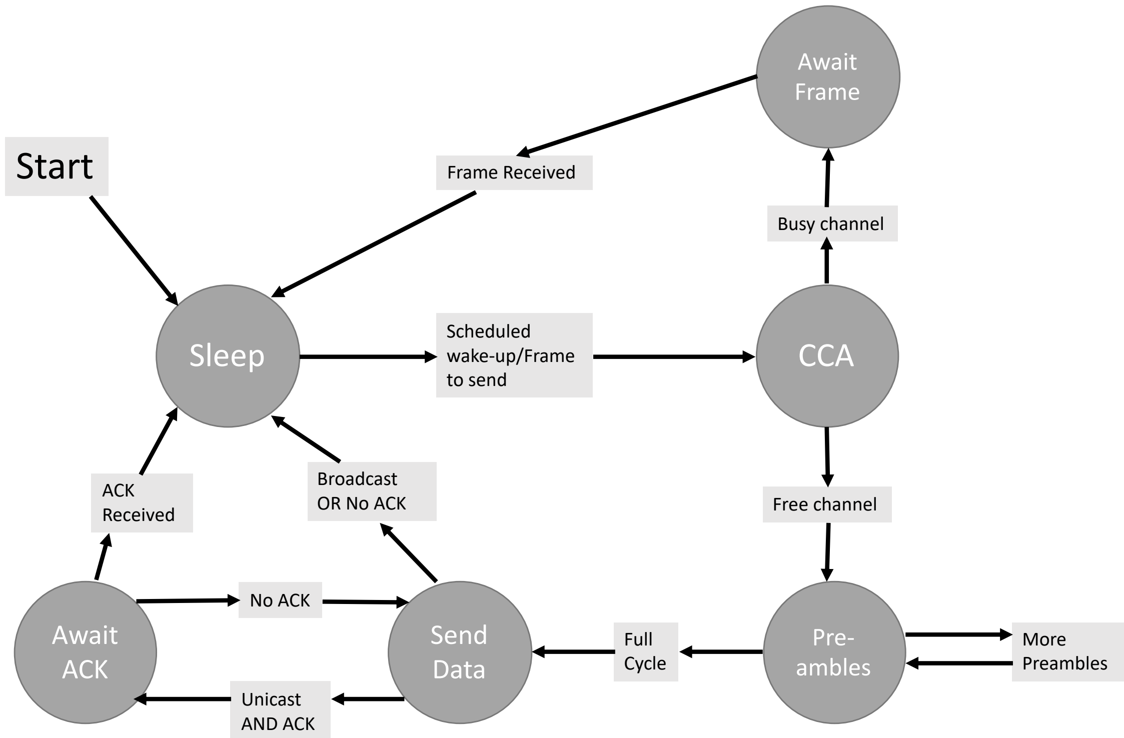


Figure 3.3: Initial MAC Implementation

3.2.6 Network Assumptions

The following properties of the WSN are assumed:

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 MADCAL, MADCaDPAL and DMEAAL independently.
4. Node power levels are consistent.
5. Transmission ranges, though variable across tests, are consistent across static nodes and the MSN.
6. Sink speed shall not be less than 2 metres per second (mps) or greater than 40mps. This in order to establish MSN effectiveness across a range of speeds, from slow to very fast.

3.2.7 Network Layer

This study does not seek to test routing protocols. However, a routing protocol is required in order to ensure final delivery is to the MSN. Otherwise, MAC layer frames would simply be sent in bursts, with the MSN behaving as any other node in the network would. Given this, the Optimized Link State Routing protocol [85] (OLSR) is utilised in this study. This is an unconventional approach as OLSR is not usually used in a WSN environment. However, OLSR is a resource heavy protocol which places a great load on the network, especially in terms of energy consumption. For the purpose of this study this proved to be of benefit and resulted in accelerated tests, requiring a lower simulation time in order to acquire the desired results.

Fourth Chapter

Mobility Aware Duty Cycling Algorithm (MADCAL)

4.1 Introduction

In this chapter a novel Mobility Aware Duty Cycling Algorithm (MADCAL) is proposed, in order that sink mobility may be used in order to influence the duty cycle of static nodes in the network in order to answer the first research question:

- To what extent can the mobility pattern of a MSN in a WSN be exploited at the MAC layer to influence the performance of static nodes, in order to increase efficiency in terms of network performance and energy consumption?

In creating the MADCAL algorithm, a predictable sink mobility pattern approach is utilised. Then, given the network parameters of sink starting position, speed and the time it has been travelling for, it is possible for each static node to accurately calculate the current sink position. This calculation made independently in each node, without the use of energy expensive beacon messages. As stated earlier, the mobility pattern utilised by MADCAL is a circular path around the network. Therefore, the nodes to be identified, the *significant* nodes, are those which are one-hop from the path of the sink based on transmission range.

Whilst the need to conserve energy is a constant factor in the function of WSNs, studies have thus far failed to utilise sink mobility directly in influencing duty cycling. Considering the benefits documented in the implementation of sink mobility in terms of countering the hotspot issue, to further benefit from this to save energy across the entire network would be significant. Given the lack of directly comparable studies, when considering comparisons in results in order to ascertain the effect of MADCAL in a quantifiable way, the approach is taken to examine a network implementing a MSN and then compare the effect of utilising MADCAL. This is covered in greater detail later in this chapter. The MADCAL algorithm features two sub-algorithms summarised as follows.

The first to create a threshold between static node and mobile sink. In taking the point closest to the circular sink path in relation to the significant node, deemed the *circlePoint*, an angle may then be calculated based in transmission range. This would be the maximum communication threshold before and after the circlePoint. However, as this would result in a large threshold when the static node is close to the sink path and a smaller threshold for nodes a greater distance away, further adjustments are required. As such, a factor is applied in order to regulate threshold size, initially by taking the distance to the sink path and dividing this by the transmission distance. Sink speed is taken into account, with a slower sink requiring larger thresholds, due to fewer opportunities to communicate with each node, and vice versa. Therefore, in order to take sink speed into account, the following calculation is made. For speeds less than 10 mps the threshold cannot be reduced by less than a factor of 0.5, for less than 20 mps this factor reduces to 0.35, reducing again to no less than 0.25 for less than 40 mps. If the speed is exactly 40 mps we take no action other than the factor remaining at the initial calculation of node distance over transmission distance.

The second MADCAL algorithm focuses on how to utilise this threshold in order to then effectively influence duty cycling within a static node. By calculating the current MSN position within the SLEEP procedure of the MAC layer implementation in use, the node can calculate if the sink is within its threshold or not. If it is not then the node may delay the move from SLEEP to CCA, in effect remaining asleep whilst still able to receive messages. This is achieved by calculating the time it will take for the sink node to reach the start of the threshold based on the current sink coordinates, coordinates of the threshold start, current time and sink speed. This will be compared to a standard duty cycling approach with CCA, preambles and check interval, to show an improvement of up to 15% in energy consumption, as well as keeping frame delivery at similar levels or better.

4.2 Determination of Significant Nodes

We utilise the grid and random network topologies as shown in Figure 3.1 and Figure 3.2. Test parameters shall show the transmission range of nodes is varied across four values. Within the grid topology this results in the same significant nodes each time, as can be seen in Figure 4.1. However, with the random topology in use the number of

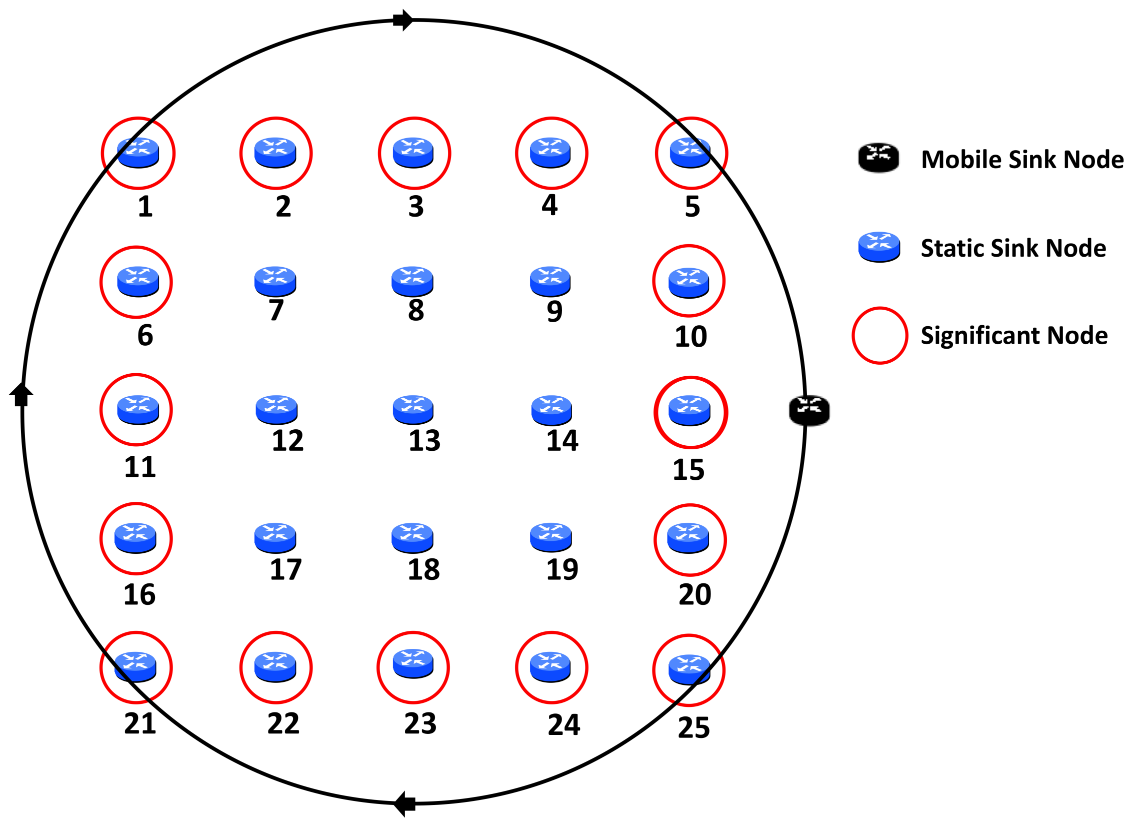


Figure 4.1: Network topology - grid with significant nodes.

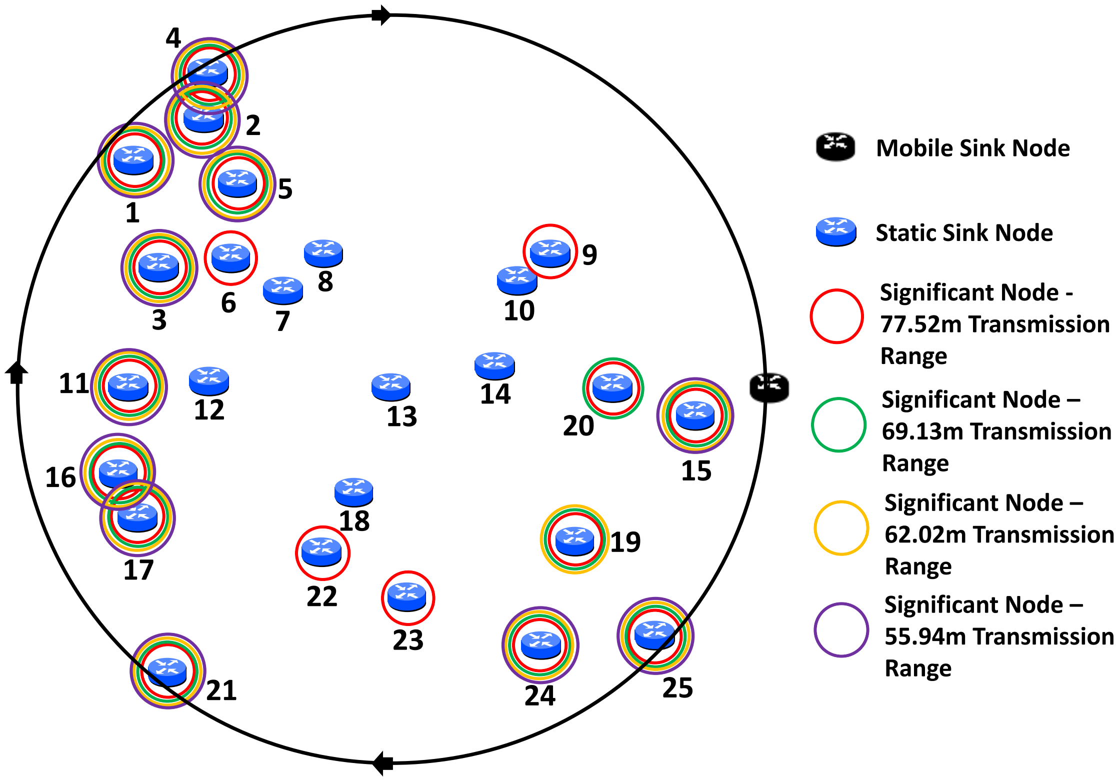


Figure 4.2: Network topology - random with significant nodes.

significant nodes reduces as the transmission range of nodes becomes smaller. This can be seen in Figure 4.2.

Among both network implementations the assumption is that static nodes will retain their positions throughout. Each static node implements MADCAL independently and although aware of its own location, is unaware of neighbouring nodes. Although four different transmission ranges are used across different test scenarios, each range is consistent across all nodes in each test.

4.3 Mobility Aware Duty Cycling Algorithm (MADCAL)

4.3.1 MAC Implementation

The MADCAL algorithm is implemented at the MAC layer, utilising MAC implementation detailed in the previous chapter, using CCA and the transmission of preambles. Figure 4.3 demonstrates this MAC implementation, with the location of the MADCAL functionality highlighted. This firstly shows where the initial communication threshold

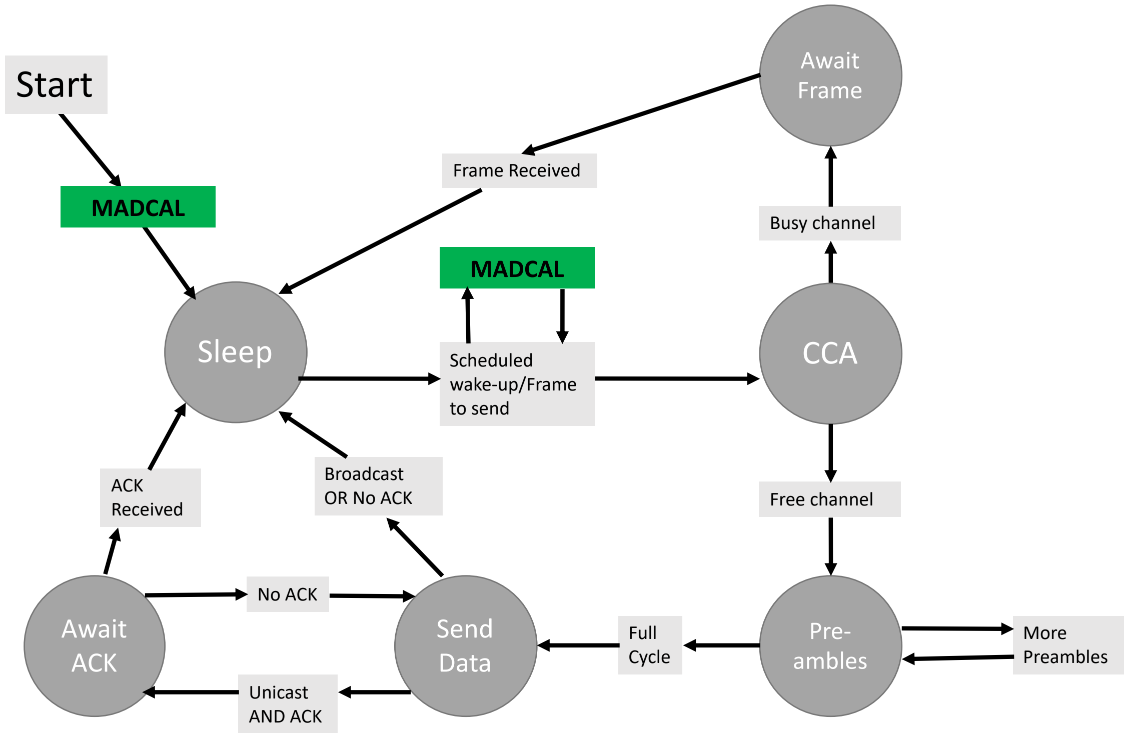


Figure 4.3: MAC Implementation with MADCAL

within the node is calculated, as covered next, and secondly to show how the normal wake-up schedule is now intercepted.

4.3.2 The Communication Threshold

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 transmission 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 threshold is calculated during the initialisation stage of each static node and is detailed in Algorithm 1.

Algorithm 1: In the Initialisation procedure the MSN speed is set from input at 2 metres per second (mps), 10mps, 20mps or 40mps with whether this is a significant node not yet established. The transmission distance, *transDist*, of the node is calculated

Algorithm 1 Communication Threshold

```
procedure INITIALISATION
    set sinkSpeed
    significantNode ← false
    set transDist
    set Circumference
    set firstSinkPos
    set firstSinkQuartile
    set distToCircle
    if distToCircle < transDist then
        significantNode ← true
    end if
    if significantNode then
        set circlePoint
        set nodeQuartile
        set distanceBetweenPoints
        set angleOfNode
        thresholdAfter ← true
        sinkThresholdAfter ← establishThreshold(sinkRadius, thresholdAfter)
        thresholdAfter ← false
        sinkThresholdBefore ← establishThreshold(sinkRadius, thresholdAfter)
        set thresholdDistance
        set beforeQuartile
        set thresholdOpposite
    end if
end procedure
function ESTABLISHTHRESHOLD(radius, after)
    nodeDist ← (radius − distToCircle)
    angleTemp ← (radius^2 + nodeDist^2 − transDist^2) / (2 ∗ radius ∗ nodeDist)
    angleRadians ← arccos(angleTemp)
    angle ← (angleRadians ∗ (180 / PI))
    factor ← distToCircle / transDist
    if sinkSpeed < 10 then
        factorCheck ← 0.5
    else if sinkSpeed < 20 then
        factorCheck ← 0.35
    else if sinkSpeed < 40 then
        factorCheck ← 0.25
    end if
    if factor < factorCheck then
        factor ← factorCheck
    end if
    angle ← (angle ∗ factor)
    if after then
        threshAngleDegrees ← (angle + angleOfNode)
    else
        threshAngleDegrees ← (angleOfNode − angle)
    end if
    threshAngleRadians ← threshAngleDegrees / (180 ∗ PI)
    threshold.x ← circleCentre.x + (radius ∗ cos(threshAngleRadians))
    threshold.y ← circleCentre.y + (radius ∗ sin(threshAngleRadians))
    return Coord threshold
end function
```

as per algorithm previously stated. The circumference of the circular path of the MSN is calculated based on the radius, with the coordinates of the start point of the MSN set from input as *firstSinkPos*. Based on the sink start point, the quartile of the circle the sink initially resides in is calculated as *firstSinkQuartile*. These quartiles can be described as North West, North East, South West or South East. The shortest distance from the node to the circular path of the MSN is calculated as *distToCircle* in Eq (4.1):

$$distToCircle = \left| \left(\sqrt{(node.x - centre.x)^2 + (node.y - centre.y)^2} \right) - radius \right| \tag{4.1}$$

Where *node.x, node.y* denote node coordinates; *centre.x, centre.y* denote circle path centre coordinates.

If the distance to the circular path is less than the node transmission distance, then the node is deemed to be significant. In that it shall be able to communicate directly with the MSN at some point. Coding is then performed relevant to only significant nodes. The coordinates of the closest point to the circular path from the node is calculated as *circlePoint* in Eq (4.2) and Eq (4.3):

$$point.x = centre.x + \left(radius \times \left(\frac{node.x - centre.x}{\sqrt{(node.x - centre.x)^2 + (node.y - centre.y)^2}} \right) \right) \tag{4.2}$$

$$point.y = centre.y + \left(radius \times \left(\frac{node.y - centre.y}{\sqrt{(node.x - centre.x)^2 + (node.y - centre.y)^2}} \right) \right) \tag{4.3}$$

Where *point.x, point.y* denote coordinates of circlePoint.

The quartile in which the *circlePoint* resides is calculated as *nodeQuartile*. The distance between *firstSinkPos* and *circlePoint* in a straight line is calculated as *distanceBetweenPoints*. The generic equation to calculate the distance between two points can be seen in Eq (4.4):

$$distance = \sqrt{(point2.x - point1.x)^2 + (point2.y - point1.y)^2} \tag{4.4}$$

The angle of the *circlePoint* is calculated as *angleOfNode*, between 0 and 360 degrees, with zero the farthest east point of the circle and default starting point for the sink, although a zero sink start point is not compulsory. The calculation is based initially on the *firstSinkQuartile* and *nodeQuartile* in order to ascertain the angle between sink start point and *circlePoint*.

The *thresholdAfter* boolean variable is set to true in order that the coordinates of the threshold after the *circlePoint* may be calculated. The coordinates are calculated as *sinkThresholdAfter* using the *establishThreshold* function, as detailed later. This is repeated with *thresholdAfter* set to false to calculate the coordinates of *sinkThresholdBefore*. Then the distance in a straight line between the two threshold coordinates, before and after, is calculated as *thresholdDistance* as per Eq (4.4).

The quartile in which *sinkThresholdBefore* is located is established as *beforeQuartile*, followed by the coordinates of the opposite point to the threshold as *thresholdOpposite*, for use later in determining the sink position in relation to the node. This concludes the initialisation.

The Establish Threshold function is passed the size of the radius of the circular path and whether whether the coordinates to be calculated are before or after the *circlePoint*. The threshold point is based upon a combination of node location and the point at which communication with the sink should no longer be possible, based on transmission distance.

The distance from the centre of the circle to the static node is calculated as *nodeDist*. The angle of the circle centre to furthest point of communication and the circle centre to the node is then calculated, using radius, *nodeDist* and *transDist* to form a triangle. This is illustrated in Figure 4.4 in relation to Node 15.

In order to avoid excessively large thresholds, a factor is determined by which how much this angle should be reduced based upon *distToCircle* divided by *transDist*. However, a factor check is required based upon the speed of the MSN. The faster the sink speed may be, the more the angle of the threshold may be reduced by. If the factor calculated is less than the factor check then the factor check value becomes the factor.

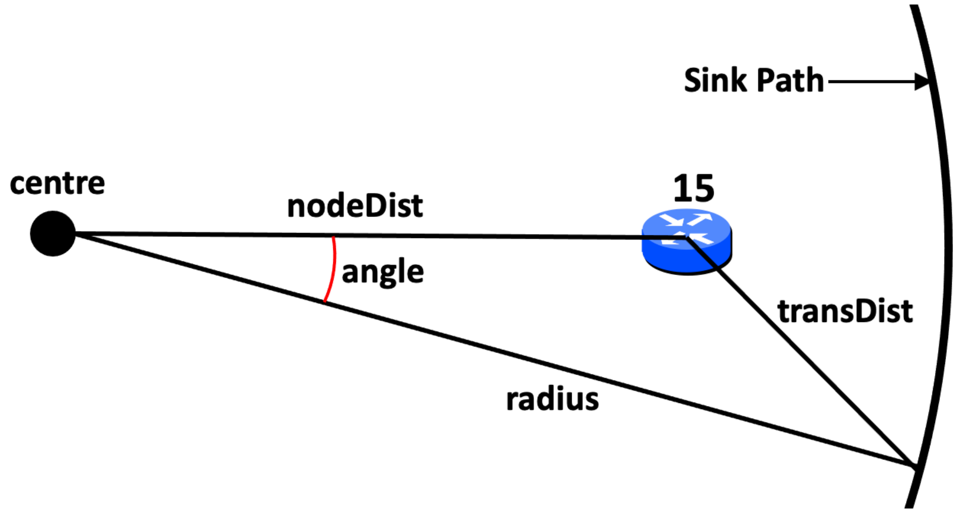


Figure 4.4: Illustration of Calculation of Initial Angle of Threshold - MADCAL

The value of the angle of the threshold is then multiplied by the factor in order to reduce it accordingly. If *sinkThresholdAfter* is being calculated then the angle calculated is added to *angleOfNode*, otherwise the angle is subtracted from *angleOfNode*. This results in *threshAngleDegrees*, which is then converted to radians. The function finishes by calculating the x and y coordinates of the threshold, returning the coordinate value *threshold*.

4.3.3 Initial Calculation of Threshold

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 transmission range of the node and the radius of the circle, an initial maximum threshold before and after the circlePoint can be calculated. This is demonstrated in Figure 4.5 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 and divided by the transmission distance, in order to create a factor by which the threshold angle shall be multiplied. Secondly, in order to now avoid an extremely small threshold for nodes closer to the path, a factor check is utilised. This lessens how much the threshold is reduced based upon the speed of the sink node, which is constant throughout each scenario. As the sink speed increases a smaller threshold is more efficient, given that the sink shall pass through this threshold more often. For example, if the sink node is travelling at 40mps it shall pass through the threshold of any significant node 40 times the same time it would take to pass just twice at 2mps. Giving each significant node many opportunities to communicate with the sink. However, at 2mps it would make sense for thresholds to be larger to give each significant node as much chance as possible to communicate with the sink.

Therefore, as a first approach, the factor check is utilised such that for speeds less than 10mps the threshold cannot be reduced by less than a factor of 0.5, for less than 20mps this factor reduces to 0.35, reducing again to no less than 0.25 for less than 40mps. If the speed is exactly 40mps no action is taken to use a factor check, allowing the threshold to be reduced based upon the initial calculation of node distance over

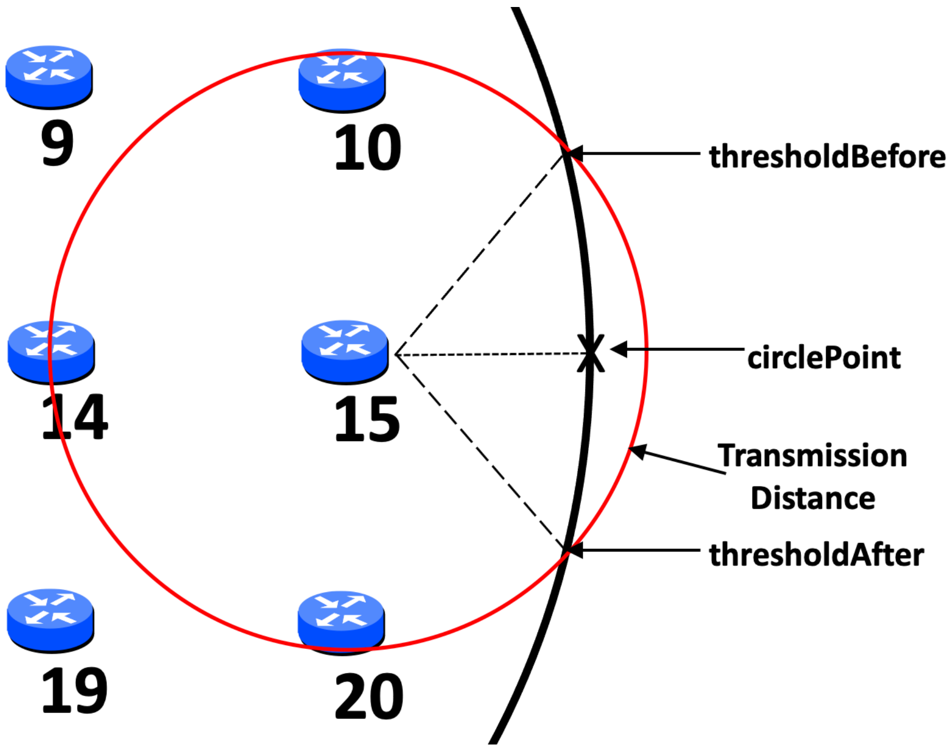


Figure 4.5: Illustration of Initial Threshold Calculation - MADCAL

transmission distance. 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.

4.3.4 Threshold Calculation with Factoring Employed

Using Node 15 again as an example, which has a distance of exactly 50m from the path of the MSN. In the example shown in Figure 4.5 the transmission range is 77.52m which would give an initial factor calculated as thus:

$$factor = \frac{distToCircle}{transDist} \tag{4.5}$$

Which in this case would result in:

$$\frac{50}{77.52} = 0.645$$

Assuming a speed of 2mps then speed is less than 10mps, therefore the factor check is set to 0.5. This would be compared against the result of the initial calculation which, if less than the factor check, would result in the factor check value, 0.5, being assigned as the value of the factor. Ensuring it can not be reduced by less than 0.5. However, in this case this is not necessary and the angle of the threshold would be reduced by a factor of 0.645 as shown in Figure 4.6.

4.3.5 Duty Cycling Adjusted with MADCAL

Algorithm 2 is designed to be inserted within the existing MAC code in order to establish the node wake-up time. This is based on calculating the current sink position by 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 GPS [86]. The sink position is then compared to the coordinates of the start of the threshold calculated in

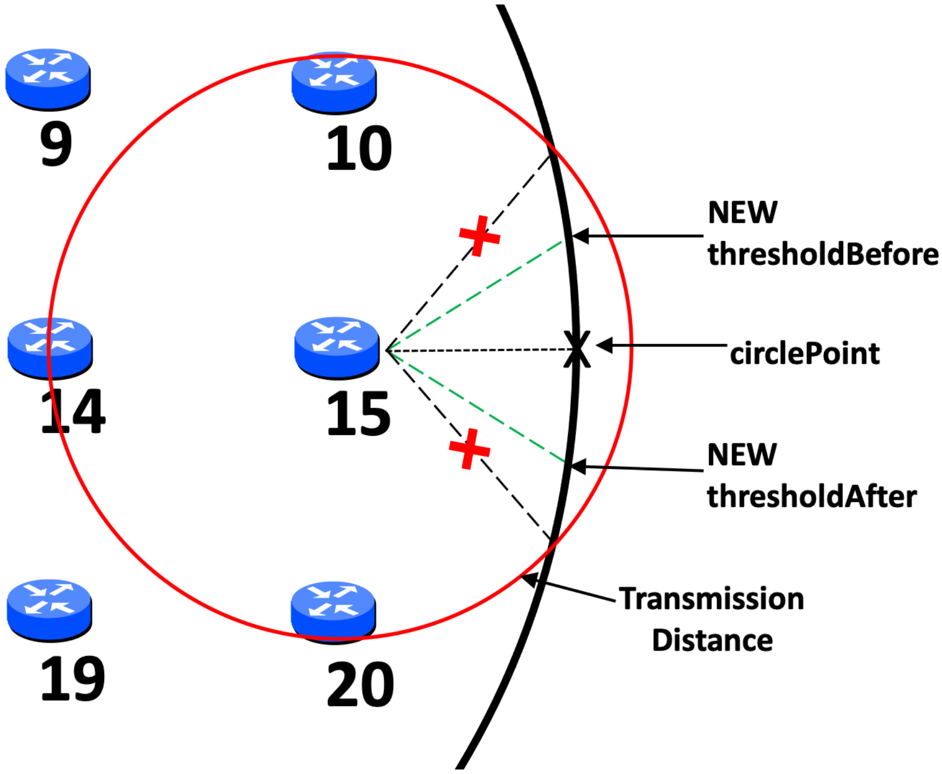


Figure 4.6: Adjusted Threshold Calculation - MADCAL

Algorithm 1, with node wake-up time determined by how long it will take the sink to reach the threshold, or else reverting to the original code using the check interval as input, in the event the sink is already within the threshold.

Algorithm 2: In the Sleep procedure, the default sleep interval is set as *checkInterval*. Now, only applying to significant nodes; the *thresholdTime* function is called to establish the time it will take for the sink to reach the threshold. If the threshold has been reached, the interval to wake-up and move to CCA reverts to the *checkInterval* otherwise, it is set to the time it will take for the sink to reach the threshold. If this is not a significant node the wake-up schedule reverts to the *checkInterval* as normal. This ends the Sleep procedure.

The *thresholdTime* function is designed to establish how long it will take the sink to reach the threshold. The current sink position is calculated as *sinkPos* based on

Algorithm 2 Threshold Interval

```
procedure SLEEP
    set checkInterval
    if significantNode then
        thresholdTime()
        if thresholdReached then
            interval ← checkInterval
        else
            interval ← timeToThreshold
        end if
    else
        interval ← checkInterval
    end if
    schedule CCA at time + interval
end procedure
function THRESHOLDTIME
    set sinkPos
    withinThreshold()
    if not thresholdreached then
        set arc to distance to sinkThresholdBefore
        set sinkQuartile
        timeToThreshold ← arc / sinkSpeed
    else
        timeToThreshold ← 0
    end if
    return timeToThreshold
end function
function WITHINTHRESHOLD
    if distance between sinkPos and thresholdAfter > thresholdDistance then
        thresholdReached ← false
    else if distance between sinkPos and thresholdBefore > thresholdDistance then
        thresholdReached ← false
    else
        thresholdReached ← true
    end if
    return thresholdReached
end function
```

simulation time and the initial sink start position as follows:

$$pos.x = start.x + radius \times \cos\left(\left(\frac{startAngle}{180}\right) \times PI\right) + \left(\frac{speed}{radius}\right) \times time \tag{4.6}$$

$$pos.y = start.y + radius \times \sin\left(\left(\frac{startAngle}{180}\right) \times PI\right) + \left(\frac{speed}{radius}\right) \times time \tag{4.7}$$

Where *pos.x, pos.y* denotes the calculated sinkPos coordinates; *start.x, start.y* denotes the sink start position; *startAngle* denotes the starting angle of the sink; *speed* denotes the sink speed; *time* denotes the current time the sink has been moving for.

The *withinThreshold* function is then called to establish if the sink is already within the communication threshold or not. If the threshold has not been reached yet, *arc* is set to the distance the sink must travel to reach the *sinkThresholdBefore* coordinates. Using the distance calculated between *sinkPos* and *sinkThresholdBefore* as per Equation 4.4, the length of the *arc* is calculated as follows:

$$arc = \left(\frac{180 - \left(\left(\arccos\left(\frac{\left(\frac{dist}{2}\right)}{radius} \right) \times \left(\frac{180}{PI}\right) \right) * 2 \right)}{360} \right) \times circumference \tag{4.8}$$

Where *dist* denotes the distance between *sinkPos* and *sinkThresholdBefore*; *circumference* denotes the circumference of the sink circular path.

The current quartile in which the sink resides is established and the time left to reach the threshold is calculated as the size of the arc in metres divided by the speed of the MSN in mps. However, if the threshold has been reached, then the time left to reach the threshold is set as zero. The *thresholdTime* function returns the *timeToThreshold*.

The *withinThreshold* function will establish the position of the sink node in relation to the static node communication threshold. The distance between the current sink position and the *thresholdAfter* coordinates is established as per Equation 4.4. If this is greater than the size of the entire threshold then the threshold has not been reached and *thresholdReached* is set to false. However, if this is not true and the distance between the current sink position and the *thresholdAfter* coordinates is less than than the size of the entire threshold, now establish if the sink is before or after the threshold. If the distance between the sink position and the *thresholdBefore* coordinates is greater than the *thresholdDistance* then the sink must be beyond the threshold and therefore *thresholdReached* is set to false. If the sink is also within *thresholdDistance* of *thresholdBefore* however, this means the sink is within the threshold. Therefore *thresholdReached* is set to true. With this, *thresholdReached* is returned and the *withinThreshold* function ends.

4.4 Evaluation and Results

4.4.1 Simulation Environment

Work has been conducted on the OMNeT++ [77] framework. Though not a simulator itself, OMNeT++ provides a platform on which simulations can be built, using compatible simulation software. This allows the user to combine different factors and provides fine-grained detail in the simulation. Each layer can be controlled within each node, with the ability to define network and MAC layer protocols and control all aspects of the physical layer. Within this study, the MiXiM [87] simulator is utilised to build the network environment including the location of nodes and the size of the actual area within which they are located. This is a simulator specifically built for use with WSNs with MAC protocols. However, as this study requires the sink node to be the final destination and, therefore, use of the network layer, a second simulator is used in combination. The inetmanet [88] simulator is used for all other factors, including physical, MAC, network and transport layer parameters. This is also beneficial in the use of mobility and energy models. The sink node and static nodes are defined separately in order that a mobility model may be applied to the sink and that all functionality to be detailed in this and subsequent chapters be coded within the static node model.

4.4.2 Energy Model

The energy module is the commonly used InetSimpleBattery module, found within inetmanet [88]. This module adds little to overhead in terms of computation with a lightweight estimation of energy consumption. As such, this is utilised by the physical layer to receive energy level values in order to facilitate operation of the wireless adaptor [89, 90, 91, 92]. In reference to the simulation parameters in Table 4.1, the simulation time is calculated to ensure an exact number of circuits of the network by the MSN. Hence, with the speed at 2mps (meters per second) the sink shall complete exactly 2 circuits of the network, for 10mps, 10 circuits and so on.

Table 4.1: Simulation Parameters - MADCAL

Test Parameters	Values
Number of Static Nodes	25
Playground Size	$x = 500m$ $y = 500m$
Circle Radius	$150m$
Sink Start Position	$x = 400m$, $y = 250m$
Sink Node Speed (metres per second)	$2mps$, $10mps$, $20mps$, $40mps$
Simulation Time	$942.47779607694s$
Transmission Distance * 4	$77.52m$, $69.13m$, 62.02, $55.94m$
Number of Runs	5
Path-loss Alpha * 4	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	$4dB$
Battery Capacity	$59400mWs$

Transmission distance is calculated as thus [93]:

$$transmissionDistance = \left(\frac{\left(\frac{SoL}{Freq}\right)^2 \times Power}{16 \times PI^2 \times 10^{\frac{SAT}{10}}} \right)^{\frac{1.0}{Alpha}} \tag{4.9}$$

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 lessen the risk of network failure due to node overload.

All parameters are consistent across all simulation runs and generally remain at the default values for the simulation implementation. However, the speed of the sink node and the transmission distance of the nodes are the significant metrics in this study and are altered accordingly. 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 transmission distance, which decreases as the alpha value increases.

4.4.3 Test Scenarios and Results

Each scenario utilises one of four different transmission ranges, consistent across all nodes and the sink. This demonstrates the effectiveness of the MADCAL algorithm as transmission ranges begin generously and are then contracted. To the point where they barely cover one-hop between nodes. Within each scenario four tests are performed, with the MSN speed altered between 2mps, 10mps, 20mps and 40mps for each. Therefore results can be compared between when the sink will move very slowly, only encountering individual nodes a small amount of times, to when a high sink speed means that many passes of each node are possible in the same simulation time.

Results are 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. Tests are conducted using both topologies as illustrated in Figures 3.1 and 3.2. The most significant difference across the two topologies is in the assignment of significant nodes. Node location is more stable and, although transmission range is altered across tests, the significant nodes do not change and remain as nodes 1, 2, 3, 4, 5, 6, 10, 11, 15, 16, 20, 21, 22, 23, 24 and 25. However, in the case of the random topology, as transmission ranges are altered, as are the significant nodes. This is covered in greater detail later.

4.4.4 Grid Network Formation

4.4.4.1 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 3.1. What was found was unless there is a large transmission distance which can encompass more than one node, this one node uses up most energy. However, when there is an overlap of transmission 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 is in danger of becoming redundant. Even in the event that neighbouring nodes can then take on the role of hotspot when a large transmission distance is in use, this may increase network lifetime but not avoid the ultimate conclusion, that network failure is the likely eventuality.

4.4.4.2 Results - Average Energy Consumption

Figures 4.7-4.10 show the average energy consumption across *significant nodes*. This is seen as important as these now take on the role of hotspot, therefore reducing energy consumption in these nodes is beneficial both in terms of overall network performance and network lifetime. The comparisons shown are between the evaluation results, where the standard duty cycling with CCA and check interval is in use, and results where the MADCAL algorithm is applied, such that a dynamic threshold is created within significant nodes for communication with the MSN.

Figure 4.7 illustrates significant energy saving when the MADCAL algorithm is in use. The larger transmission range in use here would normally result in considerable overlap of communication between significant nodes causing competition for communication with the MSN. This, subsequently, results in wasted energy consumption, with some nodes awake but unable to communicate with the sink. However, with a communication threshold established by MADCAL, although overlaps of threshold are still possible depending on node position, nodes are less likely to seek channel access at the same time. Hence there is less extraneous energy consumption.

In Figure 4.8 transmission range is reduced to 69.13m. Results remain a significant improvement, however, we can now observe how as transmission range reduces it becomes more difficult to improve energy consumption.

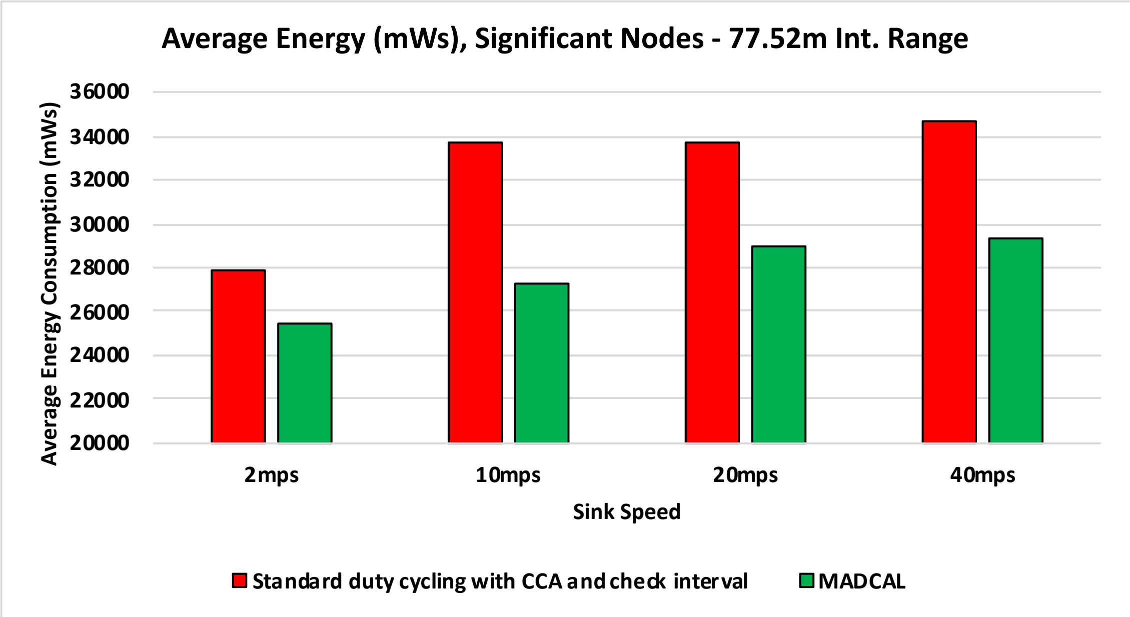


Figure 4.7: Average Energy Consumption (mWs), Grid Topology, Significant Nodes. Transmission range 77.52m. MADCAL

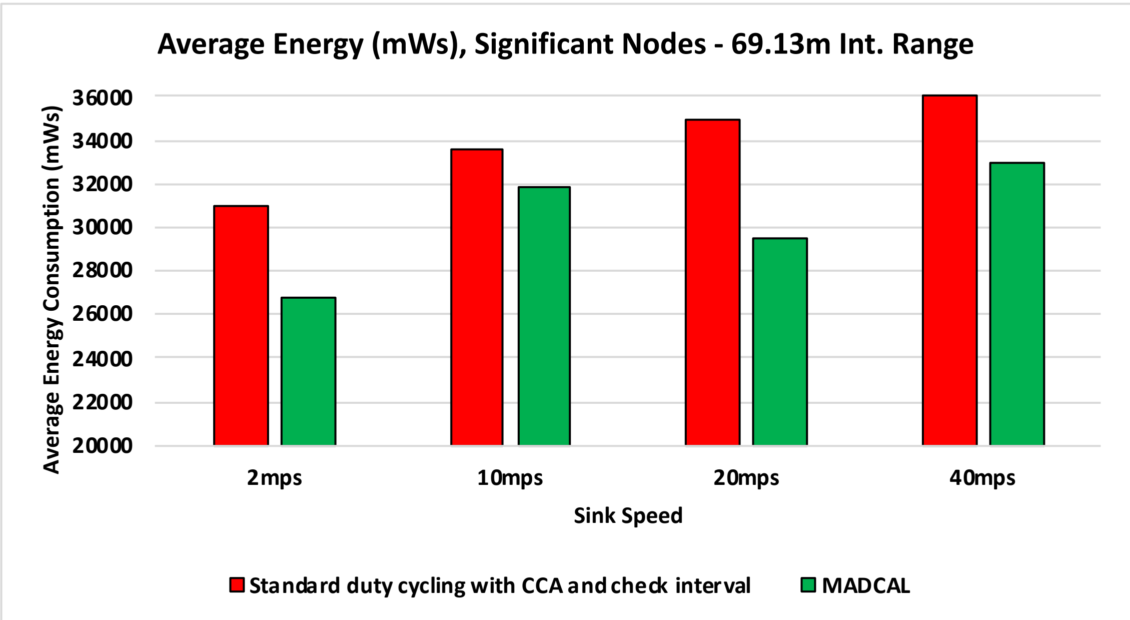


Figure 4.8: Average Energy Consumption (mWs), Grid Topology, Significant Nodes. Transmission range 69.13m. MADCAL

There is little difference between Figure 4.9 from Figure 4.8 with benefits still to be seen in energy consumption when MADCAL is in use. One main observation is that it can be seen that it is easier to save energy when the sink is moving more slowly. In this case larger thresholds are calculated, but with the sink moving at only 2mps, there is more time to put nodes to sleep before the sink node reaches. The counter to this is that it could result in increased delay of frame delivery.

Figure 4.10 is significant in that the transmission 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 improved energy consumption, which again is most in evidence when the sink node is moving slowly.

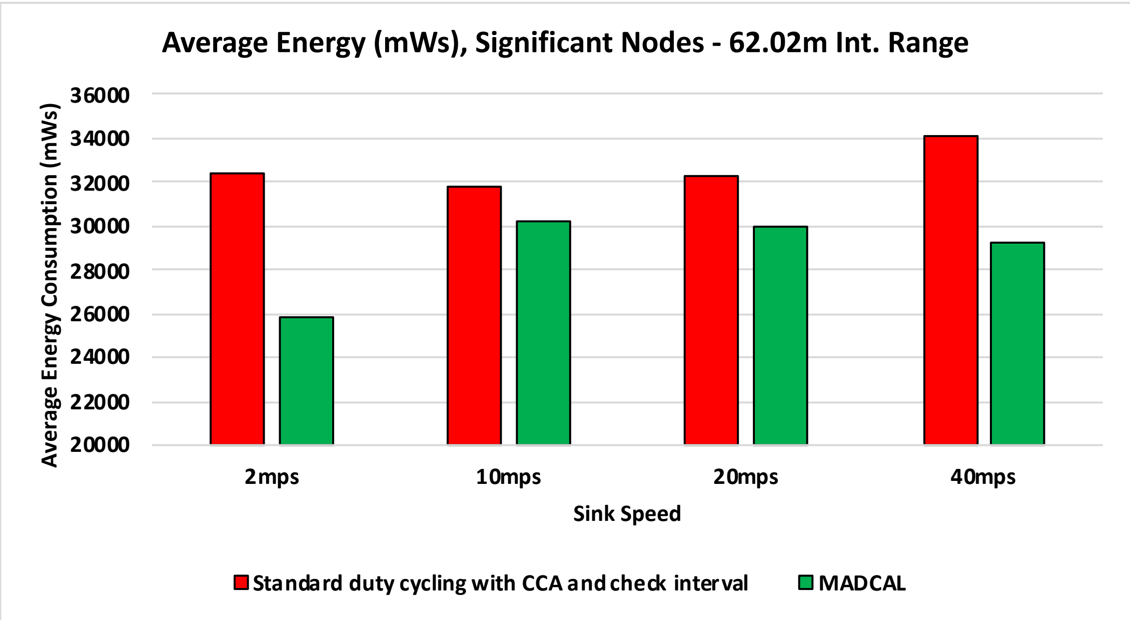


Figure 4.9: Average Energy Consumption (mWs), Grid Topology, Significant Nodes. Transmission range 62.02m. MADCAL

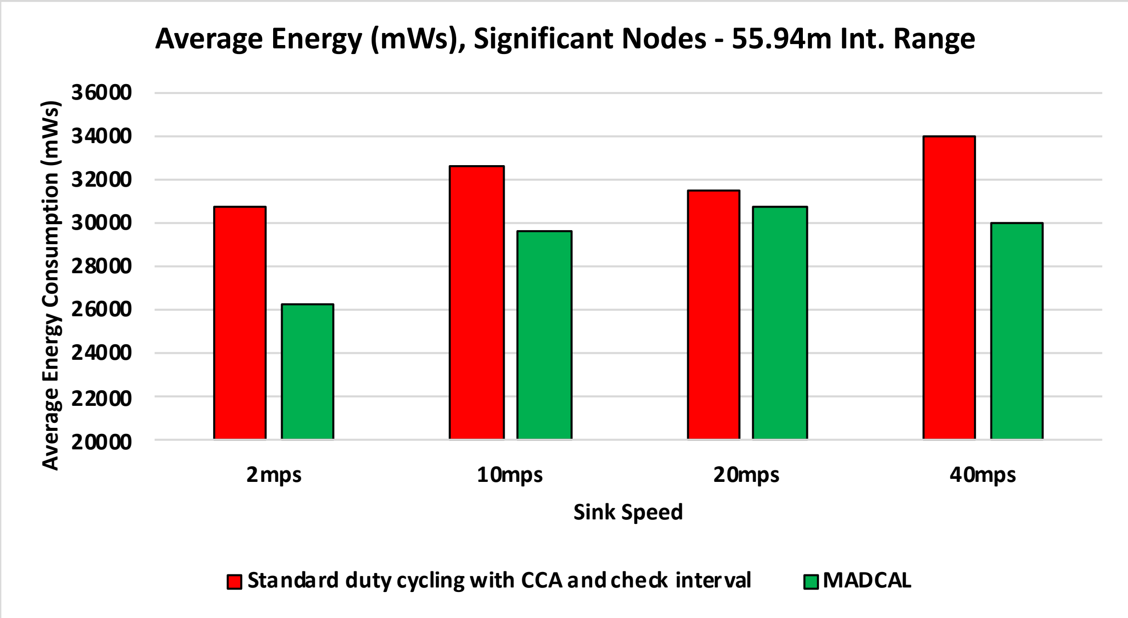


Figure 4.10: Average Energy Consumption (mWs), Grid Topology, Significant Nodes. Transmission range 55.94m. MADCAL

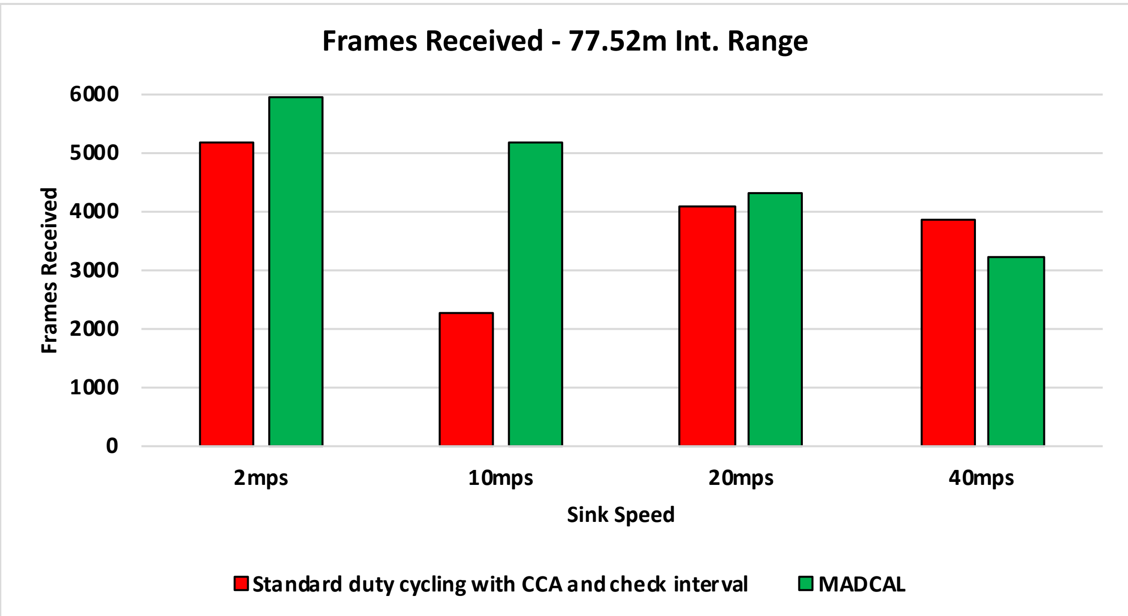


Figure 4.11: Sink Frame Reception, Grid Topology. Transmission range 77.52m. MADCAL

4.4.4.3 Results - MAC Layer Frame Delivery

Figures 4.11-4.14 illustrate the number of MAC layer frames received by the sink during each simulation scenario. This is an important comparison between when MADCAL is in use and when not, as improved energy consumption would not be acceptable if detrimental to the network's ability to function in terms of delivering frames. In observing Figure 4.11 it can be seen that much like energy consumption, 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. But frame reception which is similar, even if slightly lower, could be seen as acceptable in the event that energy consumption is significantly improved.

Figure 4.12 again shows the benefit of the sink moving more slowly, with frame reception the same or slightly worse for the faster speeds.

In Figure 4.13, again the benefits are greater when the sink moves more slowly. However, as the transmission range reduces, delivery to the sink node now becomes

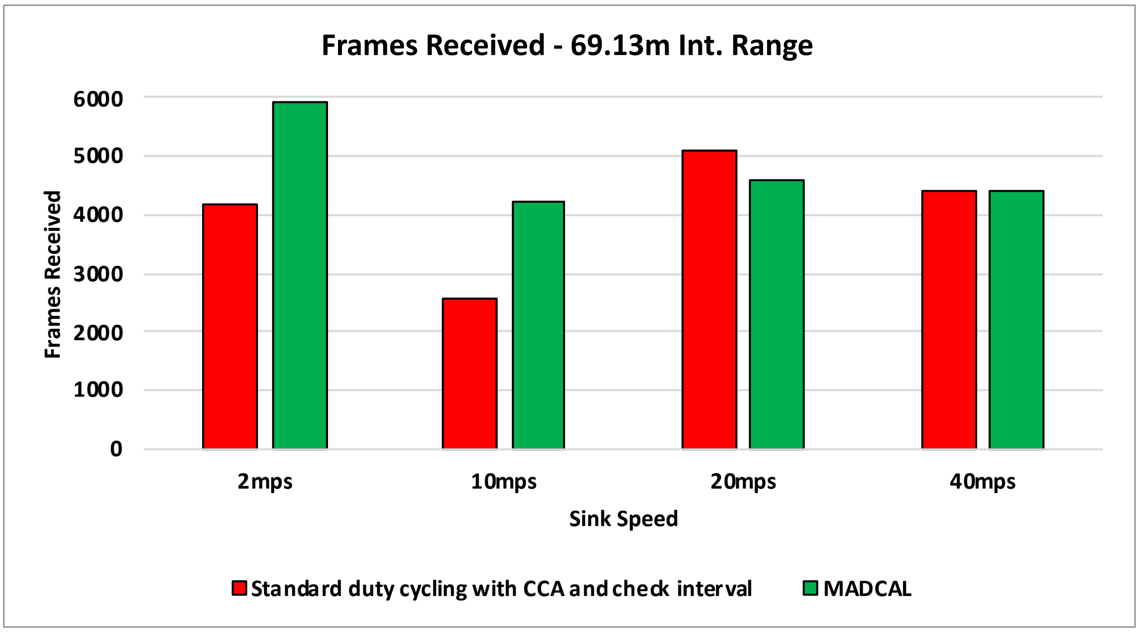


Figure 4.12: Sink Frame Reception, Grid Topology. Transmission range 69.13m. MADCAL

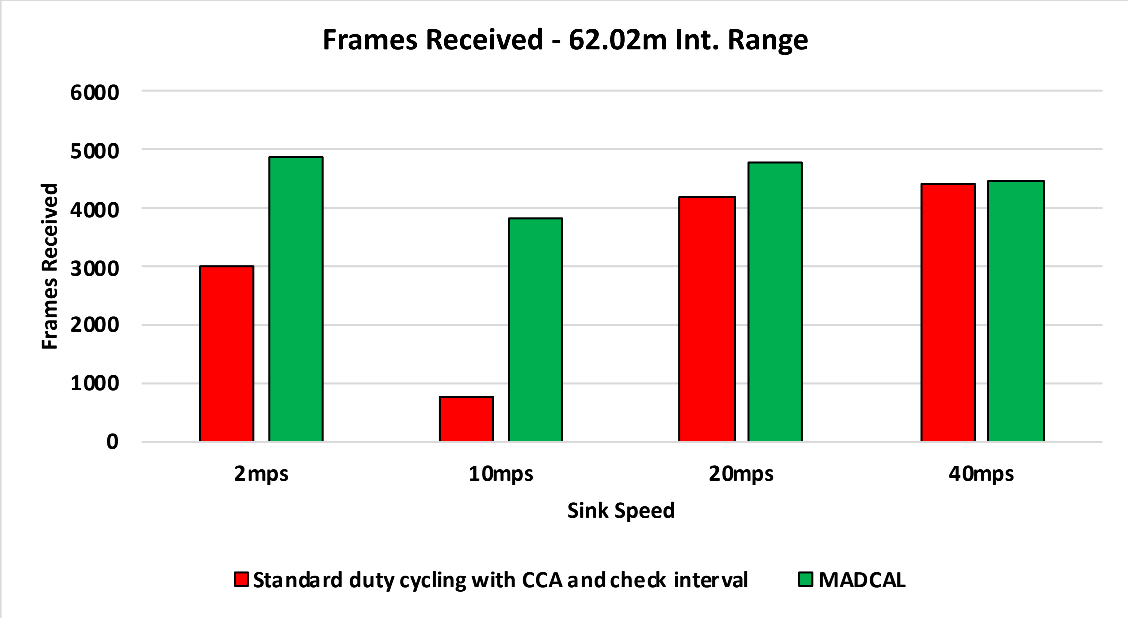


Figure 4.13: Sink Frame Reception, Grid Topology. Transmission range 62.02m. MADCAL

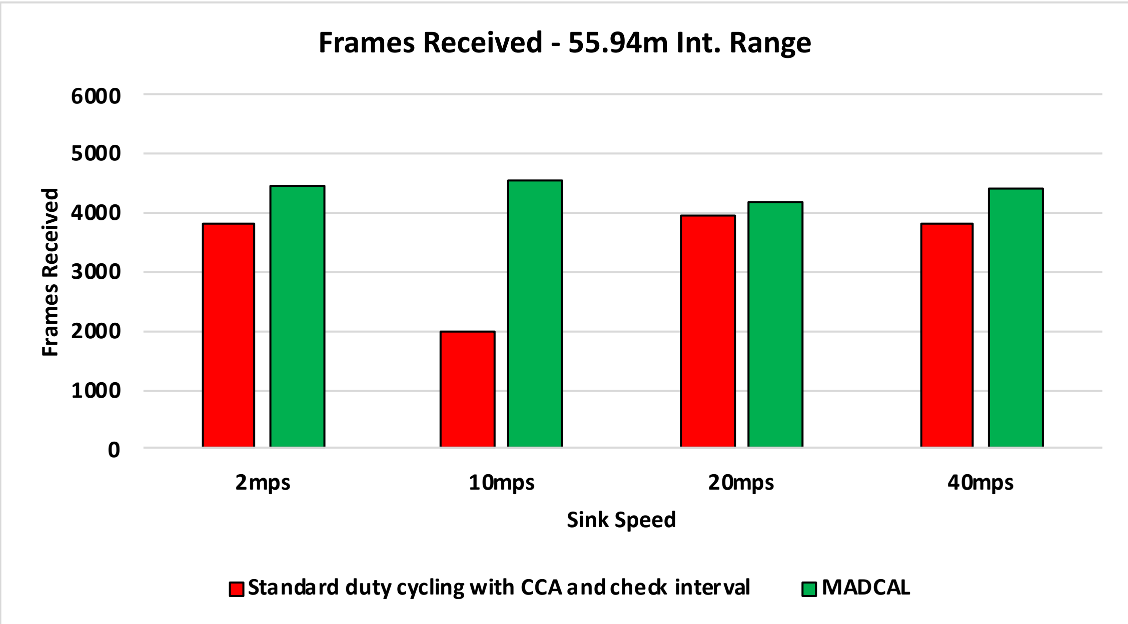


Figure 4.14: Sink Frame Reception, Grid Topology. Transmission range 55.94m. MADCAL

easier at faster speeds, with MADCAL showing a slight improvement. Therefore, despite the smaller transmission range, the subsequent reduction in overlap of communication, both with and without MADCAL, enables more efficient frame reception. Therefore, MADCAL can be seen as negating sink speed to a certain degree when considering frame delivery to the MSN.

In Figure 4.14 benefits once more are greater in lower speeds, but improvements can still be observed at faster speeds when MADCAL is in use. As in Figure 4.13, it can be observed that with MADCAL in use, frame delivery is now more consistent across all speeds.

4.4.4.4 Summary

A significant improvement in energy consumption can be observed when MADCAL is implemented, especially so when sink mobility is slow. This would result in an increase in network lifetime, with the nodes closest to the sink path consuming less energy and therefore, living longer before battery power runs out. While improvements are significant at lower speeds, once the MSN speed increases improvements are less clear in terms of frame delivery. However, over all tests there are only two occurrences of frame delivery going down and not significantly. Therefore, any slight degradation can be offset by the benefit 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 that occur in faster speeds without the use of MADCAL. 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.

4.4.5 Random Network Formation

4.4.5.1 Significant Node Variance

In the previous series of results, utilising the grid topology to be found in Figure 3.1, no matter the transmission range of the nodes, the significant nodes were unchanged throughout. With the random topology, which can be seen in Figure 3.2, this is not the case. As illustrated in Figure 4.2, as the transmission range is reduced as in the previous tests, as does the number of significant nodes reduce. This a result of some now being out of range of the circular path of the MSN as the transmission range decreases. Results are still given in reference to significant nodes, however, which nodes are of significance

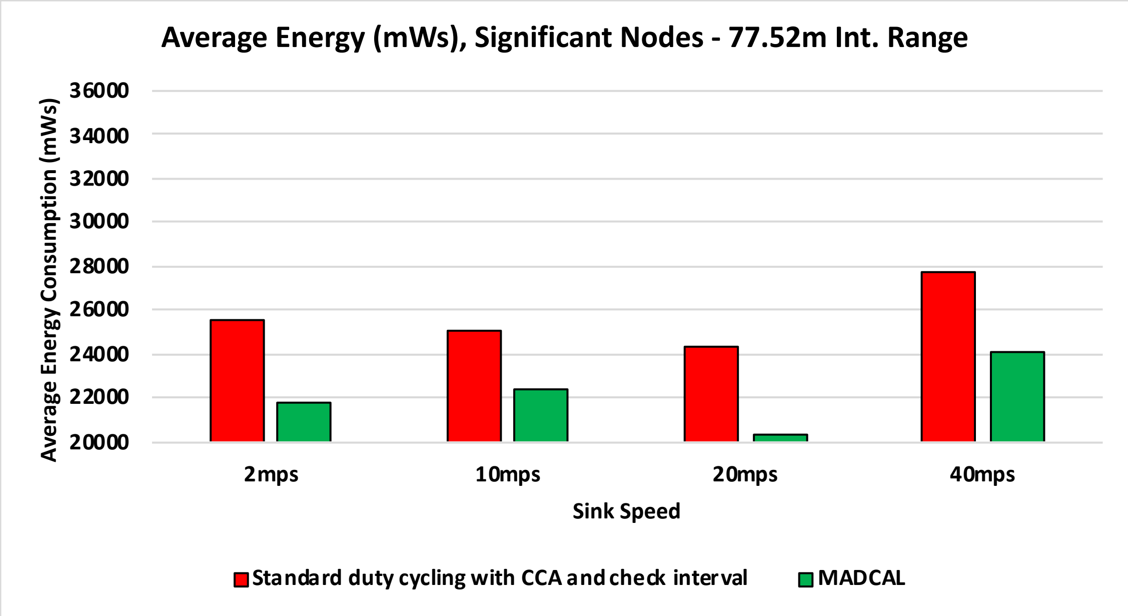


Figure 4.15: Average Energy Consumption (mWs), Random Topology, Significant Nodes. Transmission range 77.52m. MADCAL

for each scenario can be seen in Table 4.2.

Table 4.2: Significant Nodes - Random Topology

Transmission range	Significant Nodes
$77.52m$	1, 2, 3, 4, 5, 6, 9, 11, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25
$69.13m$	1, 2, 3, 4, 5, 11, 15, 16, 17, 19, 20, 21, 24, 25
$62.02m$	1, 2, 3, 4, 5, 11, 15, 16, 17, 19, 21, 24, 25
$55.94m$	1, 2, 3, 4, 5, 11, 15, 16, 17, 21, 24, 25

4.4.5.2 Results - Average Energy Consumption

Figures 4.15-4.18 show the average energy consumption across *significant nodes.* As with the grid topology, these nodes now take on the role of hotspot. The comparisons shown are between the evaluation results, where the standard duty cycling with CCA and check interval is in use, and results where the MADCAL algorithm is applied, such that a dynamic threshold is created within significant nodes for communication with the MSN.

Figure 4.15 illustrates reduced energy consumption when the MADCAL algorithm is in use. However, in comparison to the grid topology it can be seen that energy levels are lower even before MADCAL is applied. This is likely due to the large gaps between groups of significant nodes in this more random topology. Allowing for less overlap of communication over the network as a whole. This is then improved upon still by the application of MADCAL. Resultantly, it can be observed that MADCAL is again of benefit, even in a less controlled environment such as this.

In Figure 4.16 Transmission range is reduced to 69.13m. The results shown are now a significant improvement as the transmission range decreases. However, it can be seen also that energy consumption has risen considerably in comparison to Figure 14. In this test there are now 3 fewer significant nodes, therefore as each takes on a greater role, more energy is consumed overall as a result.

In Figure 4.17 there is again benefit shown when using the MADCAL algorithm as transmission range decreases again. Even more so than when a controlled, grid topology was in use.

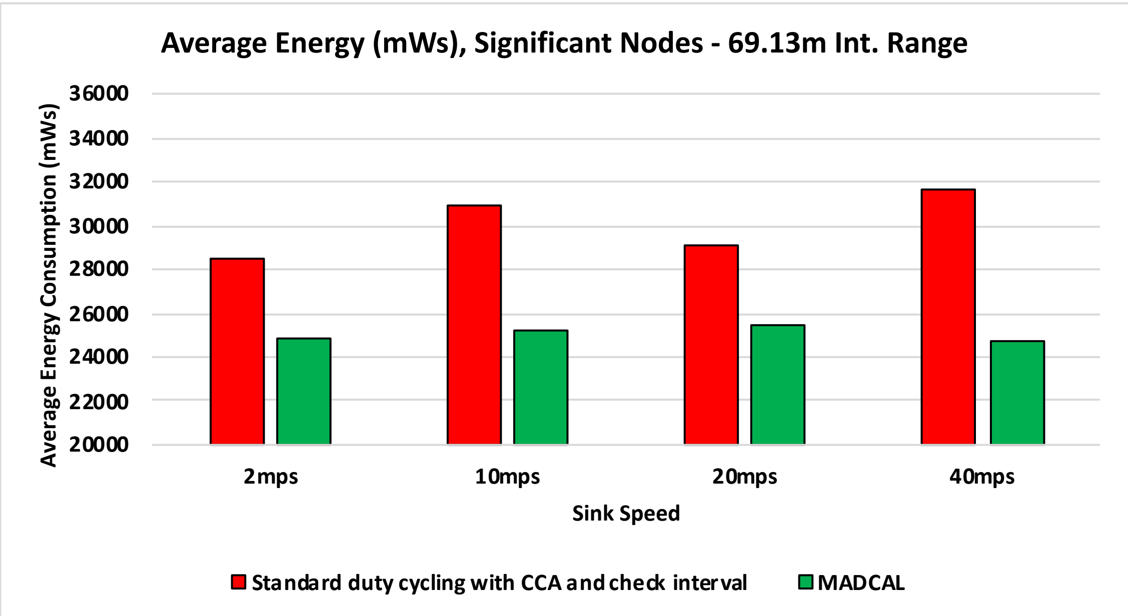


Figure 4.16: Average Energy Consumption (mWs), Random Topology, Significant Nodes. Transmission range 69.13m. MADCAL

Figure 4.18 is significant in that the transmission range is now strained to a great extent. However, benefits can still be seen with MADCAL in use. As with the grid topology, this is less evident when the sink is moving faster. However, unlike in the equivalent test, shown in Figure 4.10, the benefits in energy consumption are greater overall now.

4.4.5.3 Results - MAC Layer Frame Delivery

Figures 4.19-4.22 illustrate the number of MAC layer frames received by the sink during each simulation scenario. Again this is an important comparison between when MADCAL is in use and when not, this time to see if MADCAL improves or adversely affects frame reception now a random topology is in use.

Of interest in Figure 4.19 is that it can be observed that MADCAL brings frame reception into line across all different speeds in this scenario. 10mps was previously significantly lower than the other speeds in this regard, but not so when MADCAL is in use. This also occurred when the topology was more controlled, but only once

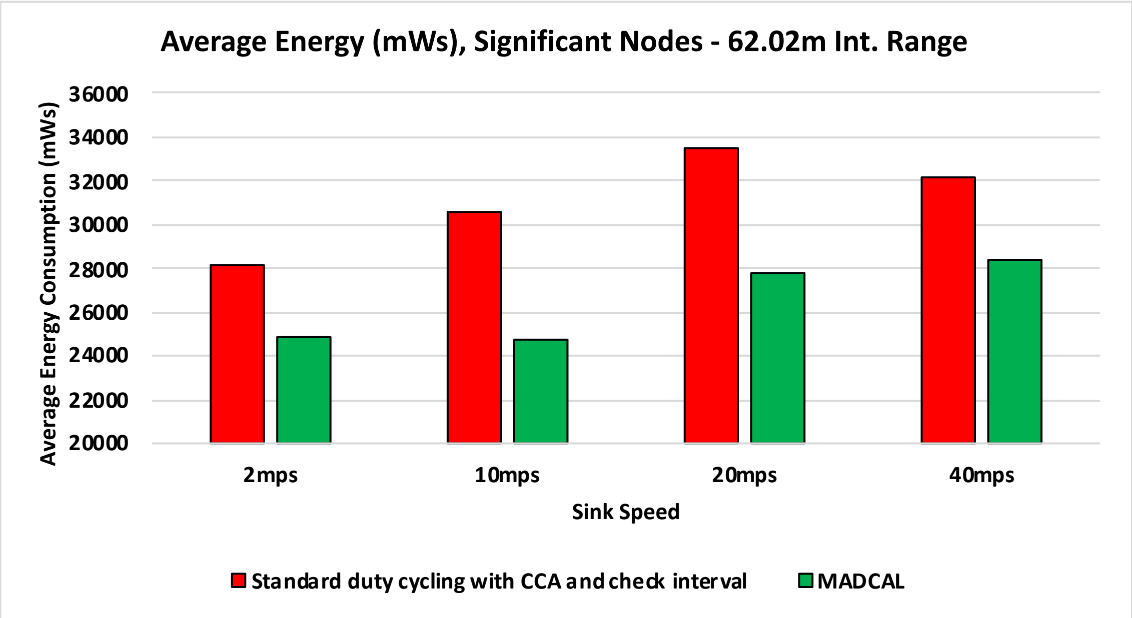


Figure 4.17: Average Energy Consumption (mWs), Random Topology, Significant Nodes. Transmission range 62.02m. MADCAL

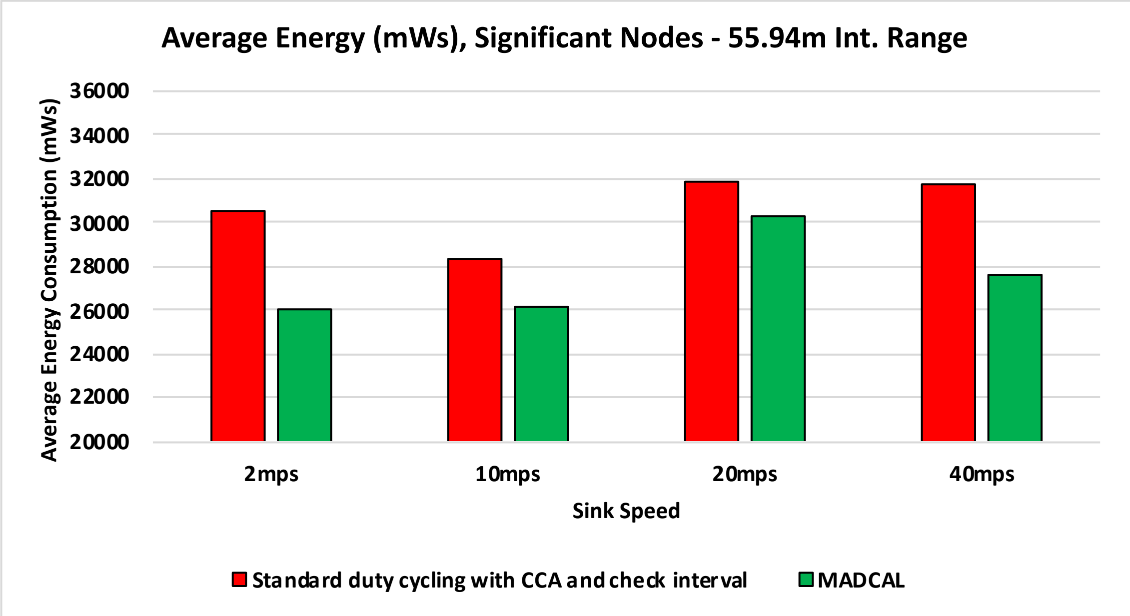


Figure 4.18: Average Energy Consumption (mWs), Random Topology, Significant Nodes. Transmission range 55.94m. MADCAL

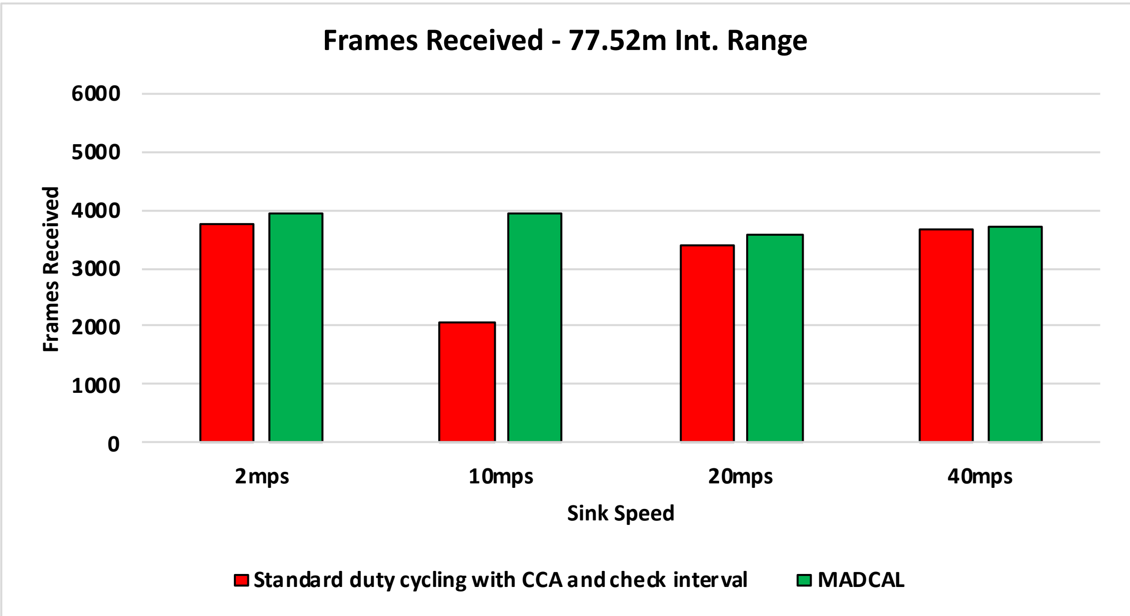


Figure 4.19: Sink Frame Reception, Random Topology. Transmission range 77.52m. MADCAL

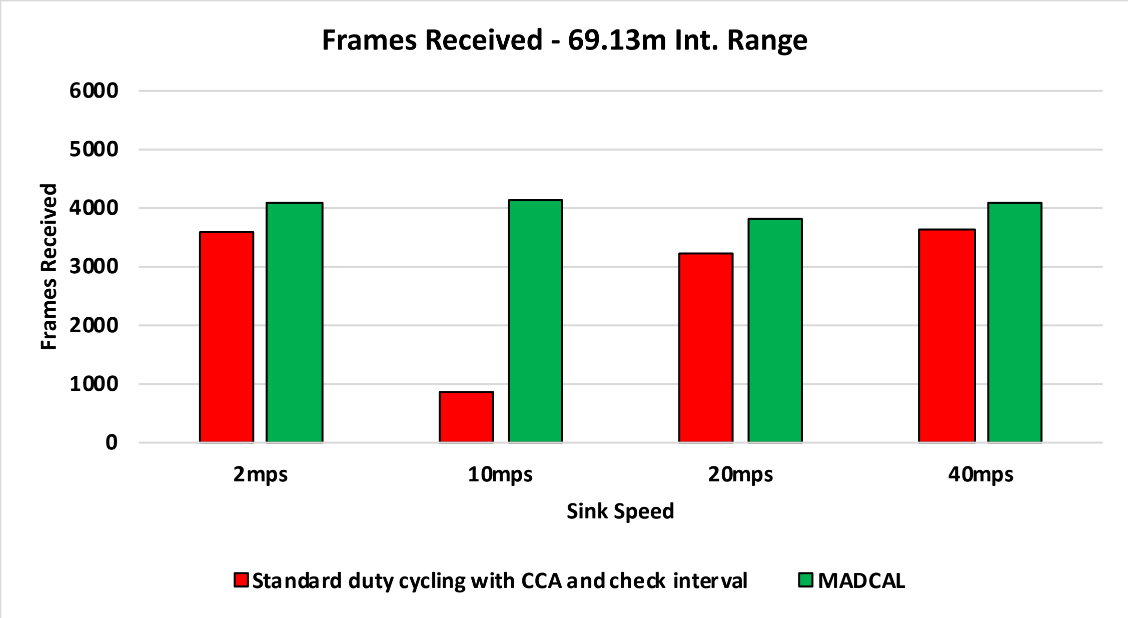


Figure 4.20: Sink Frame Reception, Random Topology. Transmission range 69.13m. MADCAL

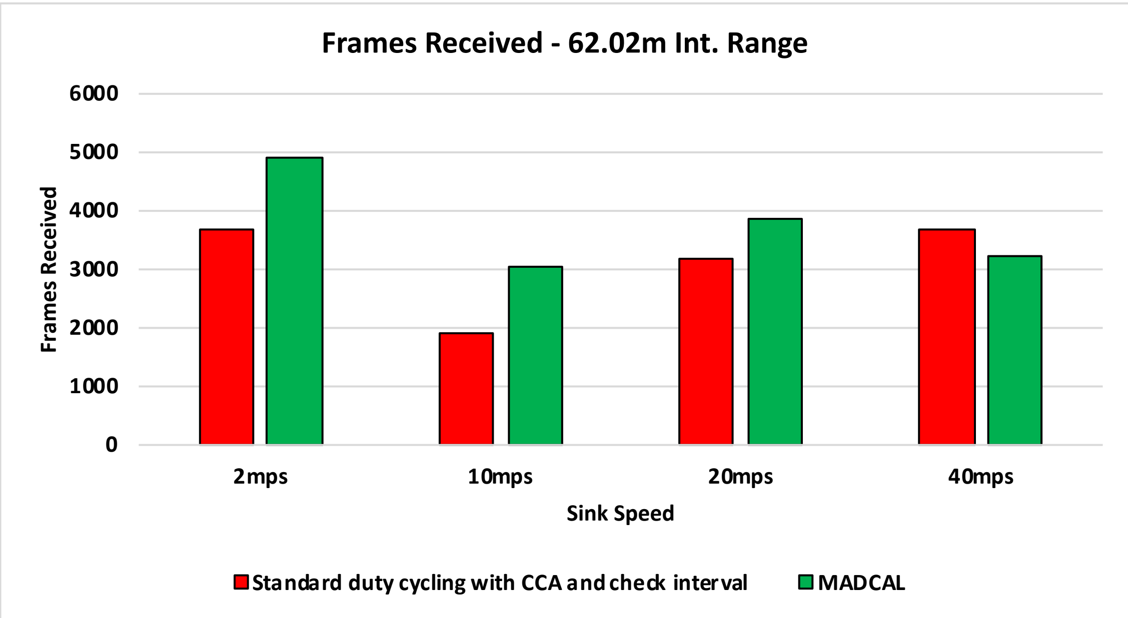


Figure 4.21: Sink Frame Reception, Random Topology. Transmission range 62.02m. MADCAL

transmission ranges became much smaller.

As with Figure 4.19, Figure 4.20 again shows that MADCAL brings the level of frame delivery to similar levels for all speeds. This time though, benefits are even more noticeable.

Now with transmission range decreasing, in Figure 4.21 it can be seen that frame

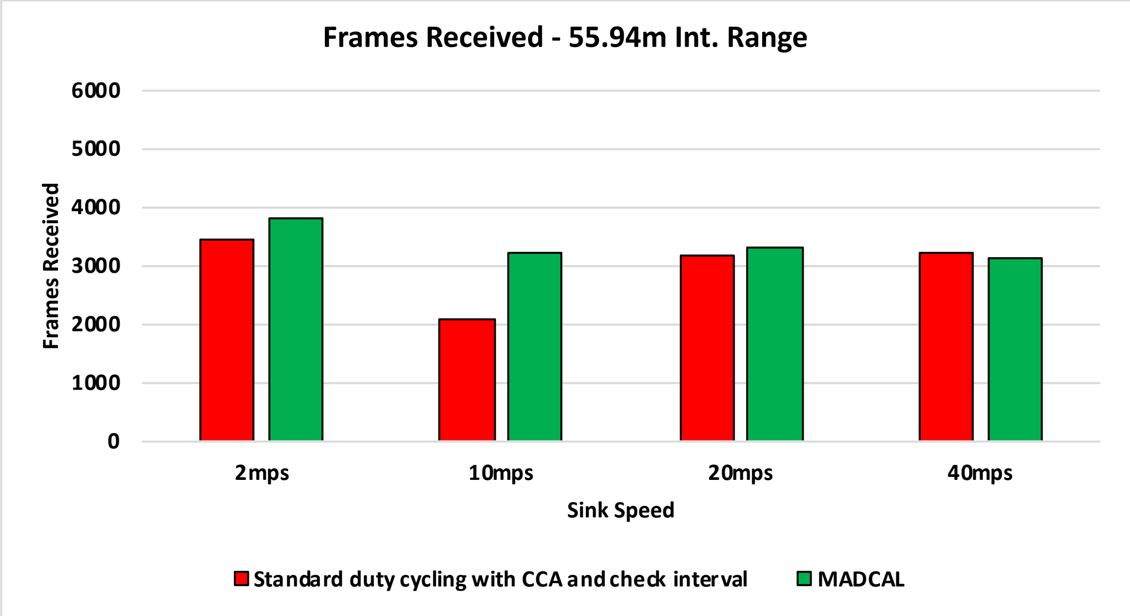


Figure 4.22: Sink Frame Reception, Random Topology. Transmission range 55.94m. MADCAL

reception decreases. MADCAL, however, shows benefit in all but one scenario, where speed is at 40mps. However, the loss is not so significant as to cause concern when the benefit in energy consumption is taken into consideration.

In Figure 4.22 it can again be seen that with MADCAL in use, frame delivery becomes consistent across all speeds. By now though, with the much smaller transmission range, frame reception overall is lower than when the grid topology was in use. This is due to large gaps between nodes that are unable to be bridged when the transmission distance is so low. A reminder of the importance of topology as a factor in the building of any network.

4.4.5.4 Summary

As with the more controlled grid topology, a significant improvement in energy consumption is again evident when MADCAL is implemented. However, in this more random scenario there is improvement across all speeds, even as the transmission range reduces. Thus, compared to the controlled grid topology, the improvements in network lifetime are greater, especially so at faster sink speeds. When considering frame delivery, there are no significant improvements when using MADCAL other than at

10mps. However, frame delivery improves slightly or stays roughly the same and is generally consistent across all speeds once MADCAL is in use. As such MADCAL can be seen as improving energy consumption considerably, whilst generally improving frame reception to the point where sink speed is mostly negated.

4.5 Conclusion

In this chapter MADCAL is proposed, a dynamic and lightweight duty cycling algorithm for use in WSNs where MSNs are utilised. Nodes which are within one-hop of the MSN path are identified as *significant nodes*, to all intent and purpose, replacing the role of nodes which previously would become *hotspots* where the sink node is static. Results demonstrate to what extent the mobility pattern of a MSN in a WSN may be exploited at the MAC layer to influence the performance of static nodes. With network topology both controlled and random, energy consumption is shown to be reduced amongst significant nodes when they are aware of their own location as well as the MSN start point and speed; this as well as improving frame delivery to the sink. These results are achieved without additional network overhead or the energy consuming exchange of messages. Therefore, MADCAL provides a crucial first step in the area of utilising predictable mobility patterns. However, issues remain to be investigated, such as whether energy consumption may be further reduced and whether individual significant nodes may be taking on too much responsibility, in effect repeating the issue of the *hotspot*. Solutions to these issues are proposed in Chapters 5 and 6 of this thesis.

Fifth Chapter

Mobility Aware Duty Cycling and Dynamic Preambling Algorithm (MADCaDPAL)

5.1 Introduction

MADCAL is shown to have benefit when considering the average energy consumption of all significant nodes. However, on examining results further it can be observed that while average energy consumption is improved there remain energy consumption *spikes*, where certain significant nodes consume far more energy than others. As such, despite the benefits of MADCAL it can be claimed that, to a certain degree, the hotspot issue remains as some nodes will still run out of energy faster than others. In this chapter the MADCaDPAL algorithm is proposed to utilise the communication threshold between static node and MSN to not only influence the move from SLEEP to CCA, but also to influence CCA itself and the sending of preambles. Such that, these should cease once the MSN reaches the end of the threshold. Hence, ensuring that communication between sink and static node is completely *closed*. As such, this approach further answers the first research question as well as the second and third questions:

- To what extent can the mobility pattern of a MSN in a WSN be exploited at the MAC layer to influence the performance of static nodes, in order to increase efficiency in terms of network performance and energy consumption?

- To what extent can a threshold built in a static node be utilised for communication with a MSN in order that said communication only occurs within the threshold?

- To what extent can network performance and energy consumption be improved by this approach?

This approach shall demonstrate a now considerable reduction in energy consumption in significant nodes, as well as a significant increase in frame delivery to the sink. This as well as reducing energy spikes in individual nodes and subsequently improving network lifetime overall.

5.2 Mobility Aware Approach to Duty Cycling and Preambles

A communication threshold between a static *significant* node and a MSN has been shown to have benefit in terms of energy consumption. Thus far, however, this has only been utilised to intercept the move from the SLEEP process to CCA at the MAC layer. Resultantly, in examining this procedure, it could be observed that once the CCA process started, this would continue between the static node and sink beyond the end of the communication threshold. The result being that some nodes could then monopolise communication with the sink, whilst others must wait for the channel to clear, even if the sink is now within their threshold. This results in the aforementioned spikes of energy. Hence, in this chapter the aim is to *close* the communication threshold, in order that as soon as the sink passes the end of the threshold, communication is no longer possible.

Consequently, whereas the MADCAL algorithm may still be utilised to influence the SLEEP function at the MAC layer, only moving to CCA when the sink is within the node's threshold, further factors are determined to be required to be taken into account.

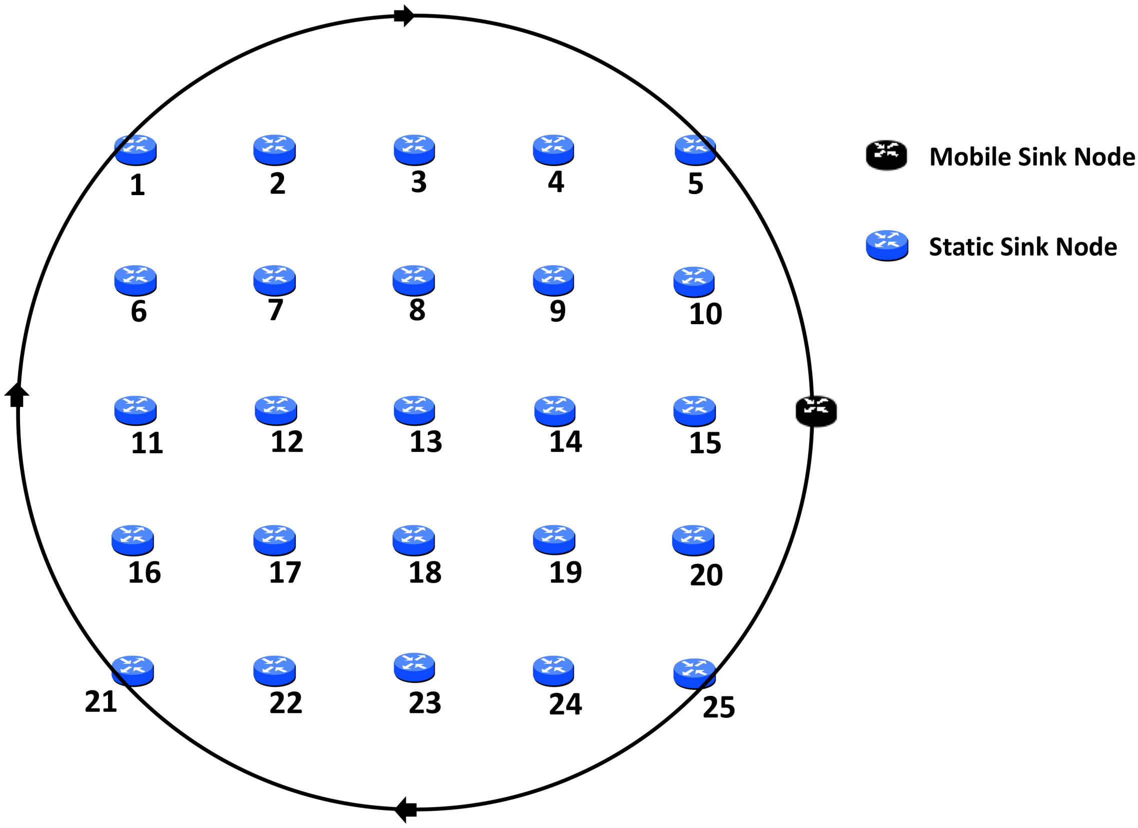


Figure 5.1: Network Topology - Grid

The position of the MSN in relation to the threshold is now used to influence the CCA process as well as the sending of preambles. With these changes, the aim is to further reduce energy consumption amongst significant nodes, while also improving frame delivery to the sink node. However, a more even distribution of energy consumption across significant nodes is of equal importance.

The Grid and random network topologies as shown in Figure 5.1 and Figure 5.2 are again utilised. This again results in the same significant nodes each time for the grid topology, as can be seen in Figure 5.3. The random topology again has significant nodes which are dependant on the transmission range of the nodes, as can be seen in Figure 5.4.

The lightweight CSMA/CA MAC implementation in use can be seen in Figure 5.5, with the original MADCAL algorithm shown to create the communication threshold, and to intercept the SLEEP function, with MADCaDPAL highlighted to intercept CCA

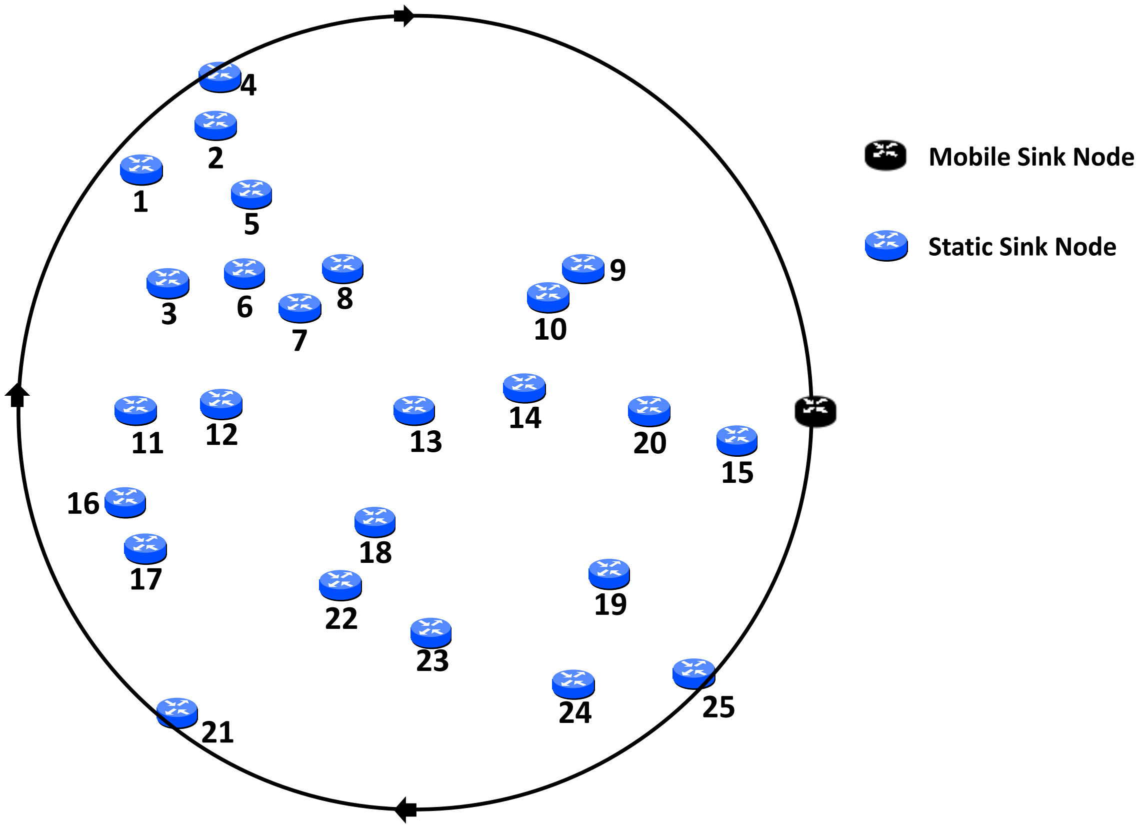


Figure 5.2: Network Topology - Random

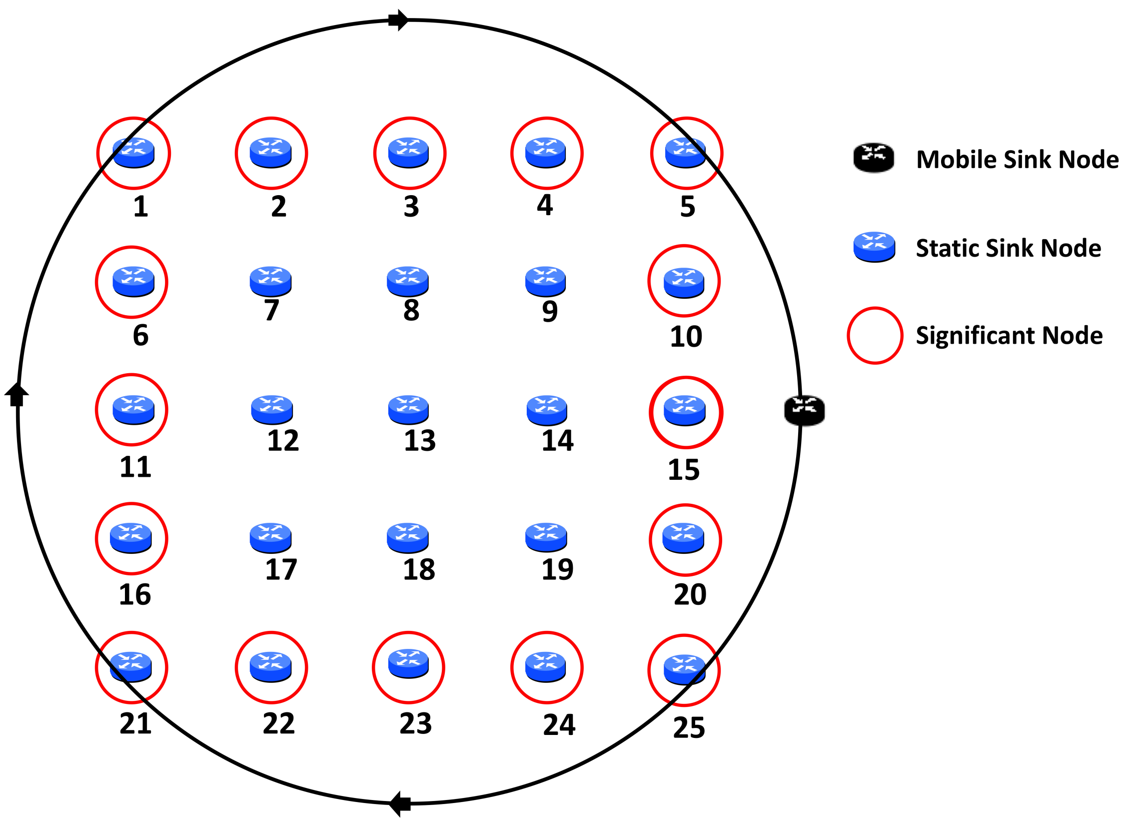


Figure 5.3: Network topology - grid with significant nodes.

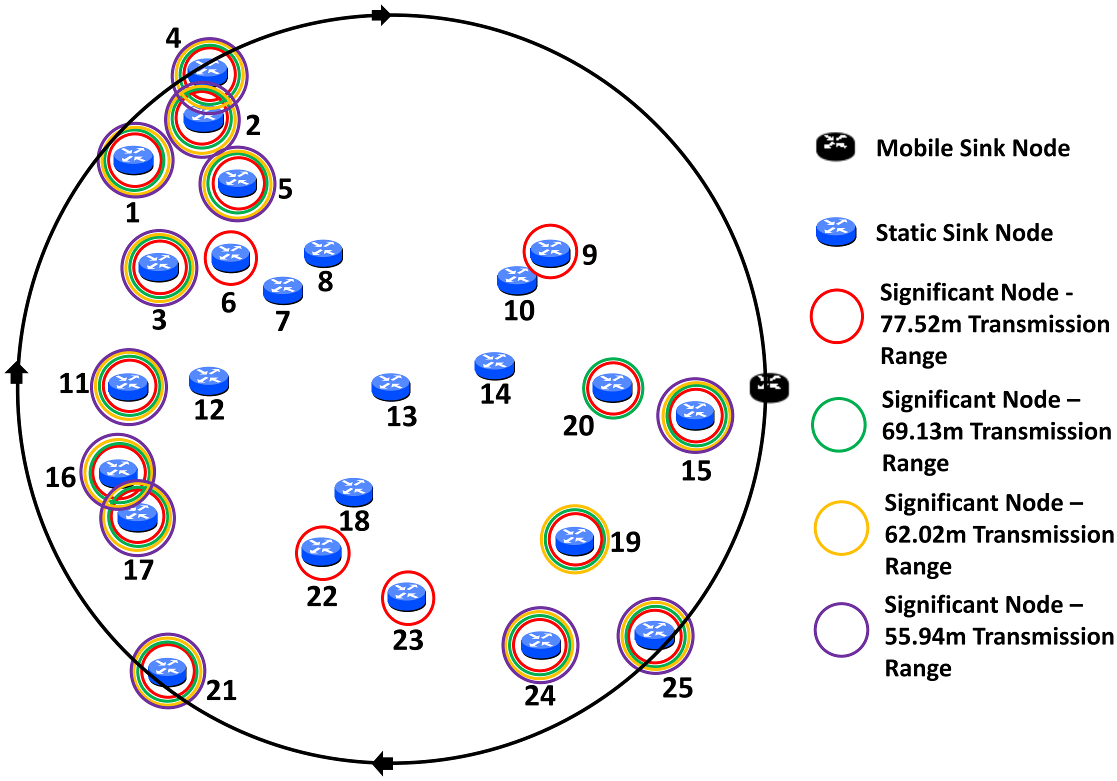


Figure 5.4: Network topology - random with significant nodes.

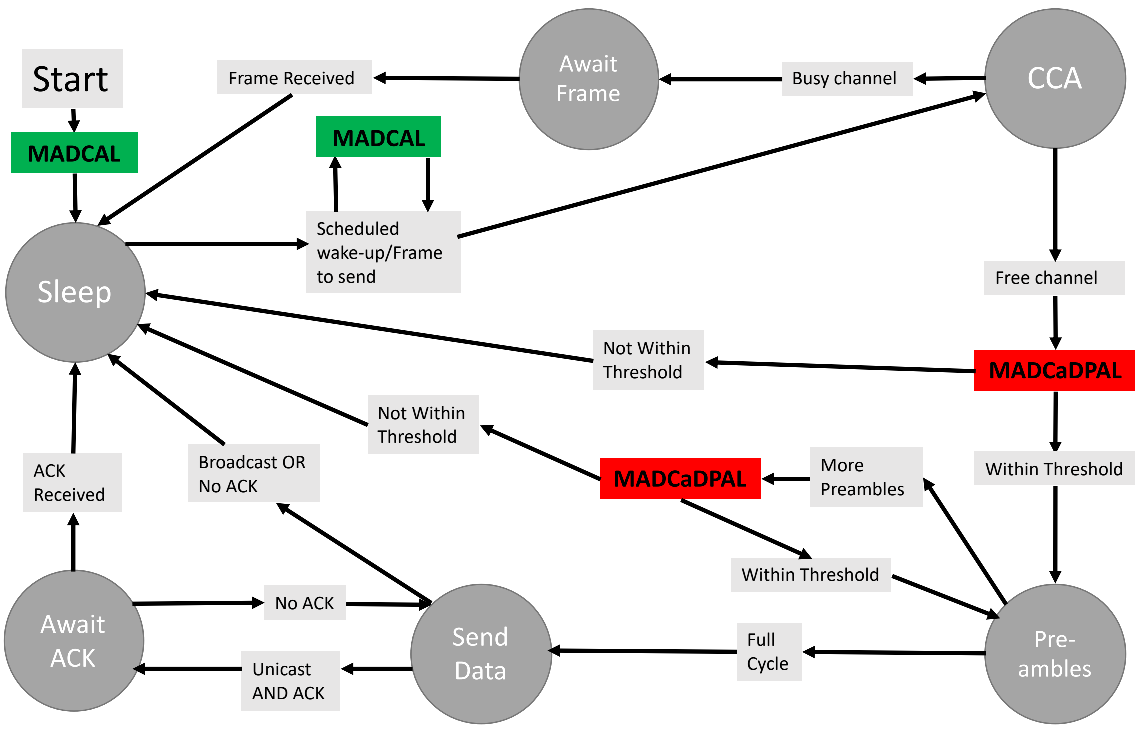


Figure 5.5: Lightweight CSMA/CA MAC Implementation with MADCaDPAL

and preamble sending.

5.2.1 New Threshold Calculation

When the MADCAL algorithm was developed, the initial threshold calculation was adjusted by a factor of the distance to the circle path divided by the node transmission distance. This was then adjusted based on sink speed, with faster speeds allowing for smaller thresholds. However, this was based around the specific speeds used, those being 2mps, 10mps, 20mps and 40mps. In this chapter, the calculation of this adjustment factor now becomes completely dynamic, with the only input required that of maximum and minimum sink speeds. In this study those are 2mps and 40mps, but this is now flexible. Any speed within that range will have an adjustment factor calculated specifically.

An illustration of the development of a dynamic communication threshold between static significant node and MSN can be seen in Figure 5.6, from the viewpoint of node 10. The new dynamic version of the algorithm to calculate this factor and how it is applied to the building of the threshold can be seen in Algorithm 3.

Algorithm 3: Within the Initialisation procedure, the MSN speed is set from input (2, 10, 20, or 40 mps), the max and min speeds in this case are 40mps and 2mps. The maximum factor we use is 0.5, utilised for slower speeds. The minimum factor is 0 which means for faster speeds there is no limit to how much the threshold may be reduced by. The *speedDiff* variable sets a range value between highest and lowest speeds.

The *establishThreshold* function, as with MADCAL, establishes the coordinates of the communication threshold, inputting the radius of the circular path and whether the coordinates to be calculated are before or after the *circlePoint*. The difference in MACDaDPAL is the new dynamic approach to determining the factor to apply to to threshold size. The distance from the centre of the circle to the static node is again calculated as *nodeDist*. As before, calculate the angle of the circle centre to the furthest point of communication, and the circle centre to the node,and to avoid excessively

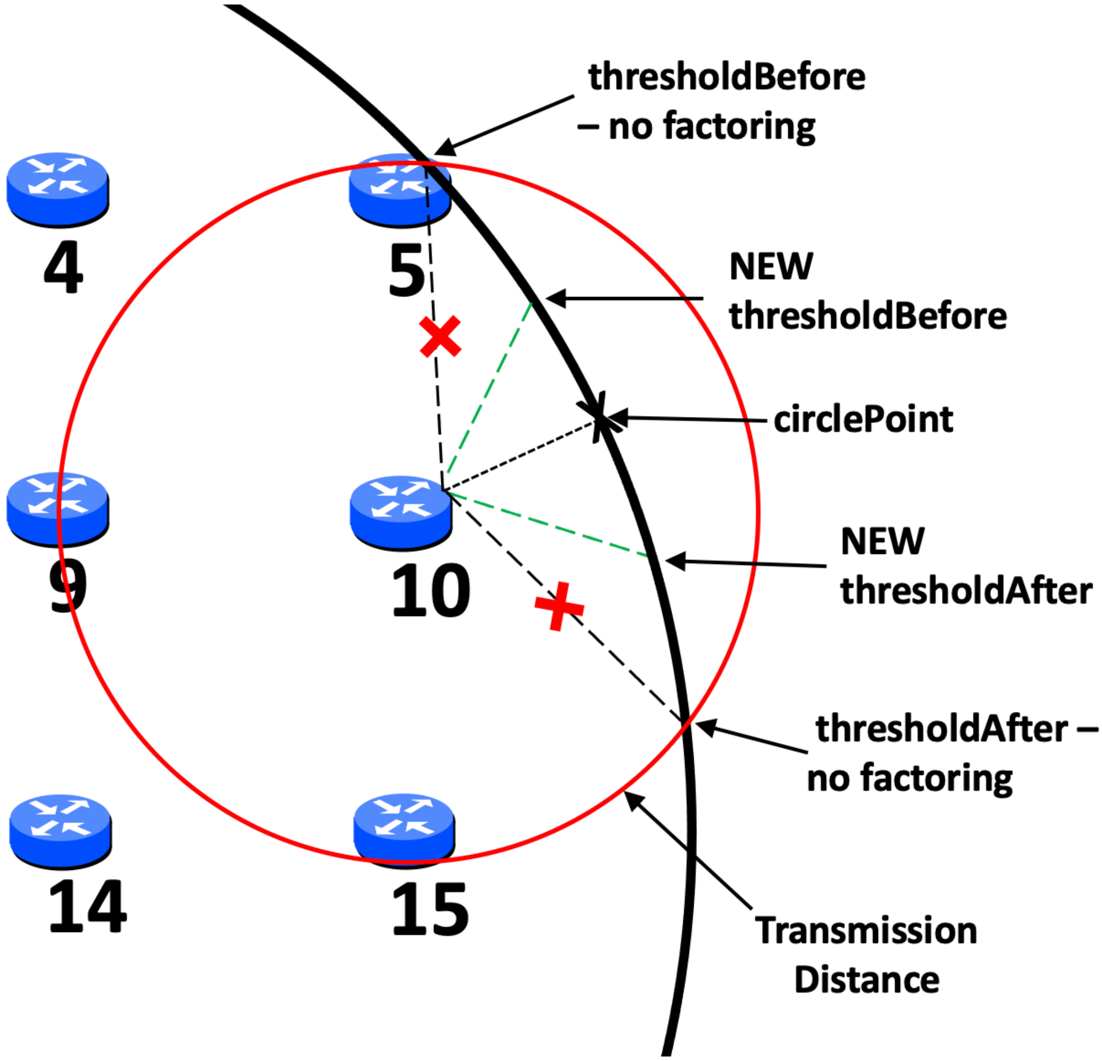


Figure 5.6: Threshold Adjustment - MADCaDPAL

large thresholds, determine a factor by which this angle should be reduced based upon *distToCircle* divided by *transDist*.

A new dynamic factor check is based on the difference of the highest and lowest factor divided by the difference in maximum and minimum speeds, multiplied by the difference between sink speed and the minimum speed. This ensures that whatever the speed, the *factorCheck* is calculated uniquely for that speed. The faster the sink speed may be, the more the angle of the threshold may be reduced by. If the factor calculated is less than the factor check then the factor check value becomes the factor. The value of the angle of the threshold is multiplied by the factor in order to reduce it accordingly. The threshold is then calculated as in MADCAL.

Algorithm 3 Communication threshold with dynamic factoring - MADCaDPAL

procedure INITIALISATION
  set $sinkSpeed$
  set $maxSpeed$
  set $minSpeed$
  set $maxFactor$
  set $minFactor$
  $speedDiff \leftarrow sinkSpeed - minSpeed$
end procedure
function ESTABLISHTHRESHOLD(radius, after)
  $nodeDist \leftarrow (radius - distToCircle)$
  $angleTemp \leftarrow \frac{(radius^2 + nodeDist^2 - transDist^2)}{(2 * radius * nodeDist)}$
  $angleRadians \leftarrow arccos(angleTemp)$
  $angle \leftarrow (angleRadians * (\frac{180}{PI})$
  $factor \leftarrow \frac{distToCircle}{transDist}$
  $factorCheck \leftarrow (\frac{(maxFactor - minFactor)}{(maxSpeed - minSpeed)} * speedDiff)$
  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

5.2.2 Mobility Aware Duty Cycling and Dynamic Preambling Algorithm (MADCaDPAL)

The novel approach the MADCaDPAL algorithm takes is in how duty cycling at the MAC layer may be influenced by the communication threshold. Beyond the MADCAL algorithm where only the SLEEP procedure is influenced in its move to CCA. This crucial first approach remains but is now extended to include a deeper examination of the actual CCA process and how the sending of preambles may be influenced. As previously, the move from the SLEEP function to CCA will only occur once the MSN is calculated to be within the significant node's threshold. However, in order to ensure that communication between the sink and static node ends as soon as possible once the MSN has exited the node's threshold, further checks are now made. Once the CCA procedure is entered a further check to ensure the sink is within the threshold is made. However, this merely ensures that the small possibility that between the SLEEP procedure and CCA the sink has moved beyond the threshold is accounted for.

Of greater importance is a further check once a clear channel is confirmed and the sending of preambles commences. Resultantly, the sending of preambles will be interrupted as soon as the MSN leaves the threshold, with any data in the cache of the static node stored until the next pass of the sink. Once interrupted, both the CCA and preamble sending procedures send the node to sleep for one slot duration. This is in line with with the existing approach to sending the node to sleep within the MAC protocol. This ensures the SLEEP procedure will then again assume responsibility for calculating the position of the sink node. This can be seen in Algorithm 4. It should be noted that the sending of data is not interrupted. Once this commences it is allowed to finish, wherever the sink may be located.

Algorithm 4: The second part of the MADCaDPAL algorithm uses some functions established within MADCAL in the previous chapter. The Sleep procedure is defined as previously in MADCAL, with CCA scheduled as before.

Within the CCA procedure, first there is a check to establish if there there is something in the queue to send. If so, the *withinThreshold* function is called to establish if the MSN is within the threshold currently. If the threshold has been reached move to sending preambles and schedule the stopping of preambles after a slot duration as normal. Otherwise, return to the Sleep procedure and awaken after a slot duration. If there is nothing in the queue, return to the Sleep procedure and awaken after a slot duration as normal. This ends the CCA procedure.

Within the Send Preamble procedure, the *withinThreshold* function is called to establish if the MSN is within the threshold currently. If the threshold has been reached then the process of sending preambles continues as normal unless Stop Preambles is reached. If the threshold has not been reached then the process of sending or stopping preambles is cancelled and the MAC protocol returns to the Sleep procedure. The node is scheduled to awaken after a slot duration. This ends the Send Preamble procedure.

Algorithm 4 Threshold Interval - MADCaDPAL

```
procedure SLEEP
    set checkInterval from input
    if significantNode then
        thresholdTime()
        if thresholdReached then
            interval ← checkInterval
        else
            interval ← timeToThreshold
        end if
    else
        interval ← checkInterval
    end if
    schedule CCA at simTime + interval
end procedure
procedure CCA
    if macQueue > 0 then
        withinThreshold()
        if thresholdreached then
            macState ← SEND PREAMBLE
            schedule STOP PREAMBLES at simTime + slotDuration
        else
            macState ← SLEEP
            schedule WAKEUP at simTime + slotDuration
        end if
    else
        macState ← SLEEP
        schedule WAKEUP at simTime + slotDuration
    end if
end procedure
procedure SEND PREAMBLE
    withinThreshold()
    if thresholdreached then
        if SEND PREAMBLE then
            sendPreamble()
            macState ← SEND PREAMBLE
            schedule SEND PREAMBLE at simTime + (0.5 ∗ checkInterval)
        else if STOP PREAMBLES then
            cancel SEND PREAMBLE
            macState ← SEND DATA #Preambles over, send data
        end if
    else
        cancel SEND PREAMBLE/STOP PREAMBLES
        macState ← SLEEP
        schedule WAKEUP at simTime + slotDuration
    end if
end procedure
function THRESHOLDTIME
    set sinkPos
    withinThreshold()
    if not thresholdreached then
        set arc to distance to sinkThresholdBefore
        set sinkQuartile
        timeToThreshold ← arc / sinkSpeed
    else
        timeToThreshold ← 0
    end if
    return timeToThreshold
end function
function WITHINTHRESHOLD
    if distance between sinkPos and thresholdAfter > thresholdDistance then
        thresholdReached ← false
    else if distance between sinkPos and thresholdBefore > thresholdDistance then
        thresholdReached ← false
    else
        thresholdReached ← true
    end if
    return thresholdReached
end function
```

5.3 Evaluation and Results

5.3.1 Network Parameters

Network parameters remain as for the MADCAL algorithm in the previous chapter. Check Interval and Slot Duration are displayed for reference and are utilised within the MADCaDPAL algorithm. The network layer routing protocol is again the Optimized Link State Routing protocol (OLSR) [85].

Network assumptions remain the same as for the previous MADCAL algorithm. Among both network implementations the assumption is made static nodes will retain their positions throughout. Each static node implements MADCaDPAL independently and although aware of its own location, is unaware of neighbouring nodes. Although four different transmission ranges are used across different test scenarios, each range is consistent across all nodes in each test. All other parameters can be found in Table 5.1.

Table 5.1: Simulation Parameters - MADCaDPAL

Parameters	**Values**
Number of Nodes (Static)	25
Grid Topology Size	$200m * 200m$
MSN Path Radius	$150m$
MSN Start Position	$x = 400m, y = 250m$
MSN Speed (metres per second)	$2mps, 10mps, 20mps, 40mps$
Simulation Time	$942.47779607694s$
Transmission Distance * 4	$77.52m, 69.13m, 62.02, 55.94m$
Number of Runs	5
Path-loss Alpha * 4	$1.85, 1.9, 1.95, 2$
Max Sending Power	$1.0mW$
Signal Attenuation Threshold	$-85dBm$
Sensitivity	$-75dBm$
Carrier Frequency	$2.4GHz$
Transmitter Power	$1.0mW$
Thermal Noise	$-85dBm$
Signal to Noise Ratio Threshold	$4dB$
Battery Capacity	$59400mWs$
Check Interval	$0.01s$
Slot Duration	$0.1s$

5.3.2 Simulation Environment and Parameters

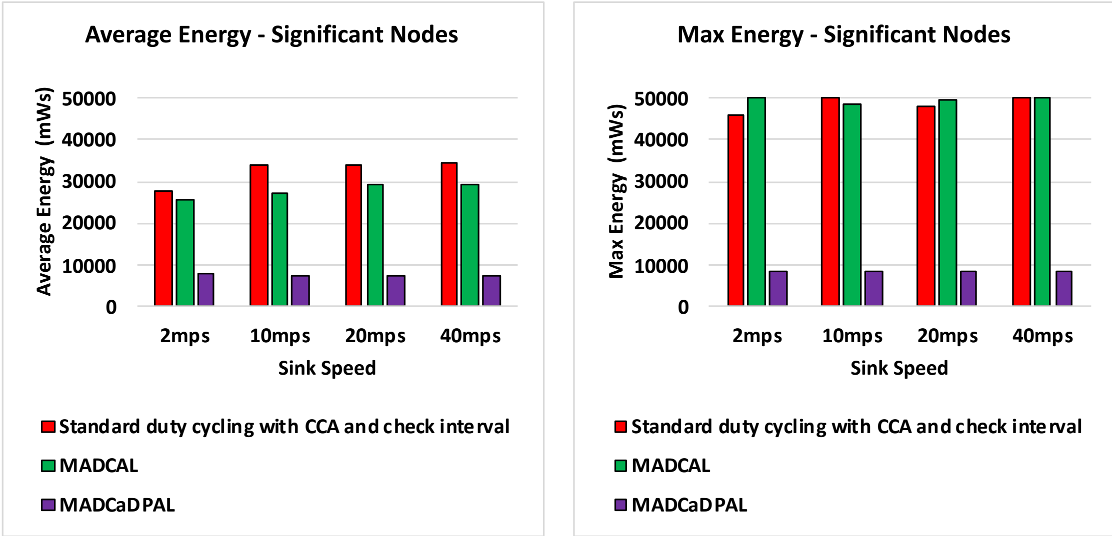
The grid and random topology networks are again built using the OMNeT++ [77] simulation framework. In this case MiXiM [87] is utilised to build the actual network layout, whilst inetmanet [88] is used to define physical, MAC, network and transport layer parameters.

Results show the effect of an MSN on the significant nodes, these highlighted in Figures 4.1 and 4.2. Each test scenario has been conducted using three different MAC implementations to establish the effect of a MSN at four different speeds and with four different transmission ranges across the network. Firstly, the existing lightweight MAC implementation using standard duty cycling techniques with no allowance for sink mobility is shown. Secondly, results are shown using the MADCAL algorithm and finally, results are presented where MADCaDPAL is in use.

5.3.3 Results - Grid Topology

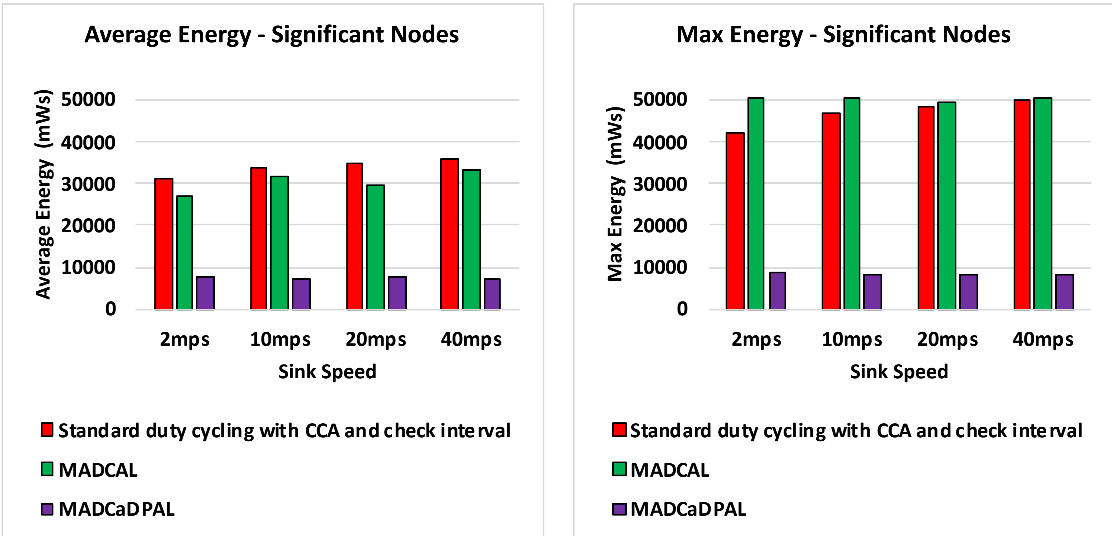
Figures 5.7-5.10 show the average energy consumption of all significant nodes. This is presented alongside the maximum energy consumption of significant nodes, this being the single significant node to consume the most energy. In examining this comparison we may observe the difference between the overall average and the greatest spike in energy consumption in a single node.

In Figure 5.7a the initial energy saving of up to 15% when MADCAL is in use can be seen versus the original MAC implementation. However, when MADCaDPAL is in use the resultant reduction in energy consumption is significant. Compared to original results with no allowance made for use of an MSN, we can observe up to 79% reduction in energy consumed. What can also be seen in Figure 5.7b is that in the original MAC implementation and again when MADCAL is in use, that spikes in energy were considerable. Now with MADCaDPAL in use, the highest single node is only slightly



(a) Average Energy **(b)** Max Energy

Figure 5.7: Average and Maximum Energy Consumption of Significant Nodes (mWs), Grid Topology. Transmission Range 77.52m. MADCaDPAL

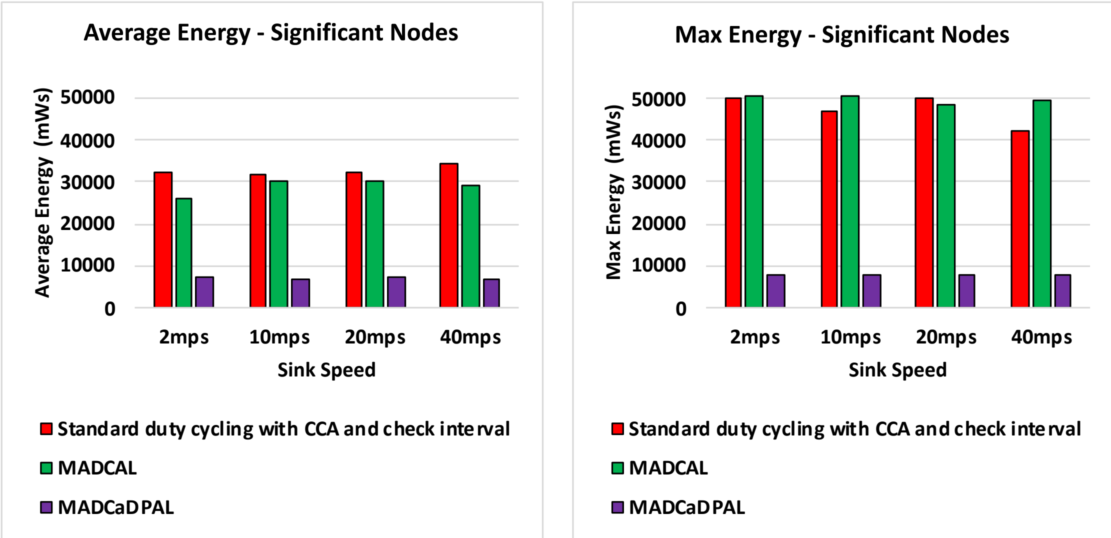


(a) Average Energy **(b)** Max Energy

Figure 5.8: Average and Maximum Energy Consumption of Significant Nodes (mWs), Grid Topology. Transmission Range 69.13m. MADCaDPAL

above average on this particular scale.

Even though the transmission range is reduced, a similar pattern of results can be seen in Figure 5.8. Again there is a significant reduction in average energy consumption as seen in Figure 5.8a, as well as with the maximum single node energy once MADCaDPAL is in use, as seen in Figure 5.8b.

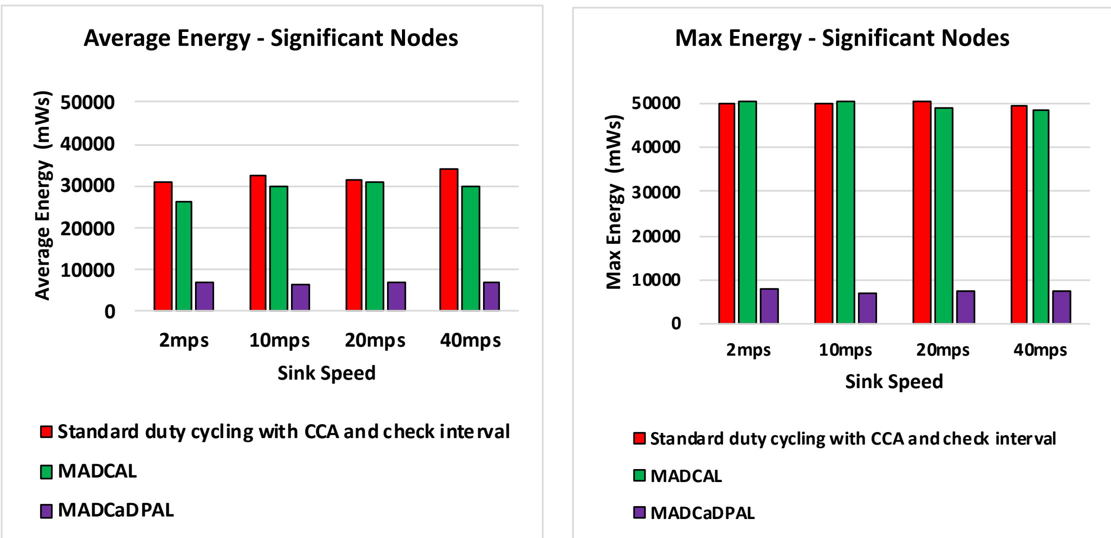


(a) Average Energy

(b) Max Energy

Figure 5.9: Average and Maximum Energy Consumption of Significant Nodes (mWs), Grid Topology. Transmission Range 62.02m. MADCaDPAL

It can be seen in Figure 5.9 that as the transmission range becomes smaller, as do the benefits achieved by the MADCAL algorithm. However, with MADCaDPAL in use, the pattern of significant improvement in average energy as seen in Figure 5.9a and maximum energy as seen in Figure 5.9b, remains.



(a) Average Energy

(b) Max Energy

Figure 5.10: Average and Maximum Energy Consumption of Significant Nodes (mWs), Grid Topology. Transmission Range 55.94m. MADCaDPAL

Even at the smallest transmission range used as in Figure 5.10, this pattern of improvements with the MADCaDPAL algorithm in use continues. Originally, as the transmission range constricted to not much more then then distance between nodes, it became more difficult for the MADCAL algorithm to make improvements. Although they are still there to be seen. This, however, does not affect the MADCaDPAL algorithm. In ensuring the threshold is closed effectively the result is a reduction in energy consumption of over 80% amongst the significant nodes as seen in Figure 5.10a. With energy consumed more evenly as the maximum figures show in Figure 5.10b.

However, a benefit such as this would be less significant if it has a detrimental effect on other factors in the network. In Figures 5.11 to 5.14 the number of MAC layer frames received by the MSN during each simulation can be seen.

In Figures 5.11 and 5.12 it can be seen that as well as a significant improvement in energy consumption, the MADCaDPAL algorithm has also improved frame delivery in all but one scenario. With a 77.52m transmission range and sink speed of 10mps frame reception decreases by around 20% from the MADCAL algorithm, although this is still an improvement on the original MAC implementation. However, in all other scenarios

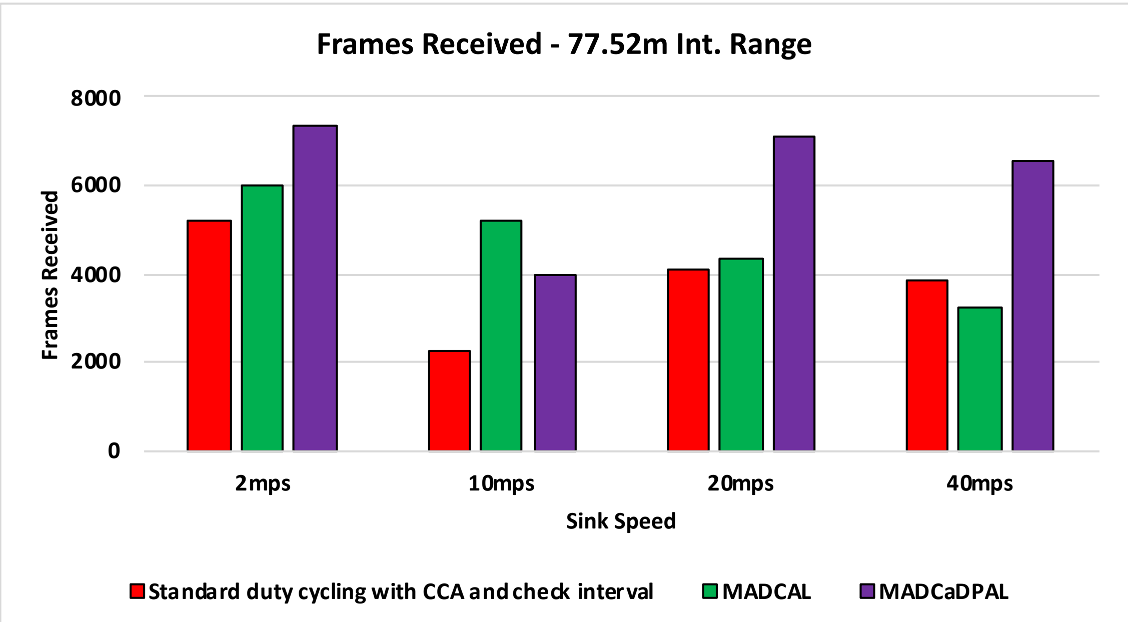


Figure 5.11: Sink Frame Reception. Transmission Range 77.52m, Grid Topology. MADCaDPAL

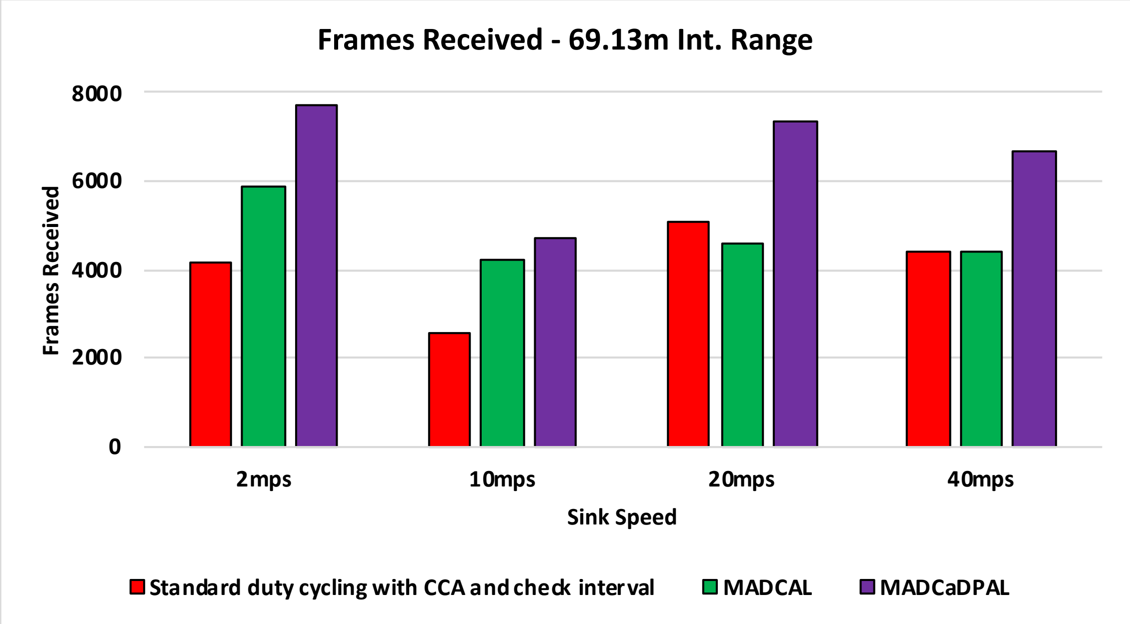


Figure 5.12: Sink Frame Reception. Transmission Range 69.13m, Grid Topology. MADCaDPAL

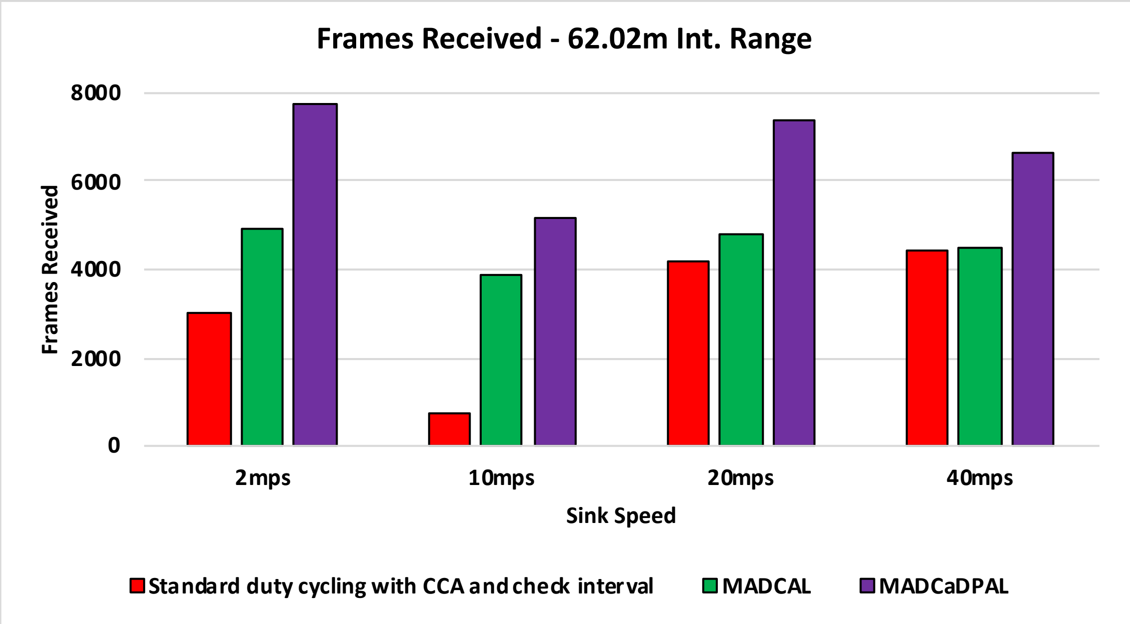


Figure 5.13: Sink Frame Reception. Transmission Range 62.02m, Grid Topology. MADCaDPAL

there is an improvement over both no change and the MADCAL algorithm. Significant improvement can be seen at 20mps, despite this being highlighted as of difficulty to achieve by the MADCAL algorithm.

Figures 5.13 and 5.14 show that even as transmission range again reduces, sink frame reception improves with MADCaDPAL in use. Across all tests a speed of 10mps proves to

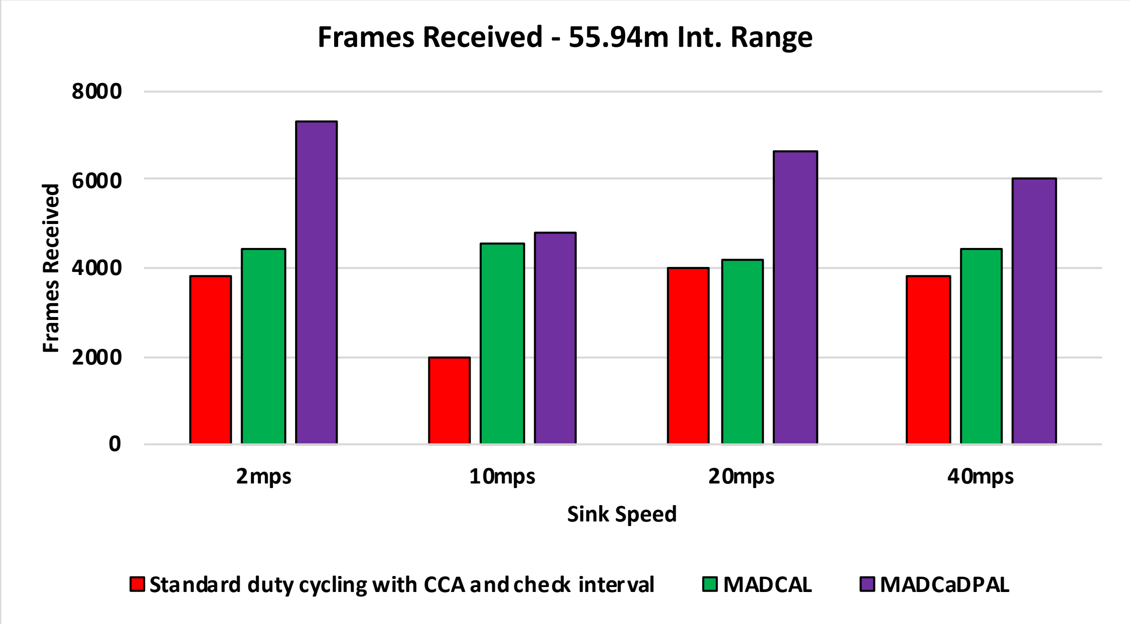


Figure 5.14: Sink Frame Reception. Transmission Range 55.94m, Grid Topology. MADCaDPAL

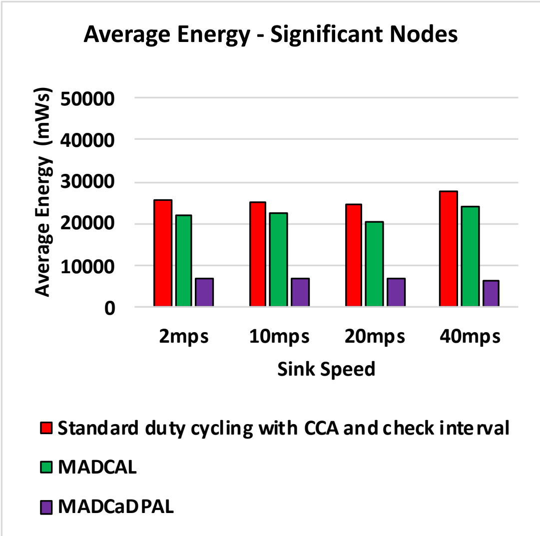
be problematic in this regard. However, considering the benefit in energy consumption, sink reception even remaining at similar levels could be seen as acceptable.

5.3.4 Results - Random Topology

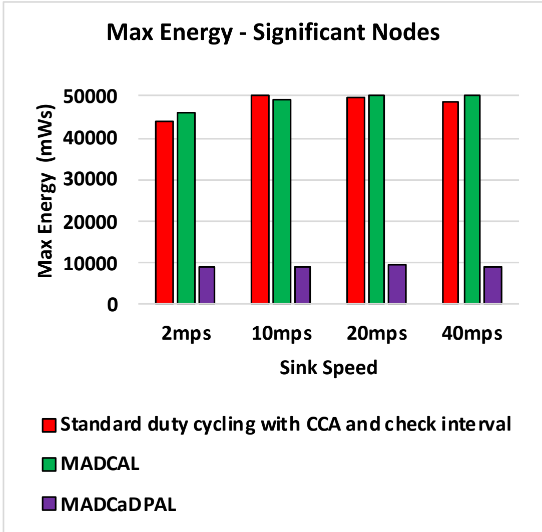
Figures 5.15 to 5.18 show the average energy consumption of all significant nodes, again presented alongside the maximum energy consumption of significant nodes.

In Figure 5.15 results are very much as with the grid topology, although it can be seen that the spike in energy consumption when standard duty cycling and MADCAL are in use is even more distinct. Again energy consumption is reduced dramatically. What is off interest is that although the benefits are great, there is still a slight difference between average energy with MADCaDPAL in use and the maximum energy consumed. Showing that even though the scale of energy consumption is reduced and network lifetime increased, there is still an opportunity to further even energy consumption across significant nodes.

This pattern repeats in Figure 5.16, even as transmission range reduces. MADCaD-

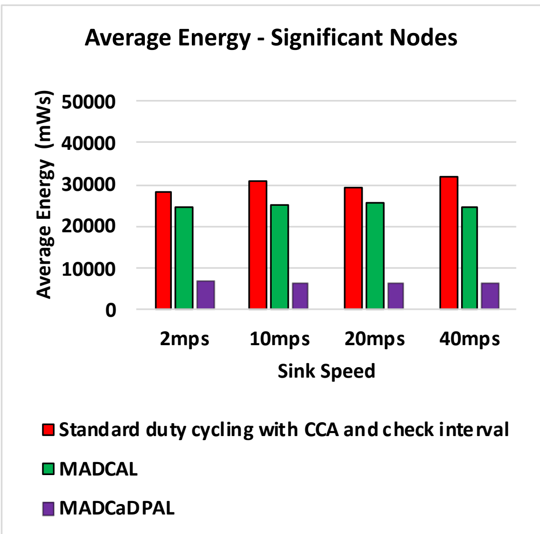


(a) Average Energy

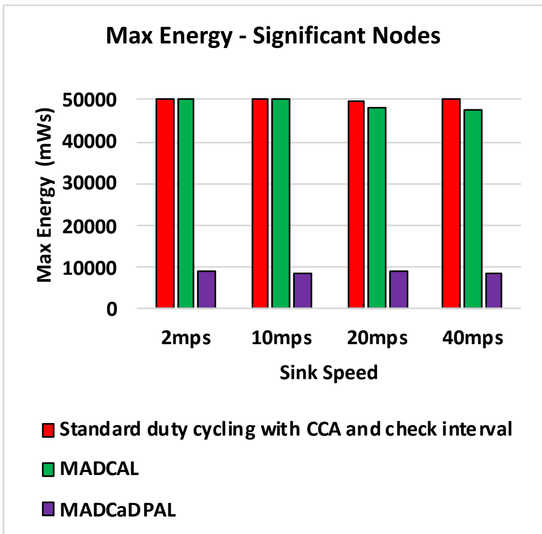


(b) Max Energy

Figure 5.15: Average and Maximum Energy Consumption of Significant Nodes (mWs). Transmission Range 77.52m, Random Topology. MADCaDPAL

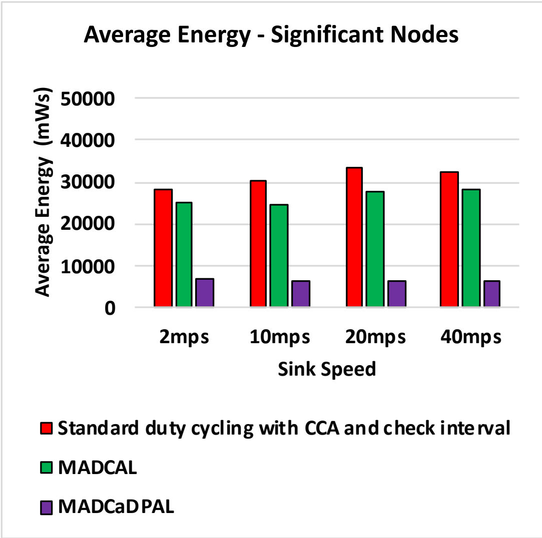


(a) Average Energy

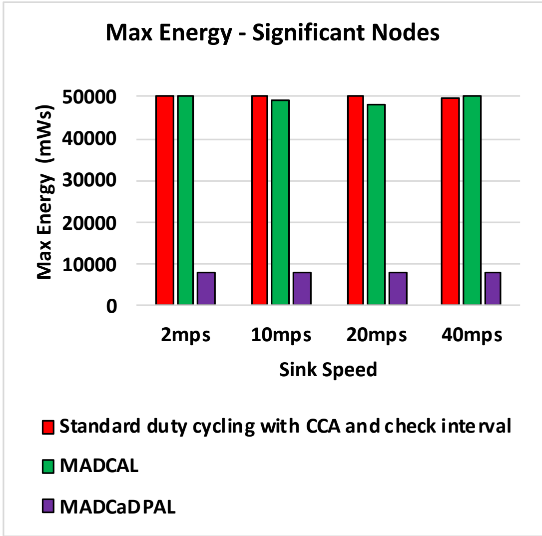


(b) Max Energy

Figure 5.16: Average and Maximum Energy Consumption of Significant Nodes (mWs). Transmission Range 69.13m, Random Topology. MADCaDPAL

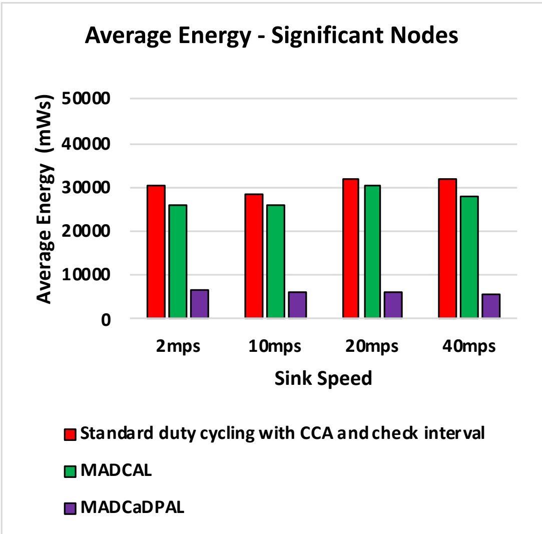


(a) Average Energy

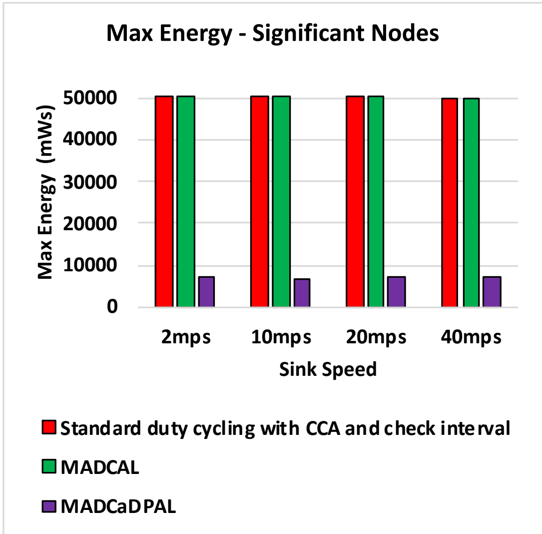


(b) Max Energy

Figure 5.17: Average and Maximum Energy Consumption of Significant Nodes (mWs). Transmission Range 62.02m, Random Topology. MADCaDPAL



(a) Average Energy



(b) Max Energy

Figure 5.18: Average and Maximum Energy Consumption of Significant Nodes (mWs). Transmission Range 55.94m, Random Topology. MADCaDPAL

PAL can be seen to again significantly reduce energy consumption.

Even at the smallest transmission ranges used as in Figures 5.17 and 5.18, this pattern of improvements with the MADCaDPAL algorithm in use continues. The improvement in reducing energy spikes is now very clear. As a result, improving network lifetime before the first node expires by a factor of up to 7.

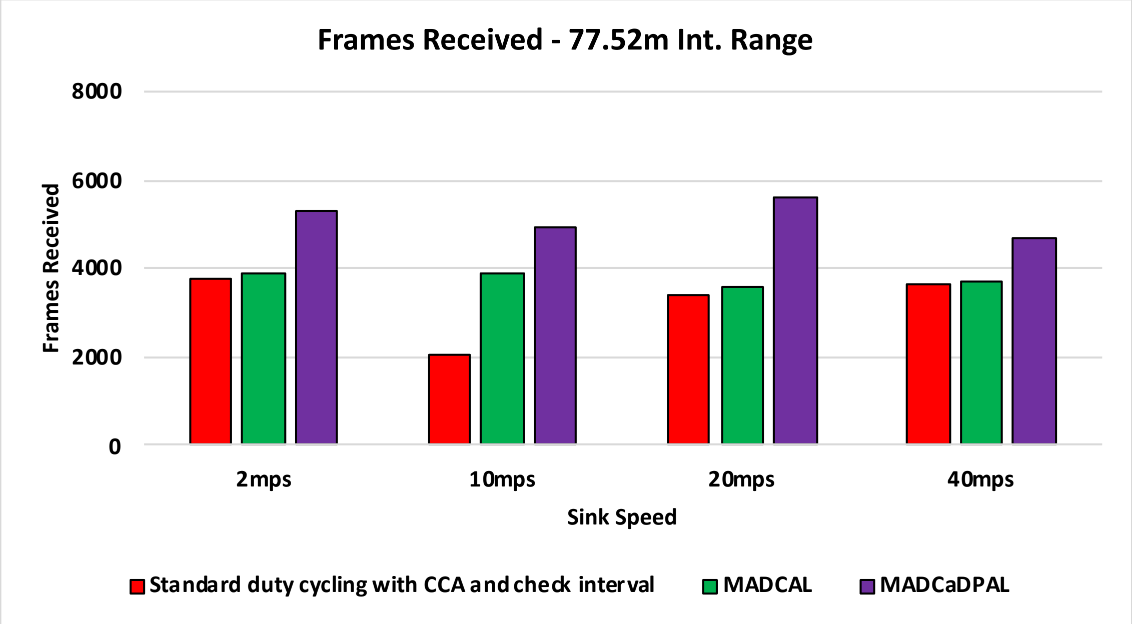


Figure 5.19: Sink Frame Reception. Transmission Range 77.52m, Random Topology. MADCaDPAL

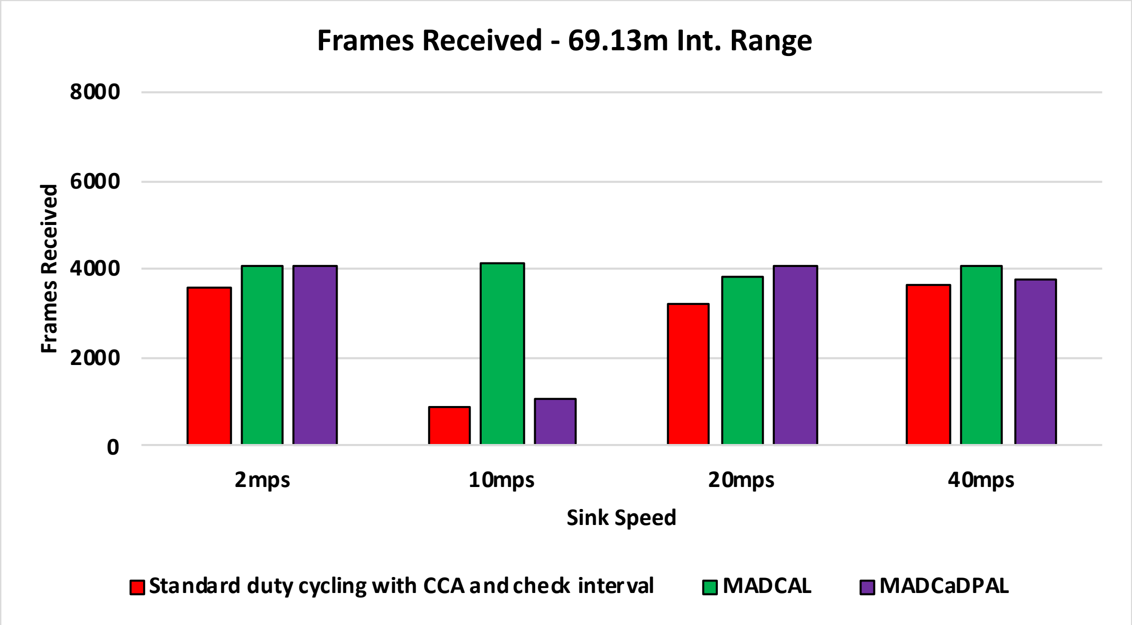


Figure 5.20: Sink Frame Reception. Transmission Range 69.13m, Random Topology. MADCaDPAL

In Figures 5.19 to 5.22 the number of MAC layer frames received by the MSN during each simulation can be observed.

In Figure 5.19 it can be seen that with the largest transmission range in use there

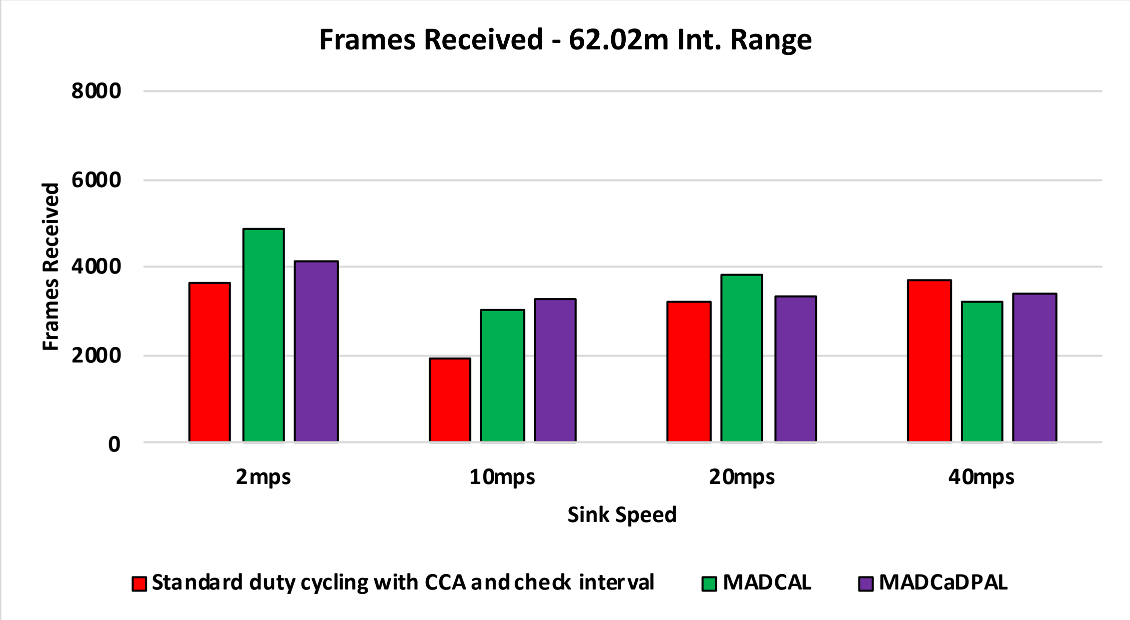


Figure 5.21: Sink Frame Reception. Transmission Range 62.02m, Random Topology. MADCaDPAL

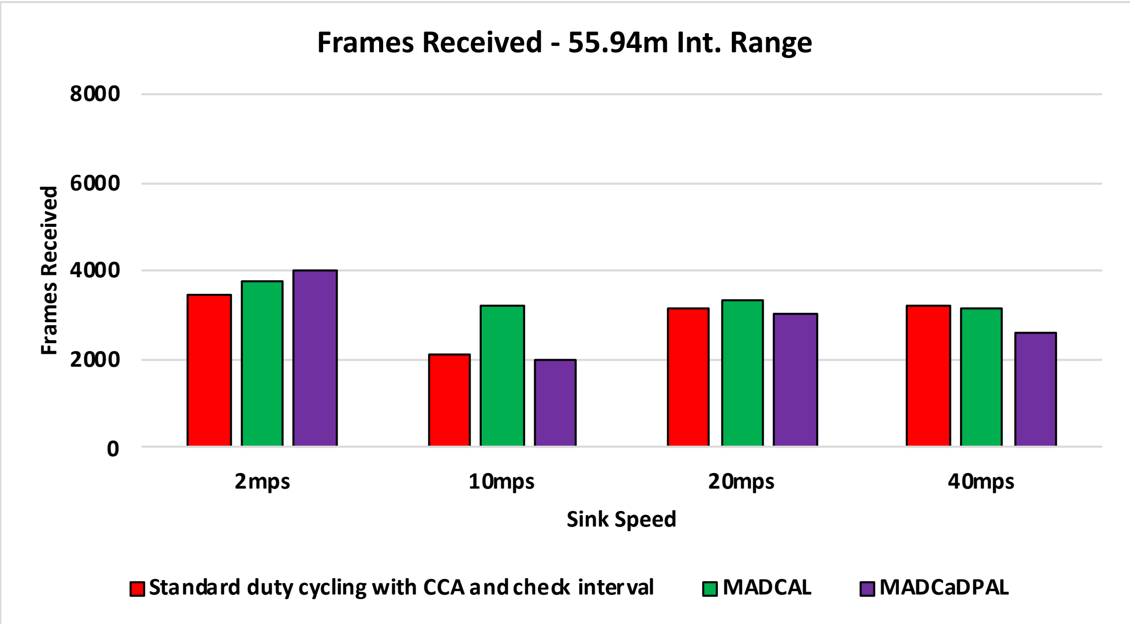


Figure 5.22: Sink Frame Reception. Transmission Range 55.94m, Random Topology. MADCaDPAL

is generally an improvement in frame delivery to the sink. As the transmission range lessens in Figure 5.20, frame delivery generally stays the same aside from when the sink is moving at 10mps. Delivery now returns to the poor level of when the standard duty cycling was in use.

Figure 5.21 shows that in the lower transmission ranges again frame delivery remains at similar levels. Again though, at the smallest transmission range as seen in Figure 5.22, there is a drop when the sink moves at 10mps.

5.4 Conclusion

In this study we propose MADCaDPAL, extending the previous MADCAL algorithm. for use with MSNs. In utilising determination of the sink node position in relation to the threshold, within the CCA and preambling procedures of the lightweight CSMA/CA MAC protocol used, and linking this to the already successful use of this approach in the SLEEP procedure, improvements of between 70% and 80% are observed in energy consumption. As importantly from a network lifetime standpoint however, the issue of excessive energy spikes is somewhat negated. Improvements are also shown in frame delivery when a controlled, grid network formation is used. When we utilise a more *strained*, random topology frame delivery is generally the same or improved aside from two scenarios. As such, the extended MADCaDPAL algorithm demonstrates how this functionality may now be used to have a significant effect on network behaviour. Whilst the initial aim was to reduce network spikes, the subsequent benefit in energy consumption is so great as to represent a major step forward in this area.

Whilst the MADCaDPAL algorithm is an effective solution to energy consumption when a MSN is in use, examination of the results provide an opportunity to further balance energy consumption across significant nodes, despite the now lower energy levels consumed. This shall be presented in Chapter 6.

Sixth Chapter

Dynamic Mobility and Energy Aware Algorithm (DMEAAL)

6.1 Introduction

Whilst the communication threshold developed has been shown to have benefit in terms of energy consumption, thus far, however, once the threshold is established it remains the same. Despite the level of battery consumption within the static node. In both the MADCAL and MADCaDPAL algorithms, the communication threshold utilised between static significant node and MSN is adjustable in that it is dependent on the sink speed, the distance between the static node and the sink path and finally the transmission range of the static node. However, once this threshold is established it then is not altered again. At this stage, therefore, it is assumed that the threshold distance is optimum for however long the WSN exists for.

Whilst this approach has demonstrated benefit in terms of energy consumption, there must also be consideration given to the fact that any nodes expending energy greater than the average consumption of all significant nodes in the network shall continue to do so, until eventually running out of power earlier than other nodes. Considering how long these networks may operate for, even a small difference over a short period could add up to a node expiring significantly sooner than necessary over the lifetime of the network. Therefore, an improvement in balancing the energy consumption of significant nodes in line with the average consumption, even if this improvement is initially small, would become more significant over a longer period. In this way this chapter answers the fourth and final research questions, that being:

- To what extent can nodes in a WSN utilising a MSN be context-aware? Such that said context, such as the energy levels of nodes, may be utilised accordingly to adjust a communication threshold and ensure even energy consumption across nodes. As such, improving network lifetime.

Hence, this chapter details the development of the DMEAAL algorithm, the final contribution of this thesis. Here, with energy consumption significantly improved by the MADCaDPAL algorithm in the previous chapter, the focus moves away from merely improving energy consumption and to how this energy is consumed and by which nodes. As such, DMEAAL takes an entirely dynamic approach. The communication threshold is adjusted in real time, with the average energy consumption figures attained in the use of MADCaDPAL utilised as a target compared against the current battery level in each particular significant node. With the threshold adjusted accordingly if the node energy consumption is above or below the target. In this way even small spikes in energy consumption may be avoided, with network lifetime improved.

6.2 A Real-time Approach to Mobility Aware Duty Cycling

In this study, the interaction between the communication and the internal workings of the MAC protocol is no longer the focus. The use of first the MADCAL algorithm remains to establish a communication threshold between static node and MSN, with the MADCaDPAL algorithm then determining whether the MSN is within the threshold or not and ending communication at the correct time. We now aim to adjust the threshold for a static significant node as time passes, such that when energy is being consumed above or below the average rate, the threshold may be altered accordingly.

As such, our hypothesis is that if the network is now operating at near full efficiency in terms of the performance of significant nodes, then the total energy consumed by all significant nodes is unlikely to be improved by any great degree. With the MADCaDPAL algorithm already seen to reduce energy consumption by 80%. However, if some nodes are consuming more energy than others, then it must be possible to lessen the reliance on these nodes and increase reliance accordingly on other nodes. Such that whilst the overall level of consumed energy remains, it is now spread more evenly across significant nodes. This without being to the detriment of frame delivery. Even given the benefit of the MADCaDPAL algorithm, at now much lower level of energy consumption then previously, it can still be observed that there is a gap in energy consumption between the highest and lowest significant nodes that could potentially be closed.

Therefore, the aim is to eliminate *dead* energy consumption. That being, energy which need not be consumed by one node and could be consumed by another instead with similar results but an increase in network lifetime. Thus, the MADCAL algorithm is first used to set up the base communication threshold between a static *significant* node and a MSN. An illustration of this can be seen in Figure 5.6, from the viewpoint of node 10. The threshold is created based on significant node distance from the circular sink path, but an adjustment is then made based on a factor which takes into account MSN speed combined with the distance to the path. The position of the MSN in relation to this threshold is used to influence the SLEEP function at the MAC layer, intercepting the move to CCA when data is to be sent, if the MSN is not within a significant node's threshold. The sink position in relation to the threshold is also utilised to influence CCA and the sending of preambles, beyond merely awakening the node. Thus, closing the threshold of communication immediately once the MSN has reached its end. Further, however, a further aim is to now integrate knowledge of energy consumption whilst utilising a cross-layer approach. Thus, the current energy consumption within the node battery may be compared with the target level of consumption - in this case built from the previous test scenarios utilising the MADCaDPAL algorithm.

6.2.1 Energy Awareness in Real-time

The focus of this chapter is DMEAAL. Such that the communication threshold is altered accordingly, dependant upon the comparison between target energy consumption and actual energy consumption of the node. Target energy consumption is achieved via the MADCaDPAL algorithm, with a database of results created per scenario. The cross-layer approach is achieved by the MAC protocol accessing the battery module after each pass of the MSN, as the network scenario runs, utilising the DMEAAL algorithm. Target energy consumption is then input at the start of each scenario. A simple adjustment value *threshAdjust* is then calculated based on the actual energy consumption's relation to target energy consumption as shown in Eq. (6.1).

$$threshAdjust = \left(\frac{targetEnergy}{\left(\frac{batteryCapacity - batteryResidual}{simTime} \right) \times 60} \right) \tag{6.1}$$

Where *targetEnergy* is the target energy consumption per minute in mWs; *batteryCapacity* is the capacity of the battery in mWs retrieved from the battery module in a cross-layer approach; *batteryResidual* is the residual energy of the battery module; *simTime* is the current simulation time in seconds.

A maximum threshold is set in the calculation of the initial threshold value, based on node transmission range and distance from the path of the sink. The threshold is then adjusted based on this factor.

It should be noted that in examining this approach it is clear that when the MSN is moving slowly, adjusting results becomes more problematic. This due to fewer opportunities to adjust the threshold on each pass of the sink as well as much greater energy consumption in general for each circuit of the network. Consequently, the minimum sink speed utilised in this chapter is 20mps.

In order to test this approach, both the grid and random network topologies used to test the MADCAL and MADCaDPAL algorithms are again utilised, with a predictable circular sink mobility pattern moving around the network in a clockwise direction as seen in Figures 5.1 and 5.2, with significant nodes remaining as seen in Figures 5.3 and 5.4.

A maximum threshold is set in the calculation of the initial threshold value, based on node transmission range and distance from the path of the sink. The threshold is then adjusted based on this factor. The lightweight CSMA/CA MAC implementation within which our work is implemented can be seen in Figure 6.1; showing where the DMEAAL algorithm is inserted to alter the threshold in real-time. As shall be detailed, this occurs during both CCA and the sending of preambles when the MSN is outside of the communication threshold. This is shown alongside MADCAL for the initial calculation of the communication threshold and implementation in the SLEEP procedure, with MADCaDPAL for the implementation of this threshold in the CCA and SEND PREAMBLE sections.

6.2.2 Dynamic Mobility and Energy Aware Algorithm (DMEAAL)

The development of a dynamic communication threshold between static significant node, from the viewpoint of node 10, has been illustrated in Figure 5.6. This is created by the MADCAL algorithm and then utilised by the MADCaDPAL algorithm in order to fully influence the SLEEP and CCA procedures as well as the sending of preambles. Resultantly, communication between static significant node and MSN is closed once the sink has moved past the threshold as shown in Figure 5.6. However, the DMEAAL algorithm is a completely dynamic, real-time solution. Figure 6.2 illustrates how the threshold may now be adjusted as long as it remains within a maximum size. It also

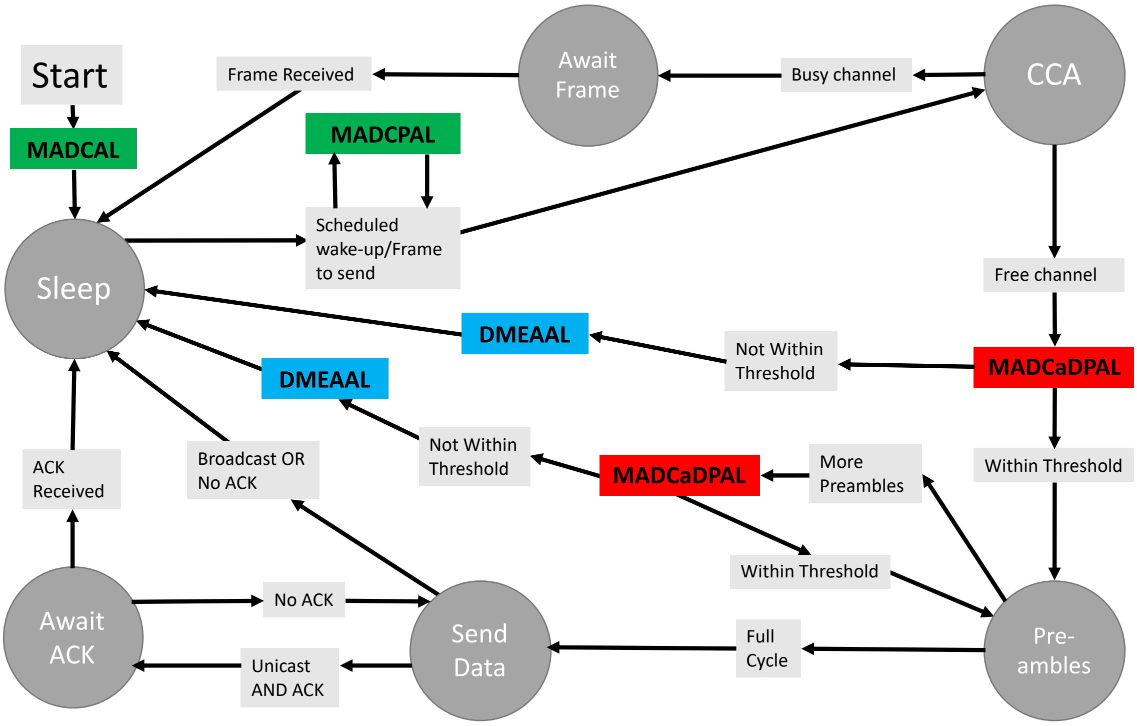


Figure 6.1: Lightweight CSMA/CA MAC Implementation with DMEAAL

demonstrates how power levels from the battery are compared to a target energy consumption level.

As such, the approach taken is to determine the initial threshold as currently determined by MADCAL. However, this now utilises a completely dynamic approach to establishing the factor by which the initial threshold calculation is reduced, based upon sink speed. This is detailed in Algorithm 5 followed by a detailed description. Algorithm 5 is run during the initialisation stage of all static nodes. However, whereas with MADCAL or MADCaDPAL in use this threshold would remain the same, in the case of DMEAAL this is merely an initial value. Hence, the value of this threshold will now only remain until the first recalculation based on current energy consumption.

Algorithm 5: Algorithm 5 details the initialisation and original establishment of a communication threshold during the initial startup of a static node, as initially developed in the MADCAL algorithm and then utilised further by MADCaDPAL. Within the *initialisation* procedure it is established whether this node is significant or not, with the

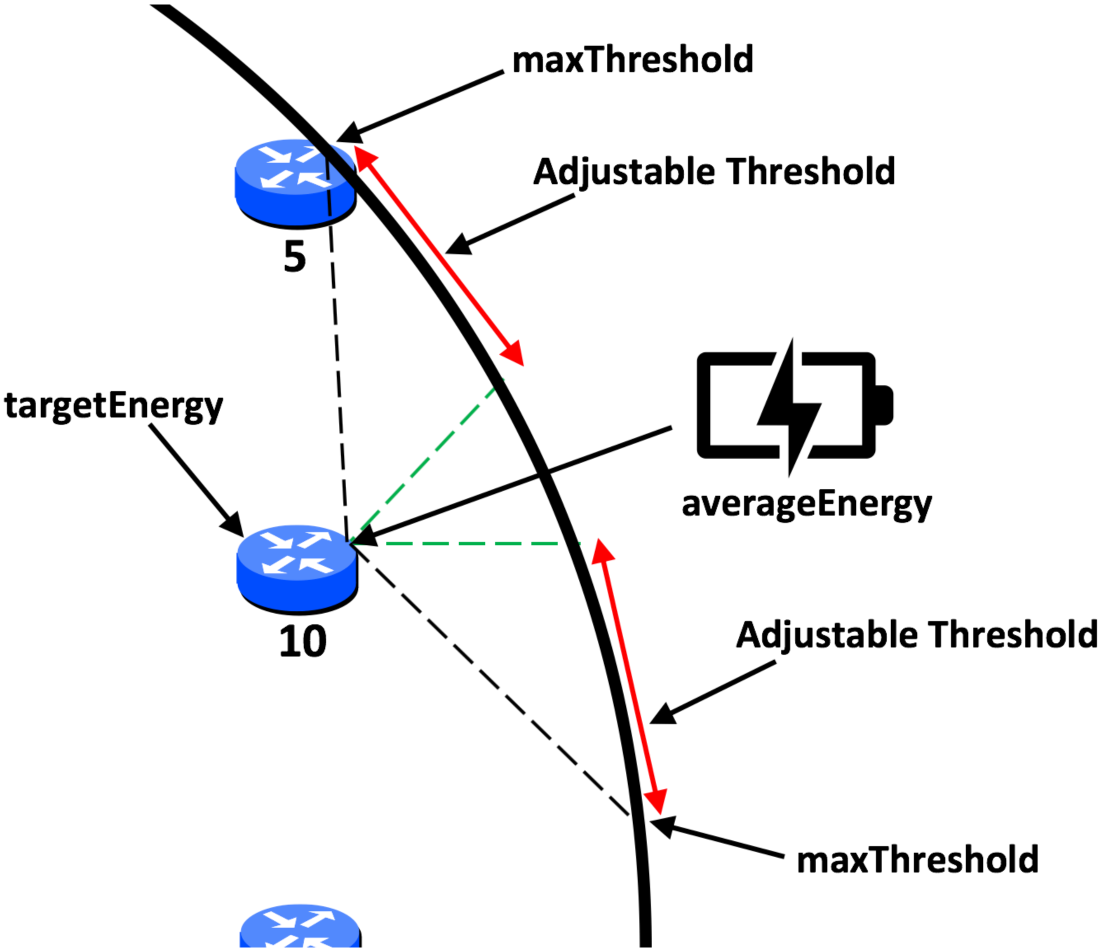


Figure 6.2: Adjustment of threshold based on average and target energy - DMEAAL

establishThreshold function then establishing the size of the actual threshold. Different from the original MADCAL algorithm are the setting of *maxAngle* on Line 36 and the now completely dynamic *factorCheck* on Line 38. *maxAngle* is set to the value of the threshold before adjustment, calculated solely on the angle created by the distance from the node to the circular path and the transmission distance. This shall then be utilised later in the DMEAAL algorithm to ensure the limit of the adjusted threshold. The new calculation of *factorCheck* now ensures the factor by which the threshold is reduced is a sliding scale between a maximum and minimum sink speed. In the case of this study those values are 40mps and 2mps. A maximum and minimum factor is also set as 0.5 and zero, meaning the most the factor may be reduced by is 0.5, for the slowest speeds, and not at all for the fastest speeds.

Figure 6.3 illustrates the flow of the procedures affected by DMEAAL, which can be

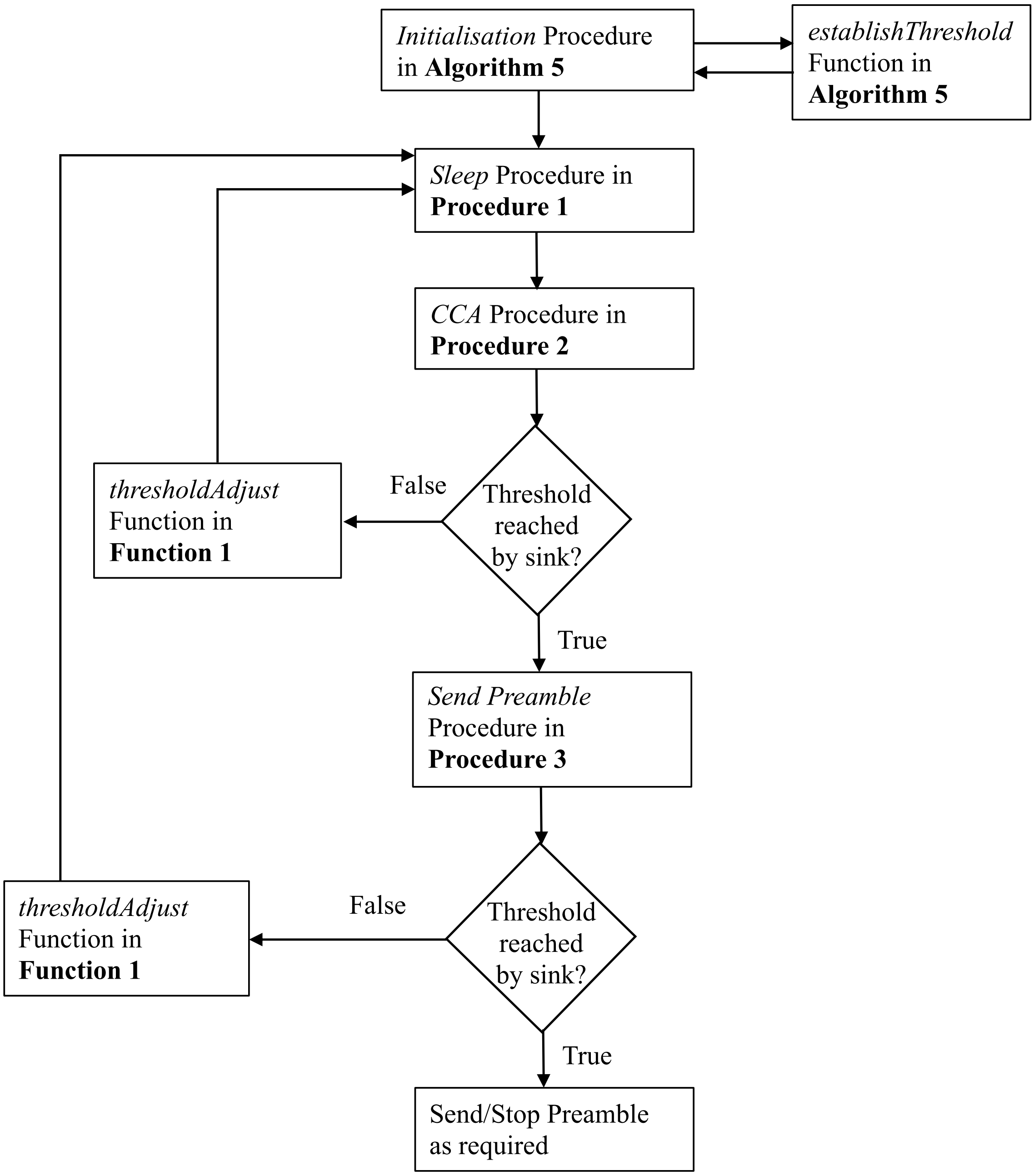


Figure 6.3: DMEAAL Control Flow

Algorithm 5 Initial Communication Threshold

```
procedure INITIALISATION
    set sinkSpeed
    set maxSpeed
    set minSpeed
    set maxFactor
    set minFactor
    speedDiff ← sinkSpeed − minSpeed
    significantNode ← false
    set transDist                                  ▷ Calculate transmission distance as in Eq. 4.9
    set Circumference
    set firstSinkPos
    set firstSinkQuartile
    set distToCircle
    if distToCircle < transDist then
        significantNode ← true
    end if
    if significantNode then
        set circlePoint
        set nodeQuartile
        set distanceBetweenPoints
        set angleOfNode
        thresholdAfter ← true
        sinkThresholdAfter ← establishThreshold(sinkRadius, thresholdAfter)
        thresholdAfter ← false
        sinkThresholdBefore ← establishThreshold(sinkRadius, thresholdAfter)
        set thresholdDistance
        set beforeQuartile
        set thresholdOpposite
    end if
end procedure
function ESTABLISHTHRESHOLD(radius, after)
    nodeDist ← (radius − distToCircle)
    angleTemp ← (radius^2 + nodeDist^2 − transDist^2) / (2 ∗ radius ∗ nodeDist)
    angleRadians ← arccos(angleTemp)
    angle ← (angleRadians ∗ (180 / PI))
    maxAngle ← angle                               ▷ To be utilised later by DMEAAL. The largest angle possible.
    factor ← distToCircle / transDist
    factorCheck ← ((maxFactor − minFactor) / (maxSpeed − minSpeed) ∗ speedDiff)
    if factor < factorCheck then
        factor ← factorCheck
    end if
    angle ← (angle ∗ factor)
    if after then
        threshAngleDegrees ← (angle + angleOfNode)
    else
        threshAngleDegrees ← (angleOfNode − angle)
    end if
    threshAngleRadians ← threshAngleDegrees / (180 ∗ PI)
    threshold.x ← circleCentre.x + (radius ∗ cos(threshAngleRadians))
    threshold.y ← circleCentre.y + (radius ∗ sin(threshAngleRadians))
    return Coord threshold
end function
```

seen in greater detail in Figure 6.1. This shows the MAC protocol Sleep, CCA and Send Preamble procedures as seen in Procedures 1, 2 and 3. Within each of the CCA and Send Preamble procedures the new *thresholdAdjust* function is utilised, as detailed in Function 1. The maximum threshold is set as the largest size the threshold could be when considering the node transmission range and distance to the path of the sink. A

Procedure 1 Sleep

```
procedure SLEEP
    set checkInterval from input                          ▷ Set the default sleep interval
    if significantNode then
        thresholdTime()    ▷ If MSN within threshold set thresholdReached otherwise set timeToThreshold as time until MSN reaches threshold
        if thresholdReached then
            interval ← checkInterval    ▷ The interval to wake-up reverts to the checkInterval
        else
            interval ← timeToThreshold    ▷ Set to the time it will take for the sink to reach the threshold
        end if
    else
        interval ← checkInterval
    end if
    schedule CCA at simTime + interval    ▷ simTime is current simulation time in seconds
end procedure
```

cross-layer solution is then implemented where the battery module is accessed after each pass of the sink node, in order to ascertain the current energy consumed. This is calculated on a per minute basis by utilising the simulation time and then compared to the target energy consumption per minute. By dividing the target energy consumption by the current energy level we create a factor by which to apply to the current size of the threshold. In this way, if energy levels are lower than the target the threshold will be increased and vice versa. Whilst nodes have no knowledge of neighbours, as the algorithm is implemented in all nodes in the network, it will be used in the same way by each significant node. Thus, it can be assumed that as one node reduces its threshold, another shall be increasing its own.

Procedure 2 CCA

```
procedure CCA
    if macQueue > 0 then
        withinThreshold()  ▷ Call withinThreshold() function to establish if the MSN is within the threshold currently
        if thresholdreached then
            macState ← SEND PREAMBLE                          ▷ Send preambles
            schedule STOP PREAMBLES at simTime + slotDuration    ▷ Schedule the stopping of preambles
        else
            macState ← SLEEP                          ▷ Return the macState to Sleep
            schedule SLEEP at simTime + slotDuration ▷ Node awakens with SLEEP Procedure at defined time
            thresholdAdjust()    ▷ Call thresholdAdjust() function to adjust the Threshold
        end if
    else
        macState ← SLEEP
        schedule SLEEP at simTime + slotDuration
    end if
end procedure
```

Procedure 3 Send Preamble

```
procedure SEND PREAMBLE
    withinThreshold()
    if thresholdreached then
        if SEND PREAMBLE then
            sendPreamble()
            macState ← SEND PREAMBLE
            schedule SEND PREAMBLE at simTime + (0.5 * checkInterval)
        else if STOP PREAMBLES then
            cancel SEND PREAMBLE
            macState ← SEND DATA                    ▷ Preambles over, send data
        end if
    else
        cancel SEND PREAMBLE/STOP PREAMBLES
        macState ← SLEEP
        schedule SLEEP at simTime + slotDuration
        thresholdAdjust()       ▷ Call thresholdAdjust() function to adjust the Threshold
    end if
end procedure
```

Procedure 1: The Sleep Procedure is utilised in the MADCAL algorithm to determine the move from Sleep to CCA based on the position of the sink node in relation to the communication threshold of the static node in which this takes place.

Procedure 2: The CCA Procedure is utilised initially in the MADCaDPAL algorithm to determine whether the node should return to the Sleep procedure or continue to sending preambles, based on the position of the sink node in relation to the communication threshold. Additionally now, for DMEAAL, as well as scheduling a return to the Sleep procedure, the *thresholdAdjust* function is called to determine of the size of the threshold should be adjusted and by how much. This function is only called if the sink node is not within the threshold anymore, as adjusting it whilst communication is taking place would be problematic.

Procedure 3: The Send Preamble Procedure is again utilised initially in the MADCaDPAL algorithm to determine whether the node should return to the Sleep procedure or continue in the process of sending preambles, based on the position of the sink node in relation to the communication threshold. Additionally now, for DMEAAL, as well as scheduling a return to the Sleep procedure, the *thresholdAdjust* function is again called to determine of the size of the threshold should be adjusted and by how much. Again, this function is only called if the sink node is not within the threshold anymore.

Function 1 Adjust Threshold

```
function THRESHOLDADJUST
    set targetEnergy from input
                        ▷ Cross-layer approach to utilise values within the battery module
    set batteryCapacity to nominalCapacity in batteryModule
    set batteryResidual to residualCapacity in batteryModule
    consumed ← (batteryCapacity − batteryResidual)    ▷ Current value of consumed energy
    if thresholdHappened then
        thresholdHappened ← False
        averageEnergy ← (consumed / simtime) × 60
        threshAdjust ← targetEnergy / averageEnergy
        angleLimitHigh ← maxAngle            ▷ As set when initial threshold established.
        angleLimitLow ← 0
        thresholdAfter ← true
sinkThresholdAfter ← adjustThresholdC(sinkRadius, thresholdAfter, threshAdjust)
        thresholdAfter ← false
sinkThresholdBefore ← adjustThresholdC(sinkRadius, thresholdAfter, threshAdjust)
        set thresholdDistance
        set beforeQuartile
        set thresholdOpposite
    end if
end function
function ADJUSTTHRESHOLDC(radius, after, adjustment)
    if after then
        angleAfter1 ← angleAfter × adjustment
    else
        angleBefore1 ← angleBefore × adjustment
    end if
    if after then
        if ((angleAfter1 < angleLimitHigh) and (angleAfter1 > angleLimitLow)) then
            angleAfter ← angleAfter1
        else
            if angleAfter1 >= angleLimitHigh then
                angleAfter ← angleLimitHigh
            else
                angleAfter ← angleLimitLow
            end if
        end if
    else
        if ((angleBefore1 < angleLimitHigh) and (angleBefore1 > angleLimitLow)) then
            angleBefore ← angleBefore1
        else
            if angleBefore1 >= angleLimitHigh then
                angleBefore ← angleLimitHigh
            else
                angleBefore ← angleLimitLow
            end if
        end if
    end if
    if after then
        threshAngleDegrees ← (angleAfter + angleOfNode)
    else
        threshAngleDegrees ← (angleOfNode − angleAfter)
    end if
    threshAngleRadians ← threshAngleDegrees / (180 ∗ PI)
    threshold.x ← circleCentre.x + (radius × cos(threshAngleRadians))
    threshold.y ← circleCentre.y + (radius × sin(threshAngleRadians))
    return Coord threshold
end function
```

Function 1: The *targetEnergy* is set from input as the desired average energy. In this case this is based on previous data when the MADCaDPAL algorithm is in use. The capacity of the battery is established in a cross-layer approach, directly from the battery module. This is then used in the *thresholdAdjust* function to calculate the current consumed energy by deleting the residual battery capacity from the actual battery capacity.

The purpose of the *thresholdHappened* variable is to establish if the threshold has been adjusted since the last time the sink reached it. There is no gain to be made from adjusting the threshold more than once in a circuit of the network as this could lead to this process recurring many times. The average energy per minute is calculated as is the threshold adjustment factor by simply dividing the target energy consumption by the current average energy consumption. We also set the highest and lowest values for the size of the angle of the threshold. In this case we use the *maxAngle* value as the highest and zero as the lowest. However, this may be adjusted according to a particular scenario. The *maxAngle* is set as the largest angle possible when calculating the initial threshold size at startup, based on transmission range and distance to sink path. The angle is then adjusted for before and after the threshold by multiplying it by the adjustment factor within the *adjustThresholdC* function. We then establish if the new angle is within the parameters of the highest and lowest angles and adjust accordingly if not. The angle is then calculated as coordinates and returned to be saved as the new threshold.

6.3 Evaluation and Results

6.3.1 Network Parameters

The network is again built using the OMNeT++ [77] simulation framework, utilising both MiXiM [87] and inetmanet [88].

As with the previous MADCAL and MADCaDPAL algorithms, within the network implementation, it is assumed that static nodes will retain their positions throughout and be aware of their own location. Each static node implements DMEAAL indepen-

Table 6.1: Network Parameters - DMEAAL

Parameters	Values
Number of Nodes (Static)	25
Grid Topology Size	200m * 200m
MSN Path Radius	150m
MSN Start Position	x=400m, y=250m
MSN Speed (metres per second)	20mps, 30mps, 40mps
Simulation Time	1884.95559215388s, 1256.63706143592s, 942.47779607694s
Number of Circuits of Network	40
Transmission Distance * 4	77.52m, 69.13m, 62.02, 55.94m
Number of Runs	5
Path-loss Alpha * 4	1.85, 1.9, 1.95, 2
Max Sending Power	1.0mW
Signal Attenuation Threshold	-85dBm
Sensitivity	-75dBm
Carrier Frequency	2.4GHz
Transmitter Power	1.0mW
Thermal Noise	-85dBm
Signal to Noise Ratio Threshold	4dB
Battery Capacity	594000 mWs
Check Interval	0.01s
Slot Duration	0.1s

dently and is unaware of neighbouring nodes. Transmission ranges remain consistent across all nodes in each test. The simulation times, however, now vary. Within this part of the study the emphasis is now to update the threshold in real-time, after each pass of the MSN. As such, in order to produce a fair and consistent comparison across the three different sink speeds, it is important to ensure that for each test scenario the node has the same number of opportunities to adjust its threshold. Therefore, simulation times are such that the MSN will complete 40 circuits of the network whatever speed it may be travelling at. As some scenarios are running for a longer time than when testing the MADCaDPAL algorithm, battery capacity has been increased accordingly. Transmission distance is calculated as before and the routing protocol remains OLSR. All other parameters can be found in Table 6.1.

Results show the effect of an MSN on significant nodes, these highlighted in Figures 5.3 and 5.4. Results are in two sections, with the first comparing the results of four different MAC implementations in order to demonstrate the effect of the DMEAAL algorithm on average energy consumption and frame delivery. The aim being that these should not be adversely affected by this new implementation. Each test scenario has been conducted using three different MAC implementations to establish the effect of a MSN at four different speeds and with four different transmission ranges across the network. Firstly, the existing lightweight MAC implementation using standard duty cycling techniques with no allowance for sink mobility is shown. Additionally, results are shown using both the MADCAL and MADCaDPAL algorithms. Finally the new DMEAAL algorithm is evaluated, with the aim now to balance energy consumption across significant nodes.

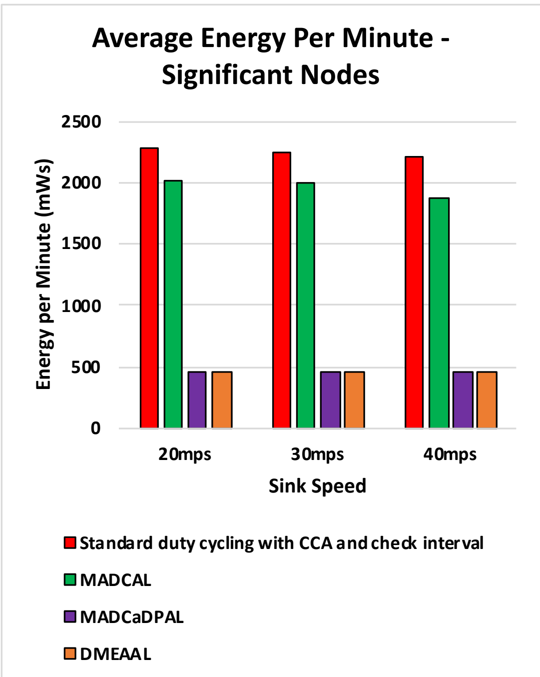
The second section of results demonstrate the energy balancing effect of the DMEAAL algorithm versus the results when MADCaDPAL is in operation. The intention here is to illustrate the difference in range of energy consumption when MADCaDPAL is in use to when the new DMEAAL algorithm is used. Tests are conducted using three different MSN speeds with four different transmission ranges across the network. Each test scenario aims to establish the effect of the algorithm when the MSN is travelling at 3 different speeds and with four different transmission ranges across the network. Of significant difference from tests conducted in the MADCAL and MADCaDPAL studies is that there is now no uniform test duration. The aim in these studies is to give each network scenario an equal opportunity to balance the energy consumption, therefore it is the number of network circuits of the MSN that is the same in each test, that being 40 times. Therefore, results are per minute.

6.3.2 Results - Average Energy Consumption and Frame Delivery, Grid Topology

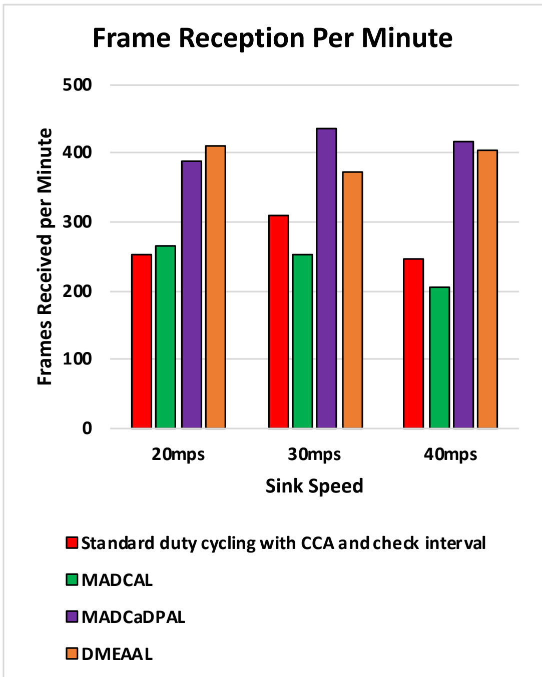
Figures 6.4-6.7 show the average energy consumption of all significant nodes for all four MAC implementations. This is presented alongside the frame delivery to the

MSN for each scenario. The purpose of these results is no longer to show improvement in energy consumption or frame delivery by any measurement. In this case these results are to show that implementing the DMEAAL algorithm does not adversely affect performance in these areas. In terms of average energy consumption, in particular, as much as possible we would seek for the results for the MADCaDPAL algorithm and those when DMEAAL is in use to be the same or very close. The subsequent results of energy balancing across significant nodes show the sought improvements that have been highlighted.

What is immediately clear from observing the average energy consumption levels in Figure 6.4a is that the same benefit to be gained from using the MADCaDPAL algorithm remains when using DMEAAL. The target of having energy consumption virtually the same is also seen to be achieved. In terms of frame delivery it can be seen that in some cases, balanced energy consumption may result in a slight decrease. This would be expected, with the possibility that neighbouring significant nodes may reduce their

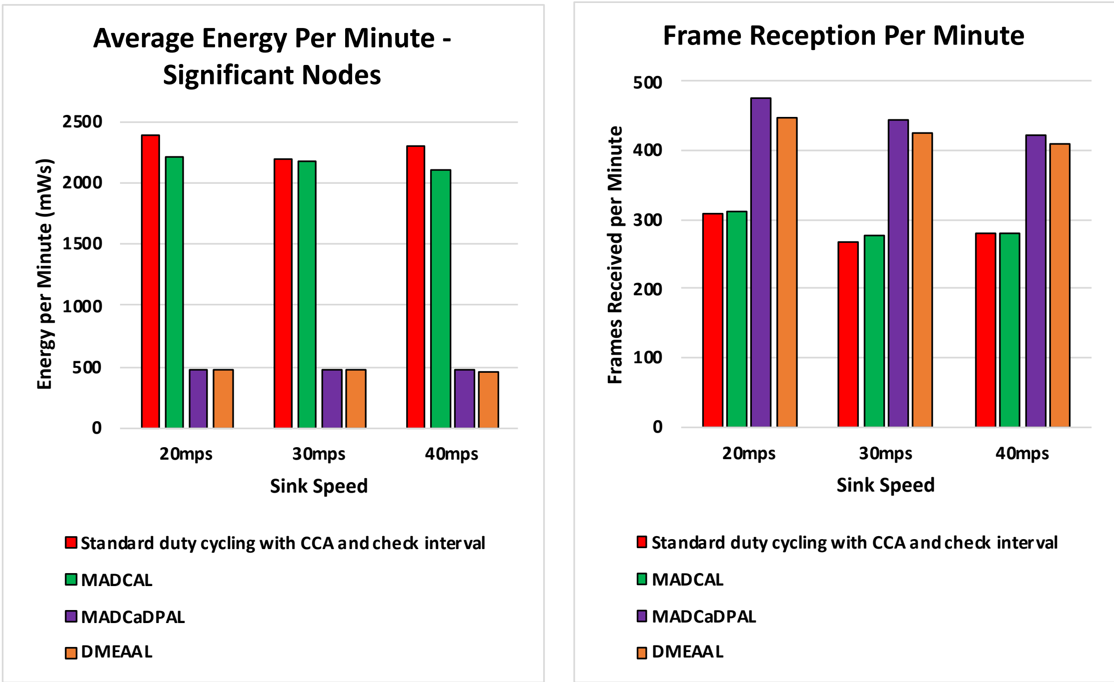


(a) Average Energy



(b) Frame Delivery

Figure 6.4: Average Energy Consumption of Significant Nodes (mWs) and Frame Delivery to Sink, Grid Topology. Transmission Range 77.52m. DMEAAL



(a) Average Energy

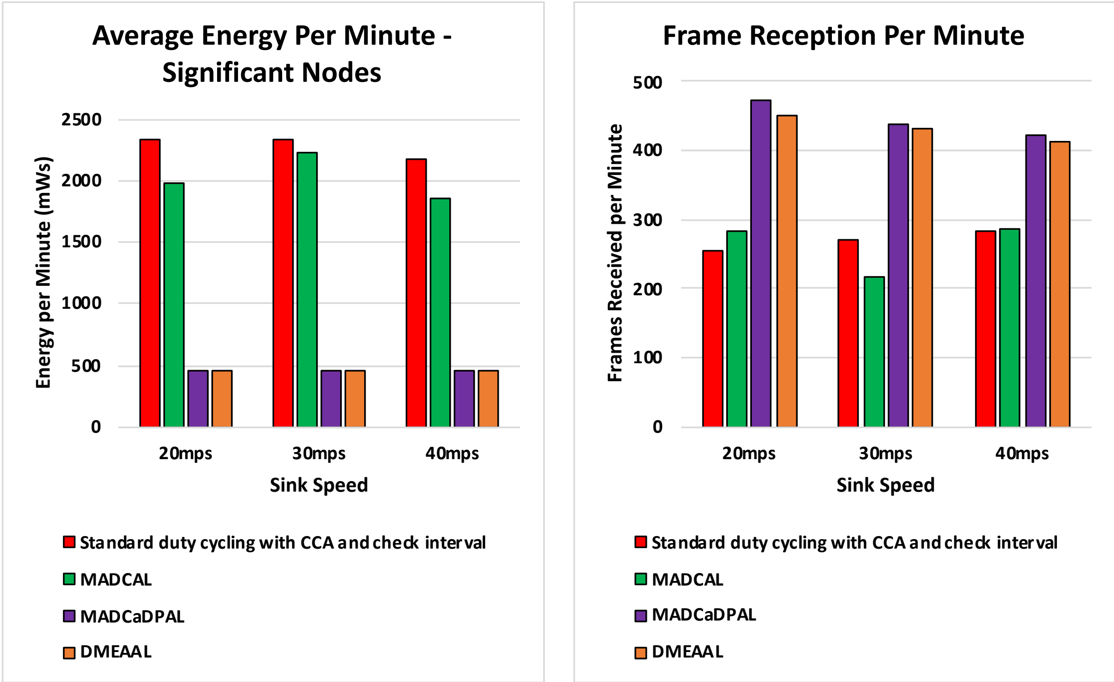
(b) Frame Delivery

Figure 6.5: Average Energy Consumption of Significant Nodes (mWs) and Frame Delivery to Sink, Grid Topology. Transmission Range 69.13m. DMEAAL

thresholds at the same time. Subsequently leaving less opportunity to deliver to the sink node, albeit temporarily. However, in all cases illustrated in in Figure 6.4b, frame delivery remains more efficient than when MADCAL and the original MAC implementation are in use.

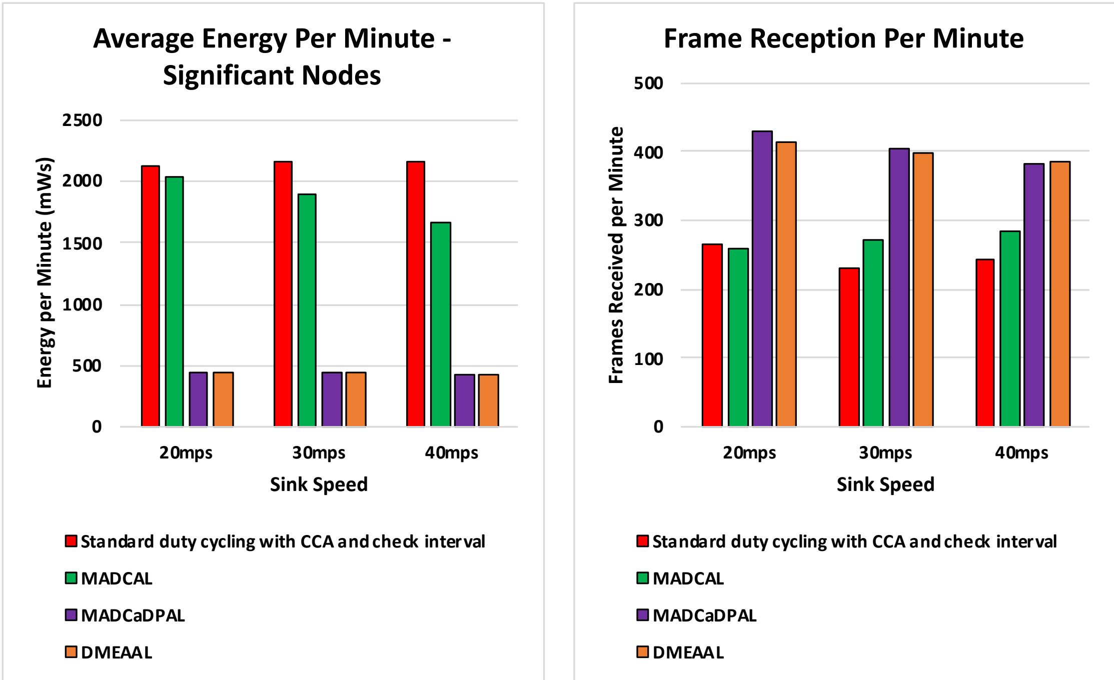
In Figure 6.5 it can be seen that this pattern repeats as the transmission range lessens. Average energy consumption remains at the same level, as seen in Figure 6.5a, but frame delivery can be seen to have decreased slightly in Figure 6.5b.

Again, when observing results in Figure 6.6, it can be observed that DMEAAL keeps average energy consumption at the same level, as can be seen in Figure 6.6a, with frame delivery only slightly less, as seen in Figure 6.6b. This even as the transmission range has constricted to just over 60m. Even as the transmission range lessens to just over 1 hop results remain much the same. as can be seen in Figure 6.7a with regard to energy consumption. In this case frame delivery is now within a small margin of MADCaDPAL,



(a) Average Energy **(b)** Frame Delivery

Figure 6.6: Average Energy Consumption of Significant Nodes (mWs) and Frame Delivery to Sink, Grid Topology. Transmission Range 62.02m. DMEAAL



(a) Average Energy **(b)** Frame Delivery

Figure 6.7: Average Energy Consumption of Significant Nodes (mWs) and Frame Delivery to Sink, Grid Topology. Transmission Range 55.94m. DMEAAL

as seen in Figure 6.7b.

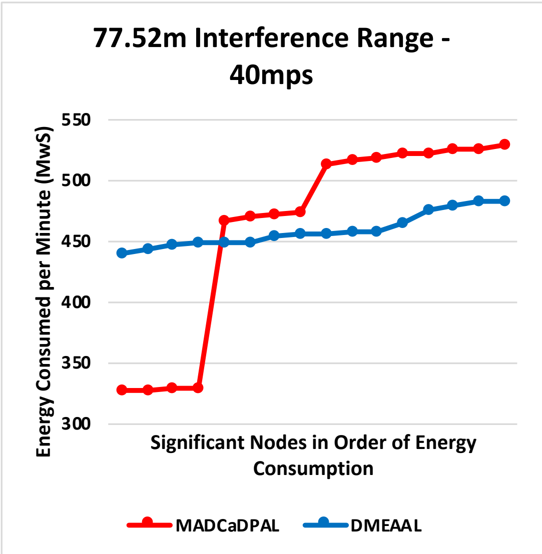
6.3.3 Results - Balanced Energy Consumption, Grid Topology

Figures 6.8-6.11 now demonstrate the ultimate result of the DMEAAL algorithm. For each different transmission range the results of each sink speed are shown with the energy consumption per minute of each significant node from lowest to highest. As such, this demonstrates the difference in energy consumption between when the MADCaDPAL algorithm is in use and the DMEAAL algorithm. The aim being for energy consumption across significant nodes to all be closer in value.

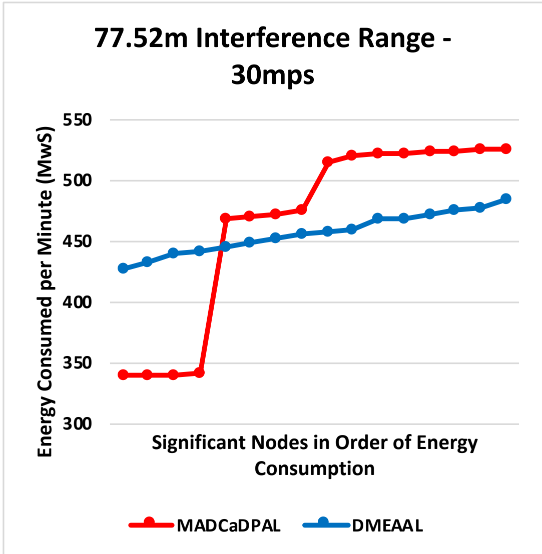
Figure 6.8 shows the results with DMEAAL in use for an transmission range of 77.52m. With the differing sink speeds shown of 40mps in Figure 6.8a, 30mps in Figure 6.8b and 20mps in Figure 6.8c. It can be clearly observed that with DMEAAL in use, the significant nodes all expend energy around the same level. As such, in all cases this brings down the energy consumption of 12 of the 16 significant nodes by utilising the excessive energy consumption of just 4. Despite how effective MADCaDPAL has been in reducing energy consumption, over an extended period of network time, a benefit such as this could have a major effect on network lifetime.

Figure 6.9 shows the results with DMEAAL in use for an transmission range of 69.13m. With the differing sink speeds shown of 40mps in Figure 6.9a, 30mps in Figure 6.9b and 20mps in Figure 6.9c. Again, similar results can be observed to the previous figure. The DMEAAL algorithm is clearly evening out energy consumption across significant nodes, unaffected by the smaller transmission range.

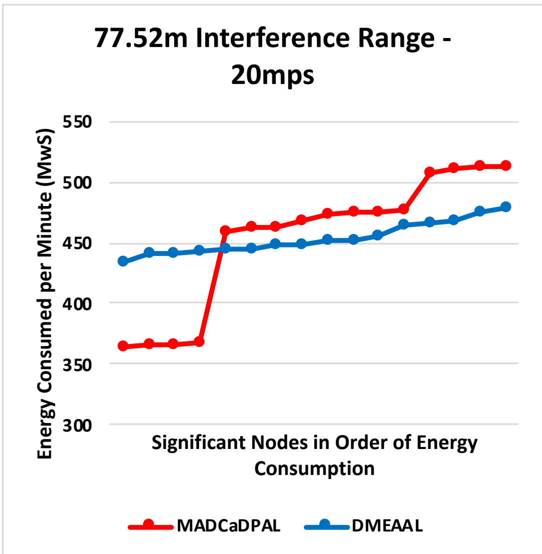
Figure 6.10 shows the results with DMEAAL in use for an transmission range of 62.02m. With the differing sink speeds shown of 40mps in Figure 6.10a, 30mps in Figure 6.10b and 20mps in Figure 6.10c. Even as the transmission range lessens this does not affect the results, with all significant nodes now much closer to the target energy consumption, that being the average energy consumed with the MADCaDPAL algorithm in use. This is repeated again in Figure 6.11 with an transmission range of



(a) 40mps

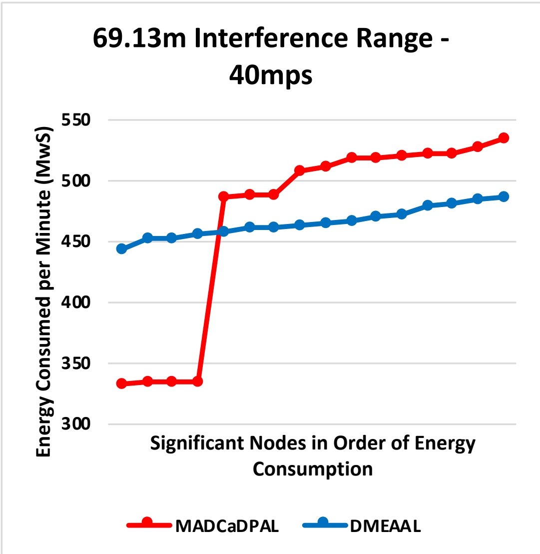


(b) 30mps

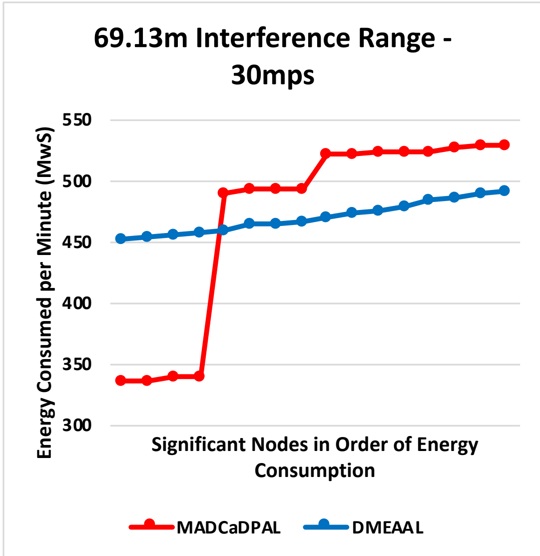


(c) 20mps

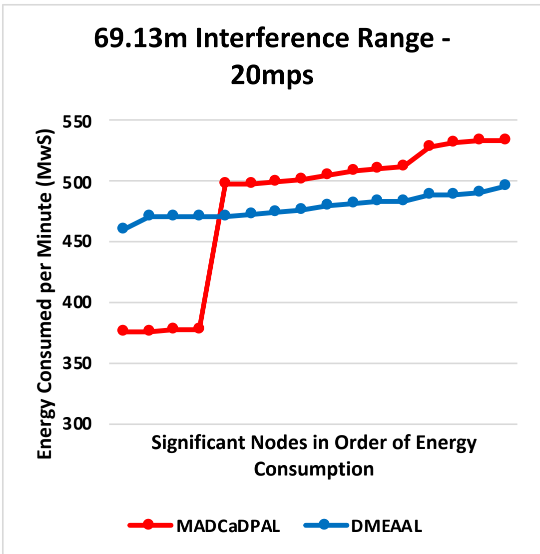
Figure 6.8: Balanced Energy Consumption of Significant Nodes (mWs), Grid Topology. Transmission Range 77.52m. DMEAAL



(a) 40mps



(b) 30mps



(c) 20mps

Figure 6.9: Balanced Energy Consumption of Significant Nodes (mWs), Grid Topology. Transmission Range 69.13m. DMEAAL

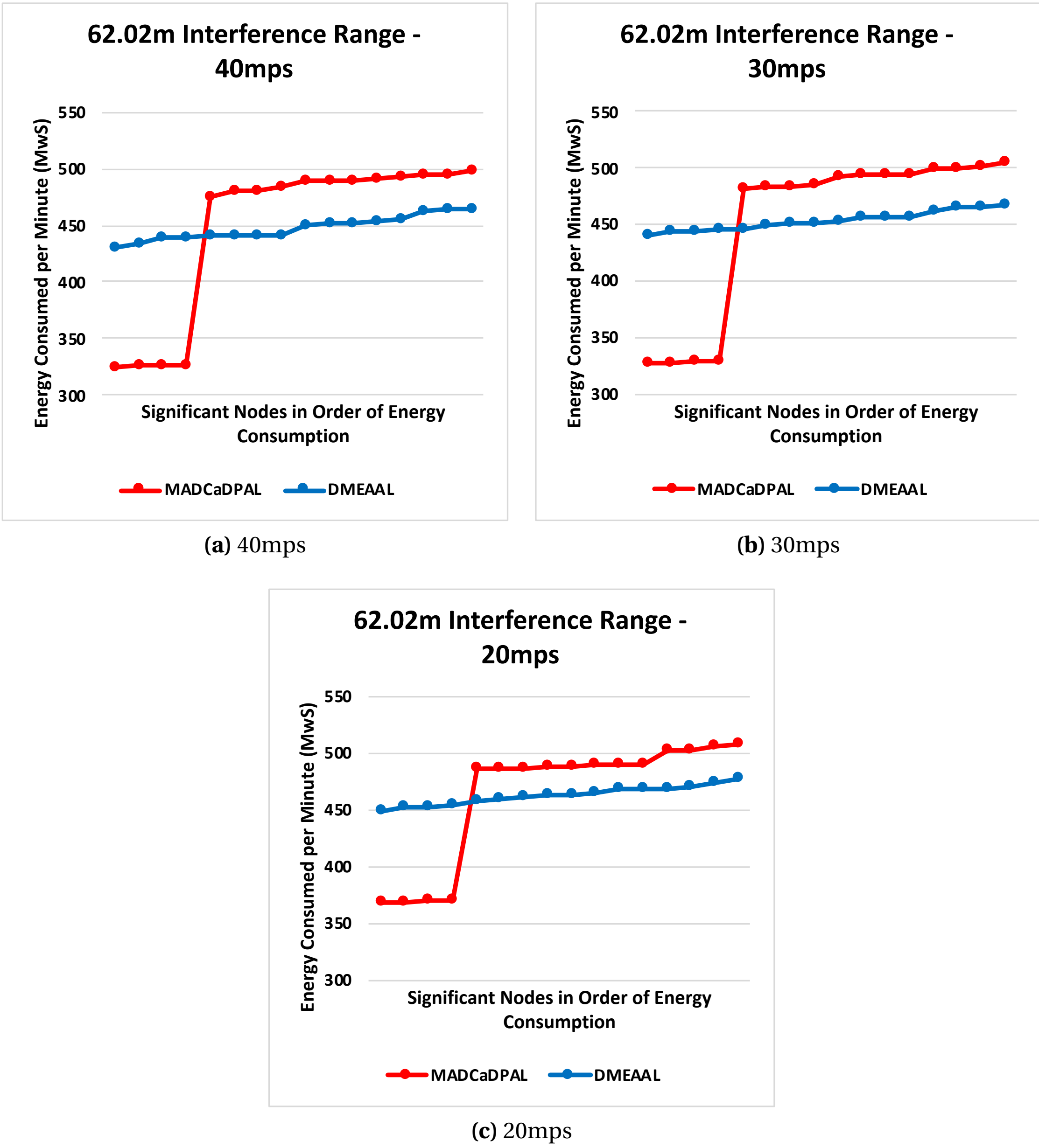
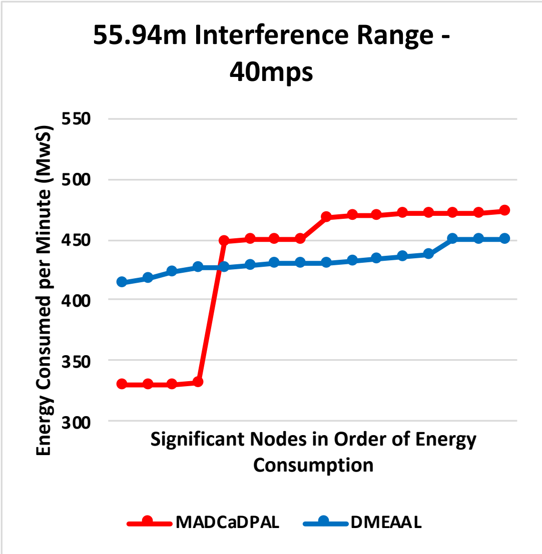
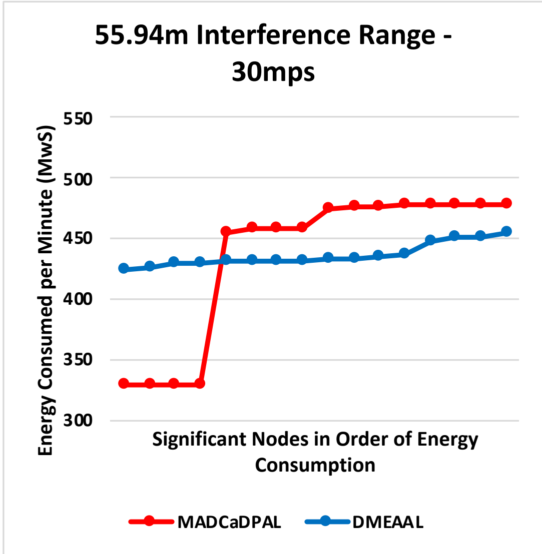


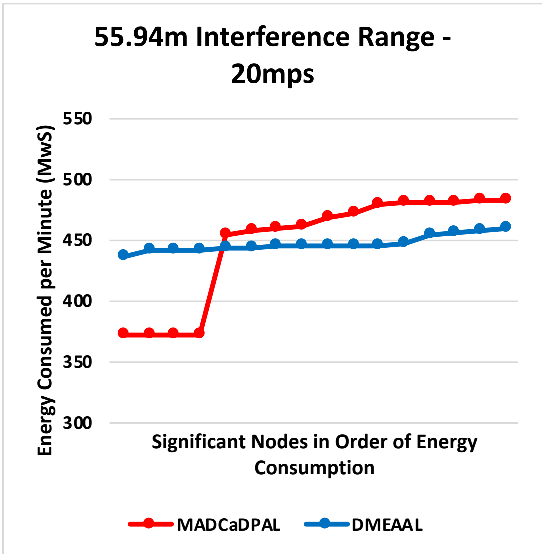
Figure 6.10: Balanced Energy Consumption of Significant Nodes (mWs), Grid Topology. Transmission Range 62.02m. DMEAAL



(a) 40mps



(b) 30mps



(c) 20mps

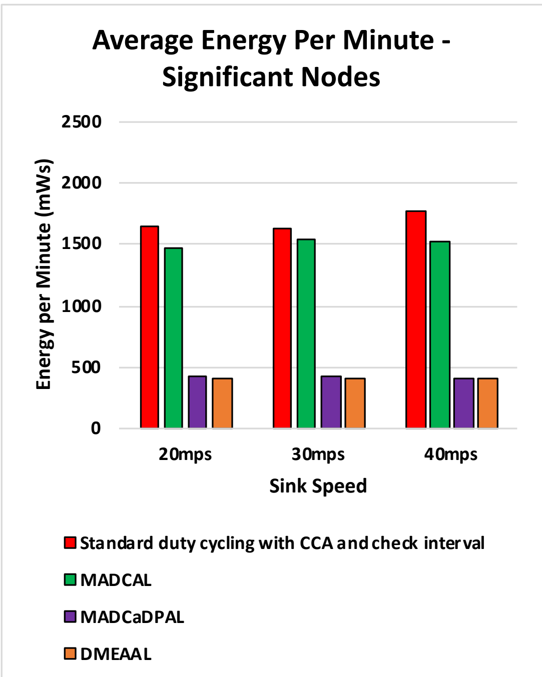
Figure 6.11: Balanced Energy Consumption of Significant Nodes (mWs), Grid Topology. Transmission Range 55.94m. DMEAAL

55.94m.

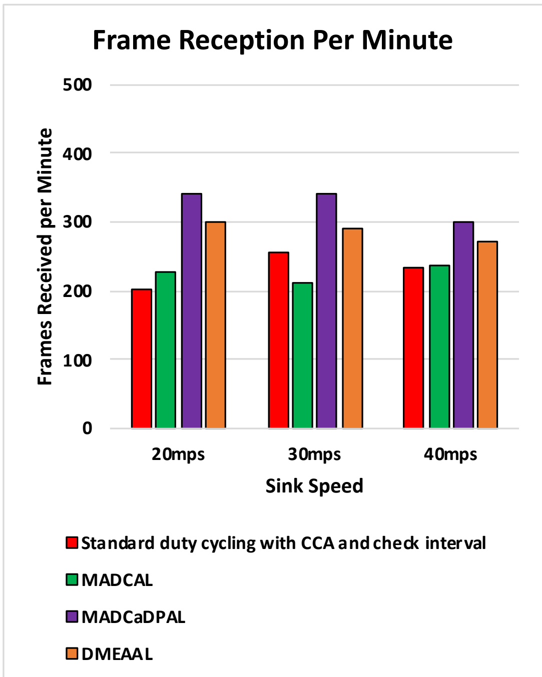
6.3.4 Results - Average Energy Consumption and Frame Delivery, Random Topology

Figures 6.12-6.15 show the average energy consumption of all significant nodes for all four MAC implementations. This is presented alongside the frame delivery to the MSN for each scenario. Again, we do not seek to show improvement in energy consumption or frame delivery by any measurement. In this case these results are to show that implementing the DMEAAL algorithm does not adversely affect performance in these areas, especially when considering a more strained topology. In terms of average energy consumption, in particular, as much as possible we would seek for the results for the MADCaDPAL algorithm and those when DMEAAL is in use to be the same or very close. With frame reception proving more problematic with this topology in use, it is also of interest to see if there has been an adverse effect on these results when using DMEAAL. However, the subsequent results of energy balancing across significant nodes remain the aim of this study.

It is clear, as with the grid topology, from observing the average energy consumption levels in Figure 6.12, that the same benefit to be gained from using the MADCaDPAL algorithm remains when using DMEAAL. The target of having energy consumption virtually the same is also seen to be achieved in Figure 6.12a. Again though, in terms of frame delivery, it can be seen in Figure 6.12b that balanced energy consumption



(a) Average Energy



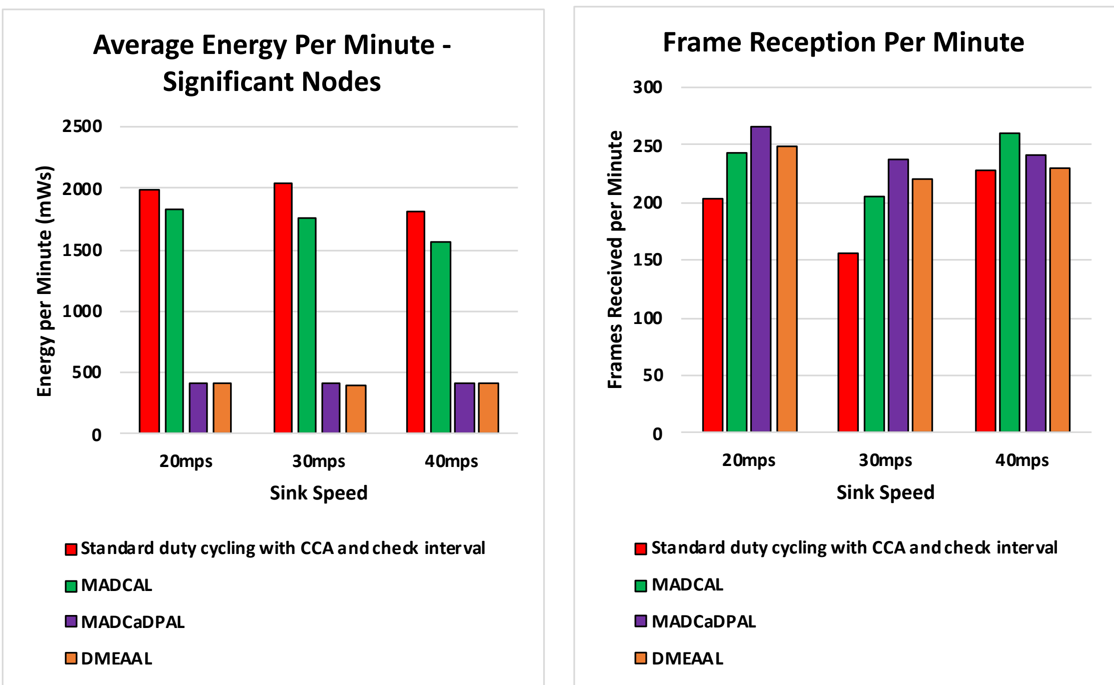
(b) Frame Delivery

Figure 6.12: Average Energy Consumption of Significant Nodes (mWs) and Frame Delivery to Sink, Random Topology. Transmission Range 77.52m. DMEAAL

may result in a slight decrease. However, in all cases illustrated frame delivery remains higher than when MADCAL and the original MAC implementation are in use.

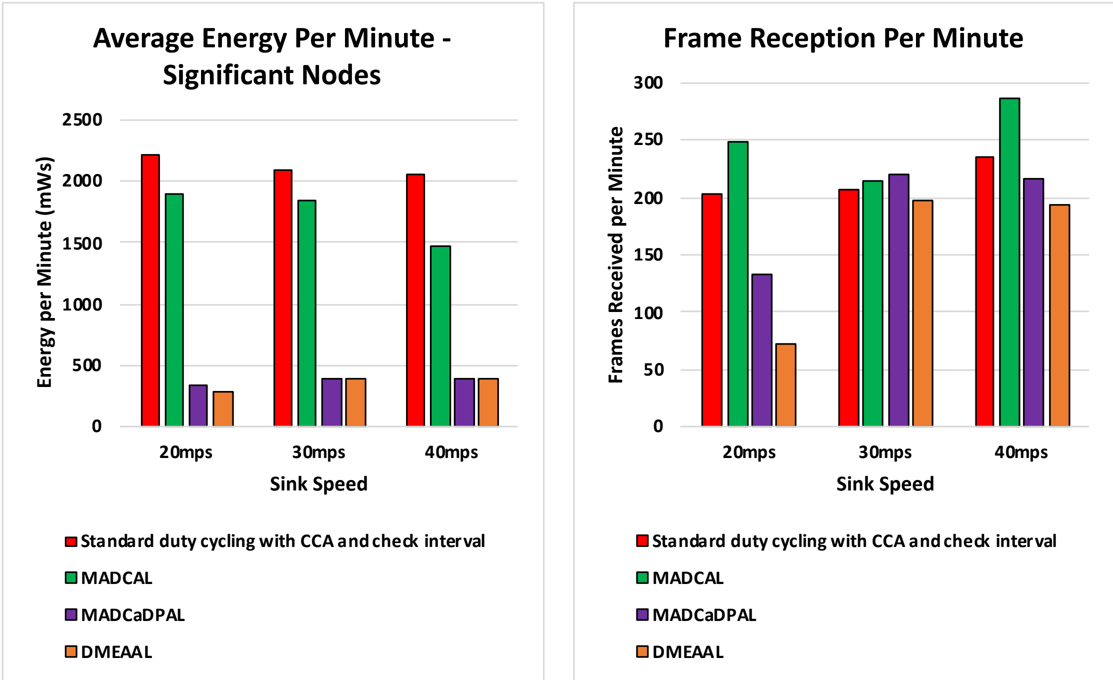
In Figure 6.13 it can be seen that this pattern repeats as the transmission range lessens. Average energy consumption remains at the same level, as seen in Figure 6.13a, but frame delivery is slightly less as seen in Figure 6.13b.

When observing results in Figure 6.14 it can again be seen that DMEAAL keeps average energy consumption at the same level, as seen in Figure 6.14a, with this also true for Figure 6.15a. This is significant as the transmission range lessens, now to just above the distance of one-hop. When considering frame delivery, this constriction of transmission range also proved problematic for the MADCaDPAL algorithm. The topology in use has clusters of nodes but also large spaces between them. As such, attempts to control energy consumption and a reduction in communication thresholds may result in a slight reduction in frame delivery. This repeats with the use of DMEAAL,



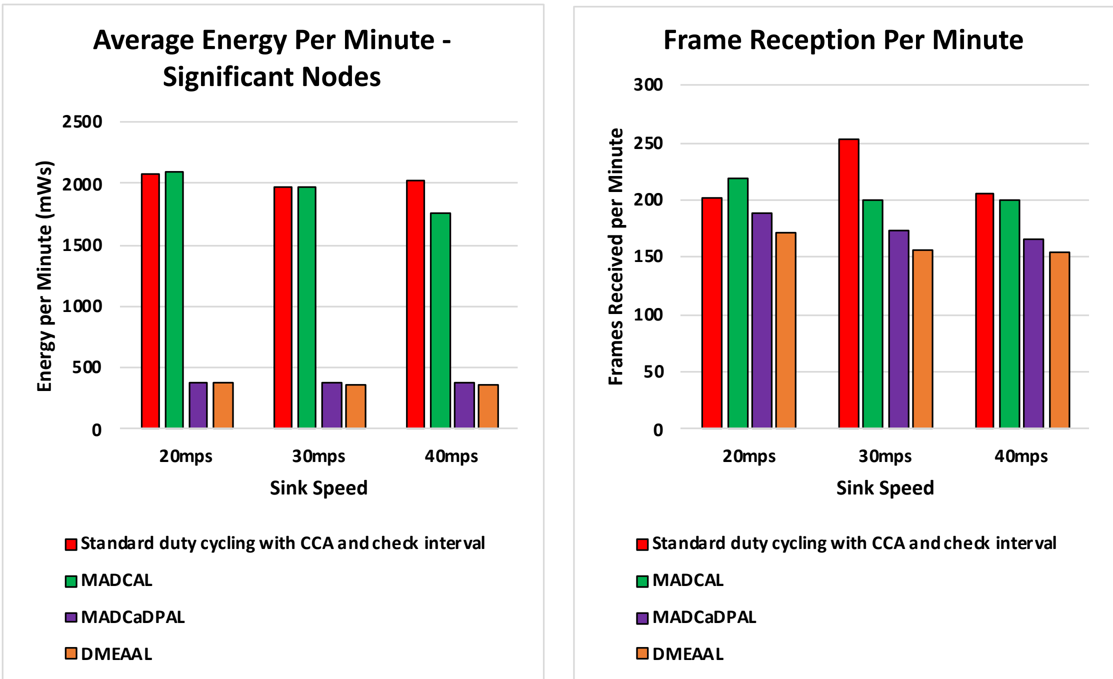
(a) Average Energy **(b)** Frame Delivery

Figure 6.13: Average Energy Consumption of Significant Nodes (mWs) and Frame Delivery to Sink, Random Topology. Transmission Range 69.13m. DMEAAL



(a) Average Energy **(b)** Frame Delivery

Figure 6.14: Average Energy Consumption of Significant Nodes (mWs) and Frame Delivery to Sink, Random Topology. Transmission Range 62.02m. DMEAAL



(a) Average Energy **(b)** Frame Delivery

Figure 6.15: Average Energy Consumption of Significant Nodes (mWs) and Frame Delivery to Sink. Transmission Range 55.94m. DMEAAL

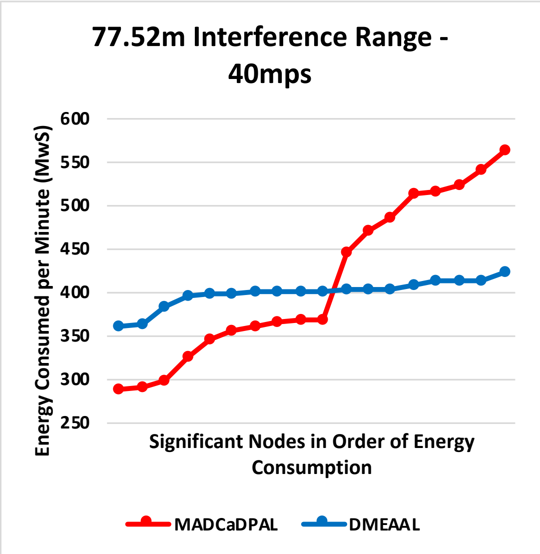
although not excessively aside from one result when the sink moves at 20mps with a 62.02m transmission range, as seen in Figure 6.14b. This combination clearly results in a threshold that is not effective in this environment. As such, this highlights the need for a certain amount of control in the density of node placement.

6.3.5 Results - Balanced Energy Consumption, Random Topology

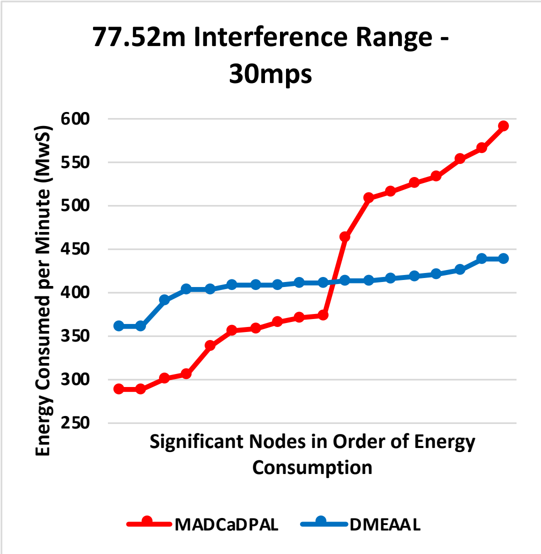
Figures 6.16-6.19 illustrate the results of the ultimate aim of the DMEAAL algorithm, that being to balance energy consumption across significant nodes. For each different transmission range the results of each sink speed are shown with the energy consumption per minute of each significant node from lowest to highest. Different from the use of the grid topology is that as the transmission range constricts, fewer nodes take the role of significant node. As before, however, results demonstrate the difference in energy consumption between when the MADCaDPAL algorithm is in use and the DMEAAL algorithm. The aim being for energy consumption across significant nodes to all be closer in value.

Figure 6.16 shows the results with DMEAAL in use for an transmission range of 77.52m. With the differing sink speeds shown of 40mps in Figure 6.16a, 30mps in Figure 6.16b and 20mps in Figure 6.16c. It can be clearly observed that with DMEAAL in use, energy consumption is spread more evenly across significant nodes. What can be observed in contrast to the grid topology, is that with MADCaDPAL in use the spread of energy is even more stark. As such, the potential to increase network lifetime is even greater and DMEAAL succeeds in this.

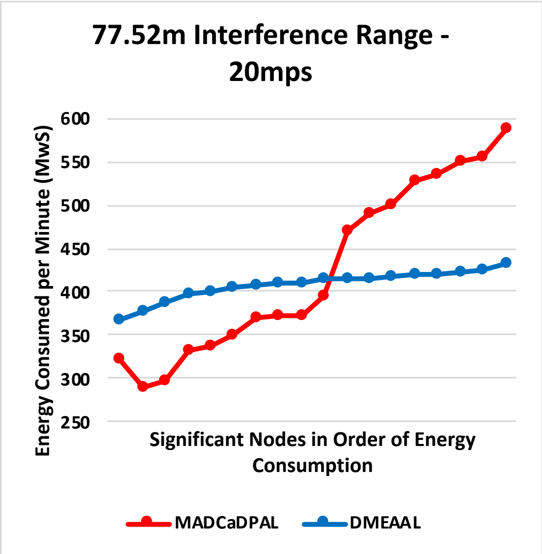
Figure 6.17 shows the results with DMEAAL in use for an transmission range of 69.13m. With the differing sink speeds shown of 40mps in Figure 6.17a, 30mps in Figure 6.17b and 20mps in Figure 6.17c. Again, similar results can be observed to the previous figure, with DMEAAL effective in evening out energy consumption across significant nodes. Despite the smaller transmission range.



(a) 40mps

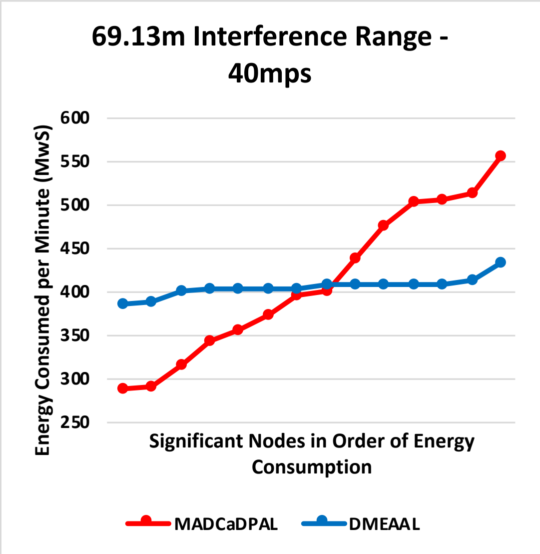


(b) 30mps

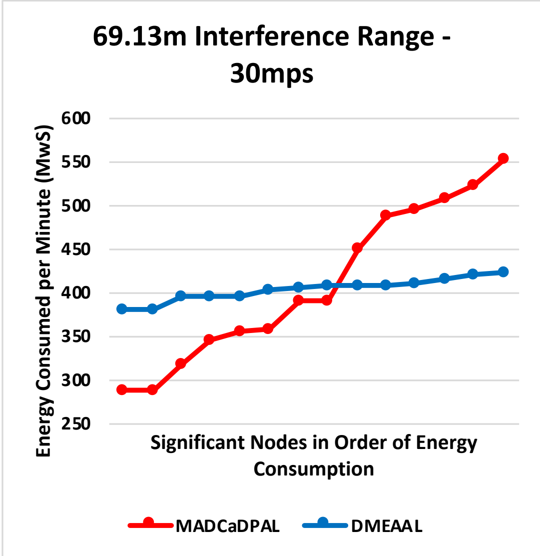


(c) 20mps

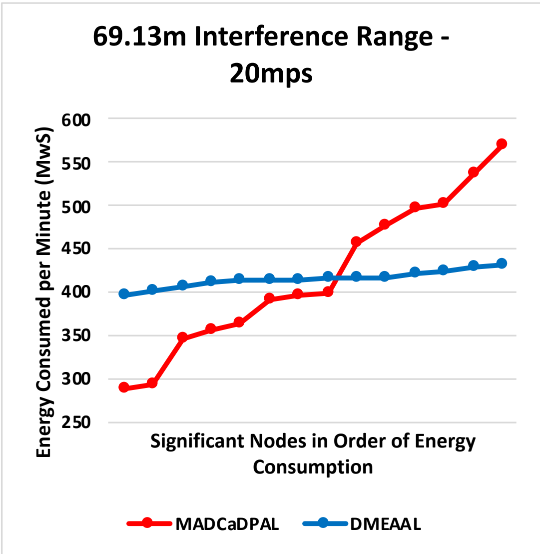
Figure 6.16: Balanced Energy Consumption of Significant Nodes (mWs), Random Topology. Transmission Range 77.52m. DMEAAL



(a) 40mps



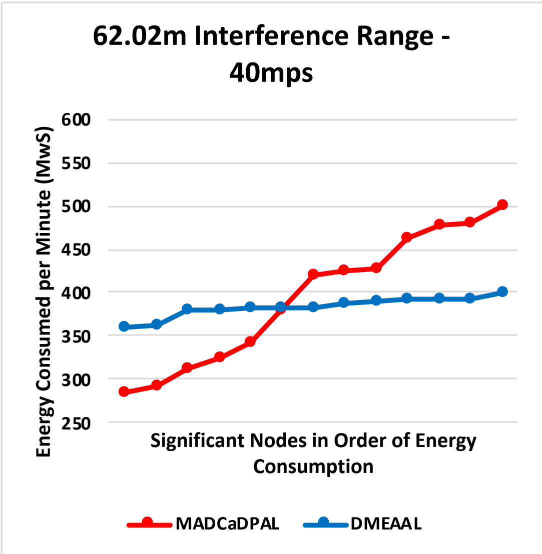
(b) 30mps



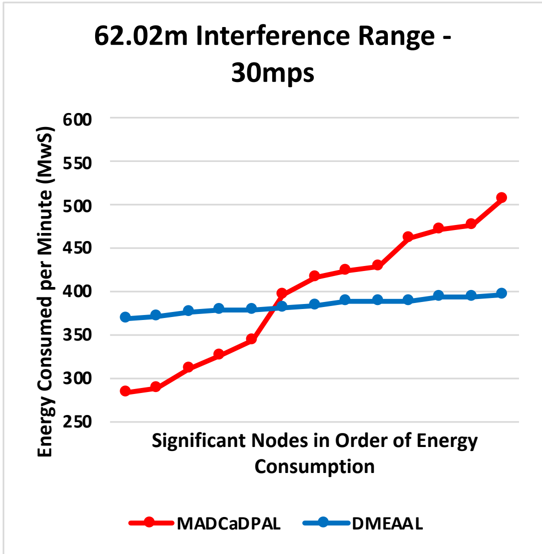
(c) 20mps

Figure 6.17: Balanced Energy Consumption of Significant Nodes (mWs), Random Topology. Transmission Range 69.13m. DMEAAL

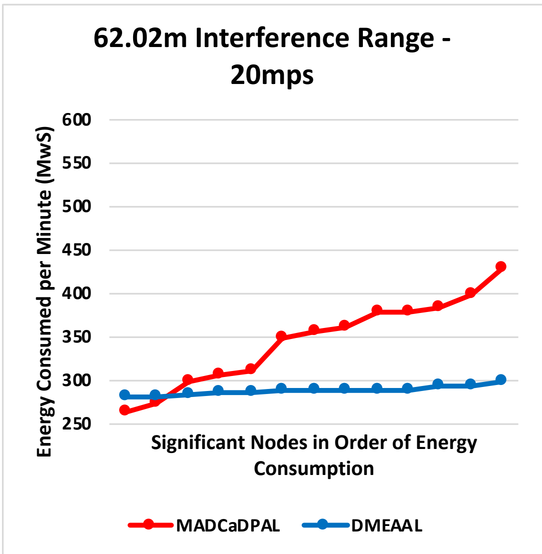
Figure 6.18 shows the results with DMEAAL in use for an transmission range of 62.02m. With the differing sink speeds shown of 40mps in Figure 6.18a, 30mps in Figure 6.18b and 20mps in Figure 6.18c. Results again follow a similar pattern. However, Figure 6.18c showing a sink speed of 20mps can be largely discounted due to the low level of frame delivery in this scenario. In this case it would have to be accepted that the MADCaDPAL algorithm is more effective. However, even in the much smaller transmis-



(a) 40mps

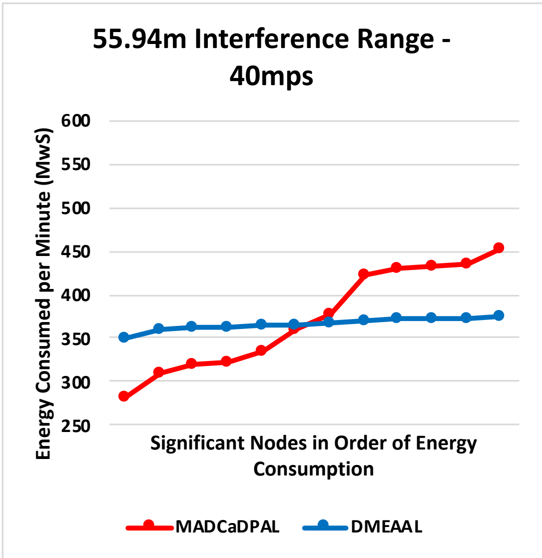


(b) 30mps

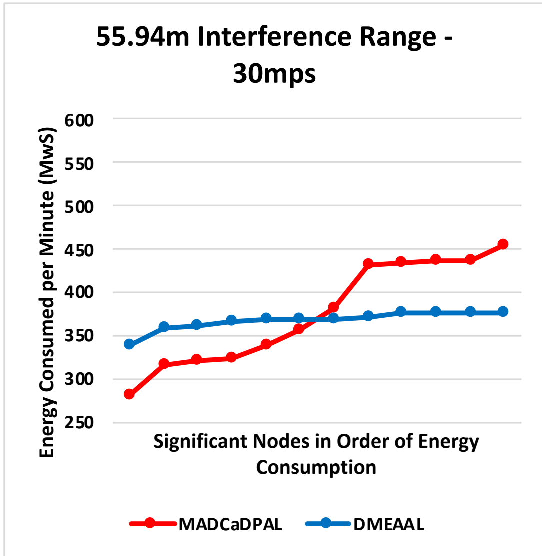


(c) 20mps

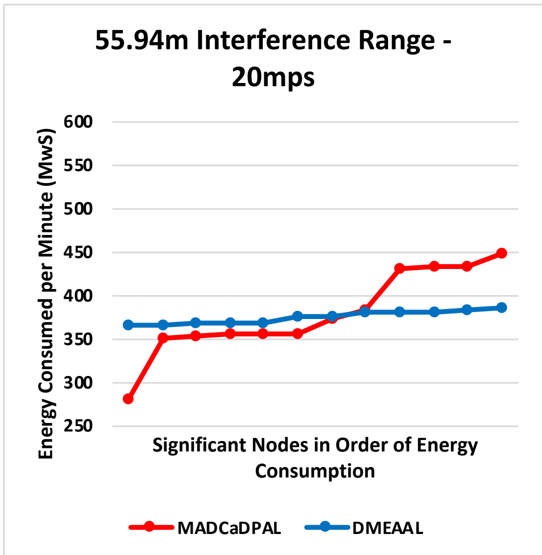
Figure 6.18: Balanced Energy Consumption of Significant Nodes (mWs), Random Topology. Transmission Range 62.02m. DMEAAL



(a) 40mps



(b) 30mps



(c) 20mps

Figure 6.19: Balanced Energy Consumption of Significant Nodes (mWs), Random Topology. Transmission Range 55.94m. DMEAAL

sion range in Figure 6.19, the effectiveness of the DMEAAL algorithm can be observed.

6.4 Conclusion

This chapter proposes DMEAAL, for use with MSNs utilising a MSN. This algorithm utilises predictable sink mobility and self knowledge of location amongst static nodes. This in order that energy consumption across significant nodes may be balanced and that *dead* energy consumption may be reduced. Results show that the improvements the MADCaDPAL algorithm achieves in terms of energy consumption and frame delivery to the sink, are generally retained. Additionally, it is demonstrated that a cross-layer approach, determining the current battery levels within a node along with knowledge of target energy consumption, results in balanced energy consumption across significant nodes. Thus improving network lifetime and finally eliminating energy spikes which contribute to the *energy hole* [17] or *hotspot* problem. Using results of the MADCaDPAL algorithm as a target level of energy consumption and altering the communication threshold accordingly, this is now an entirely dynamic solution. In consideration of frame delivery when a controlled, grid network formation is used, results remain largely similar to when MADCaDPAL is in use. This remains when a more *strained*, random topology is utilised, aside from one scenario.

Seventh Chapter

Conclusion and Future Work

7.1 Overview

The aim of this thesis is to demonstrate that knowledge of sink mobility may be utilised as a network metric in order to positively affect duty cycling in static nodes. It is demonstrated that this is a completely novel approach, with a gap in literature demonstrating sink node mobility to not have been used in this way before. As such, the development of a communication threshold between the sink and static node is initially shown of benefit in terms of energy consumption and frame delivery. Subsequently, it is demonstrated that this threshold may be further utilised to influence MAC layer performance to the point where energy consumption is improved by up to 80%. It has also been shown that the threshold may be utilised along with target energy consumption to even out energy consumption across multiple *significant* nodes, extending network lifetime. Consequently, this project has been a success in proving the benefit in predictable sink mobility and providing a base for further study in this area.

7.2 Summary of Work

This thesis outlines and discusses the state-of-the-art in ND in WSNs, in particular with regard to mobility. Highlighting the issue of node duty cycle overlaps being required in order to guarantee ND may take place, with mobility adding yet another layer of complexity in this regard. This forms the basis of this study, with literature review illustrating the journey ND has taken from a static approach in legacy networks, through utilising probabilistic and deterministic approaches in WSNs, both static and with mobility in use, toward new approaches such as opportunistic and those implementing MA. It is in the area of MA where the first potential gap is identified, in that knowledge of mobility may be used to directly influence duty cycling. It is determined that in this way, sink mobility would be an ideal candidate for this use.

The ultimate conclusion of the review of literature is that sink mobility may be used to influence duty cycling in a WSN. Hence, this thesis proposes three novel CSMA/CA based MAC algorithms to utilise a MSN with predictable mobility pattern in this way. As such, the proposed techniques and resultant algorithms answer the research questions posed in the introduction to this thesis (Section 1.3). These can be detailed as follows:

- Chapter 3, which details the testbed used, and Chapter 4, which details the initial MADCAL algorithm, successfully answer the first research question:

  - To what extent can the mobility pattern of a MSN in a WSN be exploited at the MAC layer to influence the performance of static nodes, in order to increase efficiency in terms of network performance and energy consumption?

  It is clear that when there is knowledge of the start time, sink speed, starting position, starting angle and pattern to be followed, in this case circular, that a static node may then calculate the current sink position and base its duty cycle on this. It is shown in Chapter 4 that this can be beneficial in terms of energy consumption.

- Whilst he initial development of a communication threshold occurs in Chapter 4, it is in Chapter 5 that the second research question is truly answered:

  - To what extent can a threshold built in a static node be utilised for communication with a MSN in order that said communication only occurs within the threshold?

Utilising the threshold in conjunction with knowledge of the sink mobility to ensure the node awakens when the sink is within the threshold occurs in MADCAL in Chapter 4. Adjusting for interference range, distance from sink path and, significantly, the speed of the sink. However, this is then shown to have even greater benefit in Chapter 5 with the development of the MADCaDPAL algorithm. Using the sink position in relation to the threshold to influence CCA and the sending of preambles and, subsequently, showing a large decrease in energy consumption of up to 80%. Effectively demonstrating the extent to which the threshold may be utilised to improve network performance and answering the third research question:

  – To what extent can network performance and energy consumption be improved by this approach?

- The development of the DMEAAL algorithm in Chapter 6 demonstrates that a cross-layer approach to energy consumption is possible and can show benefit. Subsequently answering the fourth and final research question:

  – To what extent can nodes in a WSN utilising a MSN be context-aware? Such that said context, such as the energy levels of nodes, may be utilised accordingly to adjust a communication threshold and ensure even energy consumption across nodes. As such, improving network lifetime.

With the development of the communication threshold complete, it is shown that where access to current battery consumption is available, the threshold may be continually adjusted based on a target energy consumption level. In this event, energy consumption may be balanced across all significant nodes, removing any remaining energy spikes, however small. The ultimate result is that network lifetime is improved. This is also important in demonstrating the differing potential uses for a communication threshold. With, ultimately, the MADCaDPAL and DMEAAL algorithms linked in that they utilise the threshold, but providing different solutions depending on the particular WSN.

7.3 Quantitative Review

1. **MADCAL:** In comparison to MAC implementation with no allowance made for the sink node being mobile, MADCAL reduces energy consumption in all significant nodes and in each scenario, by an average of 11.61% across all scenarios in the grid topology, with a maximum improvement of 20.2%. In the random topology energy consumption is again reduced in every significant node in each scenario, this time by an average of 13.86% and a maximum of 22.02%. In consideration of frame delivery to the MSN, in the grid topology this is improved in all but two scenarios with an average improvement of 27%. The maximum improvement for a single scenario is 401.77%. In the random topology the average improvement across all scenarios is 23.1% with the maximum improvement 374.73%. Again with frame delivery improved in all but two scenarios.

2. **MADCaDPAL:** In comparison to MAC implementation with no allowance made for the sink node being mobile, MADCaDPAL further reduces energy consumption in all significant nodes and in each scenario, by an average of 78.04% across all scenarios in the grid topology, with a maximum improvement of 80.07%. In the random topology energy consumption is again reduced in every significant node in each scenario, this time by an average of 78.03% and a maximum of 81.5%. In consideration of frame delivery to the MSN, in the grid topology this is improved in all scenarios with an average improvement of 43.96%. The maximum improvement for a single scenario is 571.26%. In the random topology the average improvement across all scenarios is 18.22% with the maximum improvement 137.52%. With frame delivery improved in all but two scenarios, but less so than when MADCAL is in use, with the caveat of the considerable reduction in energy consumption with MADCaDPAL in use. In addition the difference between average energy consumption and the maximum consumption of a single node closed alongside the considerable reduction in energy overall.

3. **DMEAAL:** With average energy consumption kept the same, or within a reason-

able amount, the focus is on balancing energy consumption. Working on the basis of energy consumption per minute, taking all test scenarios together, the difference in average energy consumption per minute between highest and lowest significant node for MADCaDPAL is 165.61 mWs. Whereas for DMEAAL the difference is 37.04 mWs. An improvement of 77.64%.

7.4 Challenges and Future Work

The aims and objectives of this thesis have been met and research questions answered, however, in performing this study further challenges arise. The main focus of this study was to prove that with knowledge of a sink mobility pattern, whatever that pattern may be, that duty cycling may be positively influenced. Hence, the mobility pattern utilised in this study is a circular, two-dimensional pattern. Given that the most common application referred to for MSNs is that of UAVs, or drones [20], it should be assumed that for this work to applied practically a third dimension would have to be applied in the calculation of the sink path. Even in the event of again utilising a circular path, the height of the UAV would also have to be taken into account. Taking the sink mobility pattern into account also, as has been illustrated in literature review, many studies seek to implement an optimal path for the sink node. It is envisaged that the work in this thesis could be combined with such a method at a future date. As has been stated, the development of MADCAL, MADCaDPAL and DMEAAL is dependant upon a predictable sink mobility pattern. The pattern itself is ultimately incidental.

With the primary focus of this study on duty cycling, the MAC layer is where work takes place. As has been stated throughout, the use of the OLSR network layer protocol is merely to facilitate delivery to the sink, with the high load placed on the network by this protocol aiding in accelerating testing, with nodes expending energy at a greater rate. However, moving forward it is accepted that for the work in this study to be implemented in a practical sense, that integration with the network layer must take place to some degree. In particular, when considering WSNs specifically, consideration must be given to the standardised RPL routing protocol [94]. With no standardisation currently for the use of RPL in an environment with a MSN and little research to be found in this area, integrating this routing protocol with the approach in this thesis would appear to be of interest for future work.

Another important factor within this thesis is that of the network topology. The use of MSNs very much depends upon the particular application. As has been stated, WSNs may be placed in inhospitable environments and sink mobility may be constrained by that environment. The network size utilised in this study is sufficient to demonstrate the effectiveness of the proposed algorithms. However, a topology of greater density in terms of number of nodes will have several factors to be determined such as whether multiple sink nodes are feasible and whether these should all be mobile? Clustering may also become a factor at some point. This would raise the possibility of multiple MSNs with differing responsibilities within the network.

Finally, in implementing the new DMEAAL algorithm in Chapter 6, a target energy consumption is utilised. In future, it is envisaged that the calculation of this target energy consumption may form part of the network infrastructure. Consequently, this new algorithm shows great potential to be a practical solution for edge-computing applications with a mobile sink node. Regarding this, two possibilities are envisaged:

1. Individual nodes send current battery levels to the sink node via a message. The sink node may then calculate the average energy consumption across all significant nodes and send this figure back to each node. The nodes then implement the DMEAAL algorithm, using the figure received from the sink as the target energy consumption. This solution would be completely integrated in network operation. However, additional messaging is required, adding to network overhead and potential delay. The target energy consumption would also only be based on one network run. Whereas the target energy consumption utilised in the tests detailed in this study are derived from five tests combined. This would raise the possibility

of an outlying result influencing the calculation.

2. A second application would see target energy consumption pre-programmed in the sink node. This based on multiple tests, removing the risk of outlying results. This figure may then be sent to each node as target energy consumption, with the DMEAAL algorithm implemented. This solution requires that only the sink sends a message to each node, reducing extra load on the network. Whilst not completely integrated, this solution removes the need for each node to be pre-programmed with target energy consumption. However, there remains the possibility of data loss and delay, with significant nodes unable to implement the DMEAAL algorithm until the message is received.

References

[1] J.-P. Vasseur and A. Dunkels, "Communication Mechanisms for Smart Objects," in *Interconnecting Smart Objects with IP*, 2010, ch. 12, pp. 147–165. DOI: 10.1016/B978-0-12-375165-2.00012-0 (cit. on p. 1).

[2] Scottish Power, *Smart Meters.* [Online]. Available: https://www.scottishpower.co.uk/energy-efficiency/smart-meters/ (visited on 03/03/2016) (cit. on pp. 1, 11).

[3] Department of Energy & Climate Change, *Smart meters: a guide*, 2013. [Online]. Available: https://www.gov.uk/guidance/smart-meters-how-they-work (visited on 03/03/2016) (cit. on pp. 1, 11).

[4] Amazon, *Amazon Echo - Alexa Voice Service*, 2017. [Online]. Available: https://www.amazon.co.uk (visited on 02/07/2017) (cit. on pp. 1, 11).

[5] A. Arroub, B. Zahi, E. Sabir, and M. Sadik, "A literature review on Smart Cities: Paradigms, opportunities and open problems," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, IEEE, Oct. 2016, pp. 180–186, ISBN: 978-1-5090-3837-4. DOI: 10.1109/WINCOM.2016.7777211 (cit. on pp. 1, 11).

[6] E. Ismagilova, L. Hughes, Y. K. Dwivedi, and K. R. Raman, "Smart cities: Advances in research—An information systems perspective," *International Journal of Information Management*, vol. 47, pp. 88–100, Aug. 2019, ISSN: 02684012. DOI: 10.1016/j.ijinfomgt.2019.01.004 (cit. on pp. 1, 11).

[7] H. Kumar, M. K. Singh, M. P. Gupta, and J. Madaan, “Moving towards smart cities: Solutions that lead to the Smart City Transformation Framework,” *Technological Forecasting and Social Change*, vol. 153, p. 119 281, Apr. 2020, ISSN: 00401625. DOI: 10.1016/j.techfore.2018.04.024 (cit. on pp. 1, 11).

[8] *Robot Vehicles Make Contactless Deliveries Amid Coronavirus Quarantine - IEEE Spectrum*. [Online]. Available: https://spectrum.ieee.org/automaton/transportation/self-driving/robot-vehicles-make-contactless-deliveries-amid-coronavirus-quarantine (visited on 04/06/2020) (cit. on p. 1).

[9] 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, ISSN: 1932-8184. DOI: 10.1109/JSYST.2016.2597171 (cit. on p. 2).

[10] 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, ISSN: 1936-6442. DOI: 10.1007/s12083-015-0421-4 (cit. on p. 2).

[11] 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, ISSN: 14248220. DOI: 10.3390/s18020555 (cit. on p. 2).

[12] D. R. Vincent, N. Deepa, D. Elavarasan, K. Srinivasan, S. H. Chauhdary, and C. Iwendi, “Sensors Driven AI-Based Agriculture Recommendation Model for Assessing Land Suitability,” *Sensors*, vol. 19, no. 17, p. 3667, Aug. 2019, ISSN: 1424-8220. DOI: 10.3390/s19173667. [Online]. Available: https://www.mdpi.com/1424-8220/19/17/3667 (cit. on p. 2).

[13] M. Kim, S. Park, and W. Lee, “Energy and Distance-Aware Hopping Sensor Relocation for Wireless Sensor Networks,” *Sensors*, vol. 19, no. 7, p. 1567, Apr. 2019, ISSN:

1424-8220. DOI: 10.3390/s19071567. [Online]. Available: https://www.mdpi.com/1424-8220/19/7/1567 (cit. on p. 2).

[14] J. F. Peters, M. Baumann, B. Zimmermann, J. Braun, and M. Weil, "The environmental impact of Li-Ion batteries and the role of key parameters – A review," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 491–506, Jan. 2017, ISSN: 18790690. DOI: 10.1016/j.rser.2016.08.039 (cit. on p. 2).

[15] A. Bachir, M. Dohler, T. Watteyne, I. Member, and I. S. Member, "MAC Essentials for Wireless Sensor Networks MAC Essentials for Wireless Sensor Networks," *Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010, ISSN: 1553877X. DOI: 10.1109/SURV.2010.020510.00058 (cit. on p. 2).

[16] L. Chen and K. Bian, "Neighbor discovery in mobile sensing applications: A comprehensive survey," *Ad Hoc Networks*, 2016, ISSN: 15708705. DOI: 10.1016/j.adhoc.2016.05.005 (cit. on pp. 2, 3).

[17] 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, ISSN: 1913-3715. DOI: 10.4236/ijcns.2017.105B023 (cit. on pp. 3, 18, 113).

[18] 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*, 2011, pp. 128–133, ISBN: 9781612081441. [Online]. Available: http://hal.inria.fr/hal-00647869/ (cit. on pp. 3, 5, 19).

[19] 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, ISSN: 1553-877X. DOI: 10.1109/SURV.2013.100113.00293 (cit. on pp. 3, 4, 5, 19).

[20] T. Tang, T. Hong, H. Hong, S. Ji, S. Mumtaz, and M. Cheriet, "An Improved UAV-PHD Filter-Based Trajectory Tracking Algorithm for Multi-UAVs in Future 5G IoT

Scenarios," *Electronics*, vol. 8, no. 10, p. 1188, Oct. 2019, ISSN: 2079-9292. DOI: 10.3390/electronics8101188. [Online]. Available: https://www.mdpi.com/2079-9292/8/10/1188 (cit. on pp. 3, 12, 19, 118).

[21] S. Sudhakar, V. Vijayakumar, C. Sathiya Kumar, V. Priya, L. Ravi, and V. Subramaniyaswamy, "Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires," *Computer Communications*, vol. 149, pp. 1–16, Jan. 2020, ISSN: 1873703X. DOI: 10.1016/j.comcom.2019.10.007 (cit. on p. 3).

[22] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "RFC6775: Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," 2012, [Online]. Available: http://www.hjp.at/doc/rfc/rfc6775.html (cit. on p. 3).

[23] 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. DOI: 10.1109/JSEN.2015.2449079 (cit. on p. 4).

[24] 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, Mar. 2015, ISSN: 00135194. DOI: 10.1049/el.2014.3879 (cit. on p. 4).

[25] 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, ISSN: 21693536. DOI: 10.1109/ACCESS.2015.2457031 (cit. on pp. 4, 16, 18).

[26] 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, ISSN: 10459219. DOI: 10.1109/TPDS.2014.2316159 (cit. on pp. 4, 12).

[27] 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, ISBN: 9781595939906. DOI: 10.1145/1460412.1460420 (cit. on pp. 4, 16).

[28] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" *Mobicom*, p. 185, 2012, ISSN: <null>. DOI: 10.1145/2348543.2348568. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/2348543.2348568 (cit. on pp. 4, 17).

[29] N. Chakchouk, "A Survey on Opportunistic Routing in Wireless Communication Networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2214–2241, 2015, ISSN: 0028-0836. DOI: 10.1007/978-1-4419-0304-4_2. arXiv: arXiv:1302.2563v1 (cit. on p. 4).

[30] 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*, vol. 16, no. 5, pp. 1420–1433, May 2017, ISSN: 1536-1233. DOI: 10.1109/TMC.2016.2595574. [Online]. Available: http://ieeexplore.ieee.org/document/7524719/ (cit. on p. 4).

[31] 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, ISBN: 9781479920037. DOI: 10.1109/ICC.2014.6883353 (cit. on p. 4).

[32] 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, ISBN: 9781479967292. DOI: 10.1109/ICCVE.2014.7297684 (cit. on p. 4).

[33] 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, ISSN: 15708705. DOI: 10.1016/j.adhoc.2016.05.011 (cit. on p. 4).

[34] 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, ISSN: 1229-2370. DOI: 10.1109/JCN.2015.000034 (cit. on p. 4).

[35] 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, ISBN: 9781479936359. DOI: 10.1109/COMSNETS.2014.6734890 (cit. on pp. 4, 18).

[36] R. Yarinezhad and A. Sarabi, "Reducing delay and energy consumption in wireless sensor networks by making virtual grid infrastructure and using mobile sink," *AEU - International Journal of Electronics and Communications*, vol. 84, no. April 2017, pp. 144–152, 2018, ISSN: 16180399. DOI: 10.1016/j.aeue.2017.11.026. [Online]. Available: https://doi.org/10.1016/j.aeue.2017.11.026 (cit. on p. 4).

[37] S. Redhu and R. M. Hegde, "Network lifetime improvement using landmark-assisted mobile sink scheduling for cyber-physical system applications," *Ad Hoc Networks*, vol. 87, pp. 37–48, 2019, ISSN: 15708705. DOI: 10.1016/j.adhoc.2018.10.029. [Online]. Available: https://doi.org/10.1016/j.adhoc.2018.10.029 (cit. on p. 4).

[38] 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, ISSN: 0163-6804. DOI: 10.1109/MCOM.2014.6852097 (cit. on p. 4).

[39] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H.-J. Kim, "Energy Efficient Routing Algorithm with Mobile Sink Support for Wireless Sensor Networks," *Sensors*, vol. 19, no. 7, p. 1494, Mar. 2019, ISSN: 1424-8220. DOI: 10.3390/s19071494. [Online].

Available: https://www.mdpi.com/1424-8220/19/7/1494 (cit. on p. 4).

[40] J.-P. Vasseur and A. Dunkels, "What Are Smart Objects?" In *Interconnecting Smart Objects with IP*, 2010, ch. 1, pp. 3–20, ISBN: 9780123751652. DOI: 10.1016/B978-0-12-375165-2.00001-6 (cit. on p. 11).

[41] *Sonos*. [Online]. Available: https://www.sonos.com/en-gb/home (visited on 06/23/2020) (cit. on p. 11).

[42] *Spotify*. [Online]. Available: https://www.spotify.com/uk/ (visited on 06/23/2020) (cit. on p. 11).

[43] M. Chen, Y. Ma, S. Ullah, W. Cai, and E. Song, "ROCHAS: Robotics and Cloud-assisted Healthcare System for Empty Nester," in *Proceedings of the 8th International Conference on Body Area Networks*, ACM, 2013, pp. 217–220, ISBN: 978-1-936968-89-3. DOI: 10.4108/icst.bodynets.2013.253922 (cit. on p. 12).

[44] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming Agriculture through Pervasive Wireless Sensor Networks," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 50–57, Apr. 2007, ISSN: 1536-1268. DOI: 10.1109/MPRV.2007.47. [Online]. Available: http://ieeexplore.ieee.org/document/4160605/ (cit. on p. 12).

[45] Aqeel-Ur-Rehman, A. Z. Abbasi, N. Islam, and Z. A. Shaikh, "A review of wireless sensors and networks' applications in agriculture," *Computer Standards and Interfaces*, vol. 36, no. 2, pp. 263–270, 2014, ISSN: 09205489. DOI: 10.1016/j.csi.2011.03.004 (cit. on p. 12).

[46] *Afimilk*, 2020. [Online]. Available: https://www.afimilk.com/cow-monitoring (cit. on pp. 12, 19).

[47] M. Salayma, A. Al-Dubai, I. Romdhani, and Y. Nasser, "Reliability and Energy Efficiency Enhancement for Emergency-Aware Wireless Body Area Networks (WBANs)," *IEEE Transactions on Green Communications and Networking*, vol. 2,

no. 3, pp. 804–816, Sep. 2018, ISSN: 24732400. DOI: 10.1109/TGCN.2018.2813060 (cit. on p. 12).

[48] W. Chen, *Vehicular communications and networks : architectures, protocols, operation and deployment*, p. 570, ISBN: 9781782422112 (cit. on p. 12).

[49] *How it works – Google Self-Driving Car Project*, 2016. [Online]. Available: https://www.google.com/selfdrivingcar/how/ (cit. on p. 12).

[50] L. A. Villas, D. L. Guidoni, G. Maia, R. W. Pazzi, J. Ueyama, and A. A. F. Loureiro, "An energy efficient joint localization and synchronization solution for wireless sensor networks using unmanned aerial vehicle," *Wireless Networks*, vol. 21, no. 2, pp. 485–498, Feb. 2015, ISSN: 1022-0038. DOI: 10.1007/s11276-014-0802-2. [Online]. Available: http://link.springer.com/10.1007/s11276-014-0802-2 (cit. on p. 12).

[51] N. Hossein Motlagh, T. Taleb, and O. Arouk, "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, Dec. 2016, ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2612119. [Online]. Available: http://ieeexplore.ieee.org/document/7572034/ (cit. on pp. 12, 13).

[52] P. Pace, G. Aloi, and G. Fortino, "An application-level framework for UAV/rover communication and coordination," in *Proceedings of the 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2015*, IEEE, 2015, pp. 229–233, ISBN: 9781479920020. DOI: 10.1109/CSCWD.2015.7230963 (cit. on p. 12).

[53] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of Industry 4.0," *Wireless Networks*, vol. 23, no. 1, pp. 23–41, Jan. 2017, ISSN: 1022-0038. DOI: 10.1007/s11276-015-1133-7 (cit. on p. 12).

[54] Q. Wang and X. Chang, “The Optimal Trajectory Planning for UAV in UAV-aided Networks,” in, Springer, Cham, 2016, pp. 192–204. DOI: 10.1007/978-3-319-48674-1_18 (cit. on p. 13).

[55] A. Sayyed, G. M. de Araújo, J. P. Bodanese, and L. B. Becker, “Dual-stack single-radio communication architecture for UAV acting as a mobile node to collect data in WSNs,” *Sensors (Switzerland)*, vol. 15, no. 9, pp. 23 376–23 401, 2015, ISSN: 14248220. DOI: 10.3390/s150923376 (cit. on p. 13).

[56] S. Rashed and M. Soyturk, “Analyzing the Effects of UAV Mobility Patterns on Data Collection in Wireless Sensor Networks.,” *Sensors (Basel, Switzerland)*, vol. 17, no. 2, p. 413, 2017, ISSN: 1424-8220. DOI: 10.3390/s17020413. [Online]. Available: http://www.mdpi.com/1424-8220/17/2/413%7B%5C%%7D0Ahttp://www.ncbi.nlm.nih.gov/pubmed/28230727 (cit. on pp. 13, 20).

[57] I. Jawhar, N. Mohamed, J. Al-Jaroodi, D. P. Agrawal, and S. Zhang, “Communication and networking of UAV-based systems: Classification and associated architectures,” *Journal of Network and Computer Applications*, vol. 84, no. 31, pp. 93–108, 2017, ISSN: 10848045. DOI: 10.1016/j.jnca.2017.02.008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1084804517300814 (cit. on p. 13).

[58] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, “UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges,” *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, Mar. 2017, ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600238CM. [Online]. Available: http://ieeexplore.ieee.org/document/7876852/ (cit. on p. 13).

[59] S. B. Hadj, S. Rekhis, N. Boudriga, and A. Bagula, “A cloud of UAVs for the Delivery of a Sink As A Service to Terrestrial WSNs,” in *Proceedings of the 14th International Conference on Advances in Mobile Computing and Multi Media - MoMM ’16*, New York, New York, USA: ACM Press, 2016, pp. 317–326, ISBN: 9781450348065. DOI: 10.1145/3007120.3007138 (cit. on p. 13).

[60] G. Solmaz and D. Turgut, "Tracking pedestrians and emergent events in disaster areas," *Journal of Network and Computer Applications*, vol. 84, pp. 55–67, 2017, ISSN: 10848045. DOI: 10.1016/j.jnca.2017.02.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517300784 (cit. on p. 14).

[61] A. Wichmann and T. Korkmaz, "Analysis of deployment and movement policies in wireless sensor and robot networks," in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, IEEE, Jun. 2015, pp. 1–9, ISBN: 978-1-4799-8461-9. DOI: 10.1109/WoWMoM.2015.7158131. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7158131 (cit. on p. 14).

[62] T. Zhang, D. Wang, J. Cao, Y. Ni, L. Chen, and D. Chen, "Elevator-assisted sensor data collection for structural health monitoring," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1555–1568, 2012. DOI: 10.1109/TMC.2011.191. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=6007139 (cit. on p. 14).

[63] A. Sher, N. Javaid, G. Ahmed, S. U. Islam, U. Qasim, and Z. A. Khan, "MC: Maximum Coverage Routing Protocol for Underwater Wireless Sensor Networks," in *2016 19th International Conference on Network-Based Information Systems (NBiS)*, IEEE, Sep. 2016, pp. 91–98, ISBN: 978-1-5090-0979-4. DOI: 10.1109/NBiS.2016.77. [Online]. Available: http://ieeexplore.ieee.org/document/7789743/ (cit. on p. 14).

[64] R. C. Carrano, D. Passos, L. C. Magalhaes, and C. V. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 181–194, Mar. 2014, ISSN: 1553877X. DOI: 10.1109/SURV.2013.052213.00116 (cit. on pp. 14, 15).

[65] *IEEE 802.15.4*, 2016. [Online]. Available: http://www.ieee802.org/15/pub/TG4.html (visited on 03/22/2016) (cit. on p. 14).

[66] N. Choudhury, R. Matam, M. Mukherjee, and L. Shu, “Beacon Synchronization and Duty-Cycling in IEEE 802.15.4 Cluster-Tree Networks: A Review,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1765–1788, Jun. 2018, ISSN: 23274662. DOI: 10.1109/JIOT.2018.2827946 (cit. on p. 14).

[67] 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, ISSN: 15708705. DOI: 10.1016/j.adhoc.2016.05.011. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2016.05.011 (cit. on p. 15).

[68] J. Kabara and M. Calle, “MAC Protocols Used by Wireless Sensor Networks and a General Method of Performance Evaluation,” *International Journal of Distributed Sensor Networks*, vol. 8, no. 1, p. 834 784, Jan. 2012. DOI: 10.1155/2012/834784. [Online]. Available: http://journals.sagepub.com/doi/10.1155/2012/834784 (cit. on p. 15).

[69] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys ’04*, p. 95, 2004, ISSN: 08858985. DOI: 10.1145/1031495.1031508 (cit. on p. 15).

[70] 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. DOI: 10.1109/WF-IoT.2015.7389054 (cit. on p. 15).

[71] M. J. McGlynn and S. A. Borbash, “Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks,” in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing - MobiHoc ’01*, New York, New York, USA: ACM Press, 2001, p. 137, ISBN: 1581134282. DOI: 10.1145/501431.501435 (cit. on p. 16).

[72] William Stein, *Elementary Number Theory: Primes, Congruences, and Secrets*, First. Springer, 2011. [Online]. Available: `http://wstein.org/ent/ent.pdf` (cit. on p. 16).

[73] S. Lai, B. Ravindran, and H. Cho, "Heterogenous Quorum-Based Wake-Up Scheduling in Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 59, no. 11, pp. 1562–1575, Nov. 2010, ISSN: 0018-9340. DOI: `10.1109/TC.2010.20`. [Online]. Available: `http://ieeexplore.ieee.org/document/5396329/` (cit. on p. 17).

[74] A. Khan, A. Abdullah, M. Anisi, and J. Bangash, "A Comprehensive Study of Data Collection Schemes Using Mobile Sinks in Wireless Sensor Networks," *Sensors*, vol. 14, no. 2, pp. 2510–2548, Feb. 2014, ISSN: 1424-8220. DOI: `10.3390/s140202510`. [Online]. Available: `http://www.mdpi.com/1424-8220/14/2/2510/` (cit. on p. 19).

[75] Y. Gu, F. Ren, Y. Ji, and J. Li, "The Evolution of Sink Mobility Management in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 507–524, 2016, ISSN: 1553-877X. DOI: `10.1109/COMST.2015.2388779`. [Online]. Available: `http://ieeexplore.ieee.org/document/7004042/` (cit. on p. 20).

[76] C. Thomson, I. Romdhani, M. Qasem, B. Ghaleb, and I. Wadhaj, "Cooja Simulator Manual," no. C, pp. 2015–2016, 2016. DOI: `10.13140/RG.2.1.4274.8408` (cit. on p. 20).

[77] *OMNeT++*, 2015. [Online]. Available: `https://omnetpp.org/` (visited on 01/09/2016) (cit. on pp. 20, 43, 71, 94).

[78] I. Wadhaj, I. Kristof, I. Romdhani, and A. Al-dubai, "Performance Evaluation of the RPL Protocol in Fixed and Mobile Sink Low-Power and Lossy-Networks," in *14th IEEE International Conference on Ubiquitous Computing and Communications*, Liverpool, 2015 (cit. on p. 20).

[79] S. Thomas and T. Mathew, "Intelligent path discovery for a mobile sink in wireless sensor network," in *8th International Conference on Advances in Computing and Communication (ICACC-2018)*, vol. 143, Elsevier B.V., 2018, pp. 749–756. DOI: 10.1016/j.procs.2018.10.430. [Online]. Available: https://doi.org/10.1016/j.procs.2018.10.430 (cit. on p. 24).

[80] N. Ghosh and I. Banerjee, "An energy-efficient path determination strategy for mobile data collectors in wireless sensor network," *Computers and Electrical Engineering*, vol. 48, pp. 417–435, 2015, ISSN: 00457906. DOI: 10.1016/j.compeleceng.2015.09.004. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2015.09.004 (cit. on p. 24).

[81] 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, ISBN: 978-1-931971-37-9. DOI: 10.1093/abbs/gmw048 (cit. on p. 24).

[82] 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. DOI: 10.1109/WF-IoT.2015.7389054 (cit. on p. 27).

[83] *IEEE 802.15.4*, 2016. [Online]. Available: http://www.ieee802.org/15/pub/TG4.html (visited on 03/22/2016) (cit. on p. 27).

[84] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "RFC4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks," 2007 (cit. on p. 27).

[85] T. Clausen, "RFC 3626 - Optimized Link State Routing Protocol (OLSR)," 2003, [Online]. Available: https://tools.ietf.org/pdf/rfc3626.pdf (cit. on pp. 28, 70).

[86] 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, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2532745 (cit. on p. 39).

[87] *MiXiM*, 2016. [Online]. Available: http://mixim.sourceforge.net/ (visited on 01/18/2016) (cit. on pp. 43, 71, 94).

[88] *Inetmanet Installation Guide*, 2016. [Online]. Available: http://omnet-manual.com/inetmanet-installation/ (cit. on pp. 43, 71, 94).

[89] M. Thangaraj, S. Anuradha, and S. Muthukala, "Energy Framework Enhancements in the WSN Simulator OMNETPP," Tech. Rep. 1, 2014, pp. 18–24. [Online]. Available: www.ijeijournal.com (cit. on p. 43).

[90] Z. Zhao and T. Braun, "OMNeT++ based Opportunistic Routing Protocols Simulation: A Framework," Tech. Rep. (cit. on p. 43).

[91] P. Dymora, M. Mazurek, and P. Płonka, "Simulation of reconfiguration problems in sensor networks using OMNeT++ software," *Annales UMCS, Informatica*, vol. 13, no. 2, pp. 49–67, 2014, ISSN: 1732-1360. DOI: 10.2478/v10065-012-0041-z (cit. on p. 43).

[92] R. Helal and A. ElMougy, "An energy-efficient Service Discovery protocol for the IoT based on a multi-tier WSN architecture," in *Proceedings - Conference on Local Computer Networks, LCN*, vol. 2015-Decem, IEEE, 2015, pp. 862–869, ISBN: 9781467367738. DOI: 10.1109/LCNW.2015.7365939 (cit. on p. 43).

[93] *IntRange*, 2019. [Online]. Available: https://github.com/inetmanet/inetmanet/blob/master/src/underTest/%20wpan/linklayer/ieee802154/phyLayer/Ieee802154Phy.cc (visited on 04/29/2019) (cit. on p. 44).

[94] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RFC6550: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," 2012, [Online]. Available: https://datatracker.ietf.org/doc/rfc6550/ (cit. on p. 119).

Appendix A

A.1 Best Paper Award

2019 IEEE International Conference on
Smart Internet of Things (IEEE SmartIoT 2019)

BEST PAPER AWARD

Presented to:

Craig Thomson, Isam Wadhaj, Zhiyuan Tan and Ahmed Al-Dubai

For the paper titled

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

Presented on August 10, 2019
Tianjin, China

Mohammed Atiquzzaman

Xiaobo Zhou

Dapeng Oliver Wu

Appendix B

B.1 Peer-reviewed Publications During this Study

The details of the publications achieved throughout this PhD are shown as follows:

- C. Thomson, I. Wadhaj, Z. Tan, and A. Al-Dubai, “Mobility aware duty cycling algorithm (MADCAL) in wireless sensor network with mobile sink node,” in Proceedings - **2019 IEEE International Conference on Smart Internet of Things (SmartIoT 2019), August 2019**, pp. 189–196, doi: 10.1109/SmartIoT.2019.00037. **WINNER OF BEST PAPER AWARD**

- C. Thomson, I. Wadhaj, Z. Tan, and A. Al-Dubai, “Mobility Aware Duty Cycling Algorithm (MADCAL) A Dynamic Communication Threshold for Mobile Sink in Wireless Sensor Network,” **Sensors, vol. 19, no. 22, p. 4930, Nov. 2019**, doi: 10.3390/s19224930.

- C. Thomson, I. Wadhaj, A. Al-dubai, and Z. Tan, “A New Mobility Aware Duty Cycling and Dynamic Preambling Algorithm for Wireless Sensor Network,” in **The 2020 IEEE 6th World Forum on The Internet of Things (WF-IoT 2020), Oct 2020**. doi:10.1109/wf-iot48130.2020.9221036.

- C. Thomson, I. Wadhaj, Z. Tan, and A. Al-Dubai, “Towards an Energy Balancing Solution for Wireless Sensor Network with Mobile Sink Node,” **Journal of Computer Communications, Special Issue on Optimization of Cross-layer Col-**

laborative Resource Allocation for Mobile Edge Computing, Caching and Communication, 2020. Accepted.